Design and Implementation of a QoS-included WiMAX Module for NS-2 Simulator

Aymen Belghith
TELECOM Bretagne
2 rue de la Châtaigneraie, CS 17607, 35576
Cesson-Sévigné, France
aymen.belghith@telecom-bretagne.eu

Loutfi Nuaymi
TELECOM Bretagne
2 rue de la Châtaigneraie, CS 17607, 35576
Cesson-Sévigné, France
loutfi.nuaymi@telecom-bretagne.eu

ABSTRACT

WiMAX is a very promising technology. The main promises are the high throughput and the large coverage. As for all new technologies, performance studies are required. The network simulation is considered as a solution to test the performance of technologies and especially the wireless networks. In this paper, we propose a new WiMAX module for NS-2 simulator. This module is based on the NIST implementation of WiMAX [5]. Our contribution consists of the addition of the QoS classes as well as the management of the QoS requirements, unicast and contention request opportunities mechanisms, and scheduling algorithms for the UGS, rtPS and BE QoS classes. Simulation results provide interesting observations about the throughput and mean sojourn time values. Simulation results show that our UGS, rtPS, and BE schedulers are in accordance with the specification of the QoS classes defined in the IEEE 802.16 standard.

Categories and Subject Descriptors

I.6.5 [**Simulation and Modeling**]: Model Development—*modeling methodologies*

General Terms

Design, performance

Keywords

Implementation, QoS classes, scheduling, simulation, WiMAX

1. INTRODUCTION

Worldwide Interoperability for Microwave Access (WiMAX) is based on 802.16-2004 standard [12] and its amendment 802.16e [13]. It is a Broadband Wireless Access (BWA) technology that promises a large coverage and high throughput. Theoretically, the coverage range can reach 30 miles and the throughput can achieve 75 Mbit/s [1]. Yet, in practice the maximum coverage range observed is about 20 km and the data throughput can reach 9 Mbit/s using User Datagram Protocol (UDP) and 5 Mbit/s using File Transfer Protocol (FTP) over Transmission Control Protocol (TCP) [2]. The theoretical values do not always fit the reality. The network simulation presents a solution to test the performance of technologies. Network Simulator 2 (NS-2) [4] is a widely-used tool to simulate wireless networks. Until today, this simulator does not implement a definitive WiMAX module. Nevertheless, there are some WiMAX NS-2 modules implemented by National Institute of Standards and Technology (NIST) [5] and Network and Distributed Systems Laboratory (NDSL) [3] and [9]. These modules implement the physical (PHY) and Medium Access Control (MAC) layers of a WiMAX system. The NIST module implements the Orthogonal Frequency Division Multiplexing (OFDM) PHY while the NDSL module implements the Orthogonal Frequency Division Multiple Access (OFDMA) PHY. Both PHY modules use the same duplexing technique: Time Division Duplexing (TDD). The MAC layer of these WiMAX modules contains the management messages. The NIST WiMAX module also supports mobility, fragmentation, and reassembly of frames. The NDSL WiMAX module supports Call Admission Control (CAC).

As we consider the Wireless-OFDM PHY layer, our NS-2 developments are based on the NIST implementation. Our contribution consists of the addition of QoS classes and their requirements, mechanisms specified by the IEEE 802.16 standard, and some scheduling algorithms. These implemented schedulers take into account the QoS classes. We use the Unified Modeling Language (UML) to design our WiMAX module. UML class and sequence diagrams are introduced in the following sections to model our system in a visual manner. Then, we assess our implementation and compare our implemented scheduling algorithms with the Round Robin (RR) scheduler of the existing module. The comparison is based on two parameters: the throughput and mean sojourn time. The throughput is the amount of data that can be sent in the network per one second. The mean sojourn time is the average time a data packet spends from its generation to its delivery at the destination. Simulation results are given for three scenarios in order to compare scheduling algorithms for three different QoS classes.

This paper is organized as follows. In Section II, we present an overview of the WiMAX technology. In Section III, we describe our NS-2 module as well as a brief presentation of the existing WiMAX module. In Section IV, we present the simulation model and some simulation results. We conclude in Section V.

SIMUTOOLS 2008, March 03-07, Marseille, France

DOI 10.4108/ICST.SIMUTOOLS2008.2958

2. OVERVIEW OF WIMAX

The IEEE 802.16 standard defines four QoS classes: Unsolicited Grant Service (UGS), real-time Polling Service (rtPS), non-real-time Polling Service (nrtPS), and Best Effort (BE). The IEEE 802.16e amendment added a fifth QoS class, called extended real-time Polling Service (ertPS). The five defined QoS classes are descried as follows.

UGS supports real-time service flows that have fixed-size data packets on a periodic basis. The BS provides grants in unsolicited manner. The UGS subscribers are prohibited from using contention request opportunities.

rtPS supports real-time service flows that have variable size data packets on a periodic basis. The BS periodically provides unicast request opportunities in order to allow the SS to specify the desired bandwidth allocation. The SS is prohibited from using contention request opportunities.

ertPS supports real-time service flows. It is built on the efficiency of both UGS and rtPS. The BS provides unicast grants in an unsolicited manner like UGS. Whereas the UGS allocations are fixed in size, the ertPS allocations are dynamic. Then, the SS can request to change the size of grants by sending bandwidth change request.

nrtPS is designed to support non real-time service flows that have variable size data packets on a periodic basis. The SS can use contention request opportunities to send a bandwidth request with contention. The SS can also provide unicast request opportunities.

