SCHEDULING ALGORITHM FOR OPTIMIZED BANDWIDTH USAGE IN WIMAX NETWORKS

A DISSERTATION

Submitted in partial fulfillment of the requirements for the award of the degree
of
MASTER OF TECHNOLOGY
in
COMPUTER SCIENCE AND ENGINEERING

By

HEMSHANKAR SAHU

M.G. CENTRAL LIBRARY
ACC No. G21141
Date 9/9/11
I.I.T. ROORKEE

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE -247 667 (INDIA)

JUNE, 2011

CANDIDATE'S DECLARATION

I hereby declare that the work, which is being presented in the dissertation entitled **"SCHEDULING ALGORITHM FOR OPTIMIZED BANDWIDTH USAGE IN WIMAX NETWORKS"** towards the partial fulfillment of the requirement for the award of the degree of **Master of Technology in Computer Science and Engineering** submitted in the Department of Electronics and Computer Engineering, Indian Institute of Technology Roorkee, Roorkee, Uttarakhand (India) is an authentic record of my own work carried out during the period from July 2010 to June 2011, under the guidance of **Dr. Anjali Sardana, Assistant Professor,** Department of Electronics and Computer Engineering, IIT Roorkee.

The matter presented in this dissertation has not been submitted by me for the award of any other degree of this or any other Institute.

Date: 1/6/11
Place: Roorkee

(HEMSHANKAR SAHU)

CERTIFICATE

This is to certify that the above statement made by the candidate is correct to the best of my knowledge and belief.

Date: 1/6/11
Place: Roorkee

(Dr. ANJALI SARDANA)
Assistant Professor
Department of Electronics and Computer Engineering
IIT Roorkee.

ACKNOWLEDGEMENTS

First and foremost, I would like to extend my heartfelt gratitude to my guide and mentor **Dr. Anjali Sardana** Assistent Professor, Department of Electronics and Computer Engineering, Indian Institute of Technology Roorkee, for her invaluable advices, guidance, encouragement and for sharing her broad knowledge. Her wisdom, knowledge and commitment to the highest standards inspired and motivated me. She has been very generous in providing the necessary resources to carry out my research. She is an inspiring teacher, a great advisor, and most importantly a nice person.

On a personal note, I owe everything to the Almighty and my parents. The support which I enjoyed from my father, mother and other family members provided me the mental support I needed.

HEMSHANKAR SAHU

ABSTRACT

Interest in broadband wireless access (BWA) has been growing due to increased user mobility and the need for data access at all times. IEEE 802.16e based WiMAX networks promise the best available quality of experience for mobile data service users. Unlike wireless LANs, WiMAX networks incorporate several quality of service (QoS) mechanisms at the Media Access Control (MAC) level for guaranteed services for data, voice and video. The problem of assuring QoS is basically that of how to allocate available resources among users in order to meet the QoS criteria such as delay, jitter and throughput requirements. IEEE standard does not include a standard scheduling mechanism and leaves it for implementer differentiation. Scheduling is, therefore, of special interest to all WiMAX equipment makers and service providers. The goals of scheduling are to achieve the optimal usage of resources, to assure the QoS guarantees and to maximize throughput while ensuring low algorithm complexity.

The work in this dissertation is focused towards exploring the key issues and design factors to be considered for scheduler designers and to propose a novel approach for the optimized usage of the band width. We propose a novel Downlink Slot Reservation (DLSR) scheduling algorithm based of a new slot reservation concept for the Base Station of the WiMAX. This approach optimizes the usage of the bandwidth in the WiMAX networks by taking accurate decisions of how the bandwidth should be allocated to various types of connections.

The simulation results show that by using proposed DLSR approach the throughput of the non real time connections increased to 55% which comes at the cost of some increase in the delay of real time connections. The approach is also able to maintain the QoS requirement of the real time connections as this increase in delay is within limits. Along with that the algorithm is also computationally less complex.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION AND PROBLEM STATEMENT

WiMAX (Worldwide Interoperability for Microwave Access) also known as IEEE 802.16 is the next-generation of wireless technology designed to enable high-speed mobile Internet access to the widest array of devices including notebook PCs, smart-phones etc. It is the first all IP mobile Internet solution enabling efficient and scalable networks for data, video and voice. It enables the delivery of last mile wireless broadband access as an alternative to cable and DSL.

WiMAX is a wireless broadband technology, which supports point to multi-point (PMP) broadband wireless access. It offers both fixed and mobile broadband wireless Internet access. It has a range of up to 30 miles, and can deliver broadband at around 1gigabits per second. WiMAX covers large areas such as metropolitan, suburban, or rural, delivering mobile broadband internet access at speeds similar to existing broadband. It also provides the mobility features i.e. connection to internet will be always be there even when travelling. It allows accessing broadband internet even while moving at vehicular speeds of upto 125 kmph [1].

Major application areas of WiMAX include providing portable mobile broadband connectivity across cities and countries. WiMAX provides coverage areas in miles that can cover whole city and by using WiMAX backhaul entire nation can be covered. Also Wimax can be a wireless alternative to cable and Digital Subscriber Line (DSL) for "last mile" broadband access as WiMAX can provide speed same as that a DSL line provides, thus it will be cost effective to use WiMAX instead of deploying DSL line. WiMAX offers triple play services i.e. voice, video and data thus supporting all kind of network traffic and also maintains QoS related issues to them [2].

Like ATM, the 802.16 standard (WiMAX) was designed with variety of traffic types in mind. WiMAX has to handle the requirements of very-high-data-rate applications, such as voice over IP (VoIP) and video or audio streaming, as well as low-data-rate applications, such as web surfing, and handle extremely bursty traffic over the Internet. And it may need to handle all of these at the same time.

Some network applications simply cannot work without QoS. Some delay may be acceptable, but too much can make the application unusable. For example, the IEEE 802.16 group determined that an acceptable delay for VoIP is 120 ms, and over 150 ms delay results in noticeably impaired voice quality. Humans are intolerant of speech delays of over 200 ms [3].

The signaling and bandwidth allocation algorithms in 802.16 have been designed to accommodate hundreds of connections per channel and allow a variety of QoS requirements. The end user applications may be varied in their bandwidth and latency requirements, so 802.16 must be flexible and efficient over a range of different traffic models.

1.1 Motivation

WiMAX provides high-speed and ubiquitous access and a cost-effective solution which can be deployed quickly and easily for high bandwidth last-mile connectivity. WiMAX supports various kinds of traffic including real time traffic, non real time traffic, constant bit rate and variable bit rate. It also support a variety of applications like real time multimedia application (VoIP and IPTV) and other non real time applications (FTP). Most of these applications have quality of service (QoS) requirement like delay, jitter, throughput etc. The WiMAX standard classifies the traffic into four different classes namely Unsolicited Grand Services (UGS), Real Time Polling Services (rtPS), Non-Real Time Polling Services (nrtPS) and Best Effort (BE). These are called scheduling services. An application selects one of the scheduling services depending on its QoS requirement for requesting bandwidth. But the scheduling of the packets for the downlink has not been defined in the standard and is left for the vendor to select the best scheduling algorithm as per the requirement. The aim of the scheduling algorithm is to maximize the throughput and optimize the bandwidth usage along with maintaining the Quality of Service (QoS) of different type of connections. The factors which are under consideration while designing the scheduling algorithms are delay, jitter, throughput, starvation, packet loss etc. There have been a lot of research going on the designing of an efficient scheduling algorithm for the optimized usage of the bandwidth, but none of them is able to optimize all the parameters keeping the computational overhead low. Thus the dissertation is focused on developing a scheduling algorithm which will highly optimize the usage of the bandwidth in WiMAX networks and will be computationally simple.

1.2 Statement of the Problem

"To device a novel approach for scheduling of data packets in the MAC layer of WiMAX so as to optimize the bandwidth usage in WiMAX networks for different types of connections (real time and non real time)"

Problem Description: The bandwidth in the wireless networks is always limited as so is the case with WiMAX. Data flow in the wimax networks can be classified broadly into two types of connection – real time and non real time. Real time data flow requires data to be sent within the deadline while for non-real time data flow there is no such deadlines. If scheduling algorithm does not takes proper care of the allocating the bandwidth to the various connections the limited bandwidth gets wasted in wireless networks. So there is need of a scheduling algorithm which can accurately decide how much bandwidth should be allocated to each connection in the network. The aim is to maintain QoS of the real time connections while ensuring proper throughput for non-real time connections keeping the scheduling algorithm computationally inexpensive.

Thus the above problem has been divided into following sub problems:

(i) To maintain the QoS of real time connections.
(ii) To increase the throughput of the non real time connections
(iii) To keep the scheduling algorithm computationally inexpensive.

1.3 Organization of the Report

This dissertation report comprises of six chapters including this chapter that introduces the topic and states the problem. The rest of the report is organized as follows.

Chapter 2 gives the background study of WiMAX in terms of WiMAX MAC layer, QoS provisioning in WiMAX, frame structure for IEEE 802.16, brief literature review and outlines the research gaps.

Chapter 3 describes the slot reservation concept and the proposed solution.

Chapter 4 gives the simulation details in terms of test bed description, assumptions and parameters to be compared.

Chapter 5 discusses and analyzes the results obtained from the simulation.

Chapter 6 concludes the dissertation work and gives suggestions for future work.

Chapter 2

BACKGROUND AND LITRATRE REVIEW

In this chapter we discuss the technical details of WiMAX MAC Layer and QoS issues related to it. Later various proposed approaches for the scheduling algorithm in the WiMAX are discussed and analyzed.

2.1 MAC Layer in WiMAX

Reference Model: The MAC (Media Access Control) comprises three sublayers. The service-specific convergence sublayer (CS) provides any transformation or mapping of external network data, received through the CS service access point (SAP), into MAC service data units (SDUs) received by the MAC common part sublayer (CPS) through the MAC SAP. This includes classifying external network SDUs and associating them to the proper MAC service flow identifier (SFID) and connection identifier (CID). It may also include such functions as payload header suppression (PHS). Multiple CS specifications are provided for interfacing with various protocols. Data, PHY control, and statistics are transferred between the MAC CPS and the PHY via the PHY SAP (which is implementation specific).

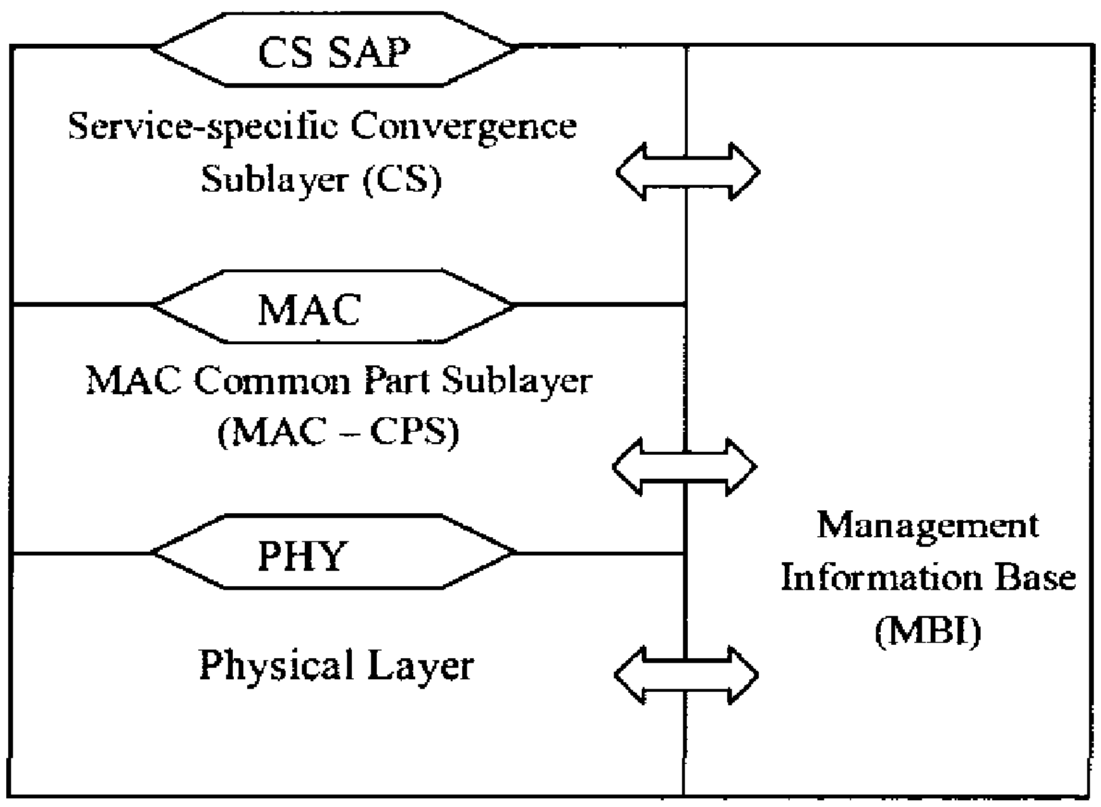

Figure 2.1 WiMAX MAC Reference Model [1]

The PHY definition includes multiple specifications, each appropriate to a particular frequency range and application. Management Information Base (MBI) is responsible for control and management of the connections. Figure 2.1 shows the reference model of the WiMAX MAC Layer.

Point-to-multipoint (PMP) operation in WiMAX

In WiMAX, in point to multipoint topology has a central Base Station (BS) and many Subscriber Station (SS). The SS requests the bandwidth form BS. The BS after authorizing the SS allocates bandwidth to the requesting SS. In PMP topology all the traffic has to go through the BS. Figure 2.2 explains the PMP operation in WiMAX networks.

Following steps take place in PMP operation:

- The DL (Down Link), from the BS (Base Station) to the user, operates on a PMP basis. The IEEE 802.16 wireless link operates with a central BS and a sectorized antenna that is capable of handling multiple independent sectors simultaneously. The DL is generally broadcast.
- In cases where the DL-MAP does not explicitly indicate that a portion of the DL subframe is for a specific SS, all SSs capable of listening to that portion of the DL subframe shall listen. The SSs check the CIDs in the received PDUs and retain only those PDUs addressed to them.
- SSs share the UL to the BS on a demand basis. Depending on the class of service utilized, the SS may be issued continuing rights to transmit, or the right to transmit may be granted by the BS after receipt of a request from the user.
- In addition to individually addressed messages, messages may also be sent on multicast connections (control messages and video distribution are examples of multicast applications) as well as broadcast to all stations.

