INSTITUTO DE COMPUTAÇÃO

UNIVERSIDADE ESTADUAL DE CAMPINAS

WiMAX module for the ns-2 simulator

J. F. Borin   *N. L. S. da Fonseca*

Technical Report - IC-11-27 - Relatório Técnico

December - 2011 - Dezembro

WiMAX module for the ns-2 simulator

Juliana Freitag Borin Nelson Luis Saldanha da Fonseca*

Abstract

The Network Simulator (ns-2) is a popular tool for the simulation of computer networks which provides substantial support for simulation of the Internet protocols over wired and wireless networks. Although some modules for WiMAX networks simulation have been proposed for the ns-2, none of them implements all MAC features specified by the IEEE 802.16 standard for bandwidth management and QoS support. This paper presents the design and validation of an WiMAX module based on the IEEE 802.16 standard. The implemented module includes mechanisms for bandwidth request and allocation, as well as for QoS provision. Moreover the implementation is standard-compliant.

1 Introduction

The IEEE 802.16 standard [1], widely known as WiMAX (*Worldwide Interoperability for Microwave Access Forum*), has been developed to accelerate the introduction of broadband wireless access into the marketplace. The advantages of this standard include easy and low-cost deployment, high speed data rate, last mile wireless access, and QoS support for multimedia applications.

Both industry and academia have been motivated to conceive novel mechanisms for 802.16 networks since some aspects of the standard are left to be defined by proprietary solutions. Several research groups are investigating QoS mechanisms, such as admission control and scheduling algorithms [5, 6, 7, 9, 10, 11].

Simulation is an essential tool in the development and performance evaluation of communication networks. Among the available tools for networks simulation, the Network Simulator (*ns-2*) is the most popular one in the research community. Much of this popularity is due to the fact that the *ns-2* is a public domain tool which implements a rich set of Internet protocols, including wired and wireless networks.

*Institute of Computing, University of Campinas, Campinas, SP, Brazil, 13089-971. This research was partially sponsored by UOL, through its UOL Bolsa Pesquisa program, process number 20060511022200a, and by CNPq.

This paper was published in J. Freitag and N. L. S. Fonseca, *WiMAX module for the ns-2 simulator*. In Proceedings of the IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications, pp. 1-6, 2007. DOI http://dx.doi.org/10.1109/PIMRC.2007.4394155

Recently, two modules were proposed for simulation of IEEE 802.16-based networks using *ns-2*. One of them, implemented by NIST [15], provides, among other features, WirelessMAN-OFDM physical layer with configurable modulation, Time Division Duplexing (TDD), Point-to-Multipoint (PMP) topology, fragmentation and reassembly of frames, but it fails to implements MAC QoS support, namely, service flows and QoS scheduling. The other 802.16 simulation module, proposed by Chen *et all* [4], uses the wireless channel implementation provided by the *ns-2*. It is also based on TDD duplexing mode and PMP topology, and it provides packet fragmentation and packing. Although this module implements the five service flow types specified in the IEEE 802.16 standard, the request/grant mechanism defined for bandwidth management is not compliant to the MAC layer specification. Moreover, users cannot configure QoS requirements, such as maximum latency and minimum bandwidth, for the high priority service flows. There is another group developing an 802.16-based simulator for the OPNET tool [13], which is a private domain simulator; however, this module is available exclusively to the consortium members.

This paper presents the design and validation of a simulation module for 802.16-based networks in the *ns-2* simulator. An extended version of this paper appears in the Simulation Modelling Practice and Theory Journal [8]. The focus of this implementation is the MAC layer and its mechanisms for bandwidth allocation and QoS support. The module implements the 802.16 five service flow types and their bandwidth request/grant mechanisms; moreover, it allows users to configure the QoS requirements of applications. Service flows are modeled by finite sate machines that capture how each service type react to different events. This module supports TDD mode and PMP topology. The wireless channel available in the *ns-2* simulator is used. We believe that the module developed is a significant contribution for the communication network research community since it allows research on 802.16 MAC layer specially those on bandwidth management and QoS provision. Although the code developed is large, containing 17 classes of objects and about 17,300 lines of code, the modularization provided by object oriented programming facilitates the inclusion of new functionalities. It is our best knowledge that no other module for WiMAX networks simulation implements bandwidth request/grant mechanism and QoS support according to the IEEE 802.16 standard.