BE is used for best effort traffic where no throughput or delay guarantees are provided. The SS can use unicast request opportunities as well as contention request opportunities.

When the BS or the SS creates a connection, it associates the connection with a service. A service flow provides unidirectional transport of packets either to uplink packets that are transmitted by the Subscriber Station (SS) or to downlink packets that are transmitted by the Base Station (BS). It is characterized by a set of parameters as a Service Flow identifier (SFID), service class name (UGS, rtPS, ertPS, nrtPS, or BE), and QoS parameters (such as Maximum sustained traffic rate, minimum reserved traffic rate, and maximum latency).

There are three kinds of dynamic service management messages. Dynamic Service Addition (DSA) for the addition of a new service flow, Dynamic Service Change (DSC) for the modification of service flow parameters, and Dynamic Service Delete (DSD) for the deletion of an existing flow service.

There are three DSA management messages: Dynamic Service Addition Request (DSA-REQ), Dynamic Service Addition Response (DSA-RSP), and Dynamic Service Addition Acknowledgment (DSA-ACK). These messages are used in order to create a new service flow between an SS and a BS, respectively. The addition of a new service flow can be initialized by the SS or the BS (see Figure 1). Once a service flow is created, some or all of its parameters can be changed by the SS or the BS across DSC management messages: Dynamic Service Change Request (DSC-REQ), Dynamic Service Change Response (DSC-RSP) and Dynamic Service Change Acknowledgment (DSC-ACK). A service flow can also be deleted across the exchange of two DSD management messages: Dynamic Service Delete Request (DSD-REQ), Dynamic Service Delete Response (DSD-RSP). The acknowledgment of DSD-RSP is not required.

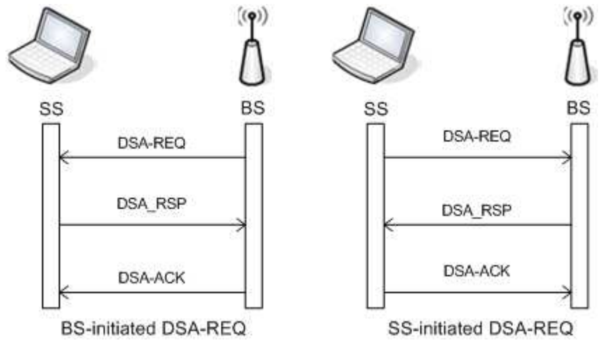

Figure 1: Addition of a new service flow.

As mentioned, the IEEE 802.16 MAC layer defines the different QoS classes and its requirements, QoS parameters, and management messages. However, it keeps scheduling algorithm(s) to be used as an open issue. Some well-known schedulers were proposed for WiMAX such as the RR scheduler in [7], the Earliest Deadline First (EDF), Weighted Round Robin (WRR), and RR schedulers in [8], and the RR and maximum Signal-to-Interference Ratio (mSIR) scheduler in [6]. There are also some schedulers specifically proposed for WiMAX such as Temporary Removal Scheduler (TRS) [6], Opportunistic Deficit Round Robin (O-DRR) [11], adaptive bandwidth request mechanism [10], and packet scheduling using the token bucket [14].

The IEEE 802.16 PHY layer defines two modes of transmission: the point-to-multipoint (PMP) and Mesh modes. In the PMP mode, the traffic occurs only between BS and SSs. In the Mesh mode, traffic can also occur between SSs. In this work, we consider the PMP mode. The PHY layer supports two duplexing techniques TDD and Frequency Division Duplexing (FDD). In the TDD mode, the uplink and downlink data are transmitted with the same frequency. In the FDD mode, the uplink and downlink data are transmitted in two different frequencies. We consider only the TDD mode.

The IEEE 802.16 standard also defines the Modulation Coding Schemes (MCS) that can be used. This allows the BS and SS a large choice of the suitable MCS used. The MCS used depends on the channel quality. When the BS performs the link adaptation, it compares the Signal-to-Noise Ratio (SNR) value of the subscriber with thresholds in order to select the Modulation and Coding Scheme (MCS) used. There are two kinds of thresholds:

- The minimum entry threshold: represents the minimum SNR required to start using an MCS that is more efficient.
- The mandatory exit threshold: represents the SNR below which the current MCS can no longer be used and it has to start using an MCS that is more robust.