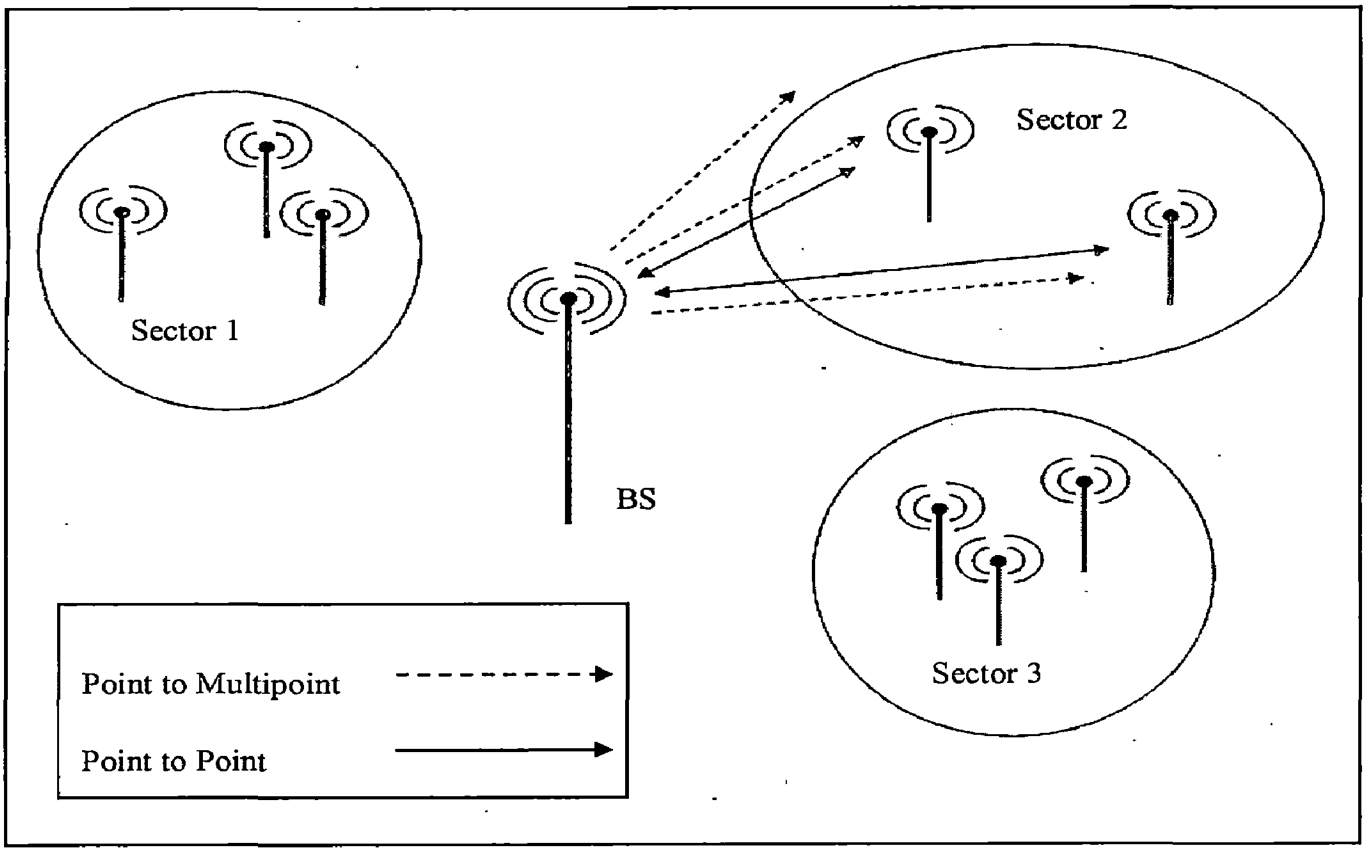

Figure 2.2 Point to multipoint operation

Transport Connection and Service flow

The MAC is connection-oriented. For the purposes of mapping various services on SSs and associating varying levels of QoS, all data communications are in the context of a transport connection. Shortly after SS registration, transport connections are associated with service flows (one connection per service flow) by the BS. This is to provide a reference to the SS using which it can request bandwidth from the BS.

The concept of a service flow on a transport connection is central to the operation of the MAC protocol. Service flows provide a mechanism for UL (Uplink) and DL (Downlink) QoS management. A SS requests UL bandwidth on per-connection basis and the service flow related to that connection defines the QoS parameters.

MAC Protocol Data Unit (PDU) Format

MAC PDUs shall be of the form illustrated in Figure 2.3. Each PDU shall begin with a fixed-length MAC header. The header may be followed by the payload of the MAC PDU. If present, the Payload shall consist of zero or more subheaders and zero or more MAC SDUs and/or fragments thereof. The payload information may vary in length, so that a MAC PDU may represent a variable number of bytes. This allows the MAC to tunnel various higher layer traffic types without knowledge of the formats or bit patterns of those messages.

MAC header	Payload	CRC

Figure 2.3 MAC Protocol Data Unit (PDU)

2.2 QoS Provisioning

The principal mechanism for providing QoS is to associate packets traversing the MAC interface into a service flow as identified by the Transport CID. A service flow is a unidirectional flow of packets that is provided a particular QoS. The SS and BS provide this QoS according to the QoS parameter set defined for the service flow

2.2.1 Service flows

A service flow is a MAC transport service that provides unidirectional transport of packets either to UL packets transmitted by the SS or to DL packets transmitted by the BS. A service flow is characterized by a set of QoS parameters such as latency, jitter, and throughput assurances.

A service flow is partially characterized by the following attributes:

a) *Service Flow ID:* An SFID is assigned to each existing service flow. The SFID serves as the principal identifier for the service flow in the subscriber station. A service flow has at least an SFID and an associated direction.

b) *CID:* The connection identifier of the transport connection exists only when the service flow is admitted or active. The relationship between SFID and Transport CID, when present, is unique. An SFID shall never be associated with more than one Transport CID, and a Transport CID shall never be associated with more than one SFID.

c) *ProvisionedQoSParamSet:* A QoS parameter set provisioned via means outside of the scope of this standard, such as the network management system.

d) *AdmittedQoSParamSet:* Defines a set of QoS parameters for which the BS are reserving resources.

e) *ActiveQoSParamSet:* Defines a set of QoS parameters defining the service actually being provided to the service flow. Only an active service flow may forward packets.

f) *Authorization Module:* A logical function within the BS that approves or denies every change to QoS parameters and classifiers associated with a service flow.

All service flows have a 32-bit SFID; admitted and active service flows also have a 16-bit CID.

The relationship between the QoS parameter sets is as shown in Figure 2.3.

The three types of serviceflows:

1) *Provisioned:* This type of service flow is known via provisioning by, for example, the network management system. Its AdmittedQoSParamSet and ActiveQoSParamSet are both null.

2) *Admitted:* This type of service flow has resources reserved by the BS for its AdmittedQoSParamSet,but these parameters are not active.

3) *Active:* This type of service flow has resources committed by the BS for its ActiveQoSParamSet. Its ActiveQoSParamSet is non-null.

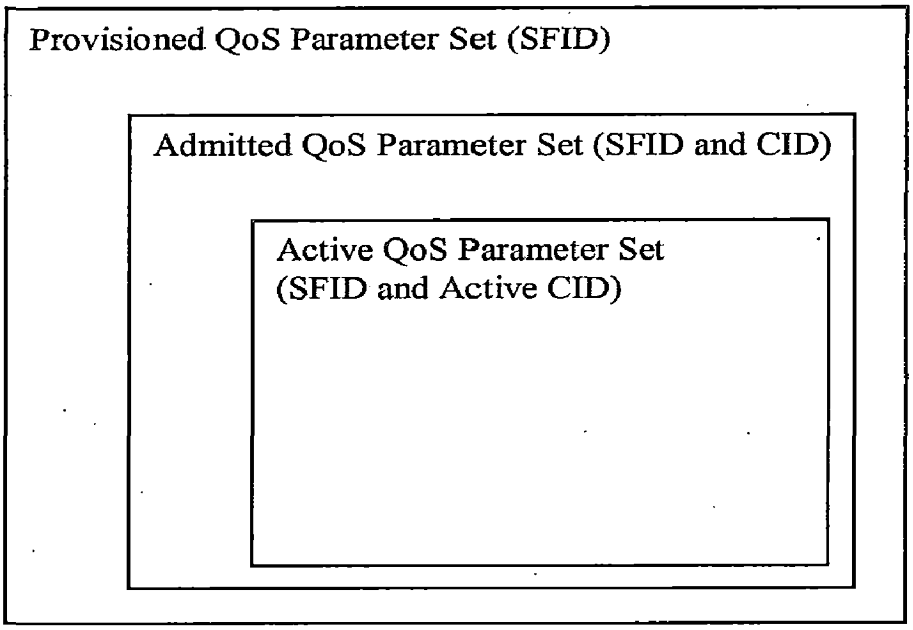

Figure 2.4 Relationship between the QoS parameter sets

2.2.2 Scheduling services

Scheduling services represent the data handling mechanisms supported by the MAC scheduler for data transport on a connection. Each connection is associated with a single scheduling service. A scheduling service is determined by a set of QoS parameters that quantify aspects of its behavior. Various parameters may be Tolerated jitter, SDU size, Minimum reserved traffic rate, Maximum Latency, Grant Scheduling Type, Unsolicited Grant Interval.

Various Up-Link Scheduling Schemes provided in MAC Layer are:

i. Unsolicited grant service (UGS): The UGS is designed to support real-time uplink service flows that transport fixed-size data packets on a periodic basis such as VoIP without silence suppression. The service offers fixed-size grants based upon the Minimum Reserved Traffic Rate of the service flow on a real-time periodic basis, which eliminate the overhead and latency of SS requests and assure that grants are available to meet the flow's real-time needs.

ii. Real-time polling service (rtPS): The rtPS is designed to support real-time UL service flows that transport variable-size data packets on a periodic basis, such as moving pictures experts group (MPEG) video. The service offers real-time, periodic, unicast request opportunities, which meet the flow's real-time needs and allow the SS to specify the size of the desired grant.

iii. *Non-real-time polling service (nrtPS)*: The nrtPS offers unicast polls on a regular basis, which assures that the UL service flow receives request opportunities even during network congestion. The BS typically polls nrtPS connections on an interval on the order of one second or less. The BS shall provide timely unicast request opportunities. The SS is allowed to use contention request opportunities.

iv. *Best effort (BE) service*: The intent of the BE grant scheduling type is to provide efficient service for BE traffic in the UL. The SS is allowed to use contention request opportunities. This results in the SS using contention request opportunities as well as unicast request opportunities and data transmission opportunities.

Various Scheduling Services and there usage rule are shown in Table 2.1.

Table 2.1 Scheduling Services and their usage rule

Service type	**Piggy Back Request**	**Bandwidth stealing**	**Description**	**QoSParameters**
UGS	Not Allowed	Not Allowed	Supports real time Constant Bit Rate services, such as VoIP without silence suppression	Maximum sustained traffic rate, Maximum latency, tolerated jitter
Rtps	Allowed	Allowed	Supports real time data with variable bit rate, such as VoIP with silence suppression, MPEG.	Minimum reserved traffic rate, Max. sustained traffic rate, maximum latency
nrtps	Allowed	Allowed	Supports non-real time services that requires variable size data grant burst on a regular basis, such as FTP	Minimum reserved traffic rate, maximum sustained traffic rate, traffic priority
BE(Best Effort)	Allowed	Allowed	For application that do not require QoS, such as web surfing	Maximum sustained traffic rate

WiMAX Forum classifies applications into five categories as shown in Table 2.2. Each application class has its own characteristics such as the bandwidth, latency and jitter constraints in order to assure a good quality of user experience. The traffic models for these applications can be also found in [16].

Table 2.2 WiMAX Application Classes

Applications	**Bandwidth Guidelines**		**Latency Guidelines**		**Suitable QoS Class**
Multiplayer Interactive Gaming	Low	50 kbps	Low	< 100 ms	rtPS and UGS
VoIP and Video conferencing	Low	32-64 kbps	Low	<160 ms	UGS and rtPS
Streaming Media	Low to High	5 kbps to 2 Mbps	NA		rtPS
Web Browsing and Instant Messaging	Moderate	10 kbps to 2 Mbps	NA		BE and nrtPS
Media Content Downloads	High	➢ 2Mbps	NA		BE and nrtPS

2.3 The IEEE 802.16 (WiMAX) Frame Structure

A frame is a unit of communication. A frame is made up of various time slots. These time slots are combination of the symbols and a symbol is group of some bits. Thus the job of the scheduler is to allocate these very bits or symbols to various connections so as to optimize the bandwidth usage. So it is very important to know what different parts of a frame are.

A frame is divided into two sub-frames: UL sub-frame and DL sub-frame as shown in Figure 2.5. In the DL sub-frame, the BS sends data and control information to the SSs and the UL sub-frame is used by the SSs for data transmission to the BS. These sub-frames may be of equal or different durations as decide by the BS. The frame duration may range from 2 ms to 200 ms.

For DL and UL sub-frames the duplexing scheme used may be Time Division Duplexing (TDD) and Frequency Division Duplexing (FDD). In TDD mode, both the UL and DL sub-frame transmissions occur in same frequency but in different time. First DL sub-frame is transmitted followed by the UL sub-frame. WiMAX has been developed with an aim of providing wireless broadband internet connectivity. Therefore it is assumed that most of the traffic will be from BS to the SS. Because of this generally the DL sub-frame size is greater than the UL Sub-frame but it may vary as per the conditions. In FDD mode, the DL and UL are transmitted using different frequencies. Thus they are sent concurrently in time. We focus on 802.16 systems operating in TDD mode.

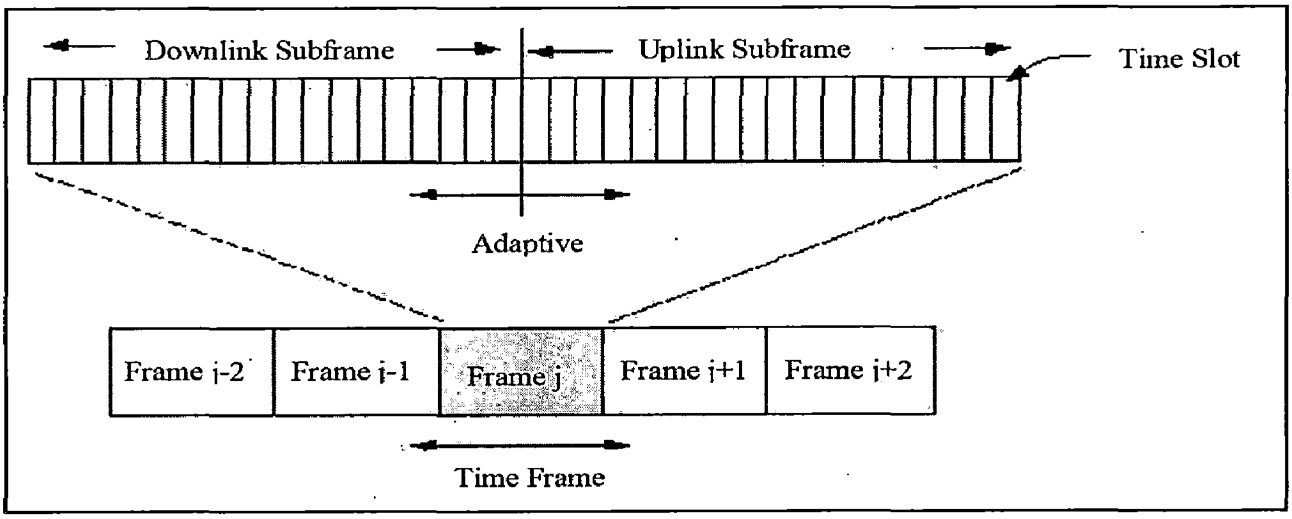

Figure 2.5 UL and DL sub-frame of a WiMAX frame [2]

Figure 2.6 shows the OFDM (Orthogonal Frequency Division Multiplexing) frame structure in TDD mode. An OFDM Physical Layer (PHY) DL sub-frame transmits one downlink PHY Protocol Data Unit (PDU), which is possibly shared by more than one SS. For the synchronization of the SS with the BS there is a long preamble at the starting of the downlink PHY PDU. A Frame Control Header (FCH) burst follows the preamble which contains the Downlink Frame Prefix (DLFP). DLFP specifies the burst profile and length of at least one downlink burst immediately following the FCH. Last byte of the DLFP is an HCS field.

After FCH a DL-MAP and UL-MAP messages are sent by the BS to describe that which part of the frame belong to which SS. DL-MAP is transmitted first followed by the UL-

MAP message. One or many downlink bursts are transmitted in order of decreasing robustness of their burst profiles following the FEC. Each UL PHY transmission burst contains only one UL burst and starts with a short preamble (1 OFDM symbol). All MAC PDUs of a UL burst are transmitted by a single SS using the same PHY mode. Two gaps separate the DL and UL sub-frames: Transmit/Receive Transition Gap (TTG) and Receive/Transmit Transition Gap (RTG). These gaps allow the BS to switch from transmit to receive mode and vice versa.

