ENERGY-AWARE QOS SCHEDULING AT MAC LEVEL IN WIMAX

A THESIS IN COMPUTER ENGINEERING
Master of Science in Computer Engineering

Presented to the faculty of the American University of Sharjah
College of Engineering
in partial fulfillment of
the requirements for the degree

MASTER OF SCIENCE

By
SANABEL ALNOURANI
B.S. 2007

Sharjah, UAE
January 2011

ENERGY-AWARE QOS SCHEDULING AT MAC

LEVEL IN WIMAX

Sanabel Hassan Mai, Candidate for the Master of Science Degree

American University of Sharjah, 2011

ABSTRACT

In a mobile wireless network, energy saving of mobile devices is one of the most important features for the extension of devices' life-time and the network. In mobile networks, the device is expected to have several connections, each with different QoS (Quality of Service) requirements. Meeting the QoS requirements on such devices along with better power saving is a challenging task. Moreover, in real-time scenarios, connections are expected to join and leave the network randomly. Before admitting a connection to the network, its QoS requirements must be checked to make sure that the network has adequate resources to accommodate it. Without a proper call admission control mechanism, the system cannot provide the promised QoS to the real-time applications.

This research proposes a scheduling algorithm and a call admission control policy for IEEE 802.16e broadband wireless access standard. The proposed scheduling algorithm is designed towards minimizing power consumption at mobile stations, while maintaining different QoS requirements for real-time traffic. The proposed algorithm considers the dynamic nature of connection joining and

termination. Connections will be allowed to join the network only if their QoS parameters can be met without violating those of existing connections.

Simulation results show that when QoS delay requirements of the connections are not too restrictive, power savings of approximately 75% and 50% at the mobile station can be achieved for low- and moderate- rate Unsolicited Grant Service traffic types, respectively.

CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

ACKNOWLEDGEMENTS

بسم الله الرحمن الرحيم

I would like to thank God for all I am and all I have. I am heartily thankful to my advisors, Dr.Rana Ahmed and Dr. Taha Landolsi, who have supported me throughout my thesis with their patience and knowledge while giving me room to work in my own way. This work would not have been accomplished without their constant encouragement and effort.

I am indebted to my friends who have provided me a stimulating and fun filled environment. My thanks go in particular to Ayman, Mai, Reem, Ahmad, Tahera, Hiba, Adi, Noha and Omniah.

Finally, saving the best for the last, I extend huge, warm thanks to my family for their valuable help, and moral support. I am ever indebted to my aunt Reem and I admire her distinguished helping nature and all the positive energies she surrounded me by. I wish to thank my uncle Ibrahim for inspiring me and guiding my thoughts whenever I needed. I wish also to thank my little Hussein, for all the feelings we experienced together. I would like also to thank my mother for her unconditional love and nonstop caring.

Thank you God for a house full of people I love.

CHAPTER 1

INTRODUCTION

This chapter will give a brief introduction about WiMAX technology. It starts by providing a general overview of WiMAX networks. Then, it defines the QoS service classes in the WiMAX standard, followed by the three power saving classes defined by the standard. Finally the objective and contribution of this thesis will be presented and the general outline of the thesis will be provided.

1.1 Overview of WiMAX Technology

WiMAX is a new standard developed by the IEEE 802.16 group based on wireless metropolitan area networking (WMAN) standards. WiMAX stands for Worldwide Interoperability for Microwave Access, and it is also called WirelessMAN. It has been defined as a "last mile" technology by the WiMAX Forum [1]. WiMAX is capable of providing services for fixed, nomadic, portable or mobile wireless connectivity. WiMAX does not require a direct line of sight (LOS) with a base station [2], [3], [4].

The original 802.16 standard was based on a single carrier physical layer with a burst time division multiplexing (TDM) MAC layer. To include non-line of sight (NLOS) applications, the IEEE 802.16 group produced 802.16a (or the fixed WiMAX) that supports fixed applications and uses orthogonal frequency division multiplexing (OFDM). OFDM is a multi-carrier modulation technique in which a large number of subcarriers are used to transmit data. The data is split into several parallel data streams or channels, one for each sub-carrier. OFDM allows only one user on the channel at any given time, as illustrated in Figure 1.1a. Each color in the figure represents a different user.

WiMAX group then developed IEEE 802.16e that supports mobility using orthogonal frequency division multiple access (OFDMA)-based physical layer. OFDMA is a multi-user OFDM that allows multiple users to transmit using different subcarriers in the same channel, as illustrated in Figure 1.1b [2], [3], [4].

WiMAX is expected to provide broadband wireless access in a range up to 50kms for fixed stations, and a range of 5 to 15kms for mobile stations. It also provides a maximum data rate of up to 74Mbps [2], [3], [4].

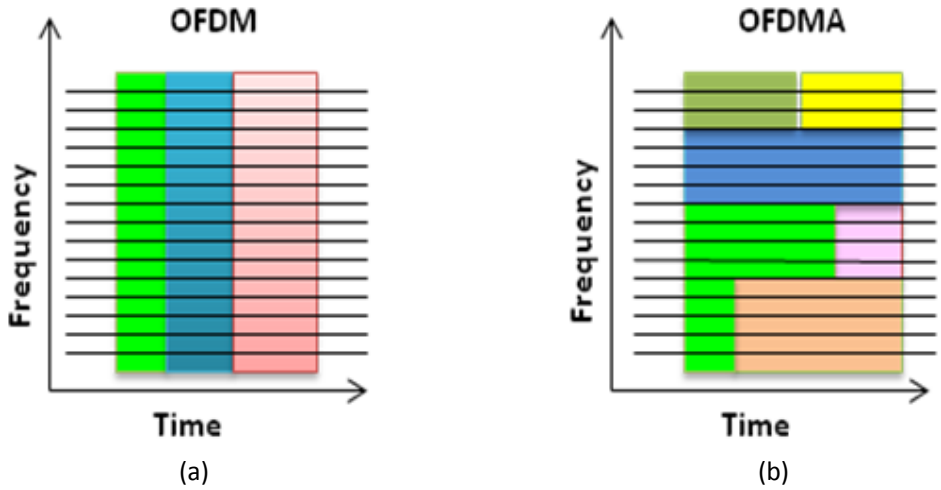


Figure 1.1: IEEE 802.16 physical: (a) OFDM and (b) OFDMA [2].

Some of the features of WiMAX network that differentiate it from other wireless broadband technology are briefly described below [2], [5]:

- WiMAX uses OFDMA-based physical layer that allows operations in NLOS conditions.
- WiMAX is capable of providing very high peak data rates up to 74 Mbps. Furthermore, WiMAX allows a scalable use of data rate according to the channel bandwidth.
- WiMAX supports adaptive modulation and coding per subscriber based on the channel conditions.
- WiMAX supports Time Division Duplexing (TDD) and Frequency Division Duplexing (FDD) that allows for low-cost implementations.

- WiMAX connection-oriented MAC layer provides QoS for different traffic types, such as best-effort (BE), real-time, non-real time, constant bit rate (CBR), and variable bit rate (VBR) traffic.
- WiMAX supports security and automatic retransmission request (ARQ).
- One important feature of the WiMAX is the power saving feature, in which mobile subscriber stations can operate for longer time without needing to re-charge their batteries.

In power saving mode, the mobile station (MS) will power down one or more of its hardware components to conserve energy when there are no packets to send or receive. By doing so, the MS enters into the sleep state or the sleep window. Sleep window is followed by a listen window, in which the MS wakes up to listen for the incoming data traffic from the base station (BS) [2].

1.1.1 WiMAX Frame Structure

The WiMAX PHY layer is responsible for slot allocation over the air. The minimum time-frequency resource that can be allocated by a WiMAX system to any user is called a slot. The slot consists of one subchannel over one, two, or three OFDM symbols, depending on the subchannelization scheme applied. WiMAX is flexible in terms of how multiple users and packets are multiplexed on a single frame. The scheduling algorithm could assign data slots to different users based on other QoS requirements, and channel conditions [2], [5].

Figure 1.2 shows an OFDM frame structure when operating in TDD mode. The frame is divided into two sub-frames: downlink (DL) and uplink (UL) sub-frame separated by a small guard interval. The downlink-to-uplink-subframe ratio is varied from 3:1 to 1:1 depending on the traffic profile. The OFDMA structure is the same as shown in Figure 1.2 except that both downlink and uplink will be transmitting simultaneously over different carriers [2], [5].

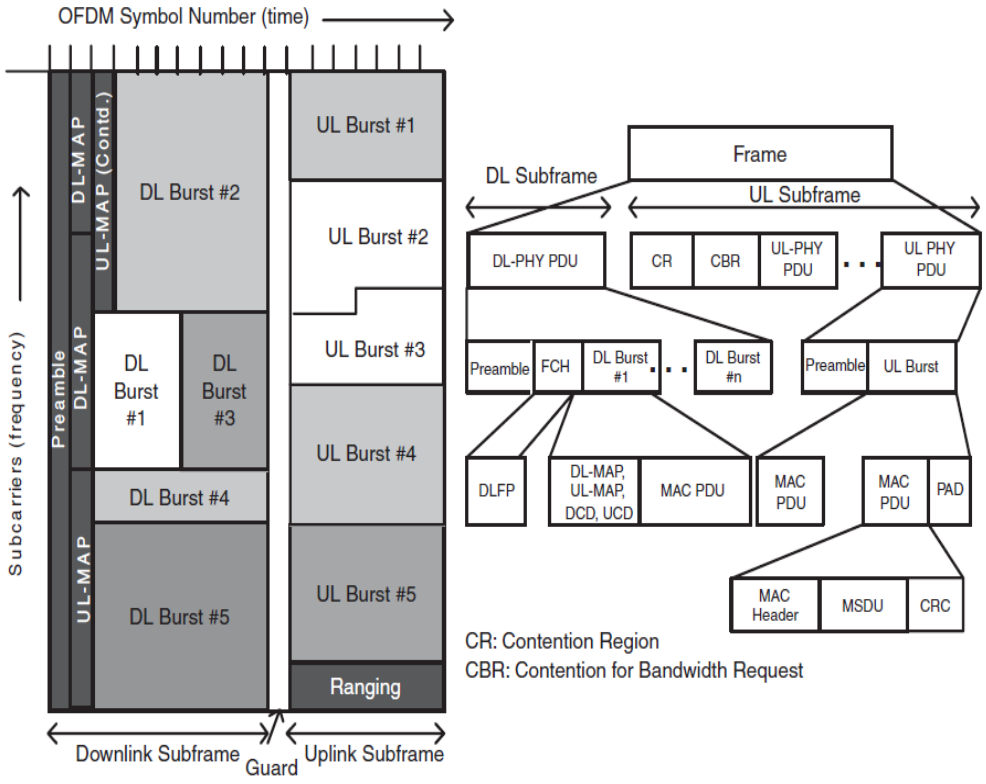


Figure 1.2: A sample TDD frame structure for mobile WiMAX [2].

A preamble in the frame is used for time synchronization and initial channel estimation. The downlink preamble is followed by a Forward Error Control (FEC), which carries system control information such as modulation type, coding scheme and the length of the DL-MAP message. The downlink and uplink MAP messages (DL-MAP and UL-MAP) define the burst profile for each user (slot allocation) [2], [5].

Each burst contains MAC protocol data units (MPDUs), and each MPDU consists of a header followed by a payload and a cyclic redundancy check (CRC).The MAC header is used to carry data and MAC-layer signaling messages. The MAC header contains connection identifier (CID) in addition to other information elements as shown in Table 1-1[2], [5].

Table 1-1: Generic MAC header fields [2].

Field	Length (bits)	Description
HT	1	Header type (set to 1 for such header)
EC	1	Encryption control (set to 0 for such header)
Type	3	Type
BR	19	Bandwidth request (the number of bytes of uplink bandwidth requested by the SS for the given CID)
CID	16	Connection indentifier
HCS	8	Header check sequence

1.2 WiMAX QoS Service Classes

WiMAX provides QoS requirements by using a connection-oriented MAC layer. In order to support a wide variety of applications, IEEE 802.16 has defined five different service classes. Each of these classes has certain QoS parameters. Table 1-2 [2] presents QoS parameters for the five service flow classes. Table 1-3 [5] further presents some of the WiMAX applications for different QoS classes, and their related guidelines.

Unsolicited grant service (UGS) class is designed for traffics with a fixed-size packet such as constant bit rate traffic [2], [5].Voice over IP (VoIP) without silence suppression is a good example of UGS class.

Real-time Polling Service (rtPS) class is designed for real time traffic. It is designed to support applications with variable-packet size, such as MPEG compressed video [2], [5].

Non-real-time Polling Service (nrtPS) class is designed for non-real-time variable bit rate traffic that requires no delay guarantees, such as File Transport Protocol (FTP) traffic. In this class, only minimum data rate is guaranteed [2], [5].

Best-effort (BE) service class is designed for data streams. In this class, delay and throughput are not guaranteed. Web browsing traffic is an example of applications using this service class [2], [5].

Extended real-time variable rate (Ert-VR) service class is designed for real time applications that have variable data rates such as VoIP with silence suppression. Applications using this class require guaranteed delay, data rate and jitter [2], [5].

Table 1-2: Services classes supported in WiMAX [2].

Service Flow Designation	Defining QoS Parameters	Application Examples
Unsolicited grant services (UGS)	Maximum sustained rate, Maximum latency tolerance, Jitter tolerance	Voice over IP (VoIP) without silence suppression
Real-time Polling service (rtPS)	Minimum reserved rate, Maximum sustained rate, Maximum latency tolerance, Traffic priority	Streaming audio and video, MPEG (Motion Picture Experts Group) encoded
Non-real-time Polling service (nrtPS)	Minimum reserved rate, Maximum sustained rate, Traffic priority	Streaming audio and video, MPEG (Motion Picture Experts Group) encoded
Best-effort service (BE)	Maximum sustained rate, Traffic priority	Web browsing, data transfer
Extended real-time Polling service (ErtPS)	Minimum reserved rate, Maximum sustained rate, Maximum latency tolerance, Jitter tolerance, Traffic priority	VoIP with silence suppression

Table 1-3: WiMAX application classes [5].

Classes	Applications	Bandwidth Guidelines		Latency Guidelines		Jitter Guidelines		QoS Classes
1	Multiplayer Interactive Gaming	Low	50 kbps	Low	< 25 ms	N/A		rtPS and UGS
2	VoIP and Video Conferenc	Low	32-64 kbps	Low	< 160 ms	Low	< 50 ms	UGS and ertPS
3	Streaming Media	Low to high	5 kbps to 2 Mbps			Low	< 100 ms	rtPS
4	Web Browsing and Instant Messaging	Moderate	10 kbps to 2 Mbps			N/A		nrtPS and BE
5	Media Content Downloads	High	> 2 Mbps			N/A		nrtPS and BE

1.3 WiMAX Power Saving Classes

The energy dissipation in a mobile station (MS) can be reduced through the "sleep mode" operation which extends the life time of the mobile station. IEEE 802.16e has defined three power saving classes. Each connection in a mobile station can select a certain power saving class (PSC). PSC can be defined for a group of connections with common properties. Each power saving class can be activated or deactivated. When the PSC is activated, MS can start alternate sleeping/listening windows of the class to save power. When the PSC is deactivated, MS will go back to normal operation of the corresponding connection.

Mobile station negotiates the parameters of a power saving class with the base station to decide the power saving parameters, such as the time to sleep and the time to listen, the length of the sleep and the listen intervals [2].

There are three types of power saving classes with different parameters and different activation/deactivation procedures as illustrated in Figure 1.3.

1) Power saving class of type I:

In this PSC, MS will first sleep for a period of time then wake up to listen if BS has any downlink data for the MS. If there are no packets buffered in the BS for the MS, then the MS will double the length of the previous sleep window, and the

process of doubling the sleep interval continues until the final-sleep window size is reached [2].

Doubling exponentially the sleep interval achieves efficient power saving; however, this will increase the delay. Therefore, type I power saving class is recommended for Best Effort (BE) and non-real time variable rate (nrt-VR) traffic type. PSC of type I is suitable for web browsing and data access services [2].

2) Power saving class of type II:

In this class, the sleep and listen period are fixed; all of the sleep and listen windows are of the same size as the initial-sleep window. In this class, MS repeats the sleep and listen period on a round-robin fashion [2].

Unlike PSC of type I, type II PSC avoids increasing the delay by keeping the sleep interval length fixed. Therefore, type II PSC is recommended for Unsolicited Grant Service (UGS), real-time variable rate (rt-VR), and extended real-time variable rate (Ert-VR) traffic. This class works well for real-time connections such as voice over IP (VoIP) and video streaming services since such applications have to send or receive packets on a periodic basis [2].

3) Power saving class of type III:

This power saving class involves only one sleep window and no listening window. This sleep window is specified by final-sleep window parameters. Mobile station will sleep for the predefined sleep period. When the sleep period is expired, the PSC is deactivated and the MS returns to normal operation [2].

Type III power saving class is suitable for management connection as well as multicast connection [2].

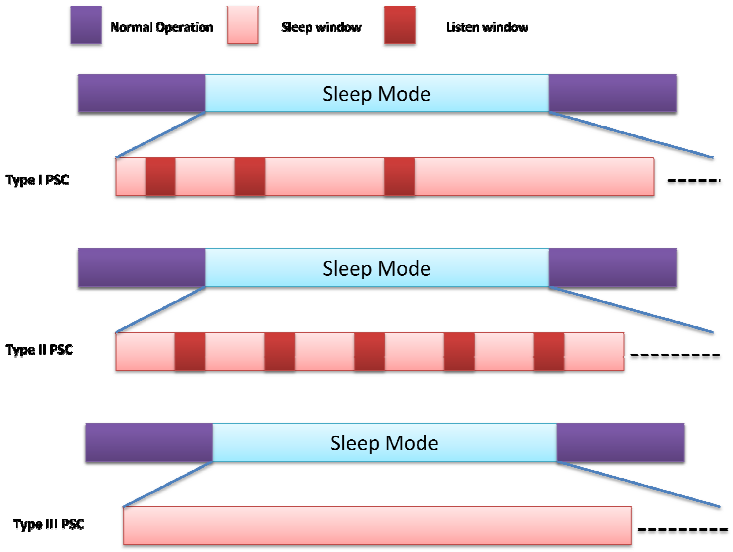


Figure 1.3: Power saving classes defined in IEEE 802.16e.

1.4 Motivation and Problem Statement

Power saving for mobile devices is a critical design aspect in wireless networks, such as WiMAX. In WiMAX systems, although three power saving classes have been defined in the standard, these classes are per-connection based. Therefore, when a MS has multiple connections, power saving efficiency can degrade drastically. MS with multiple connections should determine its actual sleep and listen intervals, by finding a common sleep interval among all connections. Managing multiple connections on a MS without violating QoS of any connection along with power saving is very challenging, and it allows MSs to operate for longer duration without having to re-charge their batteries.

Another important factor to consider is that in real-time sessions, connections are expected to join and leave the network randomly. Call Admission control (CAC) mechanism that can be used to ensure that the incoming connection'QoS parameters will not degrade the QoS of existing connections is not defined by the WiMAX standard.

In this research, we consider a WiMAX environment with multiple connections on a single MS connected to a single BS. The first objective of this research is to propose and evaluate a new energy-aware uplink packet scheduling approach at the MAC layer of WiMAX. This research proposes a new scheduling scheme to handle different connections of UGS traffic classes at a MS without

violating their QoS parameters (delay, bandwidth requirement). The proposed schedule is targeted toward minimization of power consumption for a mobile station. The second objective of this research is to propose and evaluate an efficient Call Admission Control to ensure QoS guarantees for all existing connections.

1.5 Methodology

Calls arriving at the BS will be checked first before admitting them to the network by two-step CAC (Bandwidth (BW) and delay). In the first step, CAC makes sure that the requesting call's BW can be accommodated within the network available resources. Once the incoming call passes this step, it will be forwarded to the second step of testing. In the second step, the call's delay requirement will be checked to make sure accepting the call will not degrade the QoS of other existing calls. The accepted calls requirements will be then forwarded to the UPS (Uplink Packet Scheduling) module. The UPS processes all available connections and finds the best schedule that meets their QoS parameters (BW, delay). UPS also applies power saving class of type III that will schedule the connections in the minimum possible number of awakened OFDM frames. The process of finding new schedule is invoked each time new call passes the CAC test.

The proposed UPS and CAC are designed in such a way to support multiple UGS connections on a single MS. A Matlab simulation model is developed to validate and evaluate the performance of the proposed scheduling strategies.