The remainder of this paper is organized as follows. Section 2 presents an overview of the IEEE 802.16 standard. Section 3 describes the proposed WiMAX module. Section 4 presents the simulation experiments created to validate the module. Finally, Section 5 concludes the paper.

2 The IEEE 802.16 standard

The IEEE 802.16 standard [1] primarily supports a Point-to-Multipoint (PMP) architecture, with an optional mesh topology. In PMP mode, communication is possible only between a Base Station (BS) and a Subscriber Station (SS). In mesh mode, multihop communication is possible between SSs. In this paper, we focus on the PMP mode.

The physical channel operates in a framed format. Each frame is divided in two subframes: the downlink subframe is used by the BS to send data and control information to the SSs, and the uplink subframe is shared by all SSs for data transmission. Duplexing is provided by means of either Time Division Duplexing (TDD) or Frequency Division Duplexing (FDD). In TDD mode, uplink and downlink transmissions occur at different times since both subframes share the same frequency. Each TDD frame has a downlink subframe followed by an uplink subframe. In FDD mode, the downlink and the uplink subframes are concurrent in time since they are transmitted on separate frequencies.

The IEEE 802.16 MAC protocol is connection-oriented. All traffic, including connectionless traffic, is mapped into connections which are uniquely identified by a 16-bit Connection Identifier (CID). The MAC layer defines a signaling mechanism for information exchange between the BS and SSs. This signaling mechanism allows the SSs to request bandwidth to the BS. When a connection has backlogged data, the SS sends a bandwidth request to the BS. The BS, in turn, allocates time slots to the SSs based on both bandwidth requests and QoS requirements of the requesting connection. While bandwidth requests are associated to individual connections, the BS grants uplink bandwidth to the SS as a whole. Due to this semi-distributed approach, the SSs have to implement a scheduling mechanism locally to redistribute the granted capacity among its connections. A bandwidth request can be incremental or aggregate. In aggregate requests, an SS indicates the whole connection backlog, whereas, in incremental requests, an SS indicates the difference between the connection current backlog and the one on its last bandwidth request. The self-correcting nature of the request/grant mechanism requires that SSs periodically use aggregate bandwidth requests.

A request for bandwidth can be sent as a stand-alone message, in response to a poll from the BS, or can be piggybacked in data packets. When the BS uses unicast polling, no explicit message is transmitted to poll individual SSs. Rather, sufficient bandwidth to send a request is allocated to an SS. Multicast and broadcast polling are used to poll a group of SSs when there is insufficient bandwidth to poll all the stations individually. When a group is polled, the members of the group which require bandwidth respond with a request. A contention resolution algorithm is used to resolve conflicts that arise when two or more transmission occur at the same time.

To support a wide variety of multimedia applications, the IEEE 802.16 standard defines five types of service flows, each with different QoS requirements. Each connection between the SS and the BS is associated to one service flow.

The Unsolicited Grant Service (UGS) carries constant bit rate (CBR) flows of CBR-like applications such as Voice over IP (VoIP). UGS connections receive fixed size data grants periodically.

The real-time Polling Service (rtPS) is designed for applications with real-time requirements which generate variable size data packets periodically, such as MPEG video streams. This service receives unicast polls to allow the SSs to specify the size of the desired grant. QoS guarantees are given as bounded delay and assurance of minimum bandwidth.

The extended real-time Polling Service (ertPS) is designed for real-time traffic with variable data rate, such as VoIP service with silence suppression. This service uses a grant mechanism similar to the one for UGS connections. Moreover, periodic allocated grants can be used to send bandwidth requests to inform the required grant size. The BS shall not change the size of uplink allocations until it receives another bandwidth request from the SS.

The non-real-time Polling Service (nrtPS) is adequate to better-than-best-effort services such as FTP services. The BS provides timely unicast request opportunities, besides that, the SS is also allowed to use contention request opportunities. Minimum bandwidth guarantees are also provided to nrtPS connections.

The Best Effort service (BE) is used for best-effort traffic such as HTTP. The SSs request bandwidth through contention request opportunities as well as unicast request opportunities.