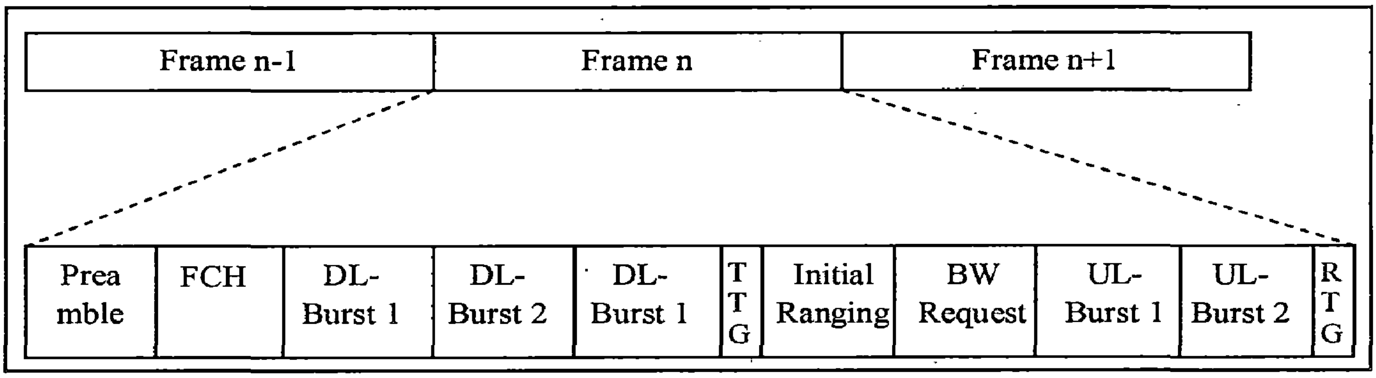

Figure 2.6 OFDM frame structure with TDD [2]

Clearly form the above description of a WiMAX frame, we can observe that the time slots are very limited in number for DL. So the decision of allocating these time slots to various connections plays an important role in optimizing the bandwidth usage in the WiMAX networks. Our proposed approach for DL scheduling is thus designed keeping this point in mind.

2.4 Functional Entities for QoS Support

Figure 2.7 shows the functional entities for QoS support, which logically reside within the MAC layer of the BS and SSs. Each downlink connection has a packet queue (or *queue*, for short) at the BS (represented with solid lines).

➢ In accordance with the set of QoS parameters and the status of the queues, the *BS downlink scheduler* selects from the downlink queues, on a frame basis, the next service data units (SDUs) to be transmitted to SSs. On the other hand, uplink connection queues (represented in Figure. by solid lines) reside at SSs.

➢ Since the BS controls the access to the medium in the uplink direction, bandwidth is granted to SSs on demand.

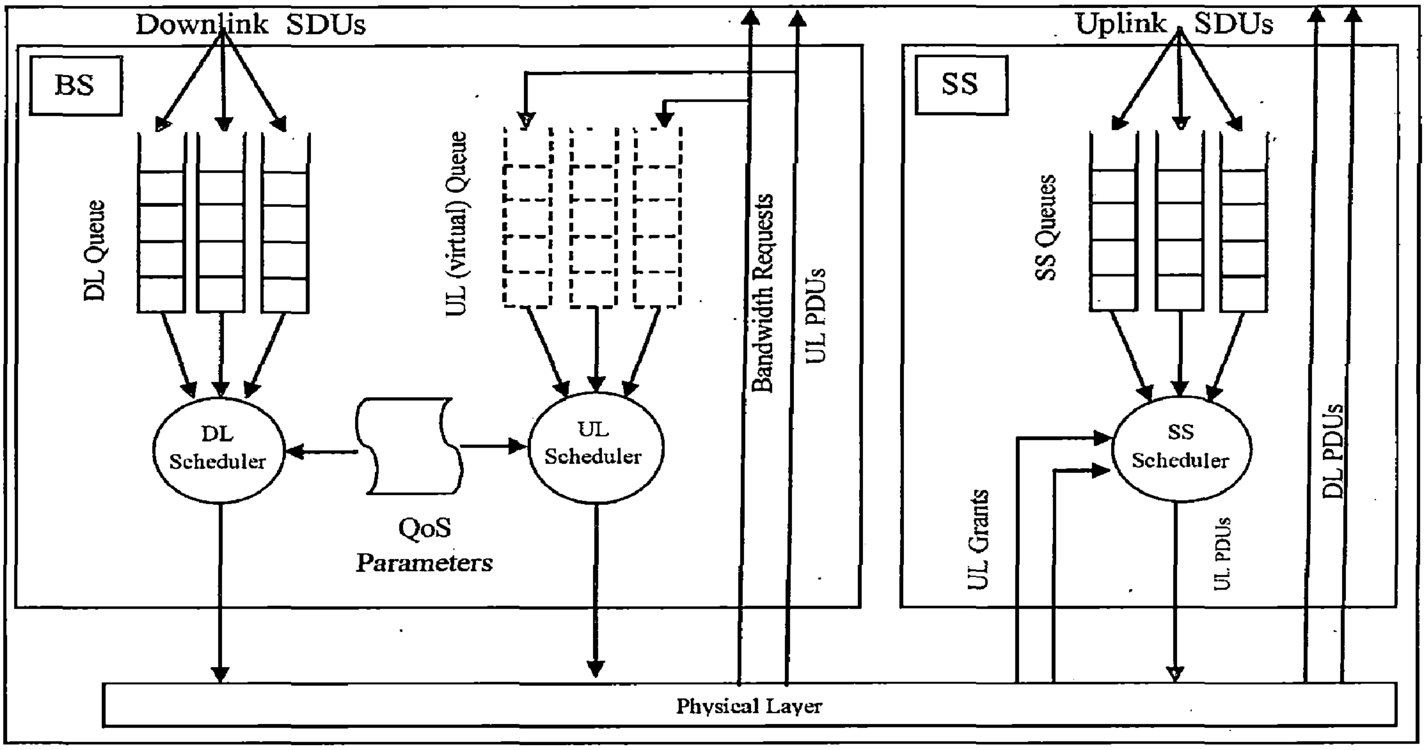

Figure 2.7 Functional Entities For QoS Support

➢ Based on the amount of bandwidth requested so far, the *BS uplink scheduler* estimates the residual backlog at each uplink connection (represented in Figure 2.7 as a virtual queue by dashed lines)

Based on this model many scheduling algorithms have been proposed and adopted according to the need and requirement which is described in next section.

2.5 Scheduling Algorithms in WiMAX

As there is no specific scheduling algorithm defined in the standard of WiMAX, many scheduling techniques have been proposed and implemented. These scheduling algorithms consider many aspects for optimized bandwidth usage like total maximum data rate, fairness, throughput, packet loss, starvation etc. Generally the existing and well known scheduling algorithms like Round Robin, Weighted Round Robin, Deficit Round Robin etc. cannot be used for WiMAX as they were originally developed wired medium. In wireless medium there are various other things especially in the WiMAX PMP architecture where there is a central BS which is responsible for handling all types of traffic having varying QoS issues and a different frame structure with different physical medium modulation. Therefore there is requirement of the scheduling algorithm specific to WiMAX so as to optimize the usage of the resources present and ensure QoS. In this section we will broadly classify various scheduling algorithms proposed specifically for WiMAX.

2.5.1 Active List Scheduling Algorithms

The Active List Scheduler [3], [4] maintains a list for active and/or eligible SS which will be selected for scheduling or transmitting their data packets form the BS. This selection of the SS in the active list is done on the basis of the packet call power, radio conditions, channel conditions and/or QoS requirements. The scheduling list contains all the SSs that can be served at the next frame. Main problem in this type of algorithms is that if the SS active list is very large than it may happen that the QoS requirement of some of the SS may not get fulfill. Also if the QoS requirement is fulfilled, chances are that the starvation of the lower priority and non real time connections will take place.

2.5.2 Fair Priority Queue Scheduling Algorithms

This type of algorithms [5], [6] aims at decreasing the delay of real-time traffic. It schedules the packets on the basis of their service types instead of scheduling as per their destination i.e. SS. The scheduler groups packets according to service type, and after sending all the packets belonging to the same service type, it moves to the next group. For example, all the packets of UGS services are scheduled first and then the packets of rtPS services are sent. Here as the UGS and rtPS services scheduled first, thus the overall delay of real-time traffic is reduced. Little variation in this type of scheduling algorithm in some of the other approaches [7], [8], [9], [10] have also considered factors like delay, jitter, waiting time, traffic rate etc. but only a small improvement of over the existing algorithm was observed. The algorithm is good for maximizing the number of SS and/or connection for real time traffic. Clearly in this type of approches the real time connections are higher in priority for getting transmitted in DL sub-frame. This causes the problem of starvation and packet loss for the non real time connections. For checking this starvation problem a threshold value for the non real time connection queues is set. If the queue length (on an average) increases above this threshold the priority of the non real time connections is increased and they get chances of transmission. But setting this threshold value do not increase the throughput of the non real time connections. Again since the packets are grouped according to the service type, it may be possible that some connections are of higher importance than other of same service type but here no provision of assigning priority to a connection of same service type is given in the first step.

2.5.3 Frame Registry Tree Scheduling Algorithms

The Frame Registry Tree Scheduler (FRTS) scheduler [11], [12] contains three operations: packet/request arrival, frame creation, and subscriber's modulation type change or connection QoS service change. This type of schedulers distributes packet transmissions in time frames. This distribution is based on the deadline of the packets. For deadline calculation of the UGS and rtPS services, both the arrival time and the latency of this packet is considered. Then the packet is sent to the sub-tree of last time frame in which it can be transmitted, if that sub tree exists else the sub-tree for that last time frame is constructed. While constructing the frames for transmission, all the packets under that frame sub-tree is collected. If the frame got over flow packets are dropped in the decreasing order of their priority i.e. BE packets are dropped first followed by nrtPS, rtPS and UGS packets. If some space remains in the frame than packets for the next frame to be transmitted is taken. Clearly there are chances of packet loss in each frame.

2.5.4 Adaptive rtPS Scheduling Algorithms

This class of scheduling algorithms [13], [14] focuses on providing better throughput to the real time connections. They are used only for the rtPS QoS class. The idea is based on the prediction of the rtPS packets arrival and their delay in the SS. The BS allocates bandwidth for rtPS traffic after receiving a bandwidth request. When the request is granted by the BS, the SS may receive from upper layers new rtPS packets. These packets will wait for the next grant to be sent and, therefore, suffer from extra delay. To overcome this delay the SS requests time slots for the data present in the rtPS queue and also for the data which will arrive. This estimation of the data arrival is done based on pattern of the traffic currently being transmitted. Again these types of algorithms are centered to ensure QoS for the real time connections and therefore leading to the starvation of the non real-time connections.

Table 2.3 shows the comparative description of the above discussed approaches.

Table 2.3 Comparison of scheduling algorithms

Scheduling Algorithm Class	Implementation Methodology	Advantages	Disadvantages	Average Complexity
Active List Scheduling Algorithms [3]	A list for eligible SS which is be selected for transmitting data packets based on radio conditions, QoS etc.	Easy To Implement	Packet Loss, Starvation, under high load some SS's QoS may get violated	$O(n^2)$
Fair Priority Queue Scheduling Algorithms [5]	The scheduler groups packets according to service type, and after sending all the packets belong to the same service type, it moves to the next group.	QoS is Guaranteed	Starvation of non real time packets, Packet Loss	$O(n)$
Frame Registry Tree Scheduling Algorithms [7]	These types of schedulers distribute packet transmissions in time frames.	Reduces starvation, QoS is Ensured	May result in packet Loss due to frame overflow, Large amount of calculation involved	$O(n)$
Adaptive rtPS scheduling algorithms [11]	SS requests bandwidth based on the present queue length as well as data that will arrive till that time.	Increases the throughput of the real time connections	Starvation of non real time packets	$O(n^2)$

In all of the above approaches the bandwidth allocated to connections based on various calculation considering the queue length and the delay requirement of the real time connections. At any time, a chunk of data is taken from the queue of a connection and is added to the frame. Thus while constructing a frame in this fashion the decision made of bandwidth allocation is little imperfect. Also a lot of overhead is involved while constructing a frame like the selection of the most appropriate connection, removing data from the queue and then adding it in the frame. Generally the scheduler has to loop again and again selecting these queues for data fetching causing large frame construction time.

Thus these approaches cannot reach to the fine granularity of a slot for the reservation of the bandwidth. In our approach we consider a packet of a connection as a basic element for bandwidth allocation. In this way we are able to decide in a better way how bandwidth should be distributed between various connections so as to optimize the bandwidth usage.

2.6 Research Gaps

- **Starvation:** Almost all of the approaches [3], [4], [5], [6], [7], [8], [10], [11], [13], [14] proposed recently suffer from the problem of starvation of non-real time connection. One reason may be that the aim of all these approaches is only to maintain QoS for different connection while less importance is give to the throughput of the connections.
- **Poor Utilization of the bandwidth:** The decision made for allocating the bandwidth to connection is taken generally using the current queue length and some threshold value [3], [4], [5], [6], [7], [9], [12], [13]. These decisions of allocation the bandwidth is not perfect as the usage of threshold value makes is less flexible. Some approach tried to calculate dynamic threshold value but turned out to be highly complex.
- **Complex Scheduling Algorithm:** Numerous scheduling algorithms [3], [4], [13], [14] were proposed in recent years, each adding some of the calculation to the previous one. Due to this the complexity of the scheduling algorithms also increased.

From the research gaps found in the earlier work, following are the goals which is to be kept in mind while designing the scheduling algorithm for WiMAX.

1. It should reduce the starvation and increase the throughput of non-real time connections.
2. It should be able to maintain the QoS of the real time connections.
3. It should be computationally simple and scalable.
4. It should be able to utilize the limited bandwidth to its full extent.
5. It should be fast.

Aiming to fulfill the above mentioned requirements, we propose a novel approach for WiMAX Base Station for Downlink direction, which is elaborated in the next chapter.

Chapter 3

PROPOSED DLSR SCHEDULER

In this chapter we propose a novel Downlink Slot Reservation (DLSR) scheduler for the Base station of WiMAX. We developed a scheduling algorithm for the downlink of the WiMAX because the WiMAX standard does not define any fixed scheduler for handling the downlink of the packets while there are provisions in the standard for the uplink of packets for varying QoS requirements. The proposed DLSR scheduler aims at increasing the throughput of the non-real time connections while maintaining the QoS requirement of the real time connections. Keeping in mind the various factors from the last chapter to be considered while designing a scheduler, we propose DLSR approach which is computationally simple and yields high throughput for the non real time connections and maximizes the bandwidth usage in the WiMAX networks.

WiMAX supports different kinds of traffic like VoIP, IPTV, FTP, web page request-response, video conferencing etc. These traffics can very broadly be classified as real time and non real time. Real Time connections have some deadline and if their packets are not transmitted within this deadline they will be useless. While for non-real time connections no such deadline is there. Because of the deadline of the real time connections there is always some minimum throughput which is to be maintained for these connections. Designing schedulers for increasing the throughput of these real time connections will be no wise, as the scheduler always have to make sure that the real time QoS requirements are being fulfilled for these connections. The increased throughput for the real time connections is likely to have no major effect on the performance of the real time applications. For example in the IPTV application, increasing the throughput can only increase the buffering rate of a TV show while the show will run in its original speed. This type of connection only requires that each packet is transmitted before its deadline and not as fast as possible.

On the other hand, throughput does highly matters, for the non real time connections. For example, the only thing under consideration when downloading a file is how fast it gets downloaded i.e. how much throughput is being achieved. The above discussion is the motivation for the designing of our DLSR scheduler. Therefore main goal of our scheduler is to increase the throughput of the non real time connections which may be at the cost of little throughput reduction of the real time connections without affecting their QoS requirements.