1.6 Thesis Contribution and Outline

The contributions of this research can be summarized as follows:

1. An Uplink Packet Scheduling residing at the BS is proposed to provide QoS guarantees in terms of delay and bandwidth for multiple UGS connection initiated by one MS.
2. A two-step efficient Call Admission Control residing at the BS is proposed to decide on accepting or rejecting the new calls requests based on the QoS constraints of the existing calls and the new calls.
3. The proposed model is simulated and it is found to save power consumption at MS by achieving good sleeping periods and bandwidth utilization of a single

MS by utilizing the type-three power saving class defined in the IEEE 802.16e.

The chapters to follow are outlined as follows:

Chapter 2 includes the summery of the literature review. It reviews similar work done in this research area. Chapter 3 discusses in details the proposed QoS architecture at the base station and its modules (UPS and CAC), and it provides the algorithms for each of the proposed modules in the QoS architecture. Chapter 4 includes the simulation results for two different traffic type scenarios, and a detailed analysis and discussion for each result. Chapter 5 summarizes the findings of this thesis and presents recommendations for future work.

CHAPTER 2

LITERATURE REVIEW

The topic of QoS scheduling in WiMAX has been extensively studied in the recent past [6-15]. However, QoS scheduling with the consideration of power saving at mobile station along with efficient CAC has received little attention from researchers.

This chapter provides the literature review done in near-field. The literature is divided into three sections. The first section reviews some of the QoS architectures that have been proposed to support QoS requirements for different traffic types, and to control the acceptance of the incoming flows. However, the proposed QoS architectures do not consider the topic of power efficiency. The second section reviews some of many algorithms that are proposed to reduce the power consumption of a mobile station. Unfortunately, some algorithms do not consider the issue of maintaining the QoS parameters. Furthermore, no call admission control algorithm is used to decide on the incoming traffic. The third section reviews the algorithms that combine scheduling the packets while maintaining their QoS parameters, and minimizing the sleeping period intervals to save more power at the mobile station. However, very few researchers have addressed this issue.

2.1 QoS Architecture

Ayman and Adlen [13] have classified the WiMAX scheduling architecture into two main categories; traditional methods and new methods. The traditional methods are based on classical scheduling algorithm, such as First In First Out, Round Robin, etc. The traditional methods are further divided into simple and hierarchal methods. Hierarchal schemes try to maintain fairness among different classes by preventing high priority connections (UGS,rtPS) from starving the bandwidth of lower priority connections (nrtPS,BE). However, the main problem in hierarchal

algorithms is their implementation complexity. New scheduling techniques can be either designed to treat all classes together in one scheme or to treat only one class.

Admission control schemes and packet scheduling algorithm are designed to offer QoS services in WiMAX network. A number of studies have investigated these subjects in IEEE 802.16. However, these scheduling algorithms do not consider the power consumption of a mobile station. For example, Wongthavarawat and Ganz [14] proposed the architecture of the uplink scheduler. Their proposed architecture completes the parts undefined by the standard to support all types of service flows. In their proposed architecture as shown in Figure 2.1, they have added a detailed Uplink Packet Scheduling (UPS) module and admission control module (at the base station). They have also added Traffic Policing module (at the subscriber station).

The proposed UPS in [14] consists of three main modules: information module, scheduling database module and service assignment module. The information module deals with traffic types that issue BW-Request (rtPS, nrtPs, BE) and provides the scheduling database with the queue length as obtained from BW-Request. The scheduling database module creates tables to save data for all type of flows. The service assignment module uses the database tables created in the scheduling database module to generate the UL-MAP. To support all types of traffic flows, their proposed UPS schedules each of the four traffic classes using strict priority bandwidth allocation discipline. Bandwidth allocation per traffic flow follows the following strict priority, from highest to lowest: UGS, rtPS, nrtPS and BE. They have proposed a set of admission control schemes for all traffic types. Admission control module accepts connections only if its QoS requirement in terms of delay and bandwidth can be met, and if the new connection's requirement will not degrade the QoS of existing connections. In their simulation, they assumed only two types of traffic (rtPS and BE), and that all the traffic is already admitted. The results reveal that their proposed UPS can provide QoS support for rtPS in terms of bandwidth and delay. Nevertheless, the proposed packet scheduling does not arrange the packets in fewer frames, therefore it results in poor power efficiency at the subscriber station.

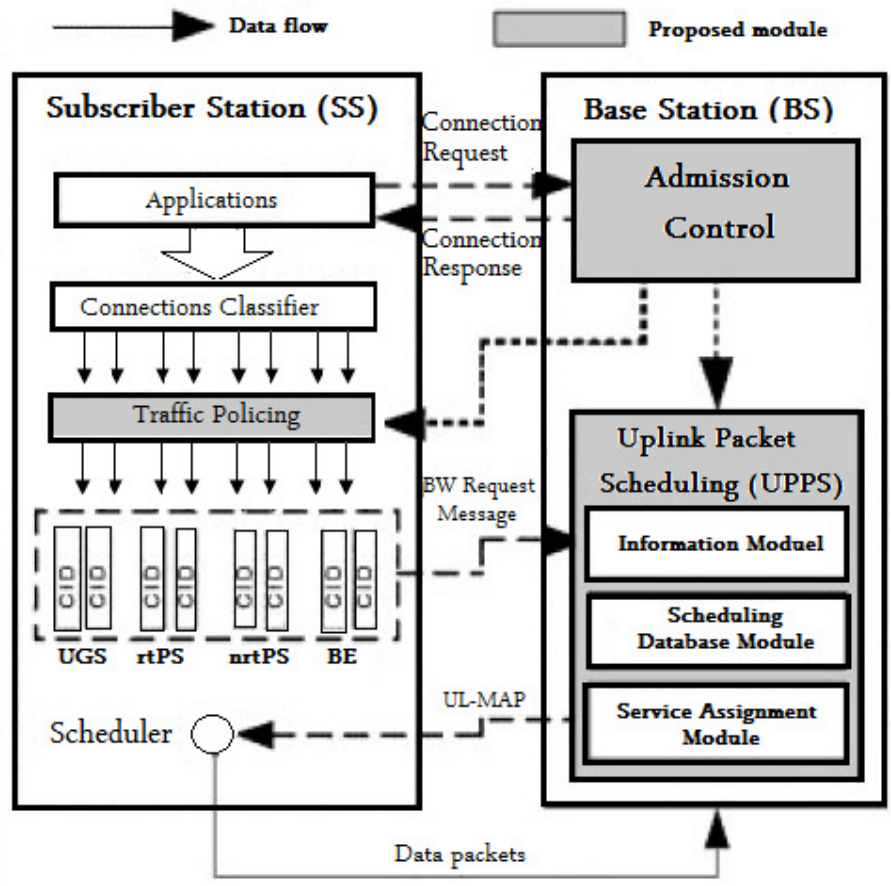


Figure 2.1: QoS architecture used in [14].

Sarat and Anirudha [15] proposed a QoS architecture at the BS and the SS. They have also proposed an efficient QoS Call Admission Control (CAC) that provides QoS guarantee in terms of bandwidth and delay to all registered connections as per service types. The traffic scheduler, which resides at the BS, decides on the allocation of connections' slots in each frame by considering the connection service type (UGS, rtPS,ntrPS or BE), the QoS parameter values of the connections, the availability of data for transmission and the capacity of the available bandwidth. Their proposed BS architecture is shown in Figure 2.2.When a new connection request is sent to the BS, the request will be classified depending on the type of service into one of the Priority Queues (with Priority Order: UGS Queue>RTPS Queue >NRTPS Queue). At the end of each scheduling interval, BE connections will be allocated within the remaining bandwidth (if any). The proposed CAC module in [15] accesses UGS, rtPS, and nrtPS queues to check if their requested QoS constraints can be met. When a connection is accepted, then the CAC will inform the scheduler at

the MS to allocate bandwidth request slots in the next scheduling interval to that connection. Authors in [15] have proposed algorithms of call admission control for UGS and rtPS connection only. For variable bit rate flows (rtPS and nrtPS), they have proposed a simple method of estimating the bandwidth which improves the performance of the system in terms of flow acceptance ratio. Simulation results show that their QoS-CAC achieves better performance than the conventional CAC-BW in terms of flow acceptance ratio. Although QoS-CAC ensures QoS for all registered connections, but it does not optimize the power consumption when scheduling them.

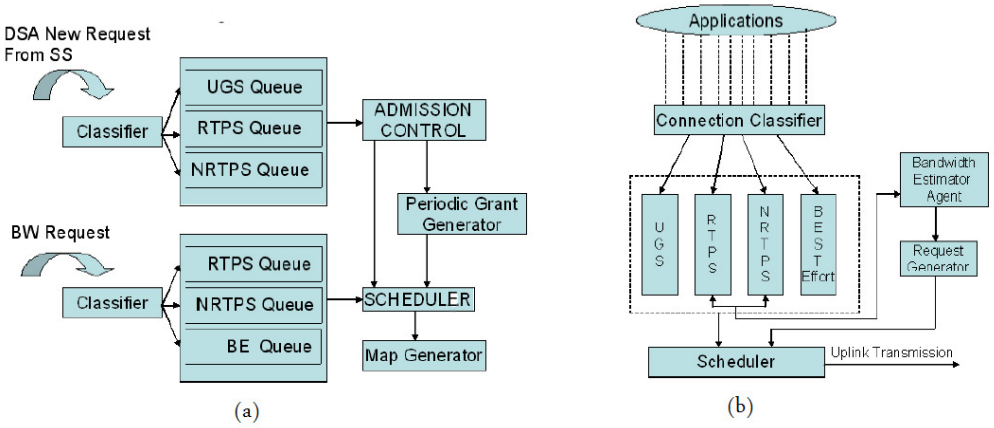


Figure 2.2: (a) BS architecture and (b) SS architecture used in [15].

In [5], Chakchai et al., presented an extensive survey of recent packet scheduling research for WiMAX networks. They have discussed the main design factors and key issues that should be considered when designing a scheduler. Many delay-based algorithms that are designed for real-time traffic have been surveyed in [5]. For example, Largest Weighted Delay First (LWDF) algorithm which chooses packets with largest delay to be scheduled first to avoid missing its deadline is proposed in [7]. Another example is found in [8], in which Delay Threshold Priority Queuing (DTPQ) algorithm is proposed to schedule packets from real-time and non real-time traffic. Other scheduling approaches for real time connections such as priority-based algorithms have been also surveyed in [5]. Packet-based algorithms guarantee the QoS to different classes. For example, packets can be prioritized based on their delay value [9]. On the other hand, connections from the longest queue will be assigned the highest bandwidth as in [6]. However, none of the surveyed

algorithms in [5] follow a certain power saving scheme to reduce the number of used OFDM frames when scheduling.

2.2 Power Saving in WiMAX Mobile Station

The power consumption issue in WiMAX has been widely investigated recently, and many algorithms have been suggested to determine the sleeping periods of a mobile station, thus improving its energy efficiency.

In [16], Min-Gon et al., proposed a new Scheduled Power-Saving Mechanism (SPSM). The mechanism synchronizes the starting time of the sleep-mode operation of a new connection with the connections that are already in the sleep-mode. This can be realized through controlling the operating parameters, i.e., the minimum and the maximum sleep intervals. This algorithm can be applied on two or more connections in an MS. However, their proposed scheme can only be applied when the QoS type of the newly initiated connection is non-real-time connection (low QoS). Simulation results show that their scheme achieves better power consumption of the MS than the traditional approach.

Tuan-Che and Jyh-Cheng in [17] proposed maximizing unavailability interval for the mixture of type I and type II power saving classes. “Unavailability interval” is defined as the time intersection of all activated power saving classes when they are all in their sleep window. During the unavailability interval, there is no communication between the MS and the BS in the downlink direction or in the uplink direction. In their proposed algorithm, the availability interval remains unchanged once it is defined. Simulation results demonstrate that their proposed algorithm reduces power consumption and average packet response time. Furthermore, because power efficiency in sleep mode is improved, the average waiting time at the mobile subscriber station decreases.

In [18] , Omanande et al., have proposed an adaptive power saving algorithm which addresses the variable traffic rate problem, and decreases the average delay for packet reception with respect to 802.16e. Their architecture considers only power saving class I, by adapting the minimum time required to sleep according to the downlink traffic pattern. Analytical model and simulation result are used to evaluate the proposed algorithm.

In [19], Jaehyuk et al., introduced new parameters of the sleep mode operation which are the initial-sleep window size, the final sleep window size and power saving threshold. These parameters were adapted according to different traffic types (CBR and FTP) to achieve efficient power saving. They applied the power saving strategies to different traffic types in order to obtain optimal operational parameters that satisfy different QoS requirements. However, in their proposed adaptive power saving strategies they have considered only Power saving Class of type I.

None of the above mentioned power saving-based approaches propose any mechanism to control the admissibility of the incoming flow that aims to maintain the QoS parameters of the new and existing connections.

2.3 Power Saving With QoS Guarantee

Very few QoS architectures have proposed a complete solution for real-time traffic in terms of reducing the number of awakened OFDM frames, scheduling packets while ensuring their stringent QoS parameters, and performing call admission control to decide on the requested connections at the same time.

Shih-chang et al., in [20] proposed three energy-efficient scheduling approaches for 802.16e. In their research, they considered multiple MS, each MS has multiple connections. Constant bit rate traffic with QoS delay constraint is considered. They introduced a QoS scheduling mechanism that finds the theoretical maximum sleeping time for single MS, which can be used as an upper bound for the energy-saving mechanisms in IEEE 802.16e. This theoretical maximum sleeping time is called the Minimum Wakeup Time scheduler. Even though they have considered multiple MS in their proposed approach, each MS has the same fixed number of connections, and all connections are of the same requirements. All established connections are assumed to last the entire duration. However, CAC unit that monitors the incoming traffic is not described. Simulation results support the advantages of the three proposed approaches in terms of sleeping ratio (number of OFDM frames in sleep mode/total number of frames) and bandwidth utilization. Comparing to the upper bound of sleeping ratio, their approaches suffer only 1.5% performance degradation.

In [21], Shiao-Li and You-Lin have proposed two energy-efficient packet scheduling algorithms for real-time connections in a Mobile WiMAX system. The proposed algorithms consider the QoS requirements and guarantee minimum power consumption for multiple mobile stations connections. The first scheme (called periodic on-off scheme) addresses power-saving class of type II by maximizing the length of a sleep period. They have proposed an algorithm to calculate the length of sleep and listen period according to the radio resources and the QoS constraints. The second scheme (called aperiodic on-off scheme) utilizes power-saving class of type III to support real-time connections. In this approach, they determine dynamically when the MS should go to sleep (on a frame basis) to maximize the length of the sleep period. Admission control is performed by the base station when a new connection is requested or a current connection is released. In these two cases, the base station will re-schedule the sleep mode operation based on the available resources. But if no connection is released or established by the mobile station, the base station will not re-schedule the resources, even if there are resources released by other mobile stations. Yet, no details of how to achieve this objective are discussed. Moreover, all MSs are assumed to have a predetermined set of connections. Simulation is used to evaluate the performance improvement achieved by the two proposed schemes over the traditional approach. The PS approach increases 80-120% more sleep periods than the traditional approach, while the AS approach gains about 15-120% more sleep periods over the traditional approach.

The research on the subject reported so far in the literature proposes no explicit complete solution designed to schedule UGS traffic with a minimum number of OFDM frames while meeting their QoS requirement, for the real-time connections that could randomly join or leave the network. This research addresses the above mentioned issues and provides a full solution that considers the UGS traffic requirement, in addition to call joining and releasing events and scheduling them in a way that improves the energy efficiency of the MS.

CHAPTER 3

RESEARCH IMPLEMENTATION

This chapter contains the original work done during the research period. The chapter starts by discussing the considerations and the main goals of the research. Then the proposed QoS architecture is broken down into its main module and each module is explained in details. The chapter states the modifications done on some work made by [14] and [15]. Finally, a detailed pseudo-code is provided for each of the QoS architecture module.

3.1 Introduction

WiMAX defines four types of service flows, each of them has different QoS requirements as shown in Table 1-2. IEEE 802.16 standard defines the UGS flows to support constant bit rate traffic. As these applications requires fixed bandwidth allocations which will not be changing with time, the bandwidth (BW) request from mobile station (MS) to base station (BS) is not required; instead, the BS will allocate a fixed number of time slots in each time frame. The IE field in the UL-MAP frame contains the number of time slots in which each connection can transmit during the uplink sub-frame. IE is sent by the BS through the UL-MAP. Upon receiving the UL-MAP, each connection will transmit data in the predefined time slots [2], [5].

This research addresses only the UGS traffic flows, and no packets scheduling algorithm will be provided for the rest of the traffic classes for the following reasons:

- rtPS and nrtPS traffics have a variable bandwidth requirement as can be seen in Table 1-2. If the network decides to guarantee only minimum rate, then more connections can be admitted to the network. If connections actually need more bandwidth, then packets of the admitted connection may encounter large delays, and it is possible that some packets will be lost because their deadline

could not be met by the network. On the other hand, if the network guarantees maximum rate, then this may result in wastage of resources when the real time connections need less than the maximum rate. To deal with this issue, a Bandwidth Estimator Agent (BEA) at the MAC layer was proposed in [15]. BEA observes the queue length of rtPS and nrtPS flow at regular interval and finds the bandwidth requirements of the flow by measuring the arrival rate of the traffic over the interval. The BEA sends a configurable BW request that is calculated as the ratio of change of queue length between current monitoring interval and previous monitoring interval. For simplicity, BEA is not applied in this research as it requires an additional calculation overhead which needs more processing power.

- BE traffic has no latency requirement to meet, and they can tolerate delay. Its packets can be scheduled in any listen period after the bandwidth is allocated to the higher priority flows (i.e., UGS,rtPS,nrtPS).

IEEE 802.16 MAC protocols is connection-oriented. In each transmission direction (uplink and downlink), the BS assigns each connection with a 16 bit unique connection ID (CID), Each CID is associated with a Service Flow ID (SFID) that determines the QoS parameters for that CID. WiMAX defines the connection signaling mechanism for information exchange between the BS and the MS. When a MS has a new connection to be established, it sends a connection request to the BS asking for its QoS needs. The BS admits or rejects this connection based on the available resources and replies back to the MS through connection response as shown in Figure 3.1. Once the connection is established between the BS and the MS, the MS sends a bandwidth request message to the BS. The connection can represent either an individual application or a group of applications sending data with the same CID [2].

In the existing QoS architecture of IEEE 802.16 as shown in Figure 3.1, the uplink packet scheduling located at the BS handles all the uplink packet transmission. Whereas the packet scheduler located at the MS retrieves the packets from different connections and transmits them in their pre-defined time slots as indicated in the UL-MAP sent by the BS. Connection classifier in Figure 3.1 differentiates connections based on their type which is determined by their CID, and then forwards them to their appropriate queue.

Call Admission control (CAC) mechanism that is responsible for accepting or rejecting connection according to the available QoS requirement is left undefined by the IEEE 802.16 standard. The standard also left the Uplink Packet Scheduling (UPS) algorithm undefined for all traffic except for the UGS. The UGS Uplink scheduling algorithm defined by the standard assigns periodically fixed-size data packets. These reserved grants are enough to send the packets generated from UGS connections. Before the connection establishment, the size and the length of the grant are first negotiated between the MS and the BS in the initialization process of the connection session. The traditional uplink scheduling algorithm defined by the WiMAX standard enables the BS to provide fixed allocations for UGS traffic without any additional signaling requirements [2].

In summary, the IEEE 802.16 standard defines the following:

(1) The signaling mechanism between the BS and the MS (i.e, connection request, connection response, BW Request, UL-MAP)

(2) The Uplink Packet Scheduling for UGS traffic.

(3) No Call Admission Control.