3. WIMAX NS-2 MODULE

We have implemented a new NS-2 WiMAX module. The version of NS-2 is 2.29 and the programming language is C++. Our module is based on the WiMAX NIST module [5]. The existing module implements the OFDM PHY and TDD MAC layers. The PHY layer has some configurable parameters such as transmission power, cyclic prefix,

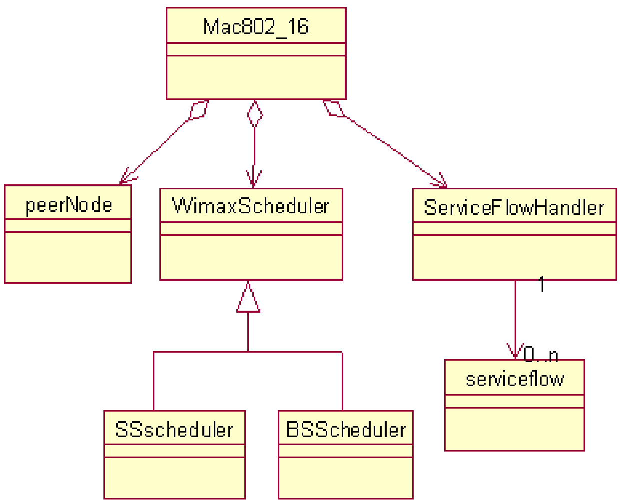

Figure 2: MAC 802.16 class diagram [14].

frequency bandwidth, and MCS. It computes some values such as sampling frequency, OFDM symbol time duration, and transmission time for a packet according to its size and MCS used.

The MAC layer contains some MAC management messages such as Downlink Channel Descriptor (DCD), Uplink Channel Descriptor (UCD), Downlink MAP (DL-MAP), Uplink MAP (UL-MAP), ranging request, ranging response, registration request, and registration response. One downlink and one uplink data connection can be added per subscriber. The BS performs the RR scheduler to allocate radio resources for the uplink connections. The existing implementation supports the mobility. The scanning and handover operations are implemented.

Our contribution consists of the addition of some QoS parameters to the service flow, the link adaptation, and some scheduling algorithms for three QoS classes: UGS, rtPS, and BE. We also implement the unicast and contention request opportunities mechanisms as specified in the IEEE 802.16 standard. The IEEE 802.16 MAC class diagram is shown in Figure 2. We are interested in the 802.16 MAC class since our contribution is applied to the MAC layer.

The *Mac802.16* class represents the MAC layer. It represents the main class and has relations with other classes: *ServiceFlowHandler*, *peerNode*, and *WimaxScheduler*. *ServiceFlowHandler* is responsible for the management of the downlink and uplink connections. Each connection has an association with a service flow that contains the QoS parameters. The QoS parameters of a service flow are set basing on the connection requirements. *peerNode* contains information about the SS or the BS. *WimaxScheduler* is responsible for the ranging and registration and performs scheduling algorithms. It includes two schedulers: one for the BS (*BSScheduler*) and one for the SS (*SSscheduler*).

3.1 Link Adaptation

In the existing implementation, all the subscribers use the same MCS. The MCS used is set from NS-2 Tool Command Language (TCL) script and does not change during the simulation. We provide the possibility of using different MCS in the same scenario. Moreover, an MCS can be changed and this depends on the SNR values. We then modify the *PeerNode* class which contains the different characteristics of a station (SS or BS). We also modify the *BSscheduler* class which manages the changes of the SNR values and we determine the suitable MCS used basing on the SNR parameter of the *PeerNode* class and the SNR thresholds. These thresholds are taken from the SNR assumptions values that are proposed in Table 266 of the IEEE 802.16e amendment of the standard (reproduced in Table 1).

Table 1: Receiver SNR assumptions (values of the IEEE 802.16e standard)

Modulation	Channel coding rate	Receiver SNR (dB)
BPSK	1/2	3.0
QPSK	1/2	6.0
	3/4	8.5
16-QAM	1/2	11.5
	3/4	15.0
64-QAM	2/3	19.0
	3/4	21.0

3.2 Dynamic Service Management

The existing implementation contains a class, called ServiceFlow, which identifies the service requirements of the associated connection. This class contains only the Service Flow Identifier (SFID) and Service Flow Scheduling Type (UGS, rtPS, ertPS, nrtPS, or BE) parameters. There is no differentiation between the QoS classes. We add the following parameters:

- CID: represents the identifier of the connection.
- Traffic Priority: defines the priority assigned to the service flow.
- Maximum sustained Traffic Rate: defines the peak information rate of the service flow.
- Minimum Reserved Traffic Rate: defines the minimum reserved rate of the service flow.
- Tolerated Jitter: defines the maximum delay variation of the connection.
- Maximum Latency: specifies the maximum latency between the reception of a packet and the forwarding of this packet.

Once the service flow parameters are initialized, the addition of a new service flow can be performed. This needs an exchange of DSA packets. These packets are defined in the IEEE 802.16 standard and already implemented in the existing NS2 module. These packets do not contain the service flow parameters as scheduling algorithm does not take into account the QoS class requirements. We modify the DSA packets and add some instructions:

- When a DSA-REQ packet is sent, the flow service parameter is added to this packet as defined in the IEEE 802.16 standard.
- When a DSA-REQ packet is received, the service flow and QoS parameters of the created data connection are filled up from the service flow parameter of the