3.1 The Slot Reservation Concept

The proposed DLSR approach for optimizing the bandwidth usage in WiMAX networks is based on a novel **slot reservation** concept. This proposed approach is inspired by the "frame registry tree scheduler" approach [10]. In [10] when a packet is received, a frame is selected in which it will be transmitted. Because the size of a frame can be fetched before hand, we can always calculate the total number of frames that are available in which the received packet can be sent without violation the QoS requirement. For example if a packet has latency of 100 ms and the frame size is 10 ms then there are 100/10 = 10 frames in which that packet can be sent before violating its latency requirement. Thus in the approach proposed in [10], the fame number for the received packet is calculated considering both the queue length of the connection of the packet and its latency. The packet is then stored in the queue of that particular connection. When a frame is selected for the transmission then data packets related to this frame are fetched from the connection queues. In this approach there are two conditions which may cause the loss in the data packet. First is when the queue of that connection is full and second while collecting the data packets of a frame and the frame gets over flow. In this case the data packets of lower priority like packets of BE and nrtPS are dropped first and then the higher priority connection packets are dropped.

The proposed 'slot reservation' is concept inspired by the Frame Registry Tree Scheduler [10]. In the approach in [10], every packet is assigned a frame and that packet with high probability will be transmitted in that frame, but it may not get transmitted though high probability. In the slot reservation concept a frame is decided and assigned to a real time packet in which the packet will be transmitted, but unlike the above approach [10], it is made sure that this packet will get transmitted in the assigned frame only. This is done by forming virtual frames and storing the packet in one of those frames instead of storing them in the queues. When time comes to transmit the actual frame, the virtual frame corresponding frame is selected and the packets stored in that virtual frame are transmitted. In this way we are able to bind or reserve the slots of the actual frame using its virtual frame. Thus some slots are reserved in the frame for the packet beforehand and hence the concept is named as 'slot reservation' concept. The maximum number of the temporary frames is decided by the scheduler, based on the maximum permissible latency. As soon as a real time packet comes in the BS its deadline is calculated, using its Maximum Latency QoS parameter, a frame is selected (based on the calculations described in section 3.3) and the packet is stored in that virtual frame reserving few slots instead of its queue. Thus these slots are reserved for the received real time packet.

3.2 Overall Approach

In this section brief overview of the proposed DLSR approach is given. Figure 3.1 shows a sequence of steps that is followed in our approach.

In DLSR approach when non-real time connection packets come they are directly stored in a common queue. Using this common queue we then calculate the total queue length of all the non-real time connections.

When a real time connection packet comes first its dead line is fetched using the Maximum latency QoS parameter. After that we find the density of the real time connections i.e. the total bandwidth required by the real time connections.

Then using the queue length of non real time connections and the density (total band width requirement) of the real time connections a frame is selected for the real time packet, which just came in, for transmission. This packet is then stored in the virtual frame, formed in the BS, before transmission. At last while transmitting a frame we take the real time packets from its corresponding virtual frame and add the non real time packets from the common non-real time Queue and finally transmit the frame.

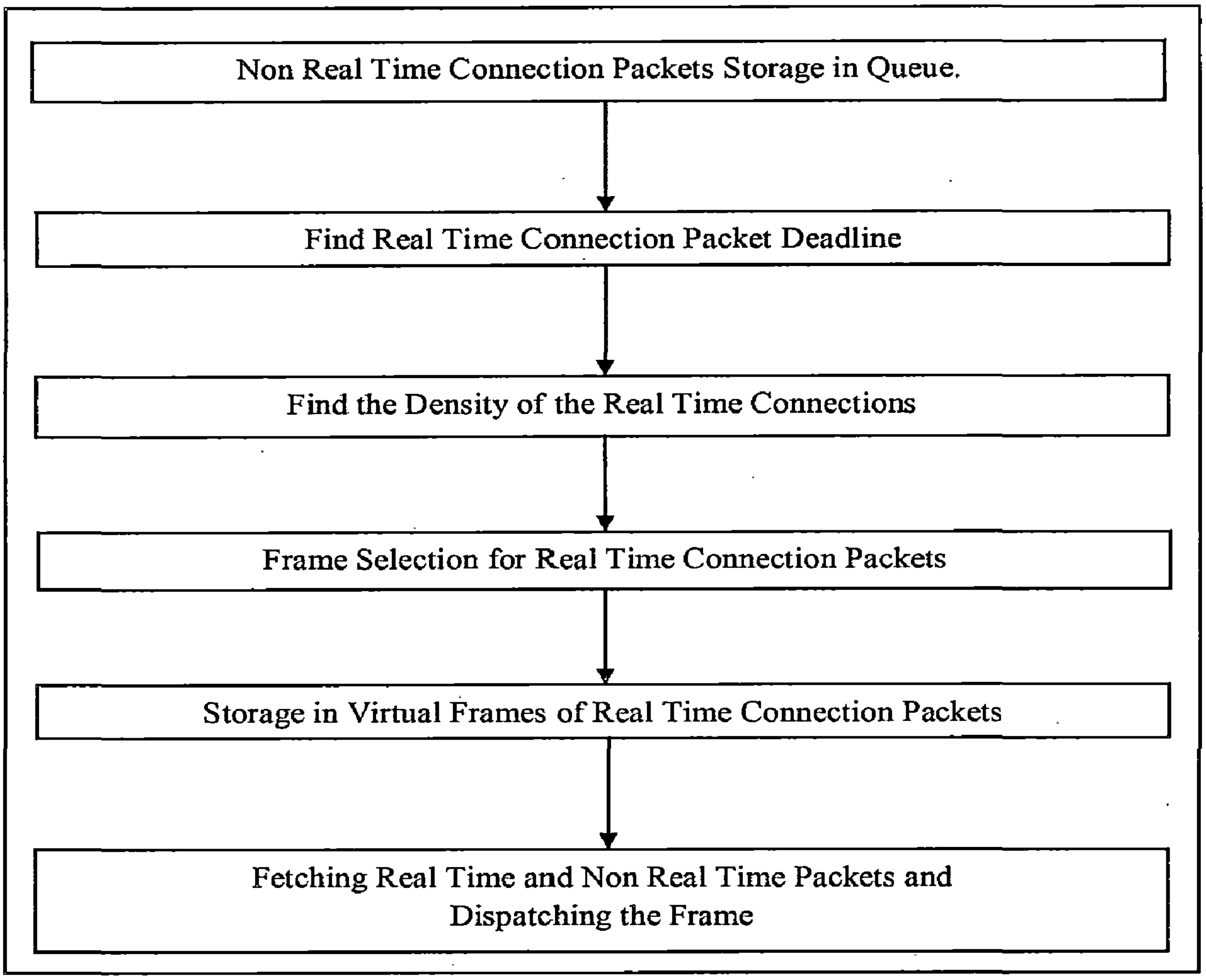

Figure 3.1 Overall DLSR Approach

The approach exploits the fact that for the real time packets there is a specified maximum delay period before which the real time data must be transmitted, called their deadline. This delay period can be used for transmitting the non-real time data packets if their queue length is more. For using this delay period we must ensure that the real time data packets will get transmitted before its deadline. Thus a deadline database is required for all the real time connection. This deadline is decided before the connection is established and this is available as a QoS service parameter for every service flow under the name Maximum Latency. For ensuring the transmission of the real time packets within their deadline a different technique is used form the one proposed in [10]. This difference is in the storing of the data packets in the BS before transmitting them. Two different approaches for storing the packet of real time connections and non real time connections is considered. Watching the drawback of the above approach [10], of packet loss while constructing the frame, the real time packets are stored directly into the frames (temporary) and not in the queue itself. While for the non real time packets we are storing the packets in their respective connection queues.

For calculating the appropriate frame number here two parameters are considered – queue length of the non-real time connections and the density of the real time connections. In Figure 3.2 shows the block diagram of the proposed DLSR approach. In this figure MAP stands for the UL and DL Maps and TOP is the temporary variable storing the information that how much virtual frame is full. Here no queue is being considered for rtPS or UGS services as they will directly be stored in the virtual frame. While queue for the nrtPS and BE is required.

When a frame is to be sent, its empty slots are then filled by the BE/nrtPS packets. These empty slots are actually the space created by the factor Q (percentage queue length) in a frame for the transmission of the non real time packets. Larger the Q larger space will be created in the frames and more number of empty slots will be present in the frames for the transmission of the non-real time packets. Here number of frames (n) is decided based on much maximum delay can be considered for rtPS services. The shaded portion in Figure 3.2 shows filled parts of frames with the real time packets are while un-shaded parts are empty.

The process of forming a DL sub-frame in our approach can be summarized under following points.

1. As soon as a packet is received if it is a non real time packet it is just stored in to a common queue and the queue length of the non real time connections is increased.
2. If the packet received is a real time packet then its frame number is calculated based on the above calculation.
3. The received real time packet is stored in its corresponding frame.
4. If the frame is full than any other frame is selected which satisfies the deadline.
5. While transmitting a DL Sub-frame if the frame to be sent is not completely filled then the non real time packets of the non real time connection queue are selected for transmission till the frame is full.
6. Step 1 to 5 is repeated.

CENTRAL LIBRARY
M.G.
ACC No.............
Date...............
I.I.T. ROORKEE

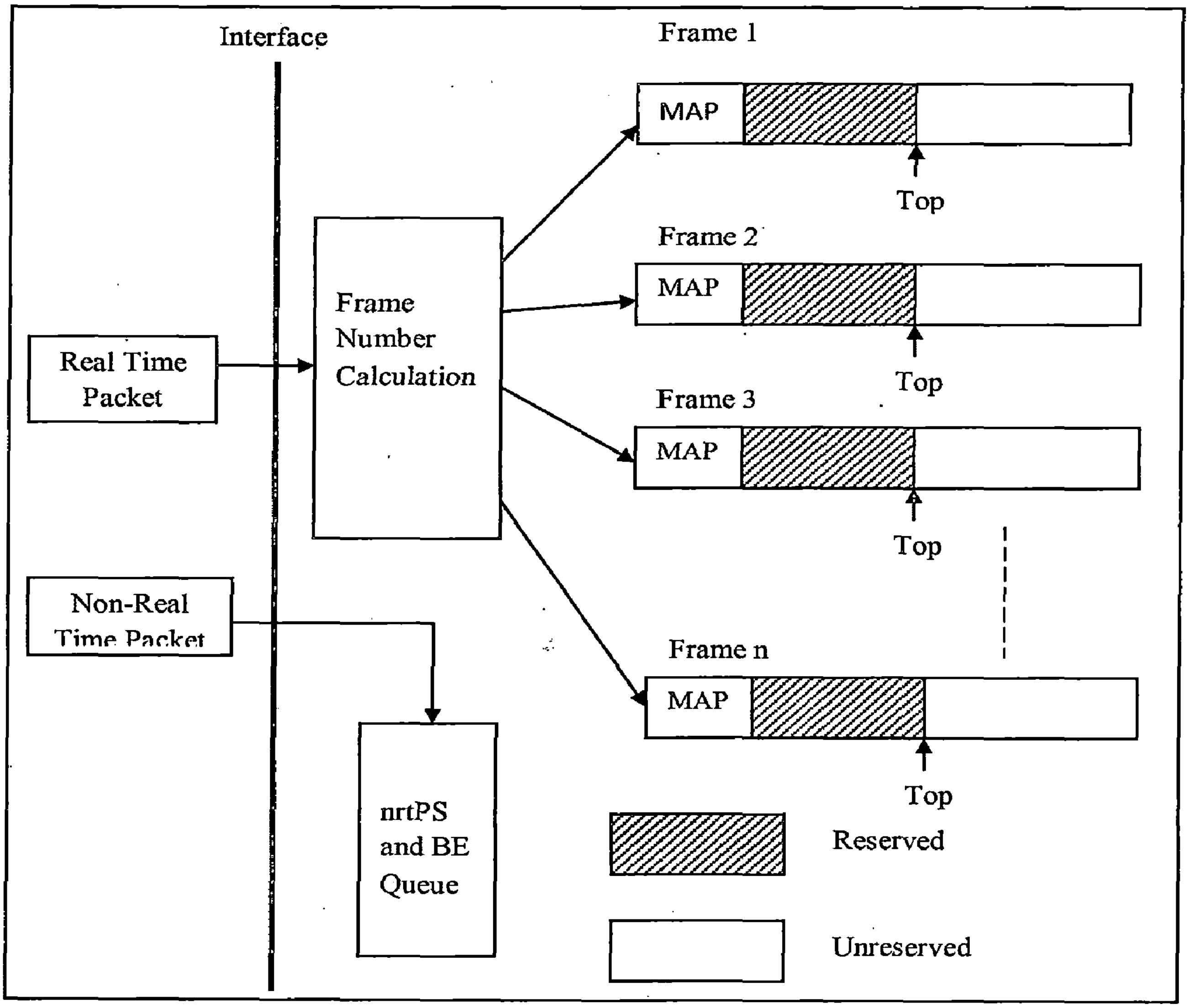

Figure 3.2 DLSR Scheduler working block diagram

3.3 Improvising QoS for Real Time Connections

In our approach we ensure that the QoS for the real time connections is not violated as we try to increase the throughput of the non real time connections. For maintaining the QoS for the real time connections we do following calculations.

3.3.1 Real Time Connection Density Calculation

Density of the real time connection is calculated as the percentage of the total bandwidth required by the real time connections out of the total available bandwidth. i.e. Density of the rtPS Traffic (D) can be calculated as

$$D = (\sum r_i / T) * 100 \tag{3.1}$$

Where r_i = bandwidth requested for ith connection

T = Total bandwidth available

3.3.2 Non-Real Time Queue Length Calculation

The queue length of the non-real time connection is calculated as the sum of all the queue length of all the non real time connections divided by the sum of maximum size of all the non real time connections i.e.

$$\textit{Non real time Queue-Length} = \frac{\sum \textit{queue length}}{\sum \textit{max. queue size}} \tag{3.2}$$

and Q is the percentage value of the non real time Queue length. Here the more the density (D) of the real time connections, more number or the real time packets will be present and thus less bandwidth can be given to the non real time connections. While the more the Queue length (Q) of the non-real time data, more will be the bandwidth required for the non-real time connections. If the density of the real time connections is less and the queue length of the non real time packets are large then the real time packet can be delayed for some frame(s) and the non real time packets can be transmitted in the mean time. This is the key idea of our approach. Here we are considering a common queue for all the non real time connections for maintaining fairness.

3.3.3 Available and Current Available Frames Calculation

The next step is to find number of available frames in which a packet can be transmitted after being received so that its Maximum Latency QoS is not violated. This is calculated by dividing the Maximum Latency of the packet by the frame time.

i.e.

$$\textit{Available Frames (AF)} = \textit{Max. Latency/Frame Time} \tag{3.3}$$

Although the packet can be transmitted in any of the following frame it is important here to consider the D factor of the real time connections before selecting any frame. For example if the total bandwidth is suppose 1000 units and the real time connections requires 100 units of the bandwidth, i.e. 10% of the total bandwidth is used by the real time connections while 90% of the bandwidth is free for other use. Suppose average latency of these real time packets is 100 ms and the frame time is 10 ms then AF will be 10 frames. But it is possible that 10% of these 10 AF (or bandwidth in other words) will already be reserved therefore we have only 9 frames free or unreserved now. So it is important to reducing this factor (number of frames), which is responsible for the delay of the real time packets. This reduced frames are denoted by Current Available Frame (CAF) and is calculated as

$$CAF = 100 - D * AF \tag{3.4}$$

Thus greater the density of the real time packets lesser will be the delay in frames. This causes more and more real time packets to be transmitted in the earlier frames and leaving later frames empty for the later large number of the real time packets.

3.4 Improvising Bandwidth for Non-Real Time Connections

After ensuring the number of frames available for the packets of the real time connection so that its QoS is not violated, we then make provision for the allocation of the bandwidth to the non real time connections. For this we select a frame for the real time packet based on the queue length of the non-real time connections making space for the non real time packets.