3 The WiMAX Module

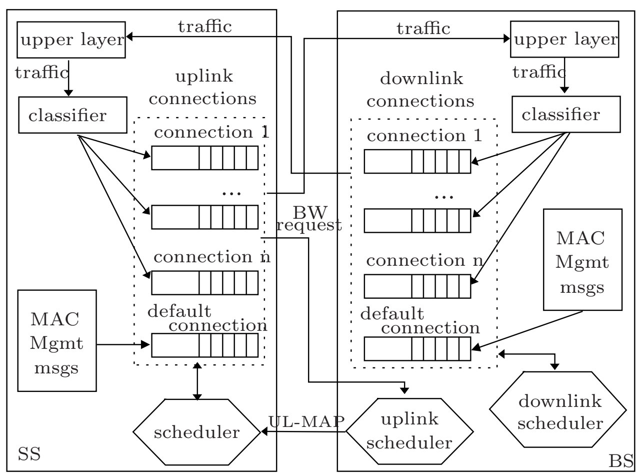

Figure 1: Structure of the WiMAX module

The WiMAX module was developed for the *ns-2* simulator, release 2.28. It is based on the specifications of the IEEE 802.16 [1] standard for PMP topology and TDD duplexing mode. The implementation was carried out in C++ using object oriented programming.

The module design was based on a module [3] designed to simulate the DOCSIS standard [2] in the *ns-2* simulator. Code reuse was possible since the 802.16 MAC is based on the MAC specification for cable modems in the DOCSIS standard.

Although the MAC layers of both standards are similar, several modifications in the DOCSIS module code were necessary to make it compliant to the IEEE 802.16 standard. The main changes were: i) implementation of nrtPS and ertPS services for the uplink traffic, ii) implementation of the five types of service for the downlink traffic, iii) changes in the interface between the MAC and the PHY layers to make the WiMAX module to use the wireless PHY implementation of the *ns-2*, iv) implementation of frames and subframes, v) aggregate requests, and vi) addition of maximum delay and minimum bandwidth QoS parameters for the rtPS service.

Figure 1 shows the module structure. When a packet arrives at the MAC layer from the higher layers, it is classified into a particular service flow. The service flows, associated to a connection, are configured by the user which, among other information, define QoS requirements. Each node have an uplink and a downlink default connection to carry management messages and all traffic that cannot be classified to any other service flow. When the simulation starts, every SS registers itself to the BS by simulating the registration phase. The BS allocates a CID to each connection and stores the service flow parameters in a table. The main flow parameters include service type, QoS requirements, and fragmentation/concatenation/piggybacking capability.

The BS has an uplink scheduler and a downlink scheduler. The downlink scheduler decides which packets coming from the upper layer will be transmitted in the next downlink subframe. This decision is based on the QoS requirements and on the queue status of the downlink connections. The uplink scheduler decides which SSs can transmit in the next uplink subframe as well as the number of slots these SSs can use. This decision is based on the QoS requirements of the uplink connections and on the bandwidth requests sent by the SSs. Bandwidth requests can be sent using polling or piggybacking. Aggregate requests are sent in intervals defined by the user.

Each SS has a scheduler to decide which packets will be sent in the data grants allocated by the BS. Scheduling is based on the information about allocated slots available in the UL-MAP, as well as on QoS requirements and on the queue status of the uplink connections.

In the IEEE 802.16 standard, the DL-MAP and the UL-MAP are both transmitted in the beginning of each downlink subframe. In the WiMAX module, the DL-MAP is not transmitted, but in the beginning of the downlink subframe some slots are reserved to produce the necessary overhead. The UL-MAP is broadcasted by the BS and it describes the slot allocation to the uplink subframe of the following frame.

According to the IEEE 802.16 standard, a frame has fixed duration, while the duration of the subframes can be dynamically determined by the BS according to the traffic load in each direction. In the WiMAX module, users can define the frame duration, and each subframe is assigned half the frame duration. The dynamic tuning of the subframes duration will be implemented in future work.

In a real system, the BS and the SSs exchange about 50 kinds of management messages, such as ranging messages, messages describing the channel characteristics, and security messages. Although several of these messages are not implemented in the WiMAX module, management messages are generated periodically to produce the corresponding overhead.