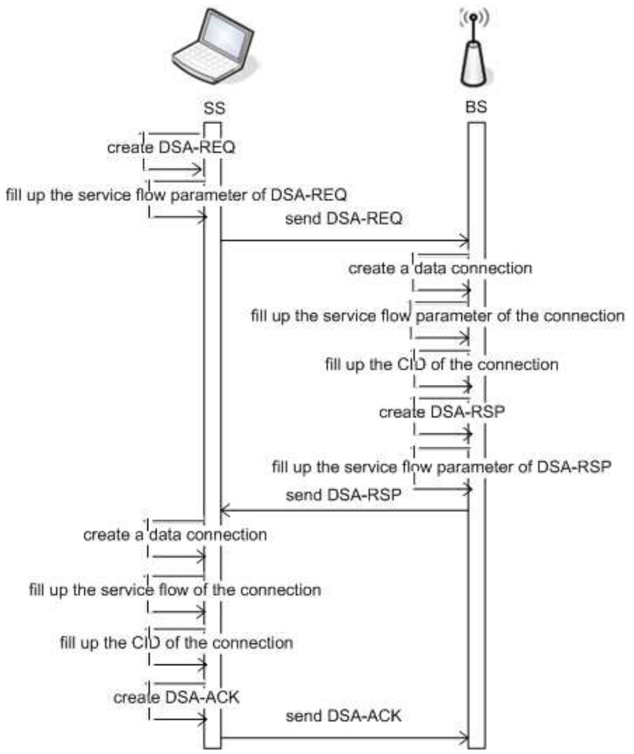

Figure 3: Steps of the creation of a new service flow.

received DSA-REQ packet. Then, a value of the CID parameter of the service flow is generated and assigned.

- When a DSA-RSP packet is sent, the service flow parameter is added to this packet.
- When a DSA-RSP packet is received, the service flow and QoS parameters of the created data connection are filled up from the service flow parameter of the received DSA-RSP packet.

The different steps of the creation of a new service flow, in our new module, are depicted in Figure 3. All those modifications are performed in a class, called ServiceFlowHandler. This class handles the list of the downlink and uplink connections and manages the creation of new connections.

3.3 Request Opportunities

3.3.1 Unicast request opportunities

We implement the unicast request opportunities mechanism in the existing implementation (see Figure 4). The provision of unicast request opportunities is done as follows. First, the BS determines the unicast polling list. A subscriber belongs to the unicast polling list when the current frame corresponds to its period of unicast request polling. Then, the BS allocates uplink bursts with an Uplink Interval Usage Code (UIUC) equal to the REQ Region Focused value. When an SS receives a burst for unicast request polling, it sends a bandwidth request. The bandwidth request contains the length of its uplink data connection queue. We assume that the subscribers are disciplined and they use the bursts of unicast request polling to only send bandwidth requests.

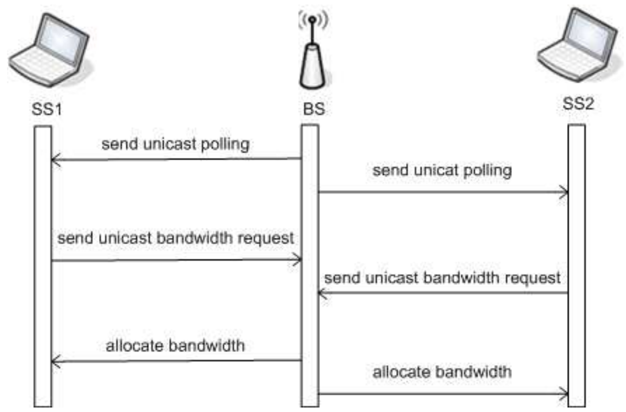

Figure 4: Unicast request opportunities.

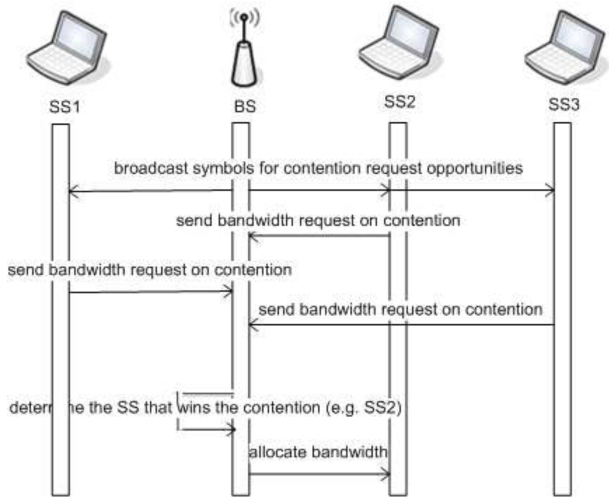

Figure 5: Contention request opportunities.

3.3.2 Contention request opportunities

The existing implementation provides contention request opportunities (see Figure 5). The BS allocates symbols for contention request opportunities at the beginning of each uplink subframe. Every frame, all SSs, having uplink data, send bandwidth requests with contention. We modify the existing implementation in order to prohibit UGS and rtPS SSs from using the contention request opportunities. Moreover, an SS sends a bandwidth request periodically and not every frame. The contention period is set by the TCL script. An SS can send a bandwidth request every frame when the contention period is equal to 1.