3.4.1 Frame Selection Calculation

Now the job is to select a frame out of the CAF. For this we need to consider a factor for queue length of the non-real time connections i.e. Q. This is done by simply selecting a frame form CAF using eq. (5)

$$Selected\ Frame = CAF * Q \tag{3.5}$$

The key here is exploiting the fact that the packet of the real time services can be made to wait in the buffer till its deadline. The WiMAX is not only developed for providing QoS to the real time traffic, but also to give proper bandwidth to the non real time traffic so that they do not starve. For the non real time traffic the timing of the bandwidth allocation is important i.e. the bandwidth should be provided when it is required. The factor Q here shifts the real time packets from current frame to some other frame and thus makes space for the real time connections. Table II shows information that is used for calculation of a frame number for a real time packet.

Note that it is not wise to store the deadline of each packet and then move that packet to corresponding frames at the time of transmitting the frame. It will require huge database to be maintained and also large time for forming a frame. Key observation here is that we need to somehow find the right packets that will be selected for transmitting in a frame. This calculation is generally done at time of transmitting the frame in other approaches which proves to be inefficient. In our approach the packets of the real time connections will directly be stored in a frame and will never be stored in queue. Therefore no deadline database for each packet is required and thus the decision of which packet to be selected for transmission in a frame is very accurate. The approach considers the 'n' number of frames which will be simultaneous filled with the real time packets.

Table 3.1 Frame Number Calculation Mechanism

Cid	**D**	**Q**	**Max. Latency (ms)**	**Total Frames Available**	**Current Frames Available**	**Selected Frame Number**
200	20 %	50%	100	10	8	4
400	50%	50%	200	20	10	5

3.4.2 Frame Formation and Dispatch

After the selection of the frames for the real time packets, we finally form a real frame and transmit it. For the formation of the frame we first select the virtual frame corresponding to the frame number to be transmitted. We then add the contents of that virtual frame to the real frame. If the frame is full then we transmit it. If it is not full then we select packets from the non real time connections queue and add them to the frame before dispatch. In this way bandwidth is allocated to the non real time connections.

3.5 The DLSR Algorithm

The DLSR algorithm is written in two parts. First part is for the formation of the virtual frames and next part is for formation and dispatch of the real frames. Figure 3.3 shows the first part of the Downlink Slot Reservation (DLSR) algorithm i.e. formation of the virtual queues. Figure 3.4 shows the second part of the DLSR algorithm i.e. Frame formation and transmission. Figure 3.5 shows the flow chart of the various steps being performed in the formation of a frame.

```
1.  Packet Received (packet)
2.      If (Packet->type == Real Time)
3.              AF = Packet->Max_Latency/Frame_Time
4.              CAF = (100-D) * AF
5.              Selected_Frame =  CAF * Q
6.              If (Selected Frame is Not Full)
7.                      Insert(Packet,Selected_Frame)
8.              Else
9.                      Selected_Frame  = (Selected_Frame + 1) mod AF
10.                     Goto 6
11.     Else
12.             Insert (Packet, Common_Queue)
```

Figure 3.3 Virtual Frame Formation in DLSR Scheduling Algorithm

For the formation of the i^{th} frame F_i

1. Select the i^{th} virtual frame V_i.
2. Add_packets (F_i,V_i)
3. If (F_i = FULL)
4.   Transmit (F_i)
5. Else
6. Add_Packet (F_i,Q) till F_i is Full
7. Transmit (F_i)

Figure 3.4 Frame Formation and Dispatch in DLSR Scheduling Algorithm

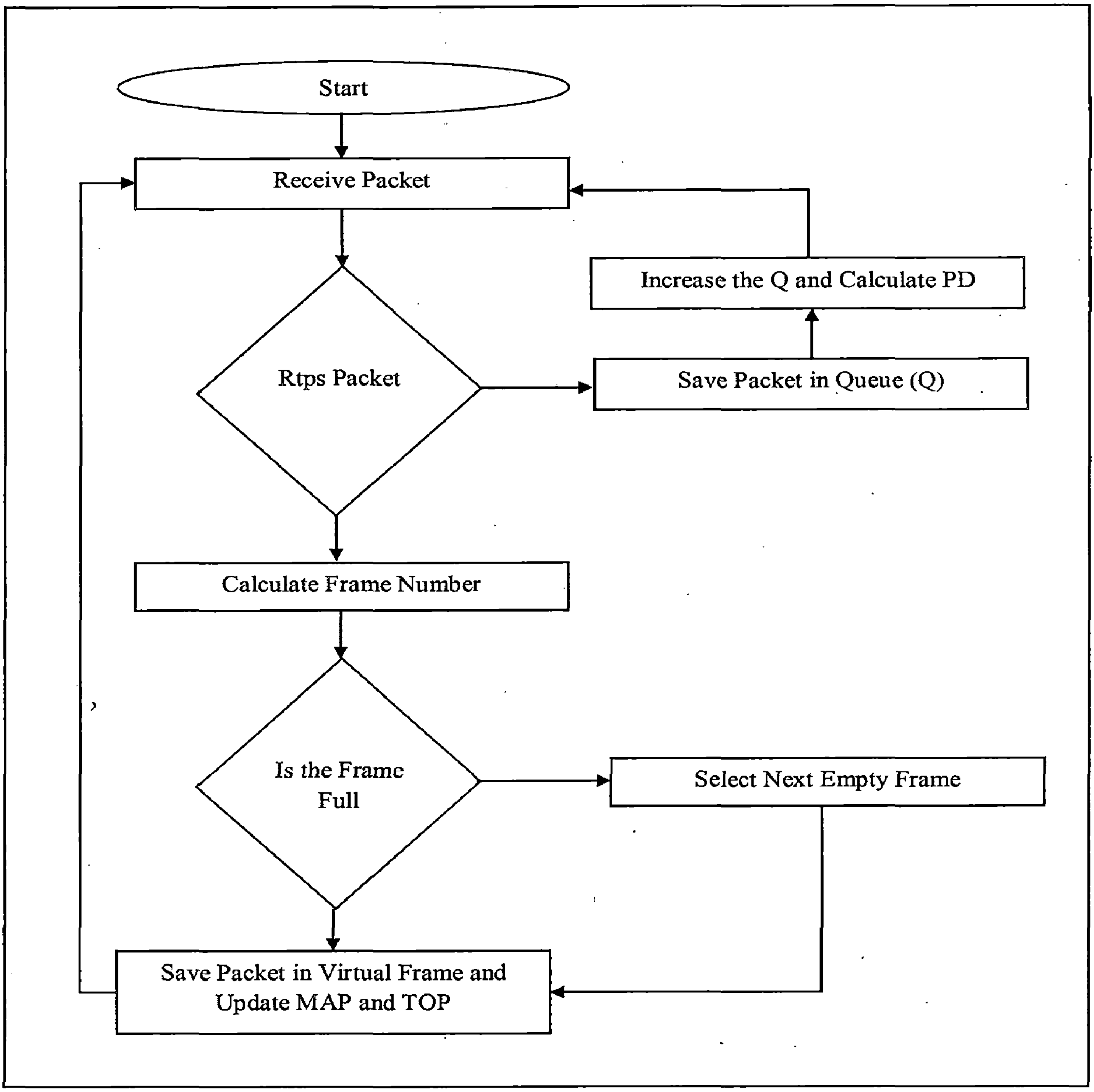

Figure 3.5 DLSR scheduler operational flow chart

Chapter 4

SIMULATION DETAILS

The proposed model has been implemented and simulated in the network simulation tool NS3 [15]. NS3 is also open source and is a relatively new simulator. NS3 provides great flexibility while simulating various scenarios. In this chapter the details of simulation topology and simulation parameters along with assumptions and performance evaluation parameters has been provided.

4.1 Simulation Topology

Figure 4.1 shows the simulated topology for the comparison of our DLSR approach with other approaches. In this simulation there are 20 SSs and 1 BS. Out of these 20 SSs, 10 SSs are made to send packets and act as sender while rest 10 SSs act as receiver. Different applications are mounted on the sender SSs for representing different QoS requirements. These applications are VoIP, IPTV and FTP/Web surfing. VoIP is modeled as constant bit rate generating application, while IPTV is modeled as variable bit rate traffic generating application. FTP and web surfing is also modeled as constant bit rate application for simplicity. The BS is in Point to Multipoint Mode. For generating different simulation environment, the type of application mounted on the SSs has been varied. For example for increasing number of real time connections, the real time application can be mounted in more number of the SSs.

4.2 Simulation Parameters

Table 4.1 shows various parameters under which the simulation has been performed. 10 connections where created having 1 UGS, 1 rtPS, 8 BE service flows. First 3 SSs were started at simulation time 0 while rest SS were started an interval of .5 seconds therefore the traffic increases linearly with the time. First connections is modeled to carry VoIP (on/off) traffic and is using UGS service flow.

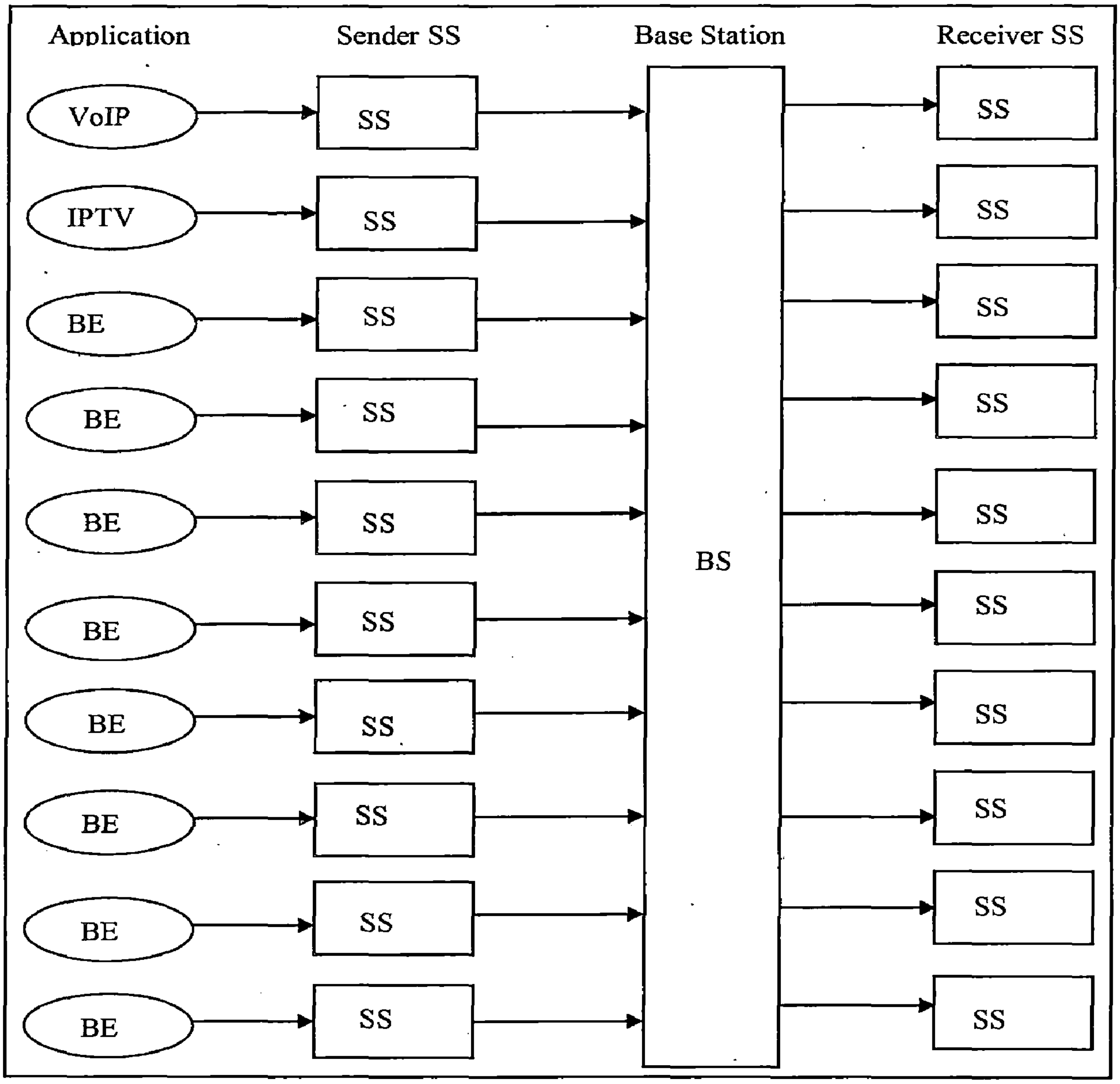

Figure 4.1 Simulation Topology

Second connection using rtPS service flow modeled to carry IPTV traffic which has variable bit rate. Rest all 8 connections are generating simple traffic with a packet size of 100 bytes and packet inter arrival time of 0.1 seconds. For varying the traffic load in the network the packet size and the inter arrival of the packet has been varied as per requirement. Specification of these variations is provided while discussing the results.

The simulation is carried out for 10 seconds in general. This is because till first 5 seconds the traffic is made to increase linearly and after that the load is at its peak for rest of the 5 seconds. This allows us to generate and store various delay and throughput values at varying traffic conditions. The simulation aims at studying the properties of the proposed scheduling algorithm and analyzing their characteristics in a network that has a variety of burst profiles.

Table 4.1 Simulation Parameters

Parameters	Values
Number of BS	1
Number of SS	20
Number of real time connections	2 (1 UGS, 1 rtPS)
No. of non-real time connections	8 (8 BE)
Frame Duration	10 ms
Modulation Type	QAM16_12
Simulation Duration	10 sec
Packet Size	100 bytes
Duplexing	TDD
WiMAX Architecture	Point-to-Multipoint
Simulator Used	NS3

Large number of simulations were carried out under varying traffic load on the network. The DLSR approach is simulated and compared with the 3 other approaches described in next chapter. For UL, a MBQ Uplink scheduler proposed in [17] is used. The variation in the delay, throughput, jitter and average delay for the real time traffic and non real time traffic is discussed in next chapter.

4.3 Simulated Applications

Three different types of applications have been simulated for modeling different types of applications. They are VoIP, IPTV and Best Effort applications.

4.3.1 Voice over Internet Protocol (VoIP)

VoIP is used to transfer real time voice over internet just like telephone or mobile. The data packets that are transmitted are small sized UDP packets. Important QoS parameter related to this application is Maximum Latency, Maximum jitter, Maximum sustained traffic rate. VoIP can be modeled as constant bit rate application generating small sized packets. To model this application in our simulation we generate small sized 50 or 100 bytes packets at a regular interval of .05 seconds.

4.3.2 Internet Protocol Television (IPTV)

IPTV is a variable bit rate application which is used to transfer the real time video packets through internet. This application produces variable sized packets at regular interval. Important QoS parameter related to this application is Maximum Latency, Maximum jitter, Maximum sustained traffic rate, Maximum reserved traffic rate. To model this application in our simulation we generated packets varying from 50 to 500 bytes at a regular interval of .1 seconds.

4.3.3 Best Effort Applications

All other applications like FTP, web browsing etc. are modeled as best effort traffic. These applications do not have any specific QoS requirement. To model these applications we generated packets of size 100 bytes at a regular interval of .1 seconds for simplicity. Variations in packet size and inter arrival time of packets can be made as per requirement. Table 4.2 show the parameter of the various applications which are simulated.

Table 4.2 Simulated Applications and Parameters.

Application	Inter Packet Arrival	Packet size (bytes)	Service Flow used
VoIP	0.05	50	UGS
IPTV	0.1	50 – 500	rtPS
BE Application	0.1	100	BE

4.4 Assumptions

Following assumptions where made while simulating the WiMAX PMP topology.

- It is assumed that there is no packet loss due to the wireless environment. In real there are several problems because of which the packet me get lost. But this assumption will not affect both TCP and UDP specific application because for

TCP the network layer is responsible for retransmitting while for UDP it is the application layer.

- The SS are considered to be static in the simulation. In real SS may be mobile which may degrade the QoS as they move away from the BS because of the change in the modulation scheme at different distances. This is a physical layer issue and is out of the scope of the dissertation. Therefore for getting best results of comparisons of different scheduling algorithms, here static SSs are considered.