3.1 Bandwidth request and bandwidth allocation

When a packet arrives from the upper layer, it is classified to a service flow and a 6 bytes MAC header [1] is added to the packet. When a packet arrives from the channel, it is classified either as a management message or as a data PDU, using the MAC header, and the payload is handled accordingly.

Each service flow has four major components:

- Classifier: the classifier uses the source IP address, the destination IP address, and the packet type to classify a packet into a certain service flow.
- Queue: all packets classified into a certain service flow are enqueued when they cannot be sent immediately.

- Allocation Table: the allocation table maintains the current and future grants for a service flow. This table is updated whenever a Map message is received.

- Finite state machine: a finite state machine (FSM) controls all the interactions of the service flow for transmissions. The FSMs are implemented using a procedure-driven approach, i.e., one function for each input state. When an event occurs, depending on the current state, the proper function is called to process the input event and update the state variable.

The definitions of the FSMs for the UGS, for the rtPS, and for the BE uplink service flows are based on the state machines proposed by Shrivastav [12] for the DOCSIS simulation module. The nrtPS uplink service flow uses the same model as the BE state machine since both service flows use unicast polling and contention polling. The difference between these two services is that the BS allocates unicast grants for the nrtPS service frequently, while the BE service receives unicast grants only when there is available bandwidth. The FSM for the ertPS uplink service is defined in order to allow the sending of both data PDUs and bandwidth requests in the periodic grants allocated by the BS.

Similarly to the UGS uplink service, the downlink service flows do not need to send bandwidth requests. Their task consists on sending data PDUs in the grants allocated by the BS downlink scheduler. Therefore, the FSMs for the five downlink service types have the same model as the UGS uplink service FSM.

Note that the QoS provided for each service flow does not depend on the FSM model, it depends on the admission control and on the scheduling mechanisms implemented in the BS as well as on those implemented in the SSs. The state machines control the transmission of data PDUs and bandwidth requests according to the information stored in the allocation table.

The scheduling mechanisms implemented in the SS scheduler and in the BS downlink scheduler follow the Strict Priority discipline. More sophisticated scheduling disciplines for both schedulers will be investigated in the future.

The BS uplink scheduler uses three queues, referred as low priority queue, intermediate queue and high priority queue. The scheduler serves the requests in strict priority order from the high priority queue to the low priority queue. The low priority queue stores the bandwidth requests of the BE service flow. The intermediate queue holds bandwidth requests sent by rtPS and by nrtPS connections. rtPS and nrtPS requests can migrate to the high priority queue to guarantee that their QoS requirements are met. Besides the requests migrated from the intermediate queue, the high priority queue stores periodic grants and unicast request opportunities that must be scheduled in the following frame [7].

In the following subsections, we present the finite state machines for UGS, rtPS, ertPS, and BE uplink service flows.

3.2 UGS FSM

Figure 2 shows the UGS uplink service finite state machine. The states have the following meaning:

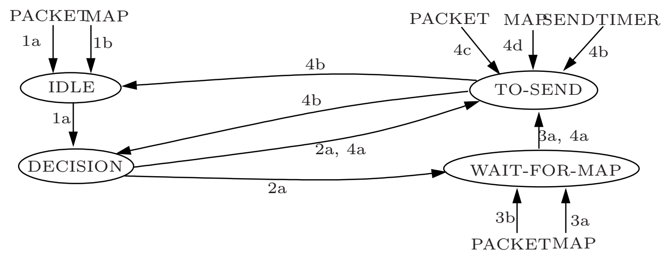

Figure 2: UGS Finite State Machine [12].

- Idle: there is no packet to transmit for this service flow.
- Decision: a temporary state in which the allocation table for the service flow is examined for grants.
- Wait-For-Map: the service flow is waiting for a Map with new allocations.
- To-Send: there is a packet pending for transmission.

The following events have been defined:

- Packet: an upper-layer packet was classified into the service flow to be sent over the channel.
- Map: an UL-MAP message was received.
- SendTimer: the send timer expired.
- SendPacket: a mandatory packet transmission.

The FSM works as follows. When the state is Idle and a packet arrives (event 1a), the MAC header is added to the packet, it is stored in a variable called *current_pkt*, and the FSM enters the new state Decision. If a Map arrives (event 1b), then the allocation table for this flow is updated and the state does not change.