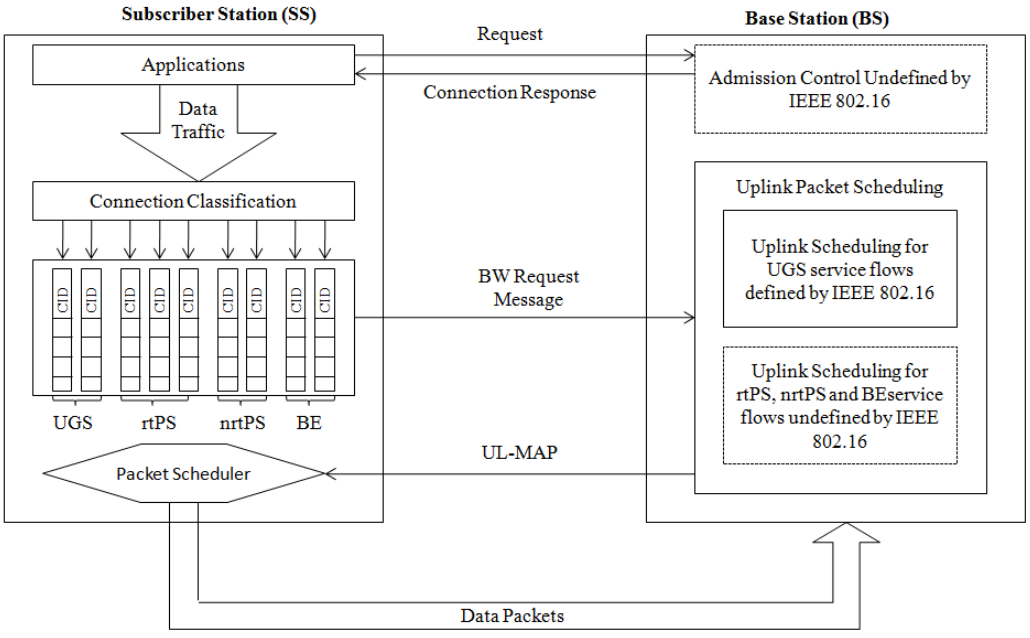


Figure 3.1: IEEE 802.16 QoS architecture [14].

Because power efficiency is one of the essential requirements of a broadband wireless access system, which is designed for portable and battery operated devices, IEEE 802.16e has defined three power saving classes as discussed in section 1.3.

Power Saving Class II is suitable for UGS flows and it allows the MS to repeat sleep and listen on a round robin basis. It also provides QoS guarantees of the connection in the MS, thus maintaining the task of uplink scheduling algorithm. Nevertheless, PSC II is not the optimized power consumption solution as can be observed in Figure 3.2. Figure 3.2 shows an example of a mobile station that has three connections. As all connections have fixed bandwidth requirements, PSC II is applied and the resulting schedule can be seen in the bottom of the figure. It can be seen in the figure that the periods that MS can sleep are determined by the sleep mode behavior associated by all connections. Clearly, without properly scheduling the sleep mode operation for multiple real-time connections on a MS, the power consumption might not be reduced, even though the sleep mode is applied.

In this research, QoS architecture is proposed with three main goals to address:

1. Provide QoS guarantee in terms of delay and bandwidth for multiple UGS connections at one MS.
2. Increase the sleeping periods of single MS by utilizing the Type-three power saving class (PSC III) defined in the IEEE 802.16e.
3. Apply efficient two-phase CAC that ensures that QoS guarantees are met.

The first and the second goals have been achieved by redefining the UPS that appears in [14] for UGS traffic only. The proposed UPS takes the connection requirements as inputs, and finds an optimal schedule (UL-MAP) which ensures the QoS requirements of all available connections while minimizing the number of allocated OFDM frames (awakened frames); hence, increasing the power saving efficiency. The third goal has been achieved by modifying the CAC proposed in [15].

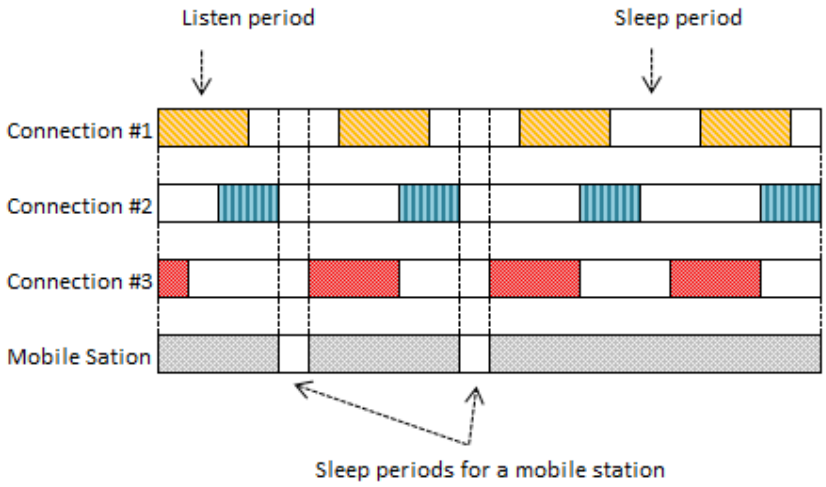


Figure 3.2: Sleep periods for a MS with three connections.

Figure 3.3 shows the block diagram of the proposed QoS architecture in this research. The blocks drawn with dotted lines are the parts undefined in the IEEE 802.16. At the BS we add a detailed description of the UPS module (scheduling algorithm which supports only UGS traffic flow), and CAC module. Connection Classifier is kept at the MS, although it is not needed as all connections are of the same traffic type (UGS). Here is a brief description of the connection establishment process using the proposed QoS architecture shown in Figure 3.3:

(1) A connection at the MS requests to join using connection signaling to the BS. The request has the traffic contract (bandwidth and delay requirement).
(2) The admission control module at the BS accepts or rejects the new connection based on the available bandwidth and delay constraints.
(3) If the admission control module accepts a new connection, it will notify the UPS module at the BS of the new connection requirement.
(4) The scheduling database at the UPS module will be updated with the new connection data.
(5) The service assignment module retrieves the information from the scheduling database module and generates the UL-MAP.
(6) The BS sends the UL-MAP to the MS.
(7) Finaly the scheduler at the MS transmits packets according to the UL-MAP received from the BS.

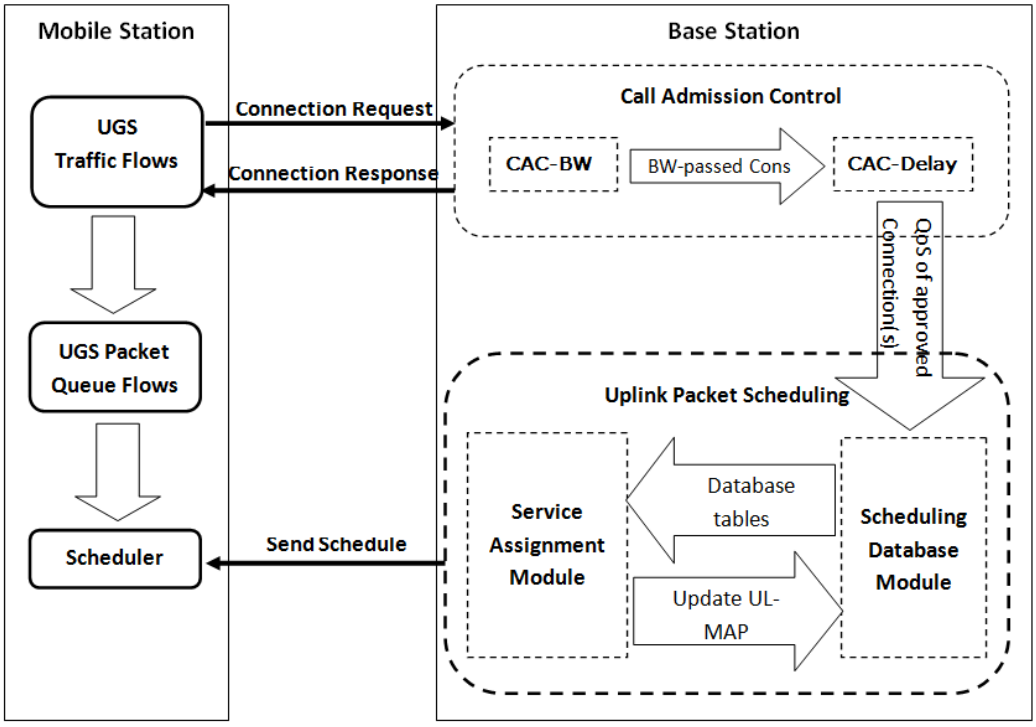


Figure 3.3: Proposed QoS architecture.

The rest of this chapter is organized as follows. The proposed Uplink Packet Scheduling is proposed and presented in section 3.2. The proposed Call admission Control is presented in section 3.3. Finally the pseudo-code algorithm of Uplink Packet Scheduling and Call Admission Control are described in section 3.4.

3.2 Proposed Uplink Packet Scheduling

In this research, a packet scheduling algorithm that provides QoS support for UGS traffic is proposed. The proposed algorithm minimizes the power consumption of a mobile station by utilizing the sleep mode feature of type III power saving class to save the energy of a mobile station with multiple UGS flows (such as VoIP connections) and to meet QoS requirements.

In a real-time environment, it is expected that calls will be joining and leaving the BS based on a certain arrival and holding time distribution. Let T_j defines the starting time of the jth OFDM frame. If at time, T_j, a call request gets granted by the BS, the schedule will be re-calculated starting from T_{j+1} (i.e., from the next frame) and up to the predefined schedule window. While at each time T_t a call session ends,

its requirement will be just removed from the current schedule and the corresponding scheduling databases will be updated accordingly. Some resources will be de-allocated after removing the terminated connection requirements and reallocating them might give a better sleeping ratio. However, re-scheduling will not be activated when a connection leaves the network, due to the following reasons: (1) the re-scheduling process costs time and power, and (2) in a moderate/heavy traffic network, calls request will be made more frequently and its mandatory to re-schedule every time a connection call gets accepted by BS. Therefore, the packet scheduling process will be only activated when a new connection call joins the BS.

In order to minimize the power consumption of a MS with multiple real-time connections, an optimum schedule needs to be followed by all connections. This schedule should guarantee both delay and bandwidth constraints specified by all connections. The delay can be satisfied by making sure that all packets will be transmitted to the BS before they expire, and this can be achieved by utilizing the UPS module. As new calls are expected to join, their delay requirements must be checked before accepting them to the system. This will be decided by the CAC at the BS. The CAC will make sure that there is enough bandwidth for any requested call before admitting it to the network.

The generated schedule (UL-MAP) determines if the MS needs to sleep or not on a frame basis. It tries to schedule all the available generated packets using the minimum number of OFDM frames without violating the QoS of all accepted connections. The schedule indicates the “ON-OFF” state of transmission frames. If the frame is fully or partially allocated then it is in “ON” (listen mode) state, otherwise, the frame is in “OFF” (sleep mode) state.

Every time the BS finds a schedule for the accepted connections, it assumes that those connections will remain in the system during the predefined schedule window (the length of the schedule in terms of OFDM frames). This is because the BS does not know when the next call request will be made by the MS. However, the predefined schedule window will be updated based on the current traffic status, as explained in section 3.2.2.

The proposed Uplink Packet Scheduling module in [14] consists of three main sub-modules; the information module, the database module, and the service

assignment module. The information module collects the queue size information of each connection from the BW-Request messages. It also determines the number of packets that arrived from rtPS in the previous time frame. Furthermore, it finds the arrival time and the deadline for rtPS packets, and passes queuing information from nrtPS and BE BW-Requests directly to scheduling database module. However, since the UGS traffics do not need to send BW-Request messages to the BS as their BW requirements are fixed, the information module is not needed for processing UGS connections.

The scheduling database module defined in [14] serves as the information database for the four different traffic types. For UGS traffic, each item in the database stores the packet size of each connection. For nrtPS and BE traffics, each item in their databases stores the queue size of each connection as provided by the information module. While for the rtPS database, it stores the packet size and the maximum delay for each connection. The service assignment module generates the UL-MAP (i.e., schedule). It uses the database tables created in the scheduling database module to determine the uplink sub-frame allocation in terms of the number of bits per connection or number of time slots. The service assignment module proposed in [14] applies strict priority service discipline for allocating bandwidth among different service flows; Earliest Deadline First (EDF) is applied to allocate bandwidth within rtPS, Weighted Fair Queue (WFQ) service, which allocates bandwidth based on the weight of the connection, is applied to allocate bandwidth within nrtPS. Strict priority service that allocates fixed bandwidth is applied within UGS connections. The remaining bandwidth is equally allocated to each BE connection.

This research modifies on the UPS proposed in [14] as the following:

First, the information module is not considered, as the UGS does not require it. Second, this research proposes a scheduling database module used to store items only for UGS flows. The proposed scheduling database module in this work saves logs of the parameters that come with accepted UGS connections. The parameters are the tolerated grant jitter (maximum tolerated delay), maximum rate (bandwidth requirement or the data burst size in terms of bits), and the nominal grant interval (*ngi*) which is the inter-packet arrival time. Furthermore, each requesting connection will be assigned a unique ID based on its arrival order. The scheduling database proposed in this research keeps track of all the generated packets and tries its best not

to drop any of the packets as a result of exceeding their delay constraints. Connection arrival time, termination, and average holding time are also stored in the scheduling database module. Third, the service assignment module proposed in this research has the scheduling algorithm. The scheduling algorithm is responsible for generating the schedule using the data stored in the database scheduling module. The proposed scheduling attempts to fit the available packets in the minimum number of OFDM frames before they expire. When the service assignment module calculates the UL-MAP or the schedule, it considers the different packets parameters such as its maximum tolerated delay and size. Moreover, it does not waste unnecessary frames as long as the main goals are maintained. A detailed pseudo code is provided in section 3.4 which explains how the service assignment module processes different information databases to generate a schedule.

If the call request gets approved by the CAC, its requirement will be forwarded to the UPS. The scheduling database module in the UPS adds the new call's data to the current database in order to help the Service assignment module in creating an optimized schedule in which both power saving and QoS support are considered.

3.2.1 Scheduling Database Module

The scheduling database module is a per connection database. It keeps a record of following:

1. The VoIP traffic generates fixed-size packets every certain period of time, called the inter-packet arrival time or *ngi*. For example, if *ngi* for a certain connection equals 4, this means that every four frames, one packet will be generated. For each frame, in a window equals to the schedule length, the schedule needs to know all the available packets and their sizes (based on their rates) and their related delay requirements to decide on transmitting them. A two-dimensional database array is used to store the pattern for each registered connection denoted as "$Cons\_Pttrn$". This pattern shows at which frame number, each connection is expected to generate its packets. Bandwidth requirement is presented by a packet burst entity denoted as Pck_Brst_i (where i is the connection ID). Pck_Brst_i will appear in the *Cons_Pttrn* log only if the connection C_i generates a packet at frame F_j.

Figure 3.4 shows an example of *Cons_Pttrn* log in which three connections are available. C_1 requests to join at F_0, C_2 requests to join at F_1, and C_3 requests to join at F_2. If connections are accepted, they are expected to join after one frame from the time they have requested to join. The *ngi* is 6,2, and 4 for C_1, C_2 and C_3 respectively.

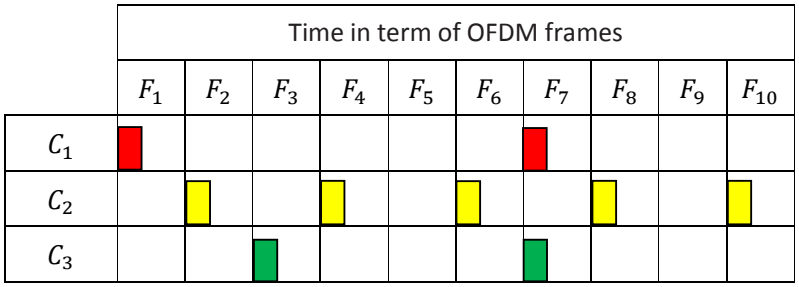


Figure 3.4: *Cons_Pttrn* log for three registered connections at the MS.

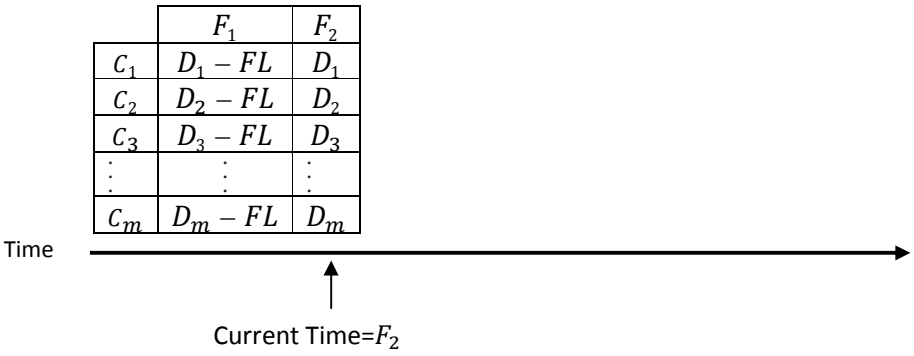
2. For each connection C_i, (i=1,2,3,…n; i is the arrival order), the scheduling database module keeps track of its generated packets (or what can also be called "data bursts"). Each connection C_i, has a QoS delay, denoted as D_i. The same Delay value D_i is applied to all packets $Cons\_Pttrn_{i,j}$ generated from connection C_i. Since VoIP is not tolerant of packet loss, each packet is dealt with and traced as individual. Therefore, the scheduling database module saves the delay value in each OFDM frame F_j for each packet $Cons\_Pttrn_{i,j}$ in a two-dimensional array, *QoS_Delay*, as shown in Figure 3.5. Once $Cons\_Pttrn_{i,j}$ is transmitted from the MS, its corresponding entity will be removed from the *QoS_Delay* array. As the time (in terms of OFDM frames) progresses, $QoS\_Delay_{i,j}$ for $Cons\_Pttrn_{i,j}$ is decremented by the OFDM frame length, *FL*, until it is sent to the BS. In other words, the packet can be sent to the BS station when it reaches its maximum tolerated delay or it can be sent earlier. Below is an example of how the *QoS_Delay* log is created by the scheduling database module. Assuming there are 'm' accepted connections in the network. The scheduling database module will build the *QoS_Delay* log from the maximum tolerated delay parameters included in each connection traffic contract.

Figure 3.5(a) shows the process of building the *QoS_Delay* Log for the packets generated at frame F_1. It is to be noted that this process is an offline process. It is performed in the BS along with other processes to generate an offline schedule. This offline schedule will be sent to the MS to start the real transmission, then the accepted connections will follow the schedule when sending their packets. The "Time" variable used in figure below is just a variable that is used to predict the values of packet delay as if the actual time goes by. This is possible since the VoIP connections have fixed bandwidth and delay requirements.

The *QoS_Delay* Log grows with time and keeps decreasing the delay constraints of previously generated packets at each following frame. It also assigns the delay requirement for the new packets that just have been issued, as can be noticed in Figure 3.5(b).

Figure 3.5(c) shows the log and its contents at any time F_j. For example, suppose $Cons\_Pttrn_{1,2}$ is generated at the second OFDM frame and has its maximum tolerated delay equals to 40ms; i.e, D_1 =40 ms. If that packet was planned to be sent in the 7th frame F_7, then at frame F_3, D_1 will be decremented by the frame length (e.g.,5ms); $\text{QoS\_Delay}_{1,2}$ = 35ms at F_3. Similarly, atF_4, $\text{QoS\_Delay}_{1,2}$=30ms, and so on. At F_7, $\text{QoS\_Delay}_{1,2}$= 10ms. The delay value at any frame F_j can be given as, $\text{QoS\_Delay}_{i,j}$=D_i-*FL*(j-1).*

(a) *QoS_Delay* log at frame F_1.



(b) *QoS_Delay* log at frame F_2.

	Time in term of OFMD frames				
	F_1	F_2	F_3	⋯	F_j
C_1	$D_1 - FL * (j-1)$	$D_1 - FL * (j-2)$	$D_1 - FL * (j-3)$	⋯	D_1
C_2	$D_2 - FL * (j-1)$	$D_2 - FL * (j-2)$	$D_2 - FL * (j-3)$	⋯	D_2
C_3	$D_3 - FL * (j-1)$	$D_3 - FL * (j-2)$	$D_3 - FL * (j-3)$	⋯	D_3
C_m	$D_m - FL * (j-1)$	$D_m - FL * (j-2)$	$D_m - FL * (j-3)$	⋯	D_m

Time

(c) *QoS_Delay* log at frame F_j.

Current Time=F_j

Figure 3.5: *QoS_Delay* log for 'm' connections, log elements can be only positive integer values.

3. For each packet in $Cons\_Pttrn$ log , the scheduling database saves the frame number *'j'* at which $Cons\_Pttrn_{i,j}$ will be transmitted to the MS in a two-dimensional array *"Pck_Tr"*. The *Pck_Tr* log is the schedule produced by the service assignment module. BS will be sending it to MS so that the registered connections at the MS can send their packets at their predetermined frame

numbers. Figure 3.6 shows a typical Pck_Tr database. Each $Pck\_Tr_{i,j}$ entity in the log corresponds to the time at which $Cons_Pttrn_{i,j}$is planned to be transmitted. When the actual transmission starts, each generated packet will be either transmitted in its pre-planned frame number, or if new connection joining event occurs, the generated packet's pre-planned frame number in Pck_Tr log will be updated so that it can fit in the new schedule.