4.5 Performance Evaluation Parameters

We compared the performance of various scheduling algorithms on the basis of average delay of the packets of a connection, throughput of a connection and average jitter of connections. Following sections describes how these evaluation parameters are calculated.

4.5.1 Average Delay

Average delay of each connection is accessed and compared as it shows the average time the packets of a connection takes from source to destination. This parameter evaluates that how quickly the packets of a connection are being transmitted on an average.

Average delay is calculated as equation 4.1.

$$\frac{\sum \textit{Delay of the Packets}}{\textit{Total number of Packets Received}} \qquad (4.1)$$

4.5.2 Throughput

Throughput or network throughput is the average rate of successful message delivery over a communication channel. The throughput is usually measured in bits per second (bit/s or bps), and sometimes in data packets per second or data packets per time slot. Here the throughput is calculated as the bytes per seconds. Equation 4.2 shows the calculation for throughput, where Packet Size_i is the packet size of the i^{th} packet reaching the destination, Packet Start_0 is the time when the first packet left the source and Packet Arrival_n is the time when the last packet arrived.

$$Throughput = (\sum i \; Packet\ Size_{i}) \,/\, Packet\ Arrival_{n} - Packet\ Start_{0} \tag{4.2}$$

The throughput is calculated for each connection in consideration. For calculation of the throughput for a particular connection, the time when the first packet is transmitted is recorded. Each time when a packet is received its size is stored. When the last packet is received, sizes of all the packets are added and the sum is divided by the difference of first packet transmitted time and last packet received time. This calculation is done each time when a packet is received thinking that it may be the last. Thus an entry is made to a file each time a packet is received and for the throughput till that time.

4.5.3 Jitter

Delay variation is the variation in the delay introduced by the components along the communication path. It is the variation in the time between packets arriving. Jitter is commonly used as an indicator of consistency and stability of a network. Measuring jitter is critical element to determining the performance of network and the QoS the network offers. Jitter calculation is done on packet basis. For each connection whenever a packet is received, the current time is subtracted for the time when the previous packet was received as per equation 4.3.

$$Jitter = |\,(Rx)_i - (Rx)_{i-1}\,| \tag{4.3}$$

Where $(Rx)_i$: is the time when the i^{th} packet was received.

Chapter 5

RESULTS AND DISCUSSIONS

In this chapter we discuss the results obtained by our DLSR scheduler and compare them with some of the well known approaches. Different scenarios were considered for testing the proposed DLSR approach under varying conditions. The parameters considered for comparison are average throughput, average jitter and average delay. We compared simulation results generated from our DLSR with three different approaches proposed recently. They are Weighted Round Robin (WRR) [4], Deficit Fair Priority Queue (DFPQ) [6] and Downlink Real Time (DLRT) scheduling algorithm [17]. We selected these approaches because they represent different class of scheduling algorithm discussed in chapter 2. We compared the results generated from these approaches with respect to average throughput for both real time and non real time connections while average delay and average jitter parameters are also compared for real time connections. In addition, analysis of variation of throughput with increasing load in the network has also been done for non real time connections. For checking starvation problem of non real time connections we also performed simulation with increasing real time connections in the network.

5.1 Experimental Scenarios

We simulated three different scenarios for conditioning different environment and testing the proposed approach with the existing ones. Scenario-I is simulated for the general comparison of throughput, delay and jitter, scenario-II is simulated for watching the behavior of the non real time connection with increasing packet rate while scenario-III is simulated to check for the starvation of the non real time connection.

5.1.1 Scenario-I

This scenario has been simulated with an aim of comparing different approaches for average throughput, delay and jitter. In this scenario we simulated 2 real time connections and 8 non real time connections. VoIP and IPTV were modeled as 2 real time connections using constant bit rate and variable bit rate respectively. The connection for

VoIP used UGS service-flow while IPTV connection used rtPS service-flow. Maximum latency requirement for both these applications has been set to 100 ms. Other 8 connections were modeled for representing best effort traffic like FTP. These connections used BE service-flow. One of these connections is selected for discussing results. Table 5.1 shows the parameters of the applications simulated for this scenario.

Table 5.1 Scenario-I Application Parameters

Application	**Inter Packet Arrival**	**Packet size (Bytes)**	**Service Flow Used**	**Number of SS Using this**
VoIP	0.05	50	UGS	1
IPTV	0.1	50 – 500	rtPS	1
BE Application	0.1	100	BE	8

5.1.2 Scenario-II

The aim of this scenario is to monitor the throughput variation of the non real time connection with increase in the number of packets sent per second. The application assigned to various SSs in this scenario has been kept same as that of scenario-I, but we simulated 9 different times for different inter-packet arrival (IPA) time for one of the BE connection. A detail of variations of IPA for one of the non-real time connection for different simulation is shown in Table 5.2.

Table 5.2 Scenario-II Parameters

Simulation No.	**IPA** (in secs)
1	1
2	0.5
3	0.1
4	0.08
5	0.06
6	0.05
7	0.04
8	0.03
9	0.02

5.1.3 Scenario-III

The aim of this scenario is to check for the starvation of the non real time connection with increase in the real time traffic in the network. For this we kept the application configuration same as the scenario-I but varied the number of real time connections. Details of this variation are given in Table 5.3. VoIP application has been used to increasing the real time traffic in the network.

Table 5.3 Scenario-III Real Time Connection Variation Details

Simulation No.	No. of real time application	No. of non-real time application
1	1	9
2	2	8
3	3	7
4	4	6
5	5	5
6	6	4
7	7	3
8	8	2
9	9	1

5.2 Scenario-I Results

5.2.1 Throughput Comparison

5.2.1.1 For Application Modeled as CBR

In this simulation, VoIP is modeled as CBR. Figure 5.1 shows the throughput comparison of our DLSR approach and other approaches for real time CBR connections i.e. VoIP traffic. It can be noted that the throughput of the real time CBR connection using our DLSR approach is slightly more (3.8%) as compared to all DLST approach which is best among the other three approaches. This small increase in the throughput is because of the accurate decision taken by our DLSR scheduler for the real time connection packets and thus the bandwidth gets better utilized. DLRT approach stands next to the DLSR approach. While WRR and DFPQ approach shows approximately the same throughput. All the other approaches shows similar type of behavior because they have been specifically developed for maximizing the throughput of the real time connections.

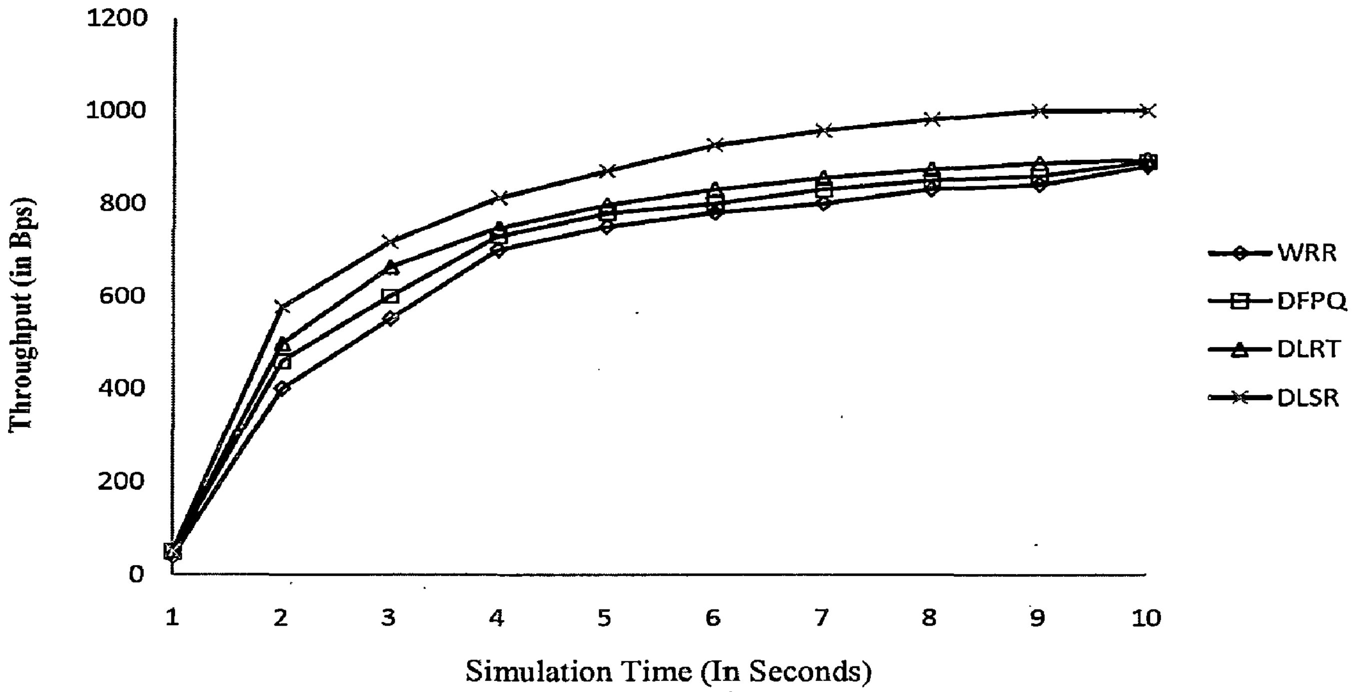

Figure 5.1 Throughput comparison real time CBR connections

5.2.1.2 For Application Modeled as VBR

Throughput Comparison for real time variable bit rate traffic i.e. IPTV is not shown because the variation in the throughput for different simulation is different. This is because throughput largely depends on the size of the packets being transmitted and in VBR connection the packets being transmitted are variable so their throughput varies and thus cannot be compared.

5.2.1.3 For Non Real Time Connections

Figure 5.2 shows the comparison of throughput of non real time connections for various approaches. We observe here that there is drastic increase in throughput of the non real time connection using our approach. On an average there is 46.34% increment in the throughput of the non real time connection. This increase in the throughput is because of the delaying of the real time packets in the BS when the queue length of the non real time connections increases for our DLSR approach. This causes the non real time packets to be transmitted while the real time packets are being delayed. As a result throughput the non real time connections increases

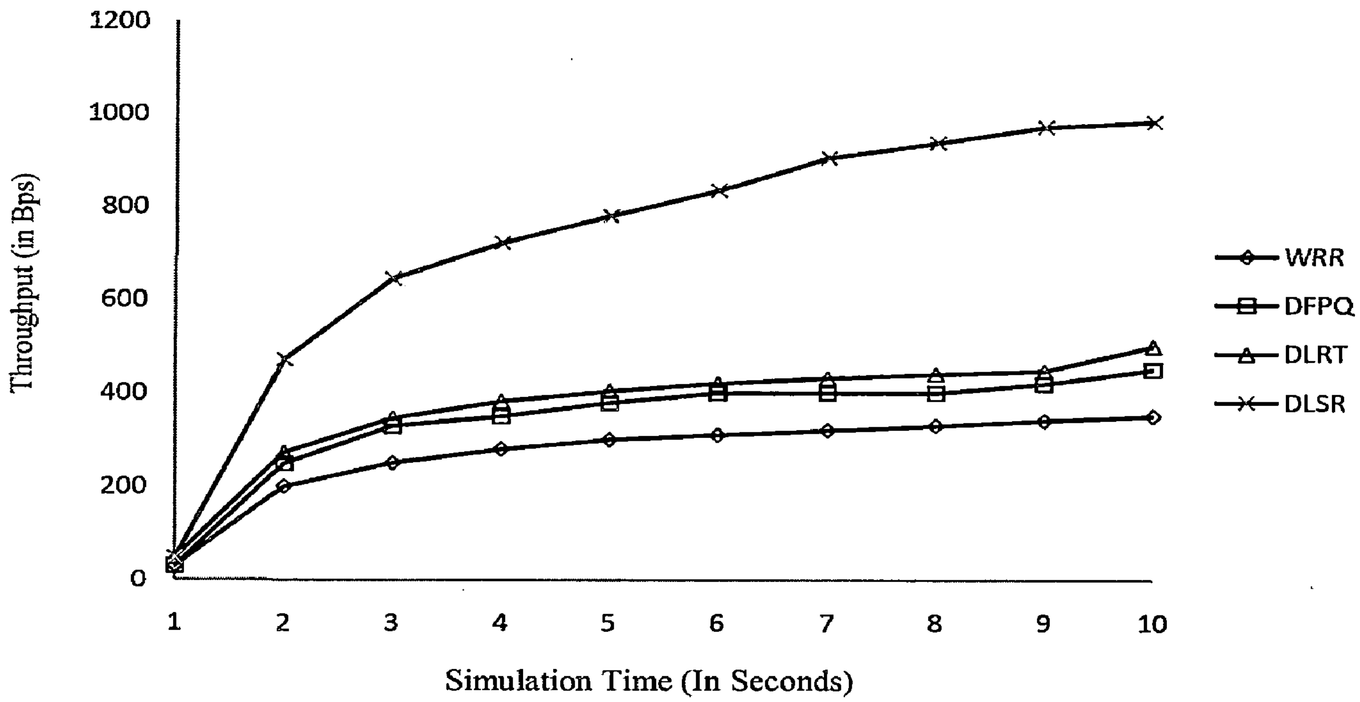

Figure 5.2 Throughput comparison of non real time connections

5.2.2 Average Delay Comparison

5.2.2.1 For Application Modeled as CBR

Figure 5.3 shows the average delay variations of the various CBR connections along with the simulation time. It may be observed that here the average delay of the real time connections have increased for our DLSR approach while the average delay for other approaches for real time connections are very low. But this increase in the delay does not violate the QoS requirements of the real time connections as the highest delay encountered is near 46 ms which is less than 100 ms (Max. Latency). This increase in average delay is again because of the delaying of the packets in the frames to be transmitted.

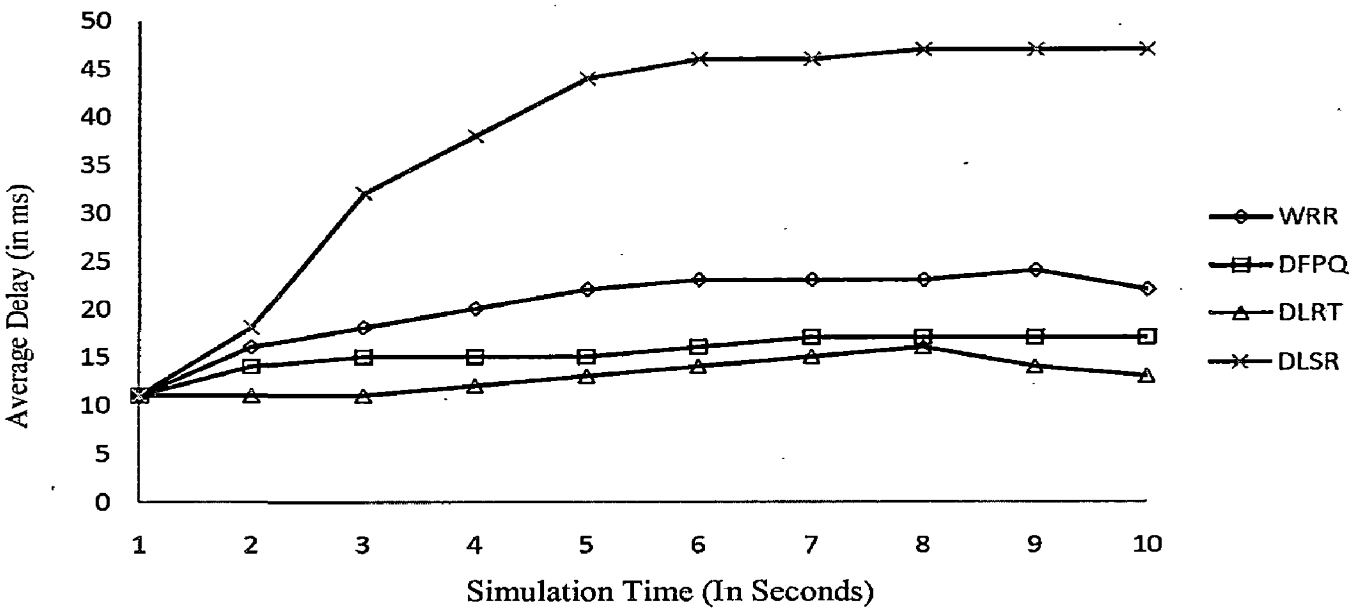

Figure 5.3 Average delay comparison of real time CBR connections

5.2.2.2 For Application Modeled as VBR

Figure 5.4 shows the average delay variations of the various VBR connections along with the simulation time. Similar kind of variation can be observed between our DLSR and other approaches. The reason for this variation in delay is again being the same as that for the CBR real time connections.