In the Decision state, if there is no data grant for the service flow in the allocation table, there is a state transition to the Wait-For-Map state; otherwise, there is a transition to To-Send state and an event SendPacket occurs.

When a Map arrives with a grant for the flow in Wait-For-Map state, the FSM enters the state To-Send and the event SendPacket occurs. If the Map has no grant for the flow, there is no state change. When a packet arrives, the MAC header is added and the packet is enqueued.

When an event SendPacket occurs in state To-Send, the send timer is set to expire at the beginning of the grant. When the timer expires, the SendTimer event is fired, and the packet in *current_packet* is sent over the channel. If the queue is empty, the FSM goes to the Idle state. Otherwise, a packet is dequeued, stored in *current_packet*, the FSM goes to the Decision state and an event Packet occurs.

When a packet arrives and the FSM is in the state To-Send, a MAC header is added and the packet is enqueued. When there is a Map arrival, the allocation table for the flow is updated.

3.3 rtPS FSM

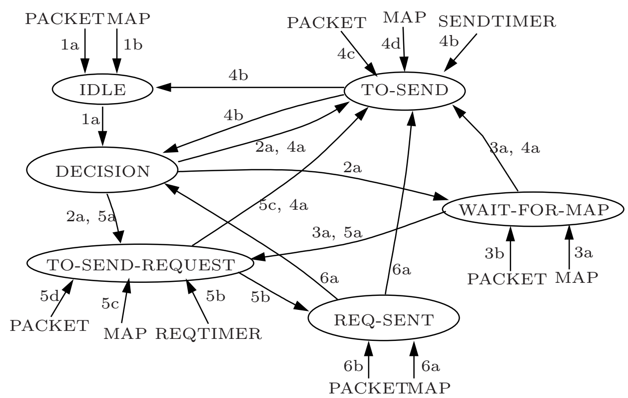

Figure 3: Finite state machine for the rtPS service. [12]

Figure 3 shows the FSM for the rtPS service. All the states have the same meaning of those of the UGS FSM. In addition, the following states and events are defined to allow the transmission of bandwidth requests.

States:

- To-Send-Req: there is a unicast request opportunity allocated to the service flow in the future.
- Req-Sent: a bandwidth request was sent.

Events:

- ReqTimer: the request timer expired.
- SendReq: a unicast request has to be sent in the future.

For the rtPS service, when the FSM is in the Decision state and there is a unicast request grant in the allocation table, the FSM goes to the To-Send-Req state and an event SendReq occurs. The same happens if the state is Wait-For-Map and a Map arrives with a unicast request grant for the service flow.

When the FSM goes to the To-Send-Req state and an event SendReq occurs, the request timer is set to expire at the beginning of the allocated unicast request opportunity. When the timer expires, the ReqTimer event is fired, a bandwidth request is sent, and the FSM goes to state Req-Sent. However, if a Map arrives with a data grant for the flow before the request timer expires, the timer stops, there is a state transition to To-Send state, and an event SendPacket occurs. If a packet arrives before the expiration of the request timer, the MAC header is added and the packet is enqueued.

If the FSM is in state Req-Sent and the SS receives a Map with a data grant for the service flow, then the FSM goes to To-Send state and an event SendPacket occurs. Alternatively, if there is a unicast request grant for the service flow, the FSM goes to Decision state and an event Packet occurs. When a packet arrives and the FSM is in state Req-Sent, the MAC header is added and the packet is enqueued.

The rtPS FSM proposed in [12] does not allow packets concatenation and fragmentation. These functionalities were added in the WiMAX module. In this way, in state To-Send several packets can be concatenated and transmitted in a single data grant, as well as a packet can be fragmented in order to fit into a grant.

3.4 ertPS FSM

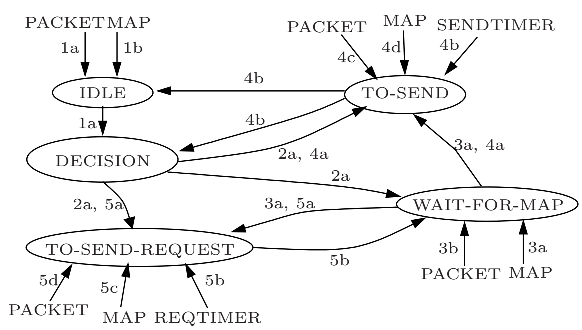

Figure 4: Finite state machine for the ertPS service.