In Figure 3.6, $Tr_{i,j}$ corresponds to the time $Cons_Pttrn_{i,j}$will wait after jth frame (the time it was generated) until it is sent in its scheduled frame number. $Tr_{i,j}$ is determined by the service assignment module.

The BS generates an offline schedule assuming that all connections will last for the pre-determined schedule interval and that no connection will be joining or leaving in the interval. But any time, T_e (e:event), a connection may join or terminate before the predefined schedule window is reached. Joining or termination event means that the schedule sent by the MS needs to be adjusted from the time T_e onwards. This requires the MS to stop running the current schedule in order to send it to the BS and get back the required updates. In this case, the current schedule running on the MS has to be stopped, and a connection request needs to be sent to the BS that will decide on updating the running schedule. In case of new connection requests joining at time T_j (i.e., the jth frame), the CAC at the BS checks if it can accept the request. If the connection request gets approved, then the new connection requirement will be passed to uplink packet scheduling. The scheduling database module will add the new connection requirements to its database, and process it to the service assignment module. The service assignment module will produce new schedule starting from T_{j+1} (i.e., the (j+1)th frame) till the end of pre-determined schedule interval. On the other hand, if termination event occurs at time T_t, then the terminated connection's requirement will be deleted from the current schedule and from all the databases starting from T_{t+1} to the end of the schedule, and no new re-scheduling process will be activated. It is one of the service assignment module's responsibilities to update the old schedule and update the required data in the scheduling database modules.

	Time (in terms of OFDM frames)				
	F_1	F_2	F_3	…	F_j
C_1	$1 + \mathrm{Tr}_{1,1}$	$2 + \mathrm{Tr}_{1,2}$	$3 + \mathrm{Tr}_{1,3}$	…	$\mathrm{j} + \mathrm{Tr}_{1,\mathrm{j}}$
C_2	$1 + \mathrm{Tr}_{2,1}$	$2 + \mathrm{Tr}_{2,2}$	$3 + \mathrm{Tr}_{2,3}$	…	$\mathrm{j} + \mathrm{Tr}_{2,\mathrm{j}}$
C_3	$1 + \mathrm{Tr}_{3,1}$	$2 + \mathrm{Tr}_{3,2}$	$3 + \mathrm{Tr}_{3,3}$	…	$\mathrm{j} + \mathrm{Tr}_{3,\mathrm{j}}$
⋮	⋮	⋮	⋮	⋮	
C_m	$1 + \mathrm{Tr}_{\mathrm{n},1}$	$2 + \mathrm{Tr}_{\mathrm{n},2}$	$3 + \mathrm{Tr}_{\mathrm{n},3}$	…	$\mathrm{j} + \mathrm{Tr}_{\mathrm{n},\mathrm{j}}$

Figure 3.6: . Pck_Tr Database for 'n' connections.

3.2.2 Service Assignment Module

The service assignment module in the UPS uses the database information retrieved from the scheduling database to build its optimal schedule. The service assignment module determines the UL-MAP and outputs it to the MS. The connections at the MS will transmit their packets according to the UL-MAP received from the BS.

The service assignment module is designed such that it gives the highest priority to the packets with the tightest QoS delay constraints. It might decide on some packets to be sent just at their deadline if this can minimize the number of used (i.e. *"ON"*) OFDM frames; hence, saving more power at MS. UPS algorithm translates the bandwidth requirements of different connections into the appropriate number of slots.

The process of creating the optimal schedule starts by summing the sizes of available unscheduled packets. At anytime, if this sum is less than the bandwidth that can be supported by OFDM frame, and none of them is going to expire at the current OFDM frame (i.e they can tolerate more delay), then they will not be chosen to be scheduled at the current frame. In order to save power at the MS, we need to design a schedule that uses a minimum possible number of "awakened" (ON) OFDM frames. One possibility to achieve such goal is to delay the unscheduled packets without violating their delay constraints. This process will help in filling the awakened OFDM frames to its full capacity along with any newly available packets. However, each generated packet $Cons\_Pttrn_{i,j}$ has to be scheduled between the jth frame and (j+D_i)th frame (packet deadline) to satisfy the delay requirement of connectionC_i.

Each time service assignment module delays a packet, its delay constraint will be decreased by the OFDM frame length. However, if at any time the sum of available unscheduled packets sizes is added up to more than what the OFDM frame can actually handle, then the bandwidth becomes the bottleneck, and regardless of their delay requirements, the UPS will select some of them to be allocated at the current frame. However, if at any time the sum of available packets sizes can fully fill the current jth frame, they will be scheduled to be transmitted in that frame.

When the sum of unscheduled packets sizes overflows the OFDM frame, UPS starts with selecting the packets with the tightest delay requirement to be scheduled first. After selecting packets from the same delay priority level, the UPS algorithm moves to the next tighter priority and so on, until the OFDM frame is completely or almost filled.

In some cases, the packets of the same delay constraints cannot all fit in one OFDM frame. This requires the UPS to choose among those packets some to schedule in the current frame and others to delay. Because all packets share the same priority, the proposed service assignment module will look into their bandwidth requirements. If the overflow is caused by only one packet, then this packet will not be scheduled in the current frame; instead, it will be delayed to be scheduled at later frames. If more than one packets cause the overflow, then the schedule will look at all unscheduled packets bandwidth requirements that share the same level of delay constraints. When the majority of packets are of high bandwidth requirement, one high rate packet will be delayed; otherwise, one low rate packet will be chosen for later scheduling. Afterwards, the UPS checks the overflow again and sees if it is caused by again one packet or more, and the process is repeated. The UPS will keep on delaying packets until the rest of requested packets will not overflow the OFDM frame. In cases where packets have the same bandwidth or delay requirements, the highest priority will be assigned to the one who came first (connection C_i with the minimum arrival order $'i'$).

The UPS always tries its best to make sure that no packets will be dropped because of exceeding their maximum tolerated delays. However, in very rare cases some packets might miss their deadlines; even though those connections have already been approved by the CAC before joining the network, and their requirement have been checked in a pre-determined interval. Nevertheless, this happens because when the CAC decides to accept a connection, its decision is based upon fitting its packets

in a temporary schedule, and the only requirement to pass is that, in the pre-defined interval, each requested packet can get enough room to meet its delay requirement as explained in section 3.3. But in the actual scheduling process (after the connection receives the grant to join), minimizing the number of awakened (ON) frames is also considered because the fewer the number of ON frames, the less the power MS will consume. This tuning between trying to schedule the packets in the minimum number of OFDM frames and at the same time meet their delay requirements might result in losing some packets especially in heavy traffic environment.

Another factor that contributes in the deadline misses is the mixture of traffic rates (low and high-bandwidth requirement). In very rare cases, when the packets sizes with the same delay constraints overflows one OFDM frame, UPS needs to choose some packets to be scheduled. On some occasions the selected packets do not completely fill the OFDM frame (i.e. there is a left-over); although there might be packets at the next level of delay priority with less strict delay constraint that could have filled this left-over. But once the UPS finds overflow in a certain delay level, it does not proceed to check from the levels. The main reasons for this logic are**:**

- First, in low to moderate loading traffic, these cases rarely happen in mixture of traffic environment.
- Second, going through different level of delay constraints costs further calculations and additional processing time. It also complicates the process of retrieving packets and scheduling them.
- Third, in cases where traffic are of the same rate, going through different levels of delay constraints will be unnecessary, since all packets have the same bandwidth requirements.

Because this schedule is designed for any traffic environment and for simplicity, the UPS will not go through multi-phases of checking once it finds an overflow at one level. However, when applying the proposed UPS for different rates, it has been found that the average deadline misses per connection is significantly low. In the worst case scenario (heavy loading high-rate traffic and tight QoS), the packet loss average is calculated to be approximately 0.004%; while the VoIP tolerates up to 1% packet loss.

BS uses the history of previous connection arrivals to decide on the length of the new schedule. The length of schedule window should not be too long so the BS will not overestimate when the next connection joining event will happen. This is important because a needlessly long schedule requires additional processing power and more time.

The pre-determined schedule length is not a fixed value. At the beginning, a random length will be assigned as the schedule length. Later, the length will be set to the average of the last n inter-call/connection arrivals (n is any integer number. In this research n is set to 10). If the schedule sent to the MS is finished and no new connection joining occurs, then the schedule will be expanded by the same previous length. The length of the schedule will be only recomputed after new 'n' connections requests.

When a new connection asks to join the network, the MS stops the running schedule and forward the joining request to the BS. The CAC module at the BS checks whether the incoming connection can be added to the network or not. When the connection gets approved to join the network, its QoS requirement will be passed to the scheduling database module, and then to the service assignment module which finds a new schedule of all available connection, in addition to the new connection request.

When a call terminates from the network, a termination request will be forwarded directly to the service assignment module without passing through the CAC. The service assignment module will remove the packets of the terminated connection from the time the connection is terminated, until the end of the current schedule. Finding a new schedule after removing the terminated connection packets might result in better sleeping periods. However, no scheduling procedure will be activated when a termination occurs unless a joining happens at the same frame, since it requires more calculations, and delay to find a new schedule. Furthermore, in moderate to heavy loading traffic, this calculation will be repeated unnecessarily, because the calls will be joining and leaving more frequently.

Figure 3.7 presents an example of how UPS finds a schedule for 'm' admitted connections and how the schedule gets updated when joining or termination occurs. At the jth frame, four connections are available at the MS and their requirements are

stored in scheduling database at the BS. It is assumed that each of C_1, C_2 and C_4 packet's burst occupies a 20% of the OFDM frame size. While C_3 is of higher rate and it occupies 40 % of the OFDM frame size. The delay constraint Di for C_1, C_2, C_3 and C_4 is 2, 2, 4, and 3 OFDM frames, respectively. The ngi for C_1, C_2, C_3 and C_4 is 4 OFDM frames.

Figure 3.7(a) shows patterns of all the available connections as they are stored in the $Cons\_Pttrn$ log. In each frame, and for each packet in the $Cons\_Pttrn$ log, the service assignment module retrieves its corresponding delay value from QoS_Delay log. Service assignment module uses the packets sizes and delay values information to decide on thier transmission time. The UPS algorithm computes the necessary number of OFDM frames to meet the QoS requirement of all connections in the minimum number of awakened frames.

Figure 3.7(b) presents the schedule produced by the service assignment module for the four resided connections. It can be noticed that although scheduled packets can tolerate more delay, they scheduled to be transmitted before their deadline expires because the sum of their sizes fill the OFDM frame.

Two events are presented in this example as the following:

- Connection Release Request:

Figure 3.7(b) demonstrates a connection termination event. At frame F_{j+4} , C_1 sends a termination request to the service assignment module at the BS. The service assignment module responses by deleting C_1 requirement from F_{j+5} and until the end of the current schedule. UPS also updates all the corresponding databases in the scheduling database module.

Figure 3.7(d) shows the schedule after removing C_1 requirement. Although new resources are released by C_1, but BS does not activate a new re-scheduling process until the next joining event.

- Connection Join Request:

In Figure 3.7(d), C_5 requests to join at frame F_{j+5}. After its QoS requirements were checked and accepted by the CAC to join the network, it joins the MS at F_{j+6} as can be seen in Figure 3.7(e). CAC module passes C_5's QoS parameters to the scheduling database module. The scheduling database module updates its logs with C_5

parameters, then it forwards the new updated tables to the service assignment module, which in turn finds new schedule considering all available connections from frame F_{j+6} to the pre-defined schedule length. The new schedule after adding C_5 is shown in Figure 3.7(f). It can be observed that two of the allocated frames in the new schedule are not fully utilized; this is because some packets have reached their deadlines and they cannot tolerate more delay.

A detailed pseudo code of the service assignment module is provided in section 3.3.

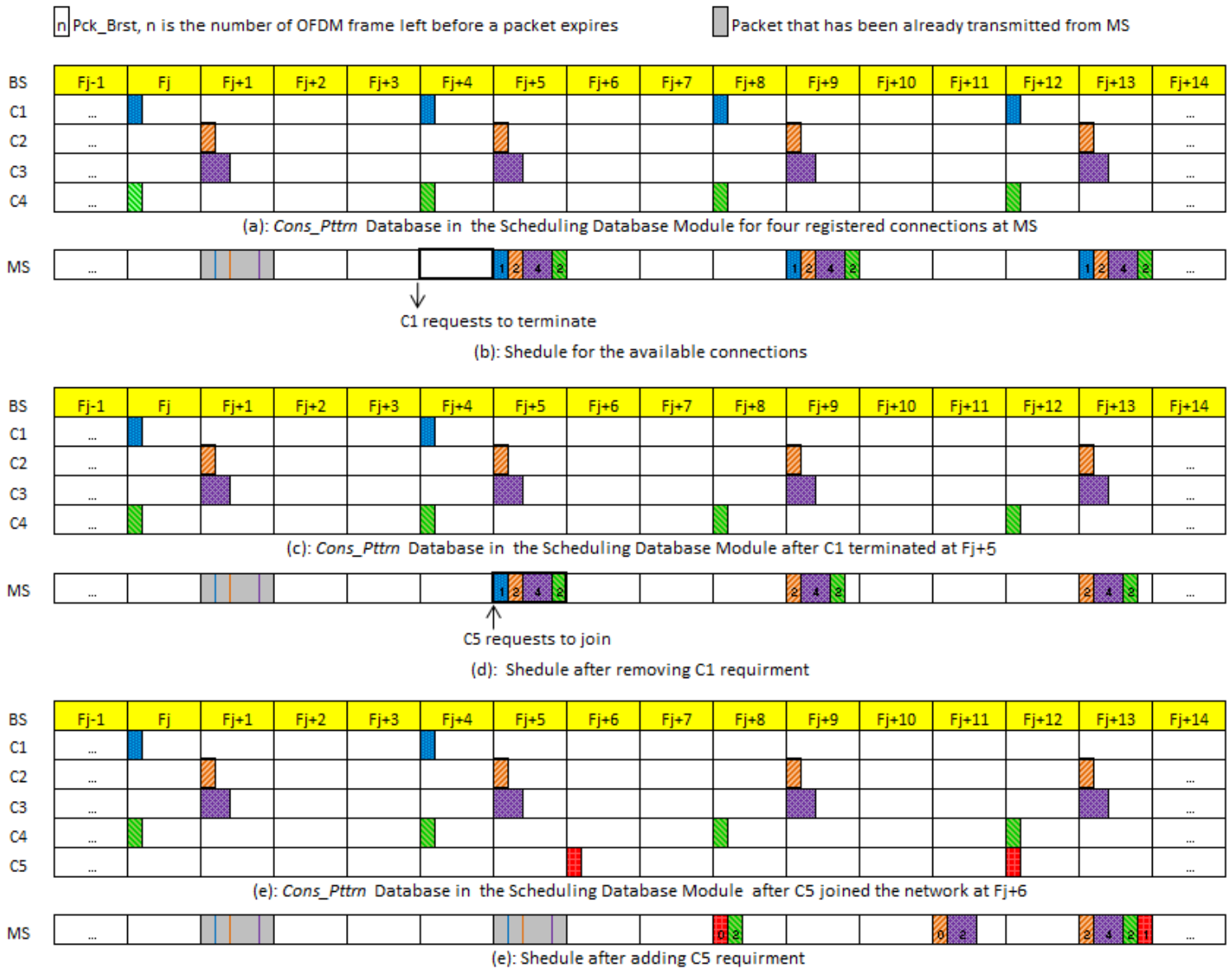


Figure 3.7:Processing a schedule at the Service Assignment Module using *Cons_Pttrn* log.

3.3 Call Admission Control

Call Admission Control running at the BS is the QoS mechanism that decides whether a new call can be established. This mechanism will ensure that existing calls' QoS will not be degraded and that the new call will be provided support for its QoS. The newly admitted connection will receive QoS guarantees in terms of both bandwidth and delay without degrading those of the existing connections.

In [15], the authors surveyed several papers for 802.16 systems in which a conventional bandwidth-based CAC (CAC-BW) is used to control the incoming flows. CAC-BW admits flows as long as there is enough bandwidth to satisfy the incoming requests, but it does not consider the deadline constraints of the connections. The work in [15] has proposed a QoS-CAC for the four traffic classes, and it accepts connection requests such that the QoS (BW and delay) guarantee is provided to all the accepted connections.

In this thesis work, the QoS-CAC module used by [15] is modified. In other words, the CAC module at the BS is divided into two sub-modules; CAC-BW sub-module and CAC-Delay sub-module. The CAC-BW is set as the primary phase for the admission of the incoming connections. Therefore, the incoming call will go through the CAC-BW module first; if it passes, it will be then forwarded to CAC-Delay to check whether its delay requirement can be met.

3.3.1 CAC-BW

For UGS traffic, when a new connection session at the MS wants to join and connect to the BS, a connection signaling starts. The connection request has the bandwidth requirement to be negotiated first with the BS to make sure that the connection BW requirement can fit within the supported network BW.

The CAC-BW makes sure that new connection will be admitted only if there is enough BW to satisfy its requirement. It does not consider the deadline constraints of the connections as this is going to be handled by the CAC-Delay. The CAC-Delay will be only activated to those connections which passed the BW test.

CAC-BW is an estimator adaptive function "BW_{Est}". Each time a new connection asks to join, the BW_{Est} calculates the current BW requirement "BW_{Crnt}" for all existing connections per OFDM frame, and finds the expected BW per OFDM

frame in case of accepting the new connection(s) request(s). The BW_{Est}function has the packet size and rate ($1/ngi$) as inputs of each existing connection, in addition to similar information about the newly requested connection(s). Moreover, CAC-BW can deal with multiple connections requested at the same time and decide which ones to accept and which ones to reject. In the case of receiving multiple connections at the same frame, it finds the expected BW in cumulative manner. It starts with BW requirement of connection that came first (connection C_i with smallest arriving order *i*). If the connection with the smallest *'i'* passes, its BW requirement will be added to the BW_{Est}function, and the CAC-BW will then check for the connection with the next smallest '*i*' and so on. Further details of the CAC-BW is described in section 3.4.

3.3.2 CAC-Delay

For UGS traffic type, the QoS-CAC proposed by [15] works as the following: (1) it makes sure that the number of requested slots within the connection's nominal grant interval is less than or equal to the total number of slots that can actually be handled within the tolerated grant jitter (*tgj*) based on bandwidth. (2) Once the first condition is satisfied, it checks if the requested slots are available within tolerated grant jitter (maximum tolerated delay) for each nominal grant interval (*ngi*) in a period equal to the Hyper-Interval. The Hyper-Interval (*HI*) is defined as follows.

$$HI^{UGS} = \forall_i LCM\ (\alpha_i), 1 \leq i \leq N^{UGS} \quad (3.1)$$

Let α_i: ngi for connection C_i, *LCM*: the Least Common Multiple, and N^{UGS}: the number of the available *UGS* connections.

The *LCM* is used in [15] for testing admissibility of connections. This makes sure that QoS requirements are met at every periodic interval of the corresponding service type. The schedule consists of 'on-off' transmission OFDM frames. For a fixed number of connections, it is expected that a pattern of on-off frames will periodically repeat each *'p'* frames. However, the *LCM* interval may not necessarily have a pattern of the allocated frames in the schedule as can be observed in Figure 3.8. Assuming that C_i passed the CAC-BW test and needs to check its delay requirement. There is a possibility that its requirement cannot be met in a window equals to the actual pattern length, which sometimes is longer than the *LCM,* even if it was able to fit within the *LCM* interval. In this case, the connection will be accepted

even though it should not be, which will result in losing some of the transmitting packets that were previously accepted by the CAC. Therefore, instead of using the *LCM* as a Hyper-Interval, we re-define the Hyper-Interval, denoted as *HI,* in which the requested connection's delay requirement is tested, needs to be large enough to show this pattern. The length of this cycle is obtained by finding the autocorrelation of the current schedule as follows.