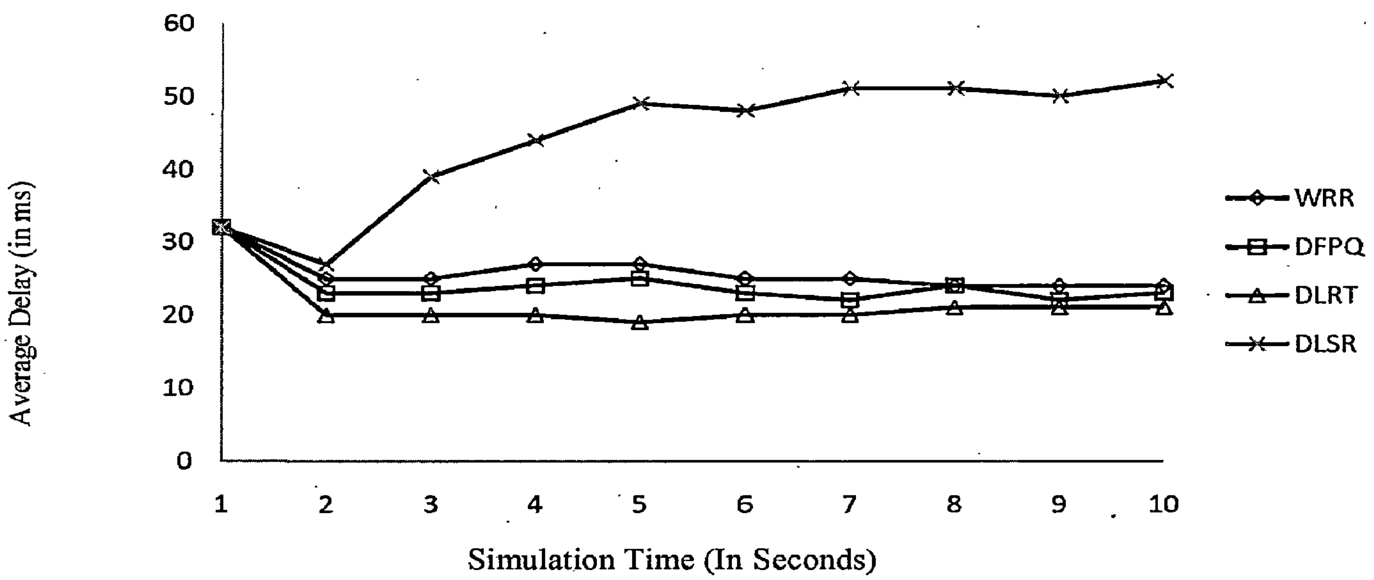

Figure 5.4 Average delay comparison of real time VBR connections

5.2.3 Average Jitter Comparisons

Figure 5.5 shows the comparison of average jitter of real time connections for various approaches. We observe here that for our DLSR approach the average jitter is quite high as compared other approaches (averaged 20.43 ms). This is because the fact that there is no regular pattern in which the packets of a real time connection are selected. They are allocated the some virtual frame as soon as they come to the BS independent of the previous packet of the same connection. But in other approaches the packets are generally taken from queue so most of the time more than one packet gets selected and transmitted in continuation in the same frame. So the average jitter for these approaches is very low. But in our DLSR approach we consider the "Maximum Latency" QoS parameter as the limit for selecting a frame for the real time packets. But if the maximum jitter QoS is less than that of the Maximum Latency than we select Maximum Jitter value as a limit for selecting the virtual frame. During our simulation there was no such type of application whose jitter requirement were more than the latency requirement.

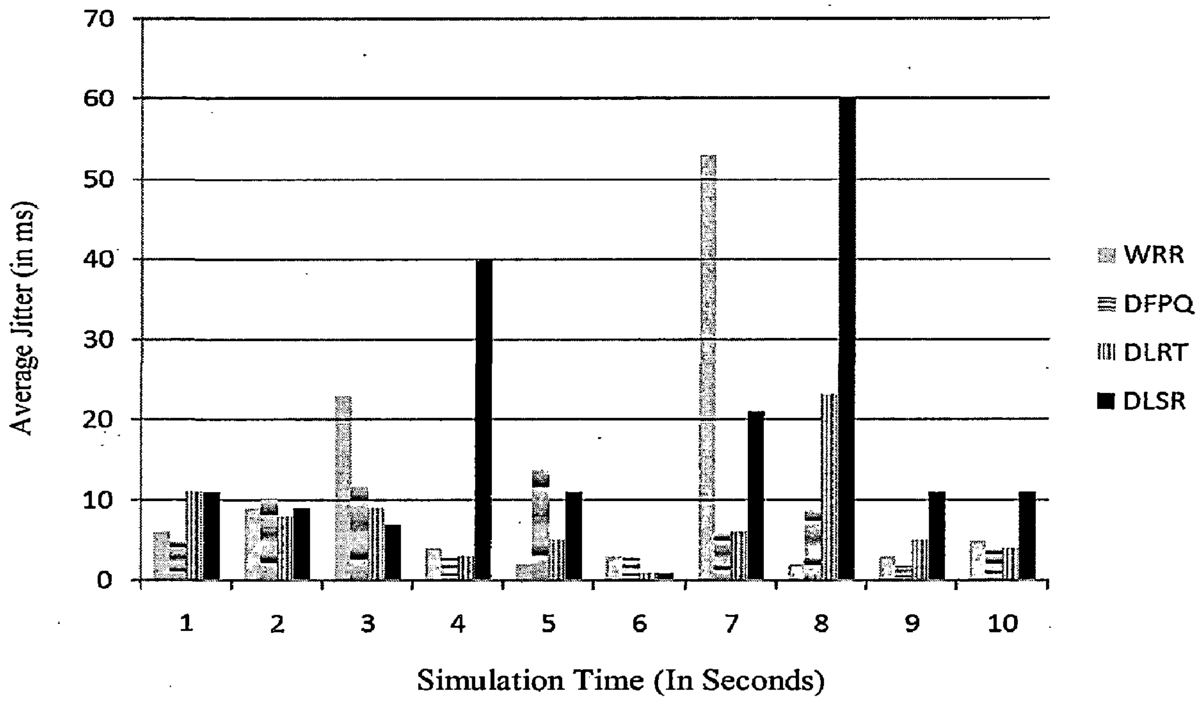

Figure 5.5 Average jitter comparison of real time connections

5.3 Scenario-II Results

(Throughput variation of the non real time connections for varying traffic load.)

Figure 5.6 shows the variation of the throughput of one of the non-real time connection with the increase in the data rate of the applications using that connection. It may be observed here that with low data rate the throughput of the non real time connection for all the approaches are same. But with the increase in the data rate of the non-real time connection, the throughput, using our DLSR scheduler increases almost twice as that of other approaches. When the inter packet arrival is set to ..03 seconds for non real time packets the throughput is double as compared to the throughput of the DLRT approach. On an average there is 55.32 percent of throughput increment for the non real time traffic.

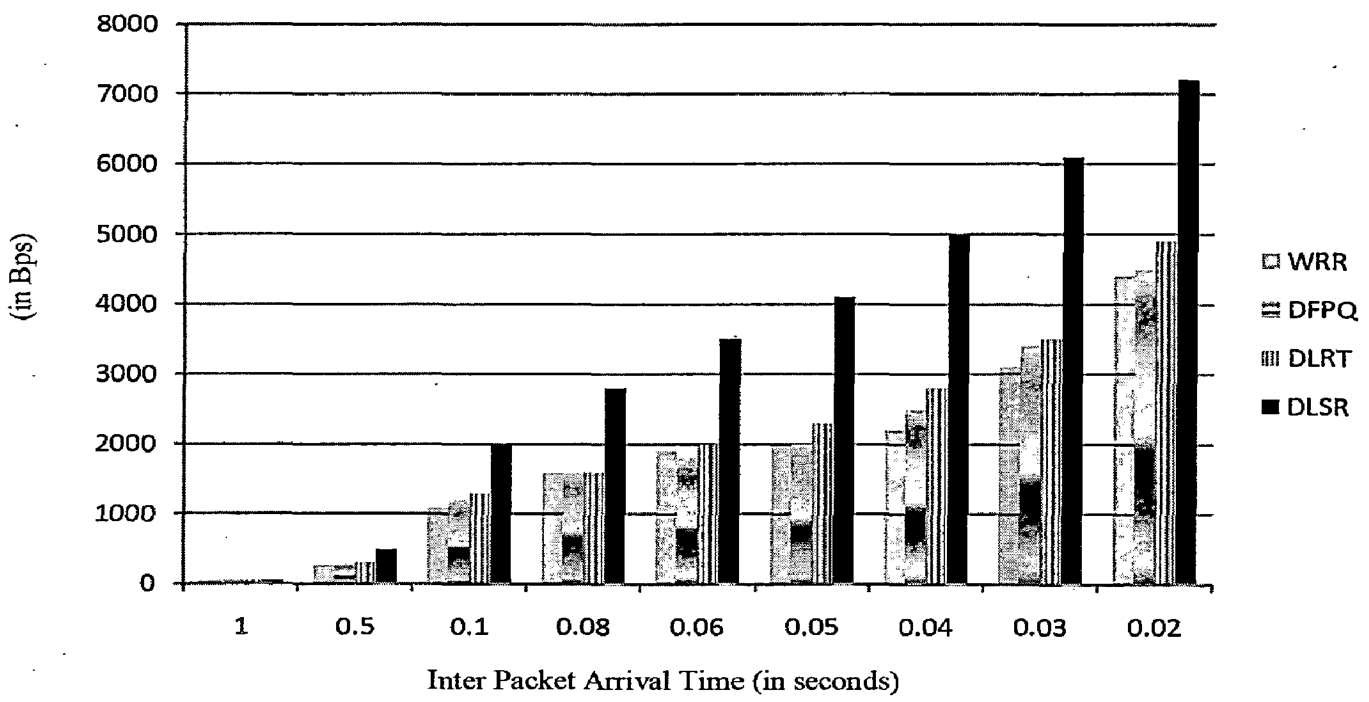

Figure. 5.6. Throughput Comparison of the non-real real time connection with the increasing traffic load.

5.4 Scenario-III Results

(Throughput variation of non-real connections with increasing real time traffic.)

A simulation is performed in which the numbers of real time connections were increased with time having only 1 non real time connection constant. Figure 5.7 shows the variation of the throughput of the non real time connection with increase in the real time connections. It may be observed that for starting 5 connections the throughput of the non real time connection was constant. After that it starts to drop with increasing number of the non real time connections. This is because the amount of the bandwidth required by the real time traffic increased and there is less bandwidth available for the real time connection. Because of this the number of frames in which the packet can be delayed reduces and so less opportunity is given to the non-real time connections. But still some bandwidth is created by delaying the real time packets for 1 or 2 frames and thus very small amount of space is created of the non real time connection. Therefore the non-real time connection never gets starve in this situation.

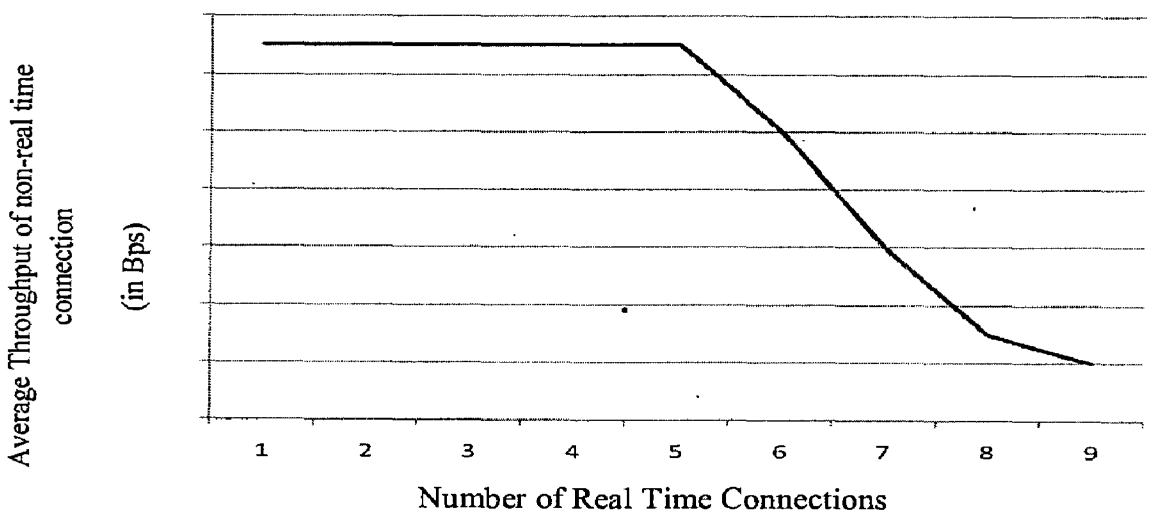

Figure. 5.7. Variation of the non-real time connection throughput with increasing number of real time connections.

\nalysis of MAP size

present at the starting of each node. It is the summary of the frame. It defines ıd what type of information is present (Uplink/downlink), which SS should listen ı part of the frame. Figure 5.8 shows 2 different cases where the map size is . for transmitting same data to same SS.

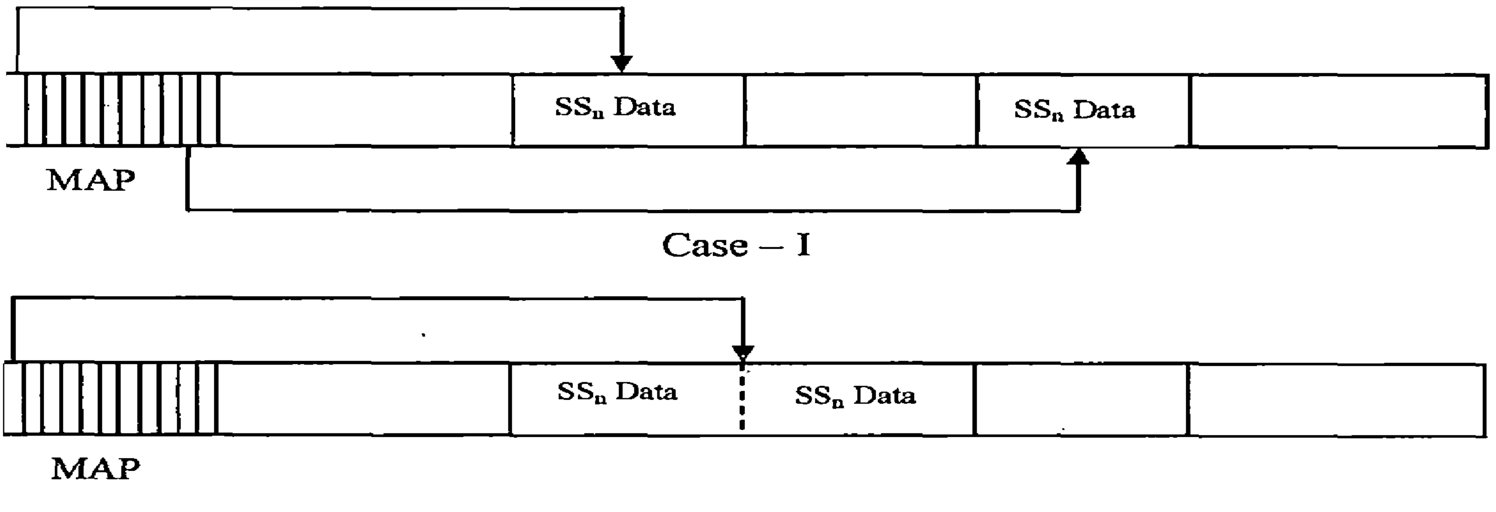

Figure 5.8 Figure showing the different cases of bandwidth allocation

– I of figure 5.8 two data packets destined for same SS are being transmitted at : parts of the frame i.e. their transmission is not continuous. Due to this BS has to two different information units in the MAP so that the SS can read the data for it. While in the second case the data being transmitted one after the other ›usly therefore BS sends only one unit of information for both data packets. Thus ase MAP size will be lesser as compared to the case–I and more of the space will for the data transmission. As a result less bandwidth is wasted in case-II.

pproach the slots are allocated to a packet of a connection independent of the of the previous packet of the same connection. Thus even if there are two of same connection continuously in a frame, the BS will issue two units of :ion in the MAP of the frame. Figure 5.9 shows this case. Thus little amount of ıth gets wasted.