3.4 Scheduling

The existing implementation does not differentiate between the different QoS classes. The scheduling algorithm of the existing implementation is described as follows. First, the BS sends its downlink packets in the downlink subframe. Then, it reserves all the remaining symbols for a single station using the Round Robin (RR) scheduler. Hence, at each frame, at most one subscriber can send its packets independently of the service classes. The implemented scheduling algorithm has to be changed in order to differentiate between the service classes and take into account the QoS parameters. In this paper, we consider three QoS classes: UGS, rtPS, and BE. We briefly describe some of scheduling mechanisms. The RR scheduler equitably distributes channel resources to all the SSs. The mSIR scheduler allocates radio resources to SSs that have the highest SNR. The WRR scheduler is an extension of the RR scheduler and it is based on static weights. The TRS scheduler [6] temporarily blocks SSs having SNR smaller than a defined threshold. We combine the TRS scheduler with the RR and mSIR schedulers (called TRS+RR and TRS+mSIR, respectively). The TRS+RR scheduler reserves $1/k$ of the whole radio resources if there are k SSs to schedule. While the TRS+mSIR reserves all radio resources for SSs that have the highest SNR.

3.4.1 Unsolicited Grant Service QoS class scheduling

The proposed UGS scheduling is described as follows (see Figure 6). The BS determines all the SSs that have UGS connections. The connection, that the BS will start serving first, is randomly picked. Since an SS is served periodically, an assignment of a period to each station is recommended. The period, called UGS scheduling period can be set using the TCL script. Otherwise, it is equal to a predefined default value. From the period of scheduling, the BS determines if an SS is served in the current frame. For each UGS connection that will be served, the BS determines the transmission time using the Maximum Sustained Traffic Rate parameter of the connection. Then, the BS determines the number of symbols to be reserved for this UGS connection from the determined transmission time and MCS used. The available symbols must be sufficient to serve all UGS connections. Once the number of reserved symbols is determined, the BS updates the uplink map (UL-MAP) and allocates an uplink burst. The UGS scheduling algorithm is performed at the beginning of each frame.

3.4.2 Real-time Polling QoS class scheduling

The proposed rtPS scheduling is described as follows (see Figure 7). The BS determines all SSs that have rtPS connections. The choice of the next connection to serve depends on the rtPS scheduling algorithm. The implemented scheduling algorithms for the rtPS connections are the Round Robin (RR), maximum Signal to Interference Ratio (mSIR), Weighted Round Robin (WRR), and Temporary Removal Scheduler (TRS) schedulers. If the SS has not a bandwidth request to satisfy, the BS determines the next SS to serve. Otherwise, the BS determines the transmission time and then the number of symbols to be reserved for this rtPS connection. If there are not sufficient symbols, the BS allocates all the remaining symbols. If the BS can satisfy the entire bandwidth request, it removes this request. Else, the BS updates the bandwidth to be allocated in the following frames. Once this SS is served, the BS determines the next SS to be served if there are yet remaining symbols (see Figure 7).

3.4.3 Best Effort QoS class scheduling

The proposed BE scheduling is described as follows (see Figure 8). First, the BS determines all the SSs that have BE connections. The connection, to start with, is determined using the RR scheduler. If the SS has not a bandwidth request, the BS checks the next SS. Otherwise; the BS determines the transmission time and the number of symbols

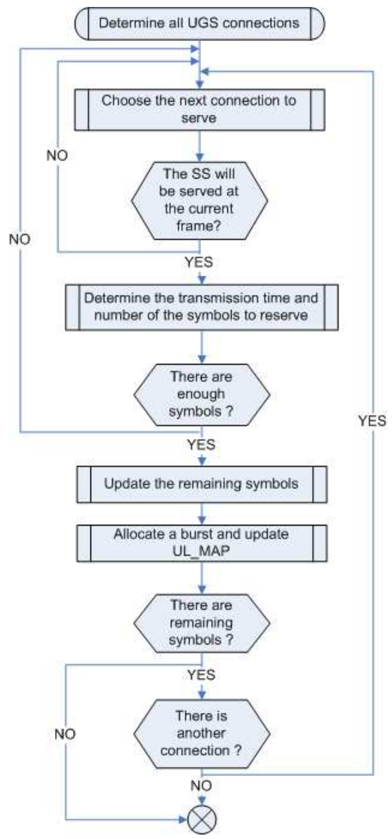

Figure 6: UGS scheduling.

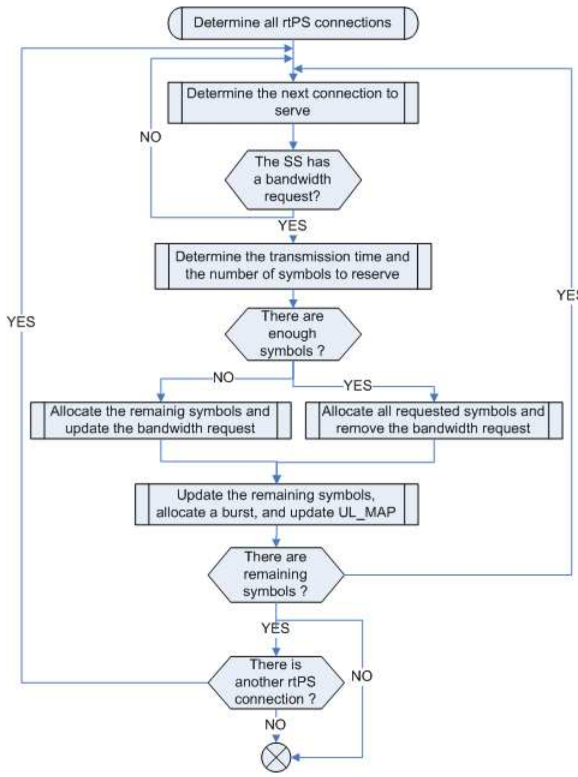

Figure 7: rtPS scheduling.