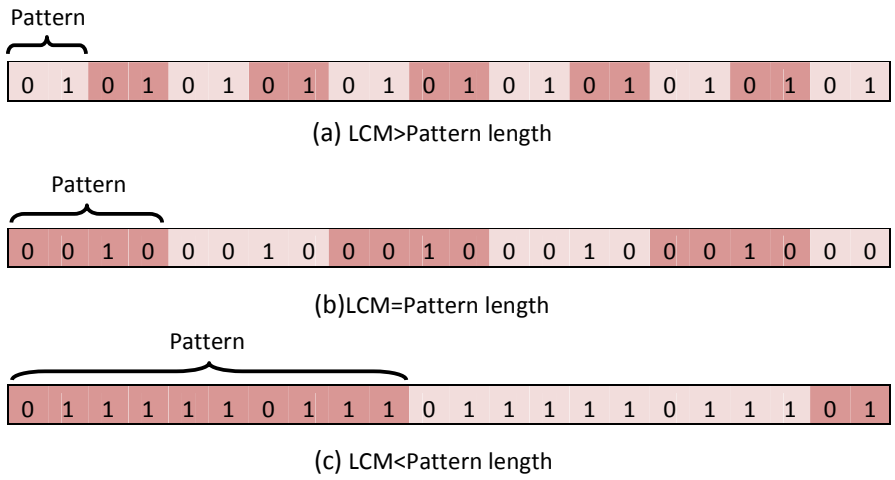


Figure 3.8: Studying the length of pattern in three schedules for different sets of connections. LCM in all cases is 4. A'0' in the schedule indicates that the current frame is in the "OFF" state, while a '1' indicates an "ON" frame.

Autocorrelations refers to the correlation of a function with itself. Correlation in itself is a measure of similarity. It is used for finding length of repeating patterns. It does so by finding the similarity between observations as a function of the time separation between them. The autocorrelation function is given in the equation below.

$$Y[k] = \sum_{k=0}^{N} X[n]X[n-k] \tag{3.2}$$

Where X is the schedule, Y is the correlation for k shifts, and N is the length of the current schedule in terms of OFDM frames.

In this research, the current schedule consisting of ON-OFF OFDM frames can be thought as number of patterns that periodically repeats each *'p'* number of frames. The length of a repeated pattern is the desired parameter for finding the period of the pattern.

Autocorrelation function is obtained by multiplying the signal with a shifted version of itself and summing or integrating the resulting signal. The value of the sum is considered as the similarity measure for that certain shift. The process repeats for all possible schedule shifts which are then stored in the autocorrelation function array.

Figure 3.9 provides good example of how the autocorrelation function finds the pattern. As shown in the figure, the pattern is found after 4 shifts (bottom part of the figure) and the resulted correlation value is the same value obtained when zero shifts is applied which means that the schedule is periodic.

Schedule	1	2	3	4	1	2	3	4	1	2	3	4
Shifted by 0	1	2	3	4	1	2	3	4	1	2	3	4
Schedule*Shifted Schedule	1	4	9	16	1	4	9	16	1	4	9	16
Corr. value at 0 shift	90											

Schedule	1	2	3	4	1	2	3	4	1	2	3	4
Shifted by 0	4	1	2	3	4	1	2	3	4	1	2	3
Schedule*Shifted Schedule	4	2	6	12	4	2	6	12	4	2	6	12
Corr. value at 1 shift	72											

Schedule	1	2	3	4	1	2	3	4	1	2	3	4
Shifted by 2	3	4	1	2	3	4	1	2	3	4	1	2
Schedule*Shifted Schedule	3	6	3	6	3	6	3	6	3	6	3	6
Corr. value at 2 shift	54											

Schedule	1	2	3	4	1	2	3	4	1	2	3	4
Shifted by 3	2	3	4	1	2	3	4	1	2	3	4	1
Schedule*Shifted Schedule	2	6	12	4	2	6	12	4	2	6	12	4
Corr. value at 3 shift	72											

Schedule	1	2	3	4	1	2	3	4	1	2	3	4
Shifted by 4	1	2	3	4	1	2	3	4	1	2	3	4
Schedule*Shifted Schedule	1	4	9	16	1	4	9	16	1	4	9	16
Corr. value at 4 shift	90											

Figure 3.9: Example of using the autocorrelation function to find pattern in the schedule. Note: Values used in the schedule are just for illustration purposes. They represent the sizes of the packet allocated at the jth frame.

This leads to the fact that the highest value of the correlation is maximum when k=0 (no shifts) $\sum_0^N X(n)X(n-k) \leq \sum_0^N (X(n))^2$, the number of shifts between two maxima (in this example 90 at 0 shifts, and 90 and 4 shifts) calculated by the autocorrelation function will be the desired parameter which indicates the pattern length.

The OFDM frames in the sleep mode are represented by zero in the schedule. Since 0 does not penalize for mismatches when multiplying the schedule with its

shifted version, we have replaced it by -1. Multiplying by -1 helps in reducing the similarity, and hence giving a better measure.

Figure 3.10 represents all the patterns for the values in Figure 3.9 above. The X axis represents the number of shifts (k), while the Y axis represents the correlation value corresponding to that k. The pattern appears each 4 frames which is the difference in k values between two consecutive peaks.

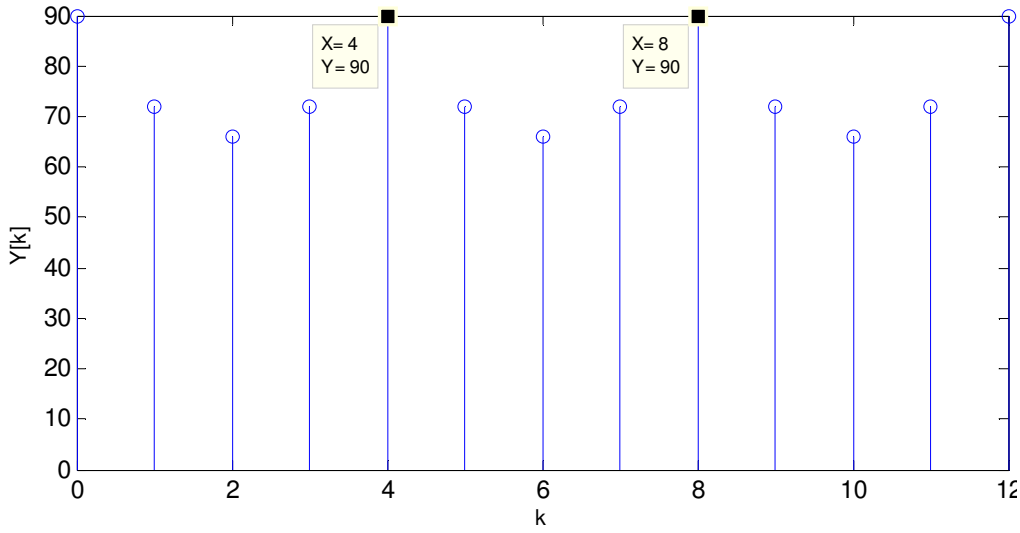


Figure 3.10: All patterns found for values in Figure 3.9.

After computing the *HI* interval that contain the pattern of ON_OFF frames in the schedule, the *HI* is then multiplied by the requested connection's nominal grant interval (*HI=HI*ngi).* This step will make sure that even in the rare cases where the *HI* is not long enough to test the requested connection's constraints, the other connections' constraints will not be violated. The *HI* length constraint is justified below.

After passing the CAC-BW constraints, the CAC-Delay tests if the connection requests delay constraints can fit within the current schedule without actually scheduling them. The proposed CAC-Delay works as follows:

First, when a new connection request is made, CAC-Delay finds the *HI* interval at which the schedule is repeated.

Second, the CAC-Delay algorithm checks if the current schedule length is long enough to cover the *HI.* If not, the current schedule will be expanded so it can

cover the *HI*. Expanding the current schedule is just to test the admissibility of connections, and it will not update any of the permanent databases.

Third, for each connection request, it is necessary to ensure that the required slots are available for each nominal grant interval "*ngi*" in a period equals to the Hyper-Interval. Within each nominal grant jitter "*ngj*" interval, the CAC-Delay searches for the requested slots in an interval between [Initial frame, Final frame], where the Initial frame is the first frame in the *ngj* interval, and the Final frame is the last frame in the *ngj* interval.

For simplicity, the allocation of slots is always assumed to be contiguous. Therefore, the number of bits generated from a certain connection is converted from slot unit to packet unit. The size of the generated packets varies depending on the UGS bit rate.

There are two different scenarios that can appear while trying to test the delay requirement of the incoming requests. The two scenarios are described as follows.

1. *ngi>ngj*: This has been implemented in [15]. Figure 3.11(a) represents this case where a new request arrives with *ngi* equals to 4 OFDM frames and *ngj* equals to 2 OFDM frames. The bottom part of the figure represents the pattern of the new requesting connection. Assuming that the connection request arrives at frame F_{Tj}, if it passes the CAC tests, then it is expected to join the network at frame F_{Tj+1}. For each ngi in the HI, CAC-Delay algorithm searches for enough slots to allocate the requested packet, such that it is assigned starting from its deadline to the left as can be seen in Figure 3.11(b). In our research, the changes made to the current schedule in Figure 3.11(b) are temporary, and this is not how the schedule will appear if the connection gets accepted, as was done in [15]. It should be noted that in [15], the power consideration was not an issue; the only issue was to deliver the packets before they expire. For example, in Figure 3.11(b). the second generated packet from the connection request is allocated in new unused frame F_{Tj+6}, although it could be handled in the previous frame F_{Tj+5} and keeping F_{Tj+6} in sleep mode, hence, saving more power. For this reason, we only use the QoS-CAC algorithm in [15] to estimate if the packets can fit in the current schedule considering their deadlines. After the connection receives the permission, it is

then the task of the UPS to place the packets in the appropriate frame locations which achieves minimal power consumption of the MS. If the UPS was involved at CAC stage of delay testing, then every time a connection request arrives, a real scheduling process needs to be established. This requires all the databases to get updated of the new request's data, and it is very possible that the requested connection cannot be served by the BS. Subsequently the schedule needs to be changed again by the UPS, and the updates in the databases as well. In other words, every time a connection request gets denied, many resources will be wasted. Consequently, the proposed CAC-Delay algorithm in this research is only used to check the delay requirements of the incoming calls and it does not actually schedule them, the real scheduling is done by UPS.

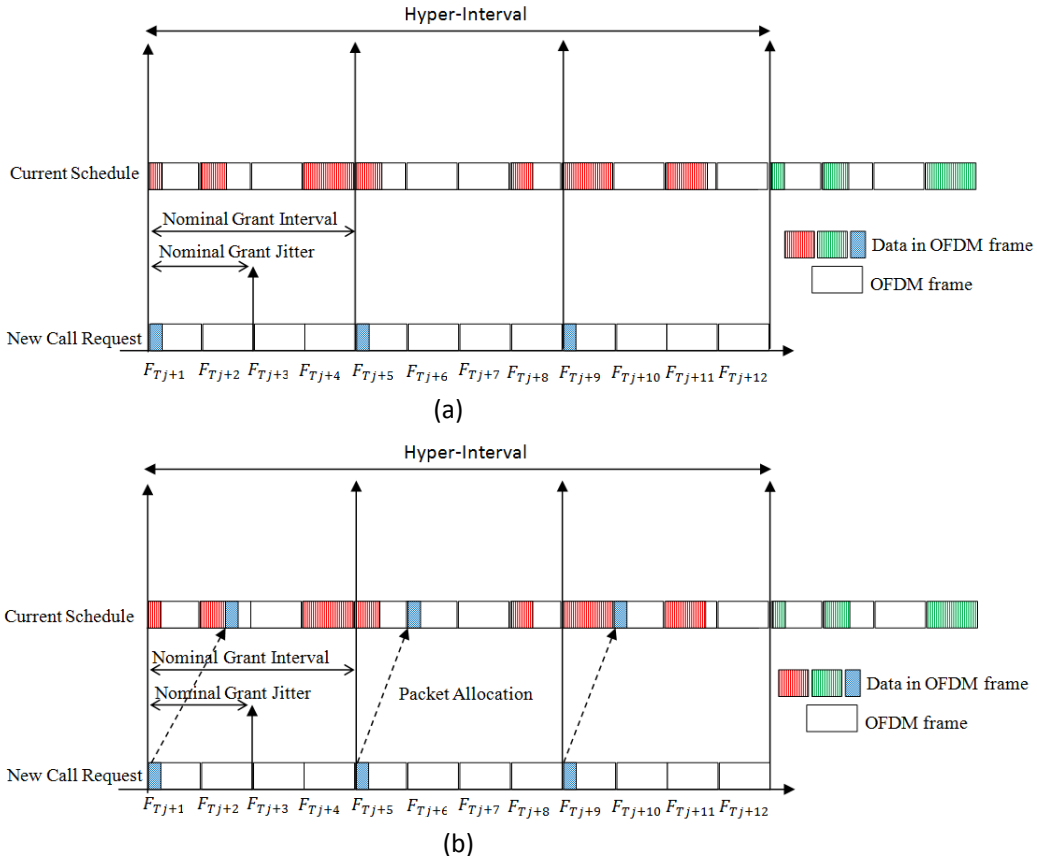


Figure 3.11: Example of checking the requirement for a new call request by CAC-Delay in case its ngi>ngj, the arrows indicate the frame chosen to serve the requested packets.

2. *ngi≤ngj:* With less strict QoS delay constraints, nominal grant jitter becomes greater than nominal grant interval. By trying to directly apply QoS-CAC used in [15], some packets might not find enough rooms to be allocated in. This

will result in rejecting the requested connection even though it should have been accepted. Figure 3.12(a) represents the case where a new request arrives with *ngi* equals to 4 OFDM frames and ngj equals to 6 OFDM frames. By applying the algorithm proposed in [15], the second generated packet as shown in Figure 3.12(b) cannot be allocated within its *ngj* because the first packet took the only room that was available. To deal with such situations, the proposed CAC-Delay algorithm is designed to check each ngi interval in the *HI* in backward manner. As shown in Figure 3.12(c), the third generated packet at frame F_{Tj+9} will be temporarily scheduled first, then the second packet, and then the first one. It is interesting to notice that the third packet was temporarily added outside the defined *HI*, as a result we make the *HI* a multiple of more than one pattern, i.e., *HI=HI*ngi* of the new call request.

The proposed CAC-Delay algorithm that runs in backward manner works properly for the above two scenarios. Therefore, instead of using two different approaches, only the proposed backward CAC-Delay is implemented in this research work.

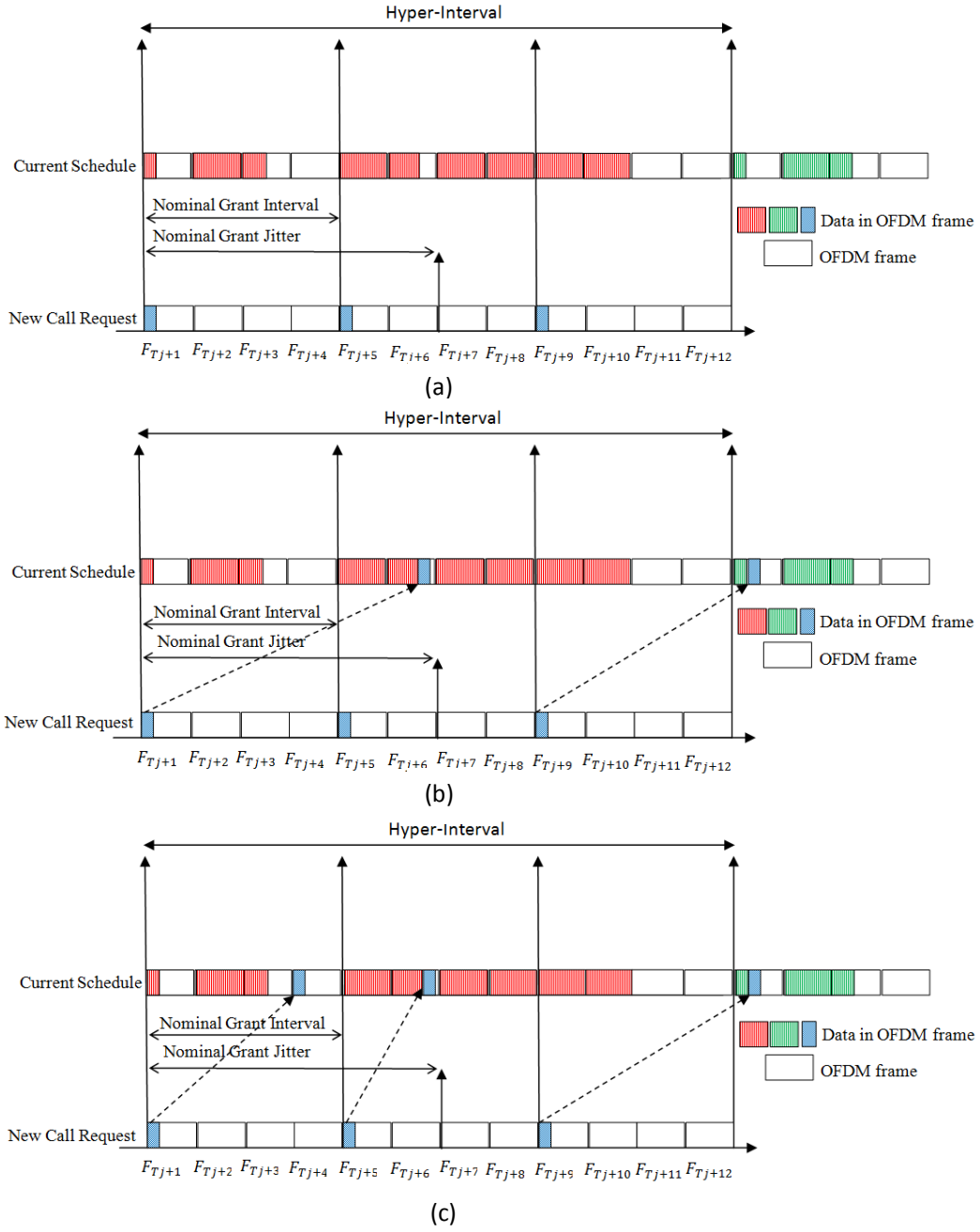


Figure 3.12: Example of checking the requirement for a call request by CAC-Delay in case its ngi≤ngj, using (b) forward approach implemented in [15] and (c) our proposed backward approach.

3.4 QoS Architecture Pseudo-codes

3.4.1 Pseudo-codes of Service Assignment Module

When a new call request is issued at T_j, where T_j is the time of the jth frame, the BS will find a new schedule one frame after the call joined the network(i.e., T_{j+1}), until the predetermined schedule length *"S_L"*. *S_L* is determined based on the last 10 inter-call arrivals.

In line 2, the actual scheduling process starts. *"Sum_Pck_Sz"* function sums the sizes of all available packets up to the jth frame (line 3). It has the inputs of *Cons_Pttrn* and *QoS_Delay* logs. *Cons_Pttrn* log is used to retrieve the generated packet burst sizes in each frame F_j , while *QoS_Delay* keeps the log of delay requirements of the unscheduled packets.

Next, the sum of unscheduled packets sizes along with their delay requirement is done, and all possible cases that could be encountered are taken care of.

If the sum of available packet sizes is less than what actually the OFDM frame can handle, but the delay value for one or more of the current unscheduled packet(s) reach '0' (line 4), this means that these packet(s) cannot wait more. In this case, packets with '0' delay constraint must be scheduled to be transmitted at the current jth frame, even though the jth frame will suffer from bandwidth wastage. Since the jth frame is *"ON"* anyway, then to make use of the frame bandwidth, all other available packets will be allocated in the jth frame, and these allocations at the jth frame will be saved in the *Pck_Tr* log (Line 5). Once a packet is scheduled, its corresponding data in the *QoS_Delay* log will be removed (Line 6). This step is needed because it helps to know which packet got scheduled and which did not.

Line 7 shows the second case which appears more frequently as the mean arrival rate increases. Up to the jth frame, if the sum of available packets sizes overflows the bandwidth that can be handled by one OFDM frame, then some of them need to be scheduled to be transmitted at the jth frame even though they could tolerate more delay. The available packets will be given priority to be scheduled at the jth frame based on their delay requirement. Packets from connections with the tightest delay requirement will be given the highest priority (Line 9) to be scheduled; the packets with the next tightest priority will be given the next higher priority and so on.