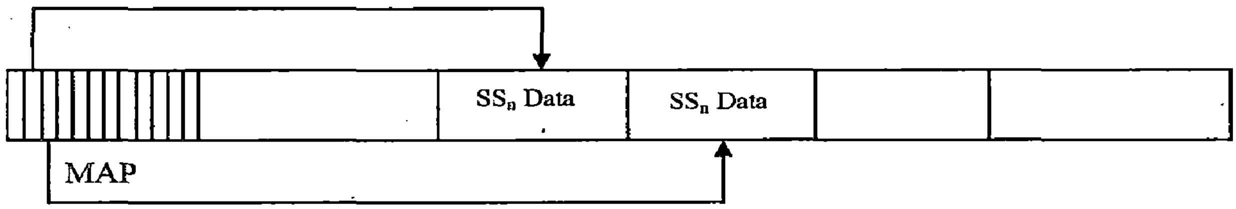

Figure 5.9 Figure showing the unnecessary unit of information being sent in the MAP

One possible solution to this approach is group together the packets of each connection and then select a frame for them. This in fact is the concept of storing the packets in the queues which is used by all other approaches. Queuing or grouping of the packets will not allow the instantaneous allocation of slots of the virtual frame to the packets. Thus this solution cannot be implemented with our DLSR approach. Till now the issue of MAP size is a limitation to this approach. But despite of this limitation and small wastage of the bandwidth in carrying extra bit of information the results in previous section shows that DLSR is the best approach in proper utilization of the bandwidth as the throughput of both the real time and non-real time traffic has increased. This is because probability of transmitting packets of the same connection in the same frame is very low under high traffic.

5.6 Complexity Analysis

Complexity of WiMAX scheduling algorithm is determined by the time taken to construct a frame. Generally in other approaches the algorithm selects packets from the queues either in one go or looping through the queues. This makes their complexity O(n) or O(n^2) respectively. But in our approach we never loop through the queues as we are not considering queues for the real time packets and considering a common queue for the non real time connections.

We store the packets in the virtual frames as soon as they come. These virtual frames will be converted into actual frames at the time of their transmission. In this way we are simultaneously constructing many frames. Suppose *m* frames are being constructed simultaneously and each frame has *n* packets. Under normal conditions these n frames will be having $n/2$ real time packets and $n/2$ non real time packets. So the calculation required to construct frame for real time packets will be $O(n/2)$ while that for non real time packets will be O(1) as all the non real time packets will be taken from the same queue.

Thus the complexity of our DLSR scheduler will be **O(n/2) + O(1)** per frame. Or it can also be written as **O(n/2) per frame**. Generally the frame time is between 10 ms to 100 ms in which number of packets will vary from 100 to 1000. Thus the complexity will be between O(50) to O(500) depending upon frame duration. In this way our DLSR approach is computationally very simple and has very low time complexity. Table 5.4 summarizes the results obtained after the simulations.

Table 5.4 Summary of the results obtained

Parameter	WRR	DFPQ	DLRT	DLSR
Average Delay	14 ms	12 ms	9 ms	**40 ms**
Avg real time Throughput	650 Bps	680 Bps	770 Bps	**810 Bps**
Avg Non Real time Throughput	260 Bps	390 Bps	410 Bps	**989 Bps**
Avg real time Jitter	10.32 ms	8.50 ms	5.4 ms	**30.54 ms**
Starvation	Yes	Yes	Yes	**No**
Complexity	O(n)	$O(n^2)$	O(n)	**O(n/2)**

Table 5.5 shows various improvement achieved by our DLSR approach.

Table 5.5 Improvement achieved by DLSR approach

Parameter	Improvement (Average)
Real Time Throughput	3.8 %
Non Real Time Throughput	55.32 %
Complexity	50 %

Chapter 6

CONCLUSIONS AND FUTURE WORK

6.1 Conclusions

In this work an approach for providing QoS at the MAC layer of the WiMAX and to optimize the bandwidth usage has been proposed. The proposed BS Downlink Slot Reservation (DLSR) approach considers the queue length of the non real time packets as a factor for deciding when to transmit the real time packet. This decision is taken in such a way that the QoS requirement of the real time traffic is not violated and non real time packets gets opportunity for transmission, frequently. The simulation results show that the proposed approach increases the throughput of the non real time connections up to 55.32% on an average. Though there is increase in the average delay of the real time connections but this does not affect their QoS requirements. Also the complexity of the proposed algorithm is O(n/2) which is lowest of all other proposed approaches.

The following conclusions can be made from the results obtained using the proposed DLSR approach:

- The proposed approach is able to increase the throughput of the non-real time connections to almost double under normal traffic conditions. This increase in throughput is at the cost of increase in the average delay of the real time connections to an extent which does not degrade their QoS requirements.
- The proposed approach minimizes the probability of the starvation of the non real time connections even in high traffic load in the network.
- The proposed approach has less computational overhead. It is also fast in terms of constructing a frame as at the time of dispatch of a frame it is already half (approx) filled.
- The proposed approach is able to accurately allocate the bandwidth to different connections as the decision of allocation of bandwidth is being taken on arrival of every real time packet.

In this way our proposed approach optimizes the usage of the bandwidth, increase the throughput of both real time and non real time connections and reduces starvation of non-real time connections keeping the computational overhead low.

6.2 Future Works

The future work includes following areas:

1. The uplink bandwidth usage can be optimized and combined to our DLSR approach for downlink with it. We hope that the combination of the two approaches would produce better results and may result in more efficient usage of bandwidth in WiMAX networks.
2. Another field could be towards managing connection establishment of SS with BS. Due to the contention method used during initial connection setup much time and bandwidth is wasted. Some approach can be found to reduce the connection setup time and bandwidth wastage.

REFERENCES

[1] IEEE 802.16-2009. "IEEE Standard for Local and Metropolitan AreaNetworks - Part 16: Air Interface for Fixed Broadband Wireless Access Systems" IEEE Std., Rev. IEEE Std802.16-2009, (Oct 2004).

[2] WimaxTutorials Available: http://www.tutorialsweb.com/wimax/wimax.htm

[3] C. Cicconetti, L. Lenzini, E. Mingozzi, and C. Eklund, "*Quality of service support in IEEE 802.16 networks,*" IEEE Network, vol. 20, pp. 50-55, 2006.

[4] A. K. F. Khattab and K. M. F. Elsayed, "*Opportunistic scheduling of delay sensitive traffic in OFDMA-based wireless networks,*" in Proc. Int. Symp. World of Wirless Mobile and Multimedia Networks., Buffalo,NY, 2006, pp. 10-19.

[5] S. Kim and I. Yeom. "*TCP-aware uplink scheduling for IEEE 802.16,*" In IEEE Communications Letters, Volume 11, pages 146-148, 2007.

[6] Lian-ying Z., Wei-wei J., "*A Study on the Application of WiMAX Access Technology in IPTV System,*" In Second International Conference on Networks Security, Wireless Communications and Trusted Computing, (NSWCTC '10), Volume 02, pp. 43-47, 2010

[7] M. Ma, P. Xie, S. K. Bose, and S. Maag, "*An efficient scheduling algorithm for QoS provisioning in WiMAX networks,*" In IEEE Communications Letters, vol. 04, pp. 240-245, 2008.

[8] J. Chen, W. Jiao, and H. Wang, "*A service flow management strategy for IEEE 802.16 broadband wireless access systems in TDD mode.*" IEEE Network, vol. 23, pp. 50-65, 2006

[9] C. Cicconetti, A. Erta, L. Lenzini, and E. Mingozzi "*Performance Evaluation of the IEEE 802.16 MAC for QoS Support*", In IEEE TMC07, Volume 6, pages 26-38, 2007.

[10] P. C. Ting, C. Y. Yu, N. Chilamkurti, W. Tung-Hsien, and C. K. Shieh, "*A proposed RED-based scheduling scheme for QoS in WiMAX networks,*" In IEEE Communications Letters, vol. 04 pp. 48-52, 2009.

[11] S. A. Xergias, N. Passas, and L. Marekos, "*Flexible Resource Allocation in IEEE 802.16 Wireless Metropolitan Area Networks*," In 14th IEEE Workshop on Local and Metropolitan Area Networks, LANMAN 2005, Chania, Greece, 18 - 21 September, pp. 45-50 2005.

[12] M. Ma, J. Lu, S.K. Bose, "*A three-tier framework and scheduling to support QoS service in WiMAX*", In IEEE Communications Letters, vol. 04, pp. 50-65, 2009.

[13] R. Mukul, P. Singh, D. Jayaram, D. Das, N. Sreenivasulu, K. Vinay, and A. Ramamoorthly, "*An Adaptive Bandwidth Request Mechanism for QoS Enhancement in WiMax Real Time Communication*," In Wireless and Optical Communications Networks, IFIP International Conference on, Bangalore, India, pp. 11 – 13, 2006.

[14] Shih-Wei L., Szu-Lin W. and Jiann-Liang C., "*Adaptive Cross-layer QoS Mechanism for Cognitive Network Applications*" In IEEE Network, vol. 10, pp. 20-25, 2008

[15] Ns-3 tutorials [Online]. Available: http://www.nsnam.org/ns-3.10/tutorial/

[16] WiMAX Forum, "WiMAX System Evaluation Methodology V2.1," Jul. 2008, 230 pp. Available: http://www.wimaxforum.org/technology/documents/

[17] Freitag, J.; da Fonseca, N.L.S., "*Uplink Scheduling with Quality of Service in IEEE 802.16 Networks*," In Global Telecommunications Conference, 2007. GLOBECOM '07. vol. no. 6, pp.2503-2508, 26-30 Nov. 2007

[18] Dosciatti E.R., Godoy W., Foronda, A. "*A New Scheduler for IEEE 802.16 with Delay Bound Guarantee,*" In Networking and Services (ICNS), 2010 Sixth International Conference , pp. 150 – 155, 2010

[19] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, P. Whiting, and R. Vijayakumar, "Providing quality of service over a shared wireless link," In IEEE Commun. Mag., vol. 39, pp. 150-154, Feb. 2001.

[20] P. Parag, S. Bhashyam, and R. Aravind, "*A subcarrier allocation algorithm for OFDMA using buffer and channel state information*," In Proc. IEEE Vehicular Technology Conf., Dallas, TX, vol. 1, pp. 622-625, 2005.

[21] S. Ryu, B. Ryu, H. Seo, and M. Shi, "*Urgency and efficiency based wireless downlink packet scheduling algorithm in OFDMA system*," In Proc. IEEE Vehicular Technology Conf., Stockholm, Sweden, 2005, vol. 3, pp. 1456-1462.

[22] W. Park, S. Cho, and S. Bahk, "*Scheduler Design for Multiple Traffic Classes in OFDMA Networks,*" In Proc. IEEE Int. Conf. Communications., Istanbul, Turkey, 2006, vol. 2, pp. 790-795.

[23] P. Viswanath, D. Tse, and R. Laroia, "*Opportunistic beamforming using dumb antennas*," In IEEE Trans. Inf. Theory, vol. 48, pp 1277-1294, Jun. 2002.

[24] S. Shakkottai, R. Srikant, and A. Stolyar, "*Pathwise Optimality and State Space Collapse for the Exponential Rule,*" In Proc. IEEE Int. Symp. Information Theory., 2002, pp. 379.

[25] V. Singh and V. Sharma, "*Efficient and Fair Scheduling of Uplink and Downlink in IEEE 802.16 OFDMA Networks*," In Proc. IEEE Wireless Communication and Networking Conf., Las Vegas, NV, 2006, vol. 2, pp. 984-990.

[26] Y. J. Zhang and S. C. Liew, "*Link-adaptive largest-weighted-throughput packet scheduling for real-time traffics in wireless OFDM networks*," In Proc. IEEE Global Telecommunications Conf., St. Louis, MO, 2005, vol. 5, pp. 5-9.

[27] Z. Liang, Y. Huat Chew, and C. Chung Ko, "*A Linear Programming Solution to Subcarrier, Bit and Power Allocation for Multicell OFDMA Systems*," In Proc. IEEE Wireless Communication and Networking Conf., Las Vegas, NV, 2008, pp. 1273-1278.

[28] Z. Yan, "*Performance Modeling of Energy Management Mechanism in IEEE 802.16e Mobile WiMAX*," In Proc. IEEE Wireless Communication and Networking Conf., Hong Kong, 2007, pp. 3205- 3209.

[29] X. Yang, "*Performance analysis of an energy saving mechanism in the IEEE 802.16e wireless MAN*," In Proc. IEEE Consumer Communications and Networking Conf., Las Vegas, NV, 2006, pp. 406- 410.

[30] J. Shi, G. Fang, Y. Sun, J. Zhou, Z. Li, and E. Dutkiewicz, "*Improving Mobile Station Energy Efficiency in IEEE 802.16e WMAN by Burst Scheduling*," In

Proc. IEEE Global Telecomunications Conf., San Francisco, CA, 2006, pp. 1-5.

[31] L. Tian, Y. Yang, J. Shi, E. Dutkiewicz, and G. Fang, "*Energy Efficient ntegrated Scheduling of Unicast and Multicast Traffic in 802.16e WMANs,*" In Proc. IEEE Global Telecomunications Conf., Washington, DC, 2007, pp. 3478-3482.

[32] H. Choi and D. Cho, "*Hybrid Energy-Saving Algorithm Considering Silent Periods of VoIP Traffic for Mobile WiMAX,*" In Proc. IEEE Int. Conf. Communications., Glasgow, Scotland, 2007, pp. 5951-5956.

[33] A. Sayenko, O. Alanen, and T. Hamalainen, "*ARQ Aware Scheduling for the IEEE 802.16 Base Station,*" In Proc. IEEE Int. Conf. Communications., Beijing, China, 2008, pp. 2667-2673.

[34] F. Hou, J. She, and P. Ho, and X. Shen, "*Performance Analysis of ARQ with Opportunistic Scheduling in IEEE 802.16 Networks,*" In Proc. IEEE Global Telecomunications Conf., Washington, DC, 2007, pp 4759- 4763.

LIST OF PUBLICATIONS

[1] Hemshankar Sahu and Anjali sardana. *"A Novel Slot Reservation Based Scheduling Algorithm Framework for optimized Bandwidth usage in IEEE 802.16 (WiMAX) Networks,"* The Third International Conference on Wireless & Mobile Networks (WiMoN-2011), Publisher: LNCS Springer (Accepted).

[2] Hemshankar Sahu and Anjali sardana. *"A Novel Downlink Slot Reservation (DLSR) Scheduling Algorithm for optimized Bandwidth Usage in IEEE 802.16 (WiMAX) Networks,"* 2nd International Conference on Computer and Communication Technology, (ICCCT-2011) Publisher: IEEE (Communicated).