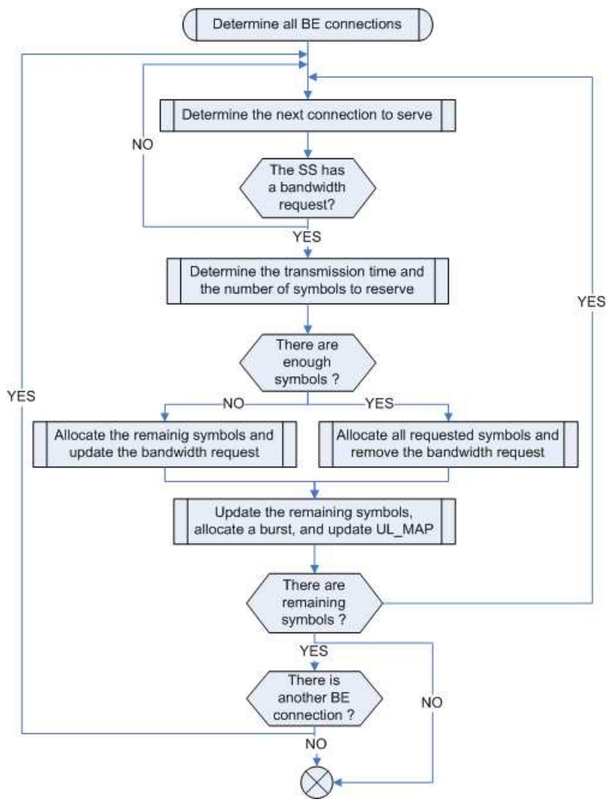

Figure 8: BE scheduling.

to reserve. If there are not enough available symbols, the BS allocates all the remaining symbols. If the BS can satisfy the entire bandwidth request, it removes this request. Otherwise, the BS updates the bandwidth to be allocated in the following frames. Once this SS is served, the BS determines the next SS to be served if there are yet remaining symbols.

4. SIMULATION RESULTS

4.1 Simulation Model

We have integrated into the existing implementation of WiMAX QoS parameters, QoS classes, unicast and contention request opportunities, five rtPS schedulers, and simple UGS and BE schedulers. The main parameters of the simulation model are represented in Table 2. We consider five UGS, nine rtPS, and two BE subscribers. The subscribers can use the QPSK 1/2, QPSK 3/4, 16-QAM 1/2, 16-QAM 3/4, 64-QAM 2/3, and 64-QAM 3/4 MCSs.

4.2 Performance of rtPS scheduling

In this section, we study the behavior of the rtPS schedulers. We consider the scheduler of the existing module (Existing_RR) as well as our implemented schedulers: RR, mSIR, WRR, TRS+RR, and TRS+mSIR. We have five UGS subscribers. Each UGS SS generates Constant Bit Rate (CBR) traffic with a rate of 160 Kbit/s s. We have also two BE SSs that generate FTP traffic.

Figure 9 shows the throughput of the rtPS connections as a function of the rtPS traffic load submitted in the network. This figure shows the low efficiency of the Existing_RR scheduler. Such a scheduler, in the quest of the simplifying of the scheduling steps, throttles the network traffic. Indeed, this scheduler allocates all the symbols to one SS even if it has not data to send. We note that the the rtPS throughput of the Existing_RR scheduler is nearly nine times worse than that of our implemented rtPS schedulers.

Table 2: Main parameters of the simulation model

Parameter	Value
Frequency band	5 MHz
Propagation model	Two Ray Ground
Antenna model	Omni antenna
Antenna height	1.5 m
Transmit antenna gain	1
Receive antenna gain	1
System loss factor	1
Transmit power	0.25
Receive power threshold	205e-12
Carrier sense power threshold	184.5e-12
Link adaptation	enabled
Frame duration	20 ms
Cyclic prefix (CP)	0.25
Packet length	1024 bytes
Simulation duration	100 s

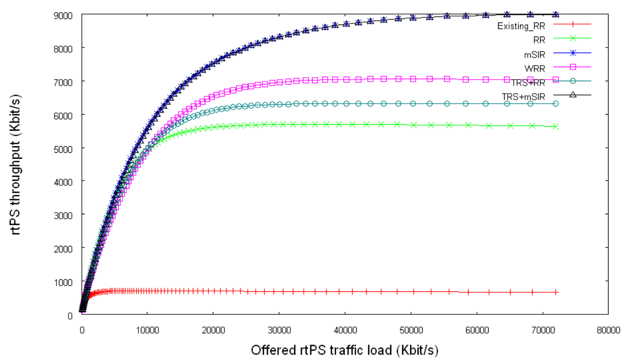

Figure 9: rtPS throughput versus offered rtPS traffic load.