At any *R*, where *R* is the range of available packets' delay constraint, if the unscheduled packets can fully fill one OFDM frame (Line 11), then there is no need to delay them and they are scheduled to be sent in the current jth frame (line 12). If at some *R* delay value, the sum of available packets sizes is less than what can actually be handled by the OFDM frame, then packets from the next higher *R* levels will be added up and the new sum of packets sizes from both *R* levels will be checked, and so on. Sometimes, even delay-based prioritization will not help to overcome the overflow at the jth frame (Line 14). In this case, the packet(s) that cause the overflow will be located, and one or more will be scheduled later than the jth frame. If the overflow equals to one of the packets sizes (line 16) , then this packet will be delayed for later frames (Line 17). If more than one packets have that same size as the overflow, then the one with the larger arriving order *'i'* will be delayed. Otherwise, if the overflow cannot be solved by postponing only one packet (Line 20), the service assignment module will group the packets based on their bandwidth requirements, then it will select one from the largest group to be delayed (Line 22), then the next smallest packet and so on till the sum of packets sizes does not overflow the OFDM frame. If the overflow problem at any *R* is solved, then the service assignment module will not proceed to the next *R* level even if the jth frame is not fully allocated. If at *R*=0 (packets cannot wait more than the current jth frame to be scheduled) , and the sum of the available packets sizes overflows the OFDM frame supported bandwidth (Line 25), then some packets will be selected to be scheduled at the current jth frame, and the rest will be dropped because they have exceeded their deadlines.

At any time, regardless of the *Delay-QoS* status, if the unscheduled packet can fully fill the OFDM frame (Line 29), then it will be scheduled in the jth frame .

After the scheduling algorithm is done finding with a new schedule for the pre-determined schedule lengh *S_L*, *Pck_Tr* log will be forwarded to the MS. Connections at the MS will follow *Pck_Tr* log to know when to transmit their packets.

Algorithm 1: Scheduling Algorithm

{/*This Algorithm takes the traffic contract of all 'm' admitted connections as input, and generates an optimum schedule that meets the QoS of all connections in the minimum number of awakened OFDM frame as output*/}
1:{/*Determine the length of the schedule *"S_L"*/}
S_L= average of the last 10 inter- call arrival times

2:{/*When a call request arrives at T_j, it is expected to join the network at T_{j+1}. ; T_j :Time of the jth frame */}

For j=T_{j+1} to T_{j+1+S_L} **do**

3:{/*Find the sum of all available (unscheduled) packet sizes generated up to jth frame that have not been scheduled yet. $Cons\_Pttrn_{i,k}$ corresponds to the packet burst Pck_Brst_i generated from connections C_i at the kth frame. $QoS\_Delay_{i,k}$ corresponds to the delay constraint value of packet $Cons\_Pttrn_{i,k}$ */}

Sum_Pck_Sz (j)=$\sum_{k=Tj+1}^{j}\sum_{i=1}^{m}(Cons\_Pttrn_{i,k}|(QoS\_Delay_{i,k} \geq 0))$

4:{/*if any of the packet reaches its deadline, it is mandatory to schedule it in the jth frame. Other available unscheduled packets will be scheduled as well to utilize the jth frame bandwidth even if they tolerate more delay*/}

If (*Sum_Pck_Sz (j)*<= OFDM frame supported BW) & (one or more (*QoS_Delay i,j*==0))

5:{/* For each packet $Cons\_Pttrn_{i,k}$ that is selected to be scheduled at the jth frame , save a value of 'j' in its corresponding entity in Pck_Tr log. Pck_Tr log saves the time at which each generated packet $Cons\_Pttrn$ is planned to be transmitted in */}

Update Pck_Tr log

6:{/* Remove the delay requirement from *Delay_QoS* array of those packets which are selected to be scheduled at the jth frame */}

Update *Delay_QoS* log

7: **Else If** *Sum_Pck_Sz (j)*> OFDM frame supported BW

8:{/*Find the maximum delay value for the available packets*/}

Max_Delay=max(QoS_Delay)

9: **For** *R*=0 to *Max_Delay* **do** {/*R represents delay range*/}

10: {/*Find the sum of packets sizes that have the tightest delay requirements and about to expire first.*/}

Sum_Pck_Sz (j)=$\sum_{k=Tj+1}^{j}\sum_{i=1}^{m}(Cons\_Pttrn_{i,k}|QoS\_Delay_{i,k} == R)$

11:{/*if the sum of available unscheduled packet sizes (that have the same *'R'* delay constraint value) completely fill the OFDM frame, then those packets will be selected to be scheduled at the current jth frame*/}

If *Sum_Pck_Sz (j)*== OFDM frame supported BW

12: Update Pck_Tr log

13: Update *Delay_QoS* log

14:{/*If one or more packets of the same delay value *'R'* cause the Overflow*/}

Else If *Sum_Pck_Sz (j)*> OFDM frame supported BW

15: Find the size of the Overflow

16:{/*check if the overflow can be solved by postponing only one packet*/}

If (Overflow==any packet from $Cons\_Pttrn_{i,j}$)

17: Postpone the packet ($Cons\_Pttrn_{i,j}|Pck\_Brst_i$==Overflow) for later scheduling {/*Pck_Brst_i is the bandwidth requirement of connection C_i, and it is fixed for all $Cons\_Pttrn_{i,j}$*/}

18: Update Pck_Tr log

19: Update *Delay_QoS* log {/*exclude the postponed packet which caused the overflow*/}

20:{/* Else if more than one packet cause the overflow*/}
Else

21: Group all packets of same delay value *R* based on their sizes (bandwidth rates)

22: Postpone the packet ($Cons\_Pttrn_{i,j}|Pck\_Brst_i$ is from the largest group rate)

23: Go to line 15

24: **End if**

25:{/*If the sum of the available unscheduled packets overflows the OFDM frame supported bandwidth, and all packets have 0 value for *R* (they cannot wait more), then some packets will be selected to be scheduled at the jth frame, and the unscheduled ones are considered lost*/}
Else If (*Sum_Pck_Sz (j)*> OFDM frame supported BW) &(R==0)

26: Repeat lines 15 to 24

27: **End if**

28: **End For**

29:{/*If at any time, the sum of available unscheduled packet completely fill the OFDM frame, then those packets will be selected to be scheduled at the current jth frame */}
Else If *Sum_Pck_Sz (j)*==OFDM frame supported BW

30: Update Pck_Tr log

31: Update *Delay_QoS* log

32: **End If**

33: {/*Decrement the delay values by the OFDM frame length (*FL*) for all residing packets that were postponed for scheduling later than the jth frame.*/}
QoS_Delay=QoS_Delay-FL

34: **End For**

3.4.2 Pseudo Code of CAC-BW

CAC-BW sub-module in the CAC makes sure that the incoming calls are able to fit within the supported network bandwidth. First, the current network bandwidth per OFDM frame "BW_{Crnt} "is calculated (Line 1). "BW_{Est}"function will be initialized with this calculated value (Line 2). $BW_{Est}$is used to predict the network bandwidth in case of accepting the call request. For *'w'* connections which request to join at the jth frame, BW_{Est}function adds their BW requirement per OFDM frame one by one (Line 4). If adding one connection bandwidth requirement overflows the OFDM bandwidth (Line 5), it will be rejected (Line 6). However CAC-BW algorithm will keep on checking the bandwidth requirement for the rest of connections. Connection(s) that are able to pass the BW test will be forwarded to the next filtering stage (Delay- CAC Test).

Algorithm 2 : Call Admission Control- Bandwidth for UGS

{/*For each call request C_w, where *w* is the number of new call request(s) at time T_j; T_j :Time of the jth frame, this algorithm takes the packet size " Pck_Brst_w", and the packet rate "Pck_Rt_w" (1/ngi) parameters as inputs and gives BW_{Est}as an output if accepted*/}

1:{/*For the available *'m'* connections, find the BW requirement per OFDM frame. */}

$$BW_{Crnt} = \sum_{i=1}^{m} Pck\_Brst_i * Pck\_Rt_i$$

2:{/* Initialize the BW_{Est} with the current network BW*/}

$BW_{Est} = BW_{Crnt}$

3: **For** k=1 to *w* **do**

4: $BW_{Est} = BW_{Est} + Pck\_Brst_k * Pck\_Rt_k$

5: **If** BW_{Est}> OFDM frame supported bandwidth

6: C_k rejected

7: **Else**

8: $C_k$accepted

9: **End if**

10: **End for**

3.4.3 Pseudo Code of CAC-Delay

When connections 'v' pass the CAC-BW test (Line 1), their delay requirements need to be tested in the current schedule in a period equals to *HI*, in order to make sure that admitting them will not cause any deadline misses. For each connection C_v that passed the BW test, the number of ngi is found in a period equals to *HI* (Line 4). In each ngi, starting with the last ngi (Line 5), the process of searching starts from the packet's deadline to the left (Line 8). In Line 9, the algorithm checks for the possibility of fitting the generated packet in one of current schedule frames that are located in the ngj interval (between the Initial and Final frame) without causing an overflow. For each ngi, if the packet requirement is met, then it will be added to a temporarily schedule "Temp_Schedule" (Line 10). But if in any ngi, the packet generated from connection C_v could not fit within its tolerated ngj, the connection will be rejected (Line 13) and the searching process will be resumed to check for the next connection constraint and so on. The connections that pass the CAC-Delay test will receive a reply from the BS to join the network, and will be forwarded to the UPS.

Algorithm 3 : Call Admission Control- Delay for UGS

{/*This algorithm takes QoS parameters of 'v' connection(s) that passed through the BW-Test , and checks if their QoS delay constraint can be met in order to make the final admission decision from time T_{j+1}; T_j :Time of the jth frame, until the end of the Hyper Interval */}

{/*Enter the loop only if one or more connection passed from CAC-BW*/}

1: **If** v>0

2: **For** i=1 to v **do**

3: Find *HI* {/**HI* is the Hyper_Interval*/}

4:{/*Find the number of ngi intervals in the HI*/}
$No\_of\_ngi = HI/ngi_i$

5:{/*Checks in each in every ngi within HI*/}
For K= *No_of_ngi* down to 1 **do**

6: Initial_Frame= T_{j+1} *K; {T_j :Time of the jth frame }

7: Final_Frame= Initial_Frame + ngi_i

8:{/* Check for the availability of the requested packet's size within tolerated grant jitter (maximum tolerated delay) starting from the packet's deadline to the left*/}
For L= Final_Frame down to Initial_Frame **do**

9:{/* Check for the possibility of fitting the requested packet in the Lth frame at Current_Schedule, Current_Schedule represents the status of OFDM frames from time T_j, until the end of the pre-determined schedule lengh. L represents the frame number, Current_Schedule(L) give information about the bandwidth left in the Lth frame. */}

If (Current_Schedule(L)+ Pck_Brst_i) ≤ OFDM frame supported bandwidth

10: Temp_Schedule(L)= Current_Schedule(L)+ Pck_Brst_i;

11: **Else**

12: Overflow!

13: C_v rejected;

14: **Break**;

15: **End if**

16: **End for**

17: **End for**

18: **End for**

19: **Else**

20: **Break;**

21: **End if**

CHAPTER 4

SIMULATION RESULTS

In order to validate and evaluate the performance of the proposed QoS architecture (described in chapter 3), a simulation model was built in Matlab. The first section of this chapter describes the environment setup and the simulation parameters chosen in the Matlab program for the proposed QoS architecture model. The second section presents two traffic-type sample scenarios. In the first scenario, the incoming calls are of the same codec type; while in the second scenario, the arriving calls are of different codec types. Different performance metrics are used to study the MS and network behavior for each scenario.

4.1 Environment Setup and Simulation Parameters

In this research, we consider a WiMAX environment with one MS connected to single BS. Multiple connections are assumed to be established at the MS. We have developed a simulation model in matlab that demonstrates that our proposed QoS architecture provides QoS support to UGS traffic applications, schedules packets in the minimum number of OFDM frames and performs call admission control.

Connection arrivals are assumed to occur at the beginning of the frame. The connection arrival process for all connections follows Poisson distribution with a certain mean arrival rate (λ). The call duration follows an Exponential distribution process with a given average holding time. Each connection has specific QoS parameters in terms of (1) bandwidth requirement which is equal to the data burst size, (2) maximum delay requirement (*ngj*), and (3) nominal grant interval (*ngi*).

Four different types of codecs for VoIP are used in the simulation for UGS traffic flows. These codecs are selected by a random function. When mixture of traffic is applied, each of these four codecs is assigned the same probability when selected. Table (4-1) shows the parameters of the VoIP codecs used in the simulation.

Table 4-1: VoIP codecs parameters [22].

Codec	Bandwidth Requirement /Bit Rate	Data Burst (Packet) Size (Byte)	Nominal Grant Interval (ms)
G.723.1	5.3 Kbps	20	30
G.728	16 Kbps	60	30
G.726	32 Kbps	80	20
G.711	64 Kbps	160	20

In this experiment we consider an OFDM frame of length 5ms, and a channel bandwidth of 5MHz. Assuming a 3:1 downlink-to-uplink bandwidth ratio, QPSK modulation scheme and ½ coding rate, the physical data rate for UL frame is 653 Kbps [2].

For simplicity, the number of slots required by each codec when transmitting their packets during each *ngi* interval is converted to frame percentage in the Matlab program. For example, G.711 codec has 160 byte to transmit each 20 ms [22]. Assuming each slot equals to 1 byte, the number of slots needed by G.711 every 20ms is 160 slots. The number of bits in 5ms UL frame is (653*5) bits, and hence the number of slots is (653*5)/8. Each 20ms the G.711 will occupy 160/((653*5)/8) which is approximately 40% of the UL frame. Similarly, the percentage of OFDM frame is found for the rest of codecs as shown in Table 4-2.

Table 4-2: Simulation details of VoIP codecs .

Connection	Description			
	Codec	Bandwidth Requirement /Bit Rate	Data Burst (Packet) Size (Percent of OFDM Frame)	Nominal Grant Interval (ngi) (OFDM Frames)
L1	G.723.1	5.3 Kbps	4%	6
L2	G.728	16 Kbps	15%	6
H1	G.726	32 Kbps	20%	4
H2	G.711	64 Kbps	40%	4

The average connection holding time is assumed to be 50 frames. The average life-time of the registered connections is selected to be small because connection releasing event will not activate the scheduling process until a new connection joining event occurs. The simulation duration is 250 frames and the simulation results are averaged from 50 trials. Results are found based on 95% confidence interval.

In the Matlab program, the length of simulation window, average holding time, λ, delay constraint, initial schedule length and codecs types are first initialized.

Each time a packet Pck_Brst is selected to be scheduled by the service assignment module at the frame F_j , its delay constraint value in the QoS_Delay array will be replaced by a null value. Then value of *'j'* will be stored in its corresponding location in the Pck_Tr array, and the jth frame in the current schedule is set to the listen mode *"ON".*

Following are the major steps of the proposed packet scheduling and admission control schemes in the simulation:

(1) A new call request arrives at the MS based on a pre-defined average arrival rate (λ). Poisson distribution function is used to determine the incoming call's arriving time. Each time a call request arrives, its arriving time will be stored. After 10 calls arrivals, the average of the inter-call arriving time is found. The length of the schedule is updated with this average.

(2) The new call is processed through two-step CAC module. Its delay requirement is checked in an interval equals to the Hyper Interval (*HI*).

   a. First, the requesting call bandwidth requirement is checked by the CAC-BW to decide whether it can be handled within the present network bandwidth. If it passes the BW-test, then the call request will

be forwarded to check its delay requirement next, otherwise, the call will be rejected.

b. For calls that passed the BW test, the CAC-Delay module tests the calls delay constraint in an interval equals to the *HI*. CAC-Delay makes sure that the current schedule length is long enough to show the *HI*.

(3) If the new call request passes the delay test, its QoS information (delay,ngi,BW) will be extracted from its request and stored in its corresponding arrays in the scheduling database module.

(4) A loop of predetermined schedule length will be started. At each frame F_j, the service assignment module starts to add up the sizes of all unscheduled available packets up to the current jth frame. At the jth frame, if:

a. The sum is less than what the OFDM frame can actually handle, but one or more packet has reached its deadline (i.e., its delay constraint value becomes '0'), then all the available packets will be selected to be scheduled at the jth frame.

b. The sum is larger than what the OFDM frame can actually handle and none of the packets has reached its deadline, then a loop of all delay constraints range will be initiated. The loop starts from '0' to the maximum delay constraint value among all available unscheduled packets. For each delay value, the size of packets will be added up. At each delay value, if:

i. The sum of packet sizes that share the same delay priority is equal to the OFDM frame supported bandwidth, then those packets will be assigned to be scheduled at the current jth frame.

ii. The sum of packet sizes that share the same level of delay priority is larger than the OFDM frame supported bandwidth, then the overflow is calculated.

1. If the overflow is equal to one of the available packets' size, then that packet will be excluded from getting scheduled at the jth frame. If more than one packets share the same size, the one who joined the network last will be delayed.

2. If there was no single packet that can solve the overflow problem, then more than one packets need to be postponed. Available packets will be classified based on their bandwidth requirement, and one packet from the larger group will be postponed for later frames. The overflow will be calculated again to check whether it can be solved by delaying one or more packets, and so on.

iii. The sum of packet sizes is more than the OFDM frame supported bandwidth and all packets have reached their deadline (i.e., '0' delay value). This means some packet will miss their deadlines. In this case, some packets will be selected to be scheduled at the jth frame. Those packets will be chosen such that they can almost fill the OFDM frame. It is to be noted that, this case is rare, and it only happens when applying tight delay constraints along with high λs.

c. The sum is equal to the OFDM frame supported bandwidth. All available packets will be assigned to be scheduled at the jth frame.

d. The sum is less than the OFDM frame tolerated bandwidth and none of the packets has reached its deadline. The jth frame is set to the sleep mode (i.e., "OFF") and packets will be delayed to the future frames.

4.2 Simulation Results and Discussion

Two different traffic types scenarios are simulated. In the first scenario, only one codec type is studied, while in the second scenario, the four different codecs are studied.

The following performance metrics are studied: the bandwidth utilization, sleeping period, acceptance ratio, and rejection ratio (by the two CAC units) of the MS. Bandwidth Utilization percentage is the fraction of "ON" OFDM frame's capacity used by connections. **Bandwidth utilization** is an indicator of how much bandwidth was wasted in each allocated OFDM frame. **Sleeping period** percentage is the number of "OFF" OFDM frames over the total number of OFDM frames in a given time period. The duration the MS sleeps is proportional to the energy it saves.

For instance, if the MS has sleeping period percentage equals 60%, this means that the MS will switch off its hardware component 60% of the time in order to save energy. **Acceptance ratio** is the number of accepted calls over the total number of call requests in a given time period.

4.2.1 Scenario 1: One codec type

In the first scenario, all the connections use the same modulation scheme, G.726. The purpose of this scenario is to study the behavior of the network when applying different QoS delay constraints and mean arrival rates.

In the first simulation, the average bandwidth utilization (BW_U) under different mean arrival rates is investigated. The highest the bandwidth utilization percentage, the minimum is the wastage. Figure 4.1 shows the average bandwidth utilization when applying the following delay constraints (*ngj*): 10ms, 30ms, 50ms and 70ms. Under the same delay constraint, it can be noticed that higher λs scores higher bandwidth utilization percentage. The UPS keeps pushing the packets until they can fill one frame or until one of them reaches its deadline. When the number of packets wishing to be scheduled increases (i.e., λ increases), the delay constraint factor will have less control on deciding when to schedule, and the probability of allocating one frame with left-over reduces, which gives higher bandwidth utilization.

Under any delay constraint, the MS has its highest bandwidth utilization when λ is the maximum (λ =200 in Figure 4.1). On the other hand, when λ is minimum, the bandwidth utilization is the least under any delay constraint. In light-loading traffic situation (small λ), it is not expected that each allocated frame will be fully used. Hence, a great part of the allocated OFDM frames is wasted.

It is interesting to observe that under any λ, the bandwidth utilizations has smallest values when the delay constraints is the minimum (10ms). On the other hand, the bandwidth utilization increases as the delay constraints become loose (i.e., larger). The reason is that when the delay constraint is larger, the UPS will have more freedom to schedule the generated packets, and the likelihood to occupy a frame in the schedule to meet packets deadlines becomes smaller. While with tight delay constraint, if at least one packet is about to expire, the schedule will be forced to send all packets available at that time, even though they might not fill the entire frame, which results in bandwidth wastage.

However, it can be noticed that the QoS delay constraint has less effect as λ increases. The bandwidth utilization becomes saturated when λ icreases (i.e., when λ ≥100. This is because that at higher λs, the scheduling process will be mostly controlled by the network capacity (bandwidth). At any OFDM frame F_j , if some connections have packets that can be accommodated in the jth frame, then they will be scheduled in the jth frame. These scheduled packets are not necessary going to expire soon, the schedule sends them in order not to overload the network although they could wait more.