We observe that the mSIR and TRS+mSIR schedulers nicely outperform the other schedulers with a maximum throughput of 9 Mbit/s. These two schedulers favor SSs having the highest SNR values and then the most efficient MCSs.

We also observe that the RR scheduler provides rtPS throughput less than that of the TRS+RR and WRR schedulers. This is due to the fact that the channel quality of the different SSs in not taken into consideration.

Figure 10 shows the mean sojourn time of the rtPS connections as a function of the rtPS traffic load submitted in the network. The mean sojourn time is a vital parameter for the real-time applications. We note that the Existing_RR scheduler requires a large average delay to deliver a data packet. This is because this scheduler does not provide sufficient symbols to the SSs. It periodically allocates all the remaining symbols of an uplink subframe to one rtPS SS. This period is equal to the frame duration multiplied by the number of all the SSs.

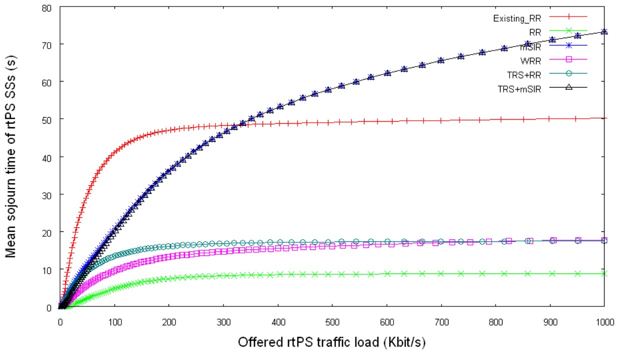

Figure 10: Mean sojourn time of rtPS SSs versus offered rtPS traffic load.

We also note that the mSIR and TRS+mSIR schedulers have a mean sojourn worse than that of the Existing_RR scheduler at high traffic load. This is due to the freezing of the traffic of the SSs having small SNR values. On the other hand, the Existing_RR scheduler does not block any SS.

We observe that the RR, WRR, and TRS+RR schedulers exhibit a much better sojourn time. These schedulers take into account the bandwidth requests of the rtPS SSs. Moreover, they do not block any SS.

4.3 UGS scheduling

In this section, we study the behavior of our UGS scheduler. We have nine rtPS subscribers that generate a mean traffic rate equal to 80 Kbit/s s. We have also two BE SSs that generates FTP traffic.

Figure 11 represents the throughput of the UGS connections as a function of the UGS traffic load submitted in the network. For a very low traffic load, our UGS scheduler and Existing_RR provide the same UGS throughput since all the SSs are entirely served. We observe that the Existing_RR scheduler throttles the network traffic and the throughput can not exceed 500 Kbit/s. Indeed, the Existing_RR scheduler does not differentiate between the QoS classes. Since there are sixteen SSs and five of them uses UGS connections, the Existing_RR scheduler always allocates five uplink subframes for each period of sixteen frames. Our UGS scheduler works as defined in the standard. All the UGS subscribers must be entirely served. The curve of our UGS scheduler is linear and the coefficient is equal to 1.

The curves of our UGS scheduler and Existing_RR are represented only for an offered UGS traffic load less than 3.5 Mbit/s. This is because we add an admission control. The admission control is such that all the UGS connections can be entirely satisfied.

Figure 12 shows the mean sojourn time of the UGS connections as a function of the UGS traffic load submitted in the network. Our UGS scheduler provides a very low mean sojourn time because it periodically allocates sufficient symbols to all the UGS SSs. Moreover, a new UGS connection is added only when it can be satisfied.

4.4 BE scheduling

In this section, we study the behavior of our BE scheduler. Since we have rtPS and UGS schedulers, we can vary the UGS or rtPS traffic load. We choose to study the relationship between the BE and UGS schedulers. In this scenario,

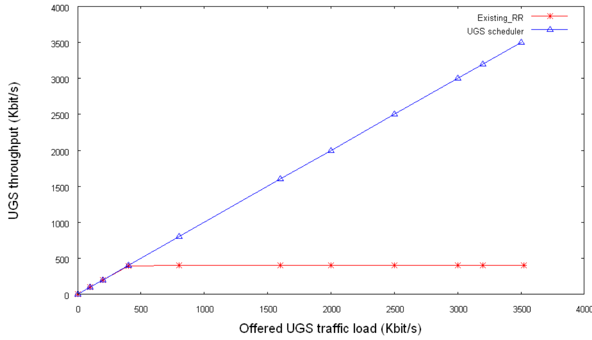

Figure 11: UGS throughput versus offered UGS traffic load.

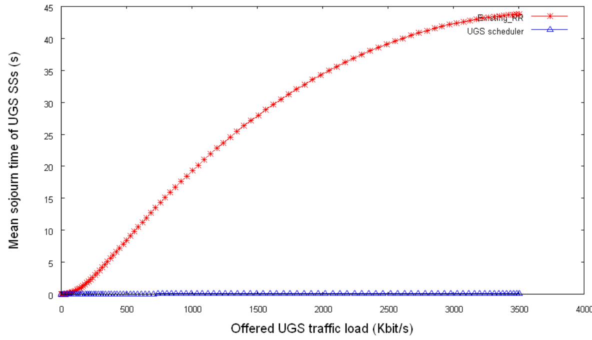

Figure 12: Mean sojourn time of UGS SSs versus offered UGS traffic load.

we have nine rtPS subscribers that generate a mean traffic rate equal to 80 Kbit/s s. We have also two BE SSs that generates FTP traffic.

Figure 13 represents the throughput of the BE connections as a function of the UGS traffic load submitted in the network. This figure shows that the Existing_RR scheduler provides the same BE throughput independently of the offered traffic load. This is because the Existing_RR scheduler is applied of all the QoS classes. In our case, it periodically allocates two uplink subframes and the period is equal to the duration of sixteen frames.

The higher the offered traffic load, the lower is the BE throughput. This is because our scheduling implementation serves the UGS, then rtPS, and finally BE connections. So, the higher the remaining symbols, the higher is the BE throughput. This behavior fits with the standard specification as the BE connections has no QoS requirements.

5. CONCLUSIONS

In this paper, we investigate the behavior of WiMAX scheduling algorithms. We consider some UGS, rtPS, and BE schedulers. We are based on the network simulation. So, we have implemented a new WiMAX module that takes into account the QoS classes and its requirements. Our module supports unicast and contention request opportunities and involves some UGS, rtPS, and BE schedulers. Extensive simulations are conducted to show that the behavior of our UGS, rtPS, and BE schedulers fits with the QoS specifications of the IEEE 802.16 standard. We show through simu-

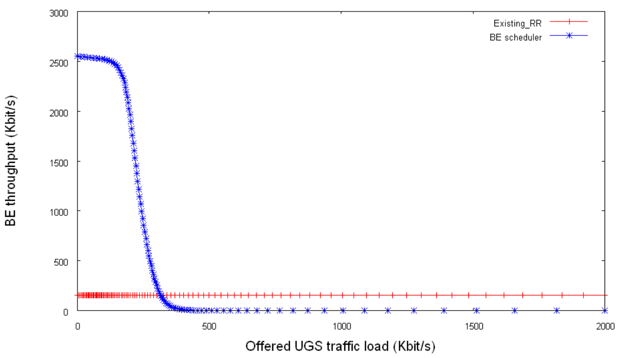

Figure 13: BE throughput versus offered UGS traffic load.

lations that our schedulers outperform the scheduler of the existing module of WiMAX; the RR scheduler. Moreover, throughput and mean sojourn time values are determined for some QoS classes and using different scheduling algorithms.

A direction for future work is the study of WiMAX capacity for other scheduling algorithms and other environments. A WiMAX environment with intelligent antennas techniques such as Multiple-Input Multiple-Output (MIMO) and smart antennas can also be a future topic of research.

6. REFERENCES

[1] Deploying license-exempt wimax solutions. Technical report, Intel, 2005.

[2] Considerations for deploying mobile wimax at various frequencies. Technical report, Nortel, 2006.

[3] The design and implementation of wimax module for ns-2 simulator. http://ndsl.csie.cgu.edu.tw/wimax_ns2.php, September 2007.

[4] The network simulator ns-2. http://www.isi.edu/nsnam/ns/, September 2007.

[5] Seamless and secure mobility. http://www.antd.nist.gov/seamlessandsecure.shtml, September 2007.

[6] C. F. Ball, F. Treml, X. Gaube, and A. Klein. Performance analysis of temporary removal scheduling applied to mobile wimax scenarios in tight frequency reuse. *European Transactions on Telecommunications*, 2(16):888–894, September 2005.

[7] C. F. Ball, F. Treml, K. Ivanov, and E. Humburg. Perfermance evaluation of ieee802.16 wimax with fixed and mobile subscribers in tight reuse. *European Transactions on Telecommunications*, 17(1):203–218, March 2006.

[8] J. Chen, W. Jiao, and Q. Guo. An integrated qos control architecture for ieee 802.16 broadband wireless access systems. *Global Telecommunications Conference*, 6:3330–3335, November 2005.

[9] J. Chen, C. Wang, C. Chang, S. Liu, J. Guo, W. Lien, J. Sum, and C. Hung. The design and implementation of wimax module for ns-2 simulator. *workshop on ns-2*, 202, October 2006.

[10] R. Mukul, P. Singh, D. Jayaram, N. S. D. Das, K. Vinay, and A. Ramamoorthly. An adaptive bandwidth request mechanism for qos enhancement in wimax real time communication. *Wireless and Optical Communications Networks*, April 2006.

[11] H. K. Rath, A. Bhorkar, and V. Sharma. An opportunistic drr (o-drr) uplink scheduling scheme for ieee 802.16-based broadband wireless networks. *International Conference on Next Generation Networks*, February 2006.

[12] I. C. Society, I. M. Theory, and T. Society. Part 16: Air interface for fixed broadband wireless access systems. October 2004.

[13] I. C. Society, I. M. Theory, and T. Society. Part 16: Air interface for fixed and mobile broadband wireless access systems. Febrary 2006.

[14] T. Tsai, C. Jiang, and C. Wang. Cac and packet scheduling using token bucket for ieee 802.16 networks. *Journal of communications*, 1(2):30–37, May 2006.