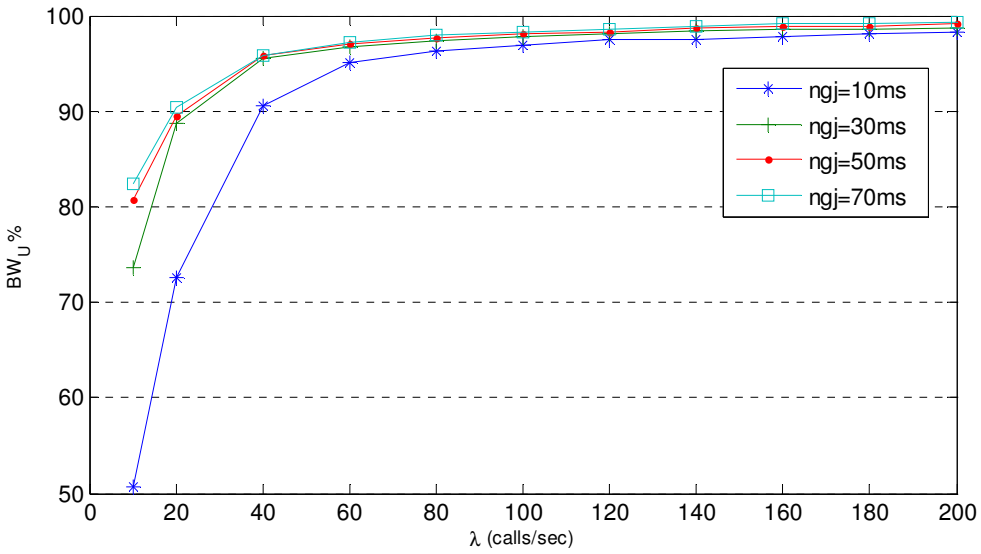


Figure 4.1: Percentage of Bandwidth Utilization for a mobile station with multiple VoIP connections for different λs and delay constraints (ngj).

Figure 4.2 presents the change in flow acceptance ratio as the average connection arrival rate changes. Flow acceptance ratio versus average arrival rate is found for four different delay constraints: 10ms, 30ms, 50ms and 70ms. Figure 4.2 demonstrates how the CAC behaves as λ increases. It can be noticed that the CAC rejects no calls when λ is very low, this is because the network can handle all the incoming calls. On the other hand, when the traffic load gets heavier the CAC cannot find the required slots to schedule the requested call, or the available slots cannot meet its QoS delay requirement in its *HI* interval, then the requested call will be rejected. As expected, when arrival rate increases, the acceptance ratio starts dropping much more rapidly.

Figure 4.2 reveals that under 30ms, 50ms and 70ms delay constraint, the flow acceptance ratio is almost 100% until the connection arrival rate is around 40. Thus,

under this environment setup, if it is desired to have a IEEE 802.16e network with no call rejection, then the network administrator should make sure that new connections arrive at a rate less than 40 calls per second.

The admission of new connection decision is controlled by the current network status and by the connection constraints. It is interesting to observe that under any traffic load the acceptance ratio always has its minimum value when delay constraint is the minimum (10ms). The acceptance decreases when the load gets heavier. In other words, tight delay requirements along with higher arriving rates adversely affect the admission of a new connection.

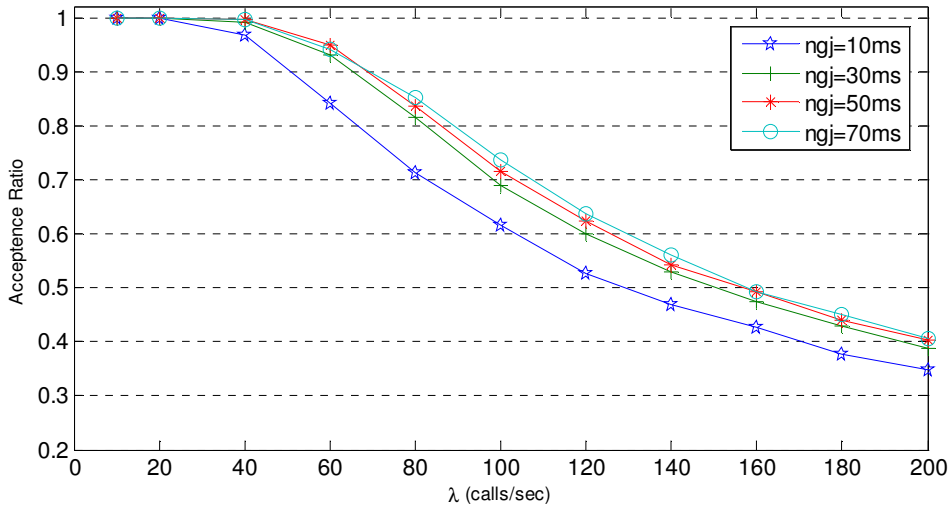


Figure 4.2: Acceptance Ration for a mobile station with multiple VoIP connections for different λs and delay constraints (ngj), for the proposed QoS architecture.

Figure 4.3 depicts the sleeping periods percentage when applying different λs under different delay constraints. As expected, the maximum sleeping periods are achieved when λ is the minimum, while the sleeping period percentage decreases as the arrival rate increases. Tight delay constrains (10ms) limit the number of calls accepted by the network as shown in Figure 4.2, hence, fewer awakened OFDM frames will be needed to schedule the accepted calls. On the other hand, when loose delay constrains are applied (70ms), more calls will be admitted to the network, and more awakened OFDM frames will be needed to accommodate the admitted calls, hence, less sleeping periods are obtained.

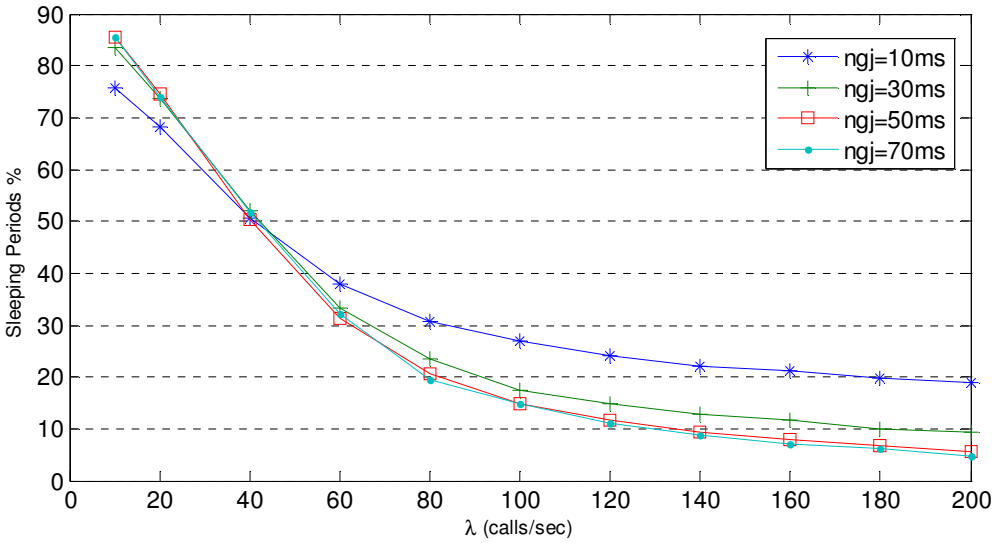


Figure 4.3: Percentage of sleeping periods for a mobile station with multiple VoIP connections for different λs and delay constraints (ngj), for the proposed QoS architecture.

In the next simulation, the percentages of sleeping periods of a mobile station under different delay constraints that G.711 connections can tolerate are investigated. Figure 4.4 shows the percentage of sleeping periods of a mobile station by applying the proposed approach. The delay constraints vary from 1 OFDM frame, that is 5ms, to 10 OFMD frame, that is 50ms.

It is interesting to observe that when $\lambda \leq 40$, the percentage of sleeping periods increases when loose delay constraints are applied. With lower λs, it is expected that the network will not be overloaded with the incoming calls. Applying loose delay constraints will give the UPS more flexibility in scheduling the packets in the minimum number of OFDM frames (by delaying them until the OFDM frame is almost fully allocated). However, the sleeping period percentage saturates when the delay constraint is larger than six OFDM frame (30 ms) because the network bandwidth constraint becomes the bottleneck.

On the other hand, when $\lambda > 40$, the percentage of sleep periods decreases when loose delay constraints are applied. When λ is higher, more calls are expected to join the network (as long as they do not violate the requirements of the existing connections).As can be seen in Figure 4.2, under any λ, the MS has the maximum acceptance ratio when loose delay constraints are applied. This means that more calls

can be admitted to the network and hence scheduling them will require more OFDM frames which means less sleeping periods.

From power saving point of view, if the delay constraints are loose, then the network administrator should make sure that calls arrive at a rate less than 40 calls per second. While for heavy traffic, tighter delay constraints need to be set in order to achieve better sleeping periods.

In [20], authors have applied very loose delay constraints (>150ms) and they have achieved 96-98% of sleeping period under relatively very low-loading traffic. While in [21], the percentage of sleeping periods is about 50-90% when the delay constraints vary from 5ms to 50ms under relatively low-loading traffic. In both works ([20] and [21]), the authors have considered multiple mobile stations and each station has a fixed number of connections. Our scheduling algorithm is more flexible as it deals with connections individually, and it offers good sleeping percentages (45-75%) under low-to-moderate traffic loading.

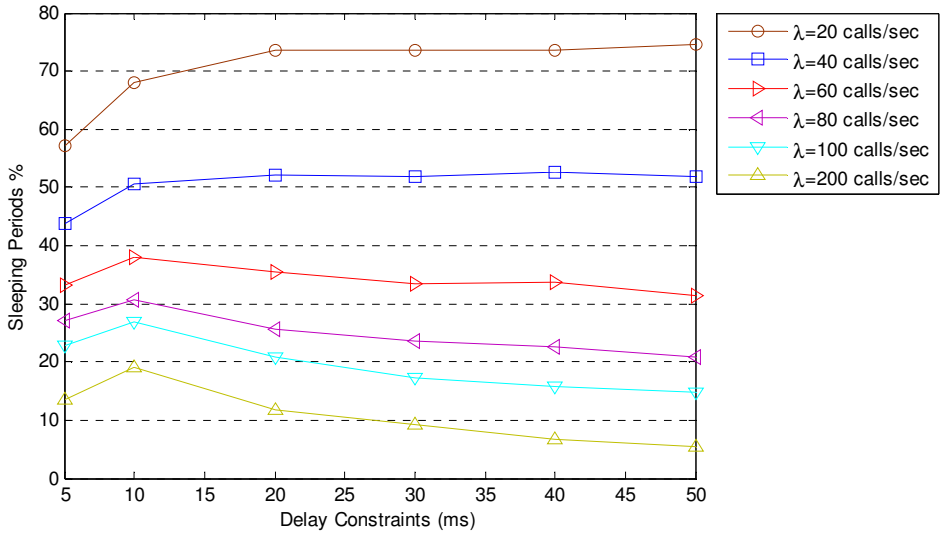


Figure 4.4: Percentage of Sleeping periods for a mobile station with multiple VoIP connections under different delay constraints (ngj).

In the proposed QoS architecture, each incoming connection is tested by two-phase Call Admission Control. In the first phase, its bandwidth requirement is checked to make sure that there is enough room in the current schedule to handle the new connection. For only those connections that passed the CAC-BW check, a second

phase of testing is performed on their delay constraints to verify whether their delay requirements can be met.

Figure 4.5 and Figure 4.6 show the rejection ratio caused by each CAC unit under 10ms and 70 ms delay constraints, respectively. When tight delay constraint (10ms) is applied, as illustrated in Figure 4.5, more calls are rejected by the CAC-Delay unit, although their bandwidth requirements are able to pass the CAC-BW test. In strict delay constrains case, CAC-Delay unit becomes the dominant in rejecting the incoming calls. On the other hand, when the delay constraints become loose (70ms) as in Figure 4.6, it is expected that more calls will receive grant to join the network. When the admitted calls increases, bandwidth becomes the bottleneck as the network will not be able to fit new calls in its schedule, hence the CAC-BW becomes the dominant in rejecting the incoming connection.

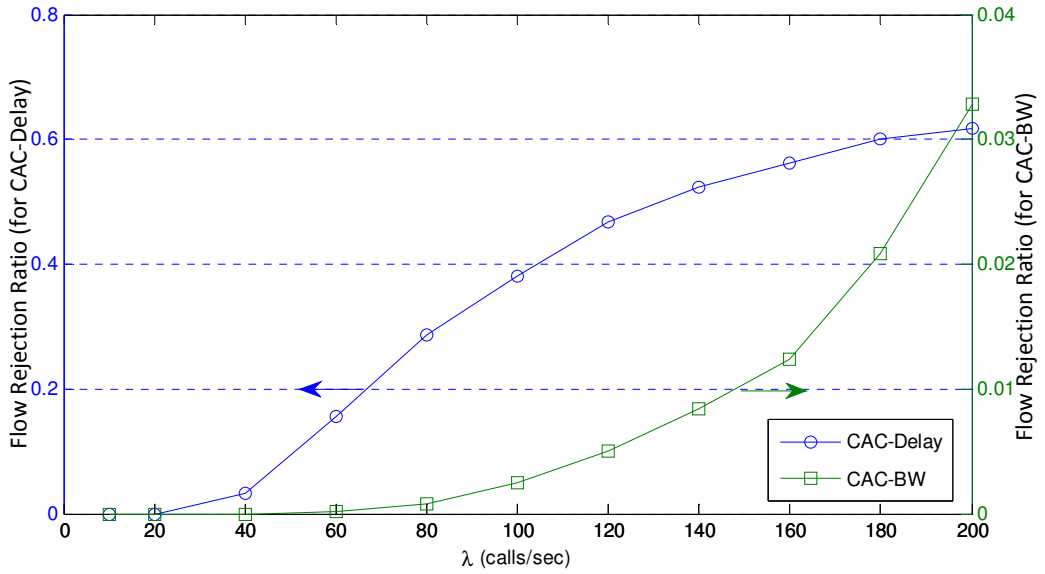


Figure 4.5: Flow Rejection Ratio of CAC-Delay and CAC-BW under different λs when applying tight delay constraints (ngj=10ms).

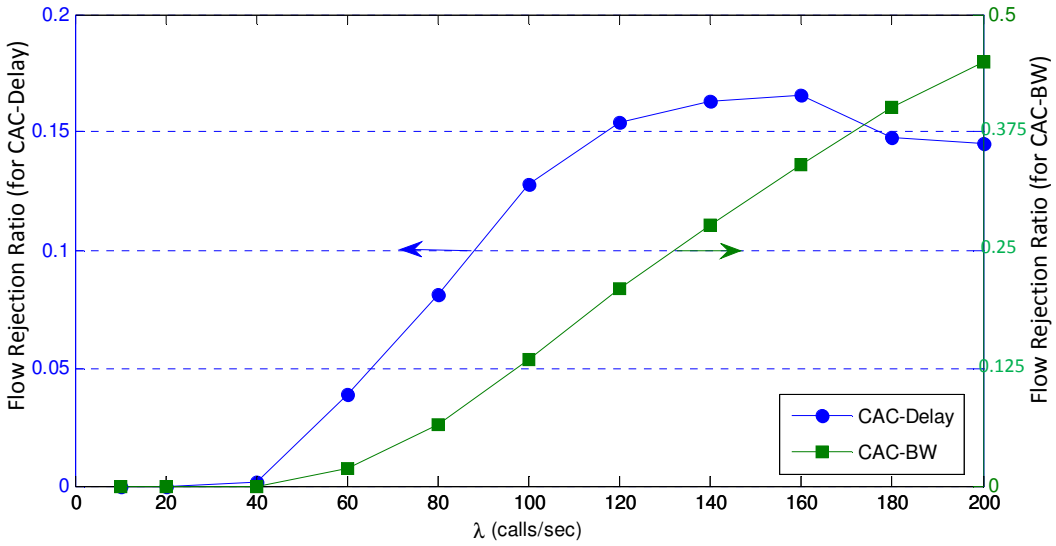


Figure 4.6 : Flow Rejection Ratio of CAC-Delay and CAC-BW under different λs when applying loose delay constraints (ngj=70ms).

4.2.2 Scenario 2: Mixture of codec types

In this scenario, the incoming traffic codec type received at the MS will be determined by a uniform distribution function. All codecs in Table 4.2 are equally probable to make a joining request. However, the actual joining of connection is dictated by CAC function ability to handle the incoming connection request at the time it requests to join. In this scenario, the acceptance ratio for each codec will be evaluated under different set of λs and delay constraints that are used in scenario 1. A comparison with scenario1 in terms of bandwidth utilization and sleeping period is also provided.

Figure 4.7 and Figure 4.8 show the acceptance ratio versus different λs per traffic codec when applying delay constraint of 10 ms and 70 ms, respectively. The acceptance ratio in this section is the total number of admitted connections from a certain codec over the total number of arriving connections. As explained earlier, when delay constraints become less strict, the calls acceptance ratio increases as long as they can fit in the available network resources. In Figure 4.7 and Figure 4.8, as λ increases, the lowest traffic rate (L1) codec scores the highest acceptance ratio because of its small bandwidth requirement that can fit without overloading the network. On the other hand, the highest traffic rate (H2) codec scores the minimum

acceptance ratio. As λ increases , it becomes more difficult for the network to handle calls with high-bandwidth requirements, and hence more high-rate calls will be rejected.

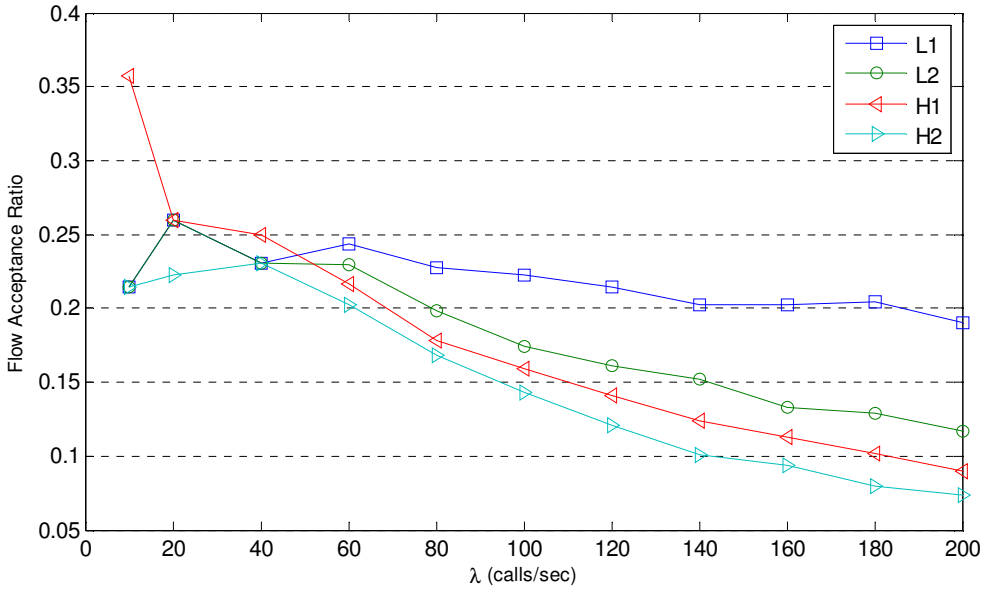


Figure 4.7: Flow Acceptance Ratio of CAC under different λs for each codec type when applying tigh delay constraints (ngj=10ms)

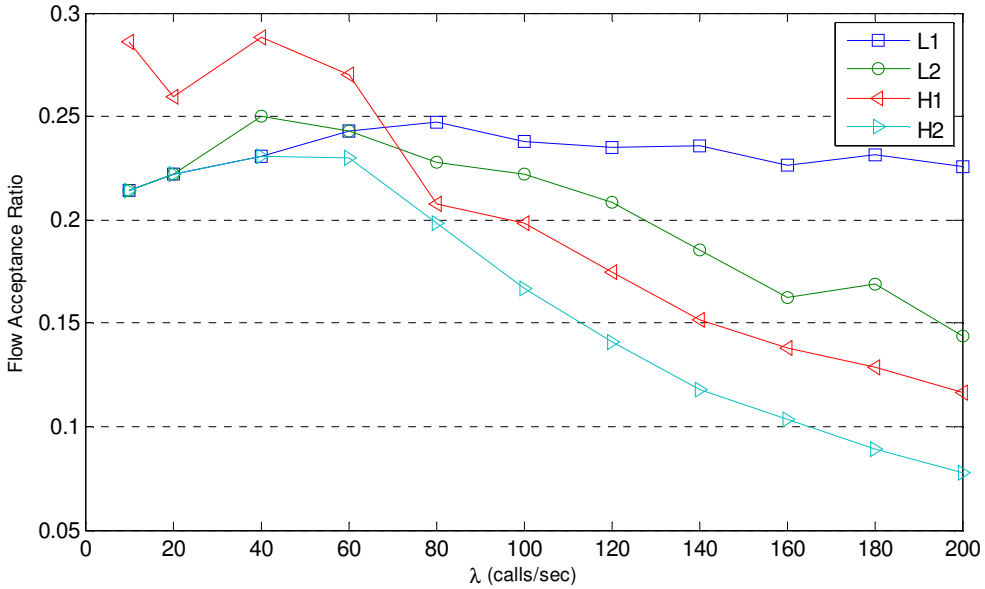


Figure 4.8: Flow Acceptance Ratio of CAC under different λs for each codec type when applying loose delay constraints (ngj=70ms).

Figure 4.9 presents the packet jitter (variation in the delay of received packets) for H1 and L1 traffic types in the second traffic-type scenario. It is to be noted that for small λs (10calls/sec), jitter increases with increasing delay constraints. When λ is small, UPS will be less controlled by the network bandwidth when scheduling the packets. This gives the UPS more freedom to schedule the packets based on their deadlines, which results in different delay variations, and hence larger jitter. On the other hand, when λ becomes larger (40 and 80), jitter value is almost fixed under any delay constraints because the scheduling decision will be heavily controlled by the bandwidth, and packet will suffer from less delay variations, hence less jitter. The maximum jitter was found to be less than 25 ms while the VoIP traffic can tolerate up to 50ms jitter.

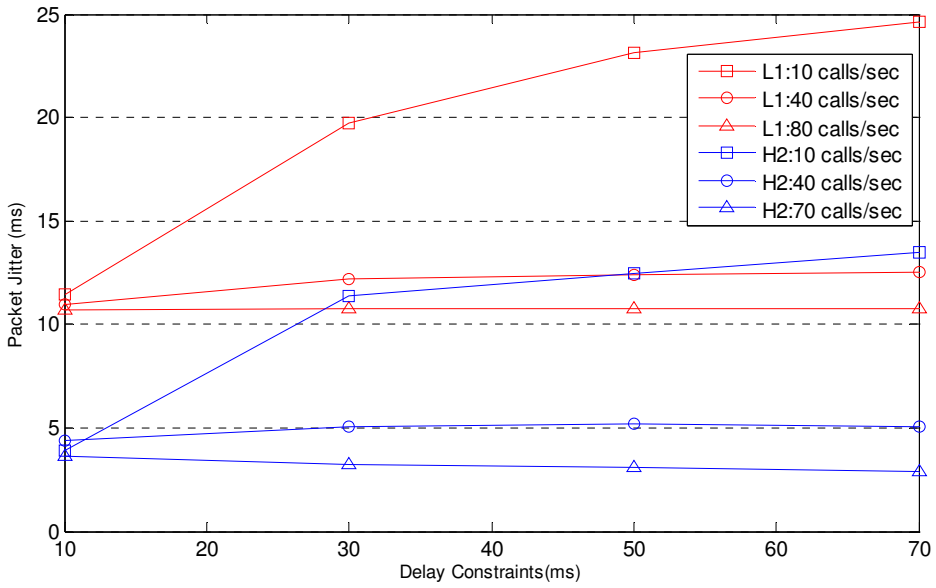


Figure 4.9: Packet Jitter for L1 and H2 connections under different delay constraints and different λs.

4.2.3 Comparison Among Three Traffic Configurations

Figure 4.10 provides a comparison in terms of bandwidth utilization between the three cases. Case 1 involves only H1 (moderate-rate) connections. Case 2 involves only H2 (high-rate) connections; while case 3, all codec types (L1,L2,H1,H2) are used in the simulation. The comparison is applied under two different delay constraints; 10ms and 70ms.

It can be observed from Figure 4.10 that under the two applied delay constraints, case 1 (H1 only) achieves the best bandwidth utilization paercentage, while case 2 (H2 only) achieves the worst percentage. In the scheduling process, the available unscheduled packets sizes are summed up to fill the frame, starting with the ones that have the minimum delay constraints. If at any delay constrain level, the sum overflows the frame, then the UPS tries to solve the overflow at the same delay level. In this mixture traffic (case 3), if the overflow is caused by a packet from higher data rate (e.g. H2), then it will be delayed for later scheduling, which might create a left-over in the scheduled frame. Although this left-over could have been filled by a lower rate packet available at the next delay level, the scheduling process does not proceed once the overflow problem is solved (in order to reduce complexity). Left-overs adversely affect the bandwidth utilization of the mobile station. However, those left-overs appear less frequently than case 1 (H1 only). In one-type traffic environment, left-overs cannot be filled from the next delay constraint level as all packets at that level share the same bandwidth requirement. Case 2 has larger left-overs than case 1 and 3 because the packet sizes are larger and they cannot fully fill the OFDM frame; hence, greater portions of the frames will be unused.

In can be noticed in Figure 4.10 that loose delay constraint under any λ scores higher bandwidth utilization. This is due to the fact that more calls are allowed to join the network, and hence, the OFDM frames are expected to have lesser left-overs, as explained earlier.

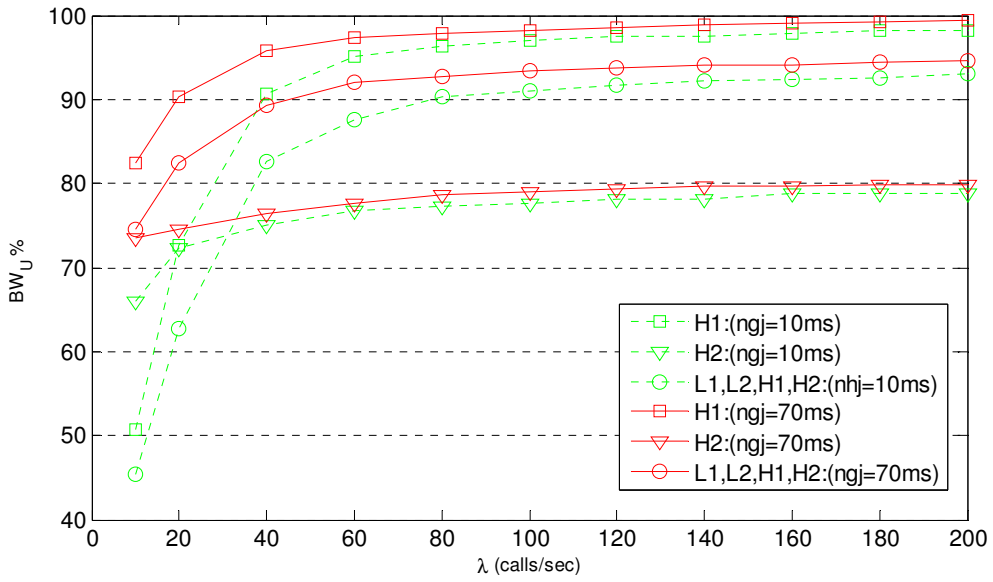


Figure 4.10: A comparison in terms of Bandwidth Utilization percentage under different λs for three traffic configurations. 10m and 40 delay constraints are applied.

Figure 4.11 provides a comparison in terms of sleeping periods between the three cases. Case 2 (H2 only) is expected to have the worst sleeping periods because of its high-rate connections .In interesting to observe that case 3 (where low to high-rate traffic exist) has a slight better sleeping periods than case 1 (where only moderate-rate traffic exist). Sleeping periods is inversely proportional to the bandwidth sum of the generated packets in the network. Higher the sum, larger the number of frames UPS needs to use for packets transmission. In the mixture traffic-type case (case 3), as shown in Figure 4.7 and Figure 4.8, the lowest traffic rate (L1) achieves the higher acceptation ratio, and the minimum acceptance ratio is achieved for the highest traffic rate (H2). Because of this, the bandwidth requirements sum of all connections in case 3 is less than the bandwidth requirements sum of connection in the first traffic type (case 1). Hence, case 3 requires less number of "ON" OFDM frames, which means more sleeping periods.

Figure 4.11 shows that loose delay constraints under any λ results in less sleeping periods because more calls are added to the network as explained earlier.

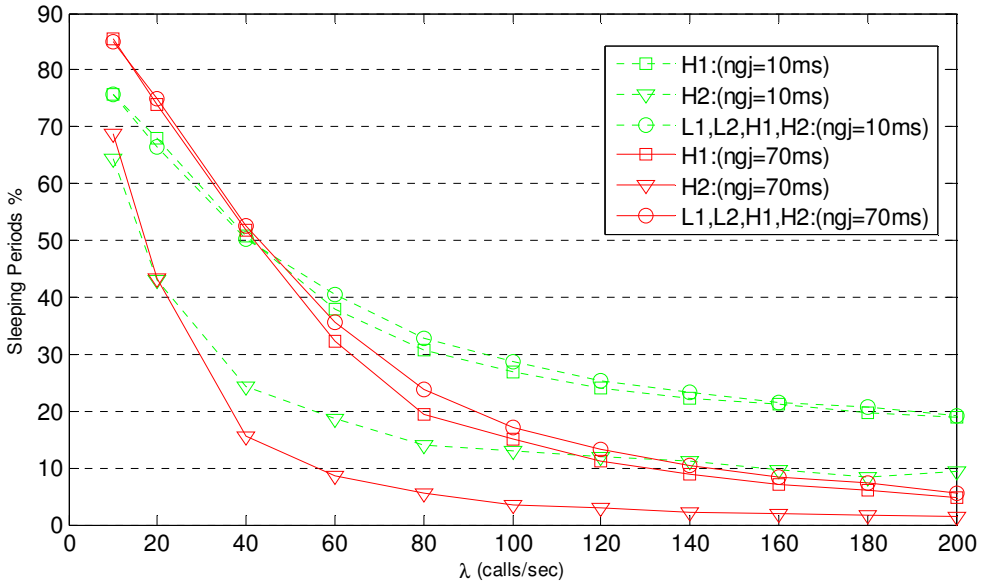


Figure 4.11: A comparison in terms of sleeping periods percentage under different λs for three traffic configurations. 10m and 40 delay constraints are applied.

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

5.1 Conclusions

Power saving function is one of the essential features for WiMAX network which is designed for portable and battery-operated mobile stations to operate for longer durations without having to recharge. Power saving is achieved by powering down parts of the MS when it does not transmit or receive data. When the MS turns itself off, it becomes unavailable and enters into the sleep mode. The main goal of this thesis is to reduce the power consumption of MS in WiMAX network by making the MS alternating between sleep and listening windows in a controlled manner. In this thesis, a new energy-aware QoS architecture at BS for IEEE 802.16e standard is proposed. The main contributions of this thesis are the following:

(1) Development of an efficient Uplink QoS Packet Scheduling algorithm for UGS traffic.

(2) Integrating the Packet Scheduling algorithm with two-phase Call Admission Control (BW and Delay) to handle the dynamic nature of calls as they join and leave the network randomly.

(3) Applying Power Saving Class of type III to extend the battery life of MS, and at the same time meet the QoS of all existing calls.

(4) Development of a simulation model to study and evaluate the performance issues in the new scheduling and CAC protocols.

We presented the performance of our QoS architecture for several different scenarios. The performance metrics studied in this work are: bandwidth utilization, sleeping periods, call acceptance ratio and packet jitter.

Simulation results show that with loose delay constraints and low to moderate traffic rates, good percentage of sleeping periods can be obtained. For example, with λ equals 20 calls/sec, and loose delay constraint (>40ms), the sleeping periods percentage of the MS is found to be approximately 75%. On the other hand, if more calls are accepted to the network, the percentage of sleeping periods drops to less than 50%, and it approaches zero as λ increases. The number of calls joining at MS and the improvement of the sleep time present tradeoff for the MS.

However, if the system traffic loading is heavy, tight delay constraints are advised to be set as to limit the number of accepted calls; hence giving better percentage of sleeping periods. But if loose delay constraints are set in heavy traffic scenarios, saving the battery consumption at a MS is not possible, as the MS has to be awake almost all the time. In this case, there is no need to implement a power saving scheme.

Bandwidth requirement also affects the sleeping periods. Simulation results show that high-rate connections achieve the minimum sleeping periods because they requires more OFDM frames to be "ON" to schedule their packets.

Under loose delay constraints, simulation results showed that the MS has its best bandwidth utilization (almost 100%) when the λ is light. The minimum bandwidth utilization (approximately 50%) is achieved when applying the tightest delay constraint (10ms) along with the lowest value of λ (10calls/sec). There is a tradeoff between the bandwidth utilization and the percentage of sleeping periods of the MS. With loose delay constraints and high λ, the bandwidth utilization scores its best percentage and the sleeping periods scores the worst percentage.

For the same values if λs and delay constraints, low-rate connections score the highest bandwidth utilization, because of their minimum bandwidth requirements that result in smaller left-overs in the used (*"ON"*) OFDM frames. When strict delay constraint (10ms) is applied in light-loading traffic (λ=10calls/sec), simulation results show that the MS has its worst bandwidth utilization (approximately 45% for mixed traffic). Tight delay constraint prevents the network from fully utilizing its used

OFDM frame because scheduling the frames is mostly controlled by the delay requirements.

Simulation results reveal that tight delay constraint limits the number of calls that can be handled by the network, while loose delay constraint allows the network to accept more calls based on the network resources. Under any traffic loading, the highest acceptance ratio is scored when loose delay constraints are set. Under any delay constraint, the acceptance ratio is 100% while the mean arrival rate (λ) is less than 40 calls/sec.

In the mixed traffic scenario with all traffic codecs having the same probability of making join requests, simulation results show that the lowest acceptance ratio is achieved for the highest-rate connections, while the highest acceptance ratio is achieved for the lowest-rate connections. This is because of the fact that low-bandwidth requirements can easily be accomodated without overloading the network resources.

Simulation results show that packet jitter has its highest value when loose delay constraint and low λs are applied. In this case, the packet jitter is found to be less than 25ms while the VoIP traffic can tolerate up to 50ms.

Packet loss is shown more frequently for higher values of λs along with very tight delay constraints. However, in this case, packet loss is found to be less than 0.004%. It should be noted that VoIP traffic tolerates up to 1% packet loss.

5.2 Recommendations for Future Work

One of the improvements that can be added to the proposed QoS architecture is to consider all different traffic types (UGS,rtPS,nrtPS,BE). Since rtPS and nrtPS have variable bandwidth requirement, then Bandwidth Agent Estimator (BAE) to monitor the queue length of rtPS and nrtPS traffic at regular intervals needs to be applied. This BAE will estimate the bandwidth requirement of the connection and subsequently make the appropriate bandwidth request for such connections.

Simulation results show that in cases where connections of high bandwidth requirement are applied, OFDM frames are likely to have left-overs. If only one UGS flow is applied, future work can involve fragmenting these packets so that it can be allocated in two consecutive frames, only if these frames are in the listen mode, and

both can meet the packet delay constraints. If BE is considered in addition to UGS, then the leftovers can be allocated to BE traffic.

In heavy traffic scenario, packets loss can be mitigated by modifying on the scheduling decision such that it does not push the packets to their deadlines. Instead, packets can be scheduled before reaching their deadlines by one or more frames based on the current traffic status. However, this will be achieved at the expense of power saving because more OFDM frames are expected to be in ON mode.

REFERENCES

[1] WiMAX Forum Technical Working Group. Available:.http://www.WiMAXforum.org [Accessed: Jan. 14, 2010].

[2] J.Andrews , A.Ghosh , R.Muhamed ,"Fundamentals of WiMAX," Prentice Hall, February 27, 2007.

[3] L. Nuaymi, "WiMAX – Technology for broadband wireless access", ed. Wiley, 2007.

[4] International Engineering Consortium, International Engineering Consortium. "Broadband Wireless and WiMax Comprehensive Report," International Engineering Consortium, 2005.

[5] C.So-In, R.Jain, A.Tamimi, "Scheduling in IEEE 802.16e Mobile WiMAX Networks: Key Issues and a Survey," *IEEE Journal on Selected Areas in Communications,* Vol. 27, No. 2,pp 156 - 171 , Feb. 2009.

[6] D. Niyato, E. Hossain, "Queue-aware uplink bandwidth allocation for polling services in 802.16 broadband wireless networks*," Global Telecommunications Conference, 2005. GLOBECOM '05. IEEE.* vol.6, 5 pp. - 3706 , Jan.2006.

[7] A. Stolyar, K. Ramanan,"Largest Weighted Delay First Scheduling: Large Deviation and Optimality," *Annals of Applied Probability*, vol. 11, pp. 1-48, 2001.

[8] D. Kim, C. Kang, "Delay Threshold-based priority Queuing Packet Scheduling for Integrated services in Mobile Broadband Wireless Access System," in *Proc. IEEE Int. High Performance Computing and Communications*, Kemer-Antalya, Turkey, 2005, pp. 305-314.

[9] Yan Wang, S. Chan, M. Zukerman, R. Harris, "Priority-based fair Scheduling for Multimedia WiMAX Uplink Traffic," *in Proc. of IEEE ICC 2008*, pp. 301-305, May 2008.

[10] L. De Moraes, P. Maciel, "Analysis and evaluation of a new MAC protocol for broadband wireless," *International Conference on Wireless Networks, Communications, and Mobile Computing - WirelessCom 2005*, Maui, Hawaii, Jun. 2005.

[11] W. Lilei, X. Huimin," A new management strategy of service flow in IEEE 802.16 systems," in *Proc. IEEE Conf. Industrial Electronics and Applications*, Harbin, China, 2008, pp. 1716-1719.

[12] J. Borin, N. Fonseca, "Scheduler for IEEE 802.16 Networks," *IEEE Commun.Lett.*, vol. 2, pp. 1142-1147.

[13] A. Khalil and A. Ksentini ."Classification of the Uplink Scheduling Algorithms in IEEE 802.16," *IWDYN'07 Workshop*. Rennes, France: INSA Rennes,2007.

[14] K. Wongthavarawat, A. Ganz, "Packet scheduling for QoS support in IEEE 802.16 broadband wireless access systems," *International Journal of Communication Systems*, vol. 16, pp. 81–96, Feb. 2003.

[15] S. Chandra, A. Sahoo, "An Efficient Call Admission Control for IEEE 802.16 Networks," *Local and Metropolitan Area Networks,* pp. 188-193, Jun. 2007.

[16] M. Kim, J. Choi, M.Kang, "Scheduled Power-Saving Mechanism to Minimize Energy Consumption in IEEE 802.16e Systems ,"*IEEE Communications Letters*, vol. 12, no. 12 , pp. 874-876, Dec. 2008.

[17] T.Chen, J.Chen ,"Extended Maximizing Unavailability Interval (eMUI): Maximizing Energy Saving in IEEE 802.16e for Mixing Type I and Type II PSCs," *IEEE Communications Letters*, vol. 13,no. 2, pp. 151-153, Feb. 2009.

[18] O.Vasta,M.raj,R.kumar,D.Panigrahy,D.Das,"Adaptive Power Saving Algorithm for Mobile Subscriber Station in 802.16e," *IEEE Communication Systems Software and Middleware, 2nd International Conference* , pp. 1-7 Jul. 2007.

[19] J.Jung, K.Han, S.Choi,"Adaptive Power Saving Strategies for IEEE 802.16e Mobile Broadband Wireless Access*," IEEE Communications, APCC '06. Asia-Pacific Conference* , pp 1-5, Dec. 2006.

[20] S.Huang, R.Jan, C.Chen,"Energy Efficient Scheduling with QoS Guarantee for IEEE 802.16e Broadband Wireless Access Networks," *in Proc. IWCMC,* Aug. 2007, pp. 547-552.

[21] S.Tsao, Y.Chen,"Energy-efficient packet scheduling algorithms for real-time communications in a mobile WiMAX system," *Computer Commnuications*, vol. 31, no. 10, pp. 2350-2359 , Jun. 2008.

[22] Cisco System, "Voice Over IP - Per Call Bandwidth Consumption".Available: http://www.cisco.com/en/US/tech/tk652/tk698/technologies_tech_note09186a0080094ae2.shtml [Accessed: Sep. 20, 2010].

VITA

Sanabel Hassan AlNourani was born on July 30, 1986, in Al-Ain, United Arab Emirates. She was educated in government and private schools and graduated from Tabareya School in 2003. She then got enrolled in the Philadelphia Private University, Jordan from which she graduated with honors, first in her batch in 2007. Her degree was Bachelor of Science in Computer Engineering. Ms. Mai received a graduate teaching assistantship to join the master's program in Computer Engineering at the American University of Sharjah in 2009.