Figure 4 presents the finite state machine defined for the ertPS service. The ertPS FSM has the same states, except for the Req-Sent state, and events of the rtPS finite state machine. However, the actions taken in some states and the state transitions are quite different.

If the FSM is in the Decision state and the allocation table for the service flow has a data grant smaller than the size of the packet stored in *current_pkt*, the FSM goes to the To-Send-Request state and an event SendReq occurs. However, if the data grant is greater or equal to the packet size, there is a transition to To-Send state and an event SendPacket occurs. The same actions are taken when a Map arrives and the state is Wait-For-Map.

In the To-Send-Request state, the data grant is used to request a new grant size to the BS. After sending the bandwidth request, the FSM goes to Wait-For-Map state.

When in the To-Send state, before sending the packet, the FSM compares the packet size (with all the overhead) to the grant size. If they are of the same size, the packet is sent. However, if the grant size is greater than the packet size, a piggyback request is sent with the data packet in order to announce the new packet size to the BS.

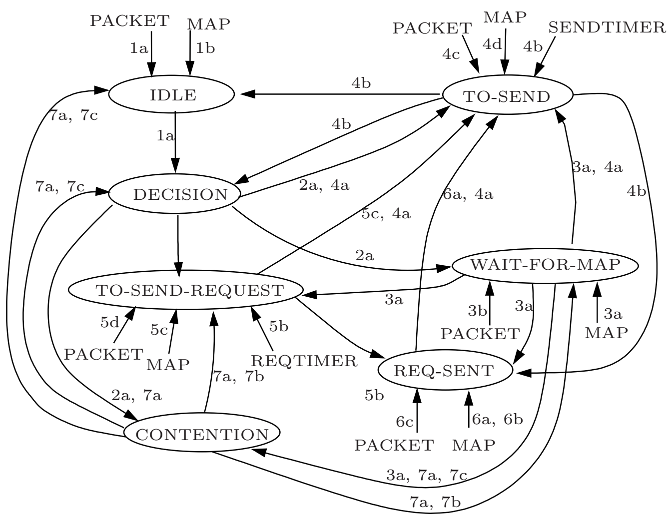

Figure 5: BE Finite state machine. [12]

3.5 BE FSM

Figure 5 shows the BE service finite state machine. Besides the states and events in the UGS, rtPS, and ertPS FSMs, the Contention state, which implements the backoff algorithm, is defined as well as the following events:

- UnicastReq: a unicast request opportunity is available for the service flow.
- ContentionReq: a contention request opportunity is available for the service flow.
- ContentionOn: the contention phase should be entered.
- ContentionSlots: the service flow is in the contention phase.
- ContentionBkoff: backoff is required as the request sent in the contention opportunity was lost.

When the FSM enters the Decision state with event Packet, it searches the allocation table for available grants. Whenever there is a data grant, the FSM goes to the To-Send state and an event SendPacket occurs. Otherwise, if there is a unicast request opportunity, there is a transition to To-Send-Request state and an event UnicastReq occurs. For both situations, the contention resolution process is interrupted. In case there is a contention request opportunity and the service flow has not entered the contention process, the FSM goes to the Contention state and an event ContentionOn occurs. If the contention process has already been started, the FSM goes to the Contention state and an event ContentionBkoff occurs.

When a Map arrives and the FSM is in state Wait-For-Map, the actions are the same taken for the Decision state, except for the case when the Map has a contention request opportunity and the contention process has already started. In this case, the FSM transitions to Contention state and an event ContentionSlots occurs.

In the To-Send state, several packets can be concatenated and transmitted in a single data grant, and a packet can be fragmented in order to fit into a grant. If the data grant is not sufficient to send all the packets stored in the queue and there is no data grant pending in the allocation table, the service flow sends a piggyback request, and the FSM goes to state Req-Sent. If the request is not sent and there is a packet in the queue, there is a transition to the Decision state and an event Packet occurs. If the piggyback request is not sent and the queue is empty, there is a transition to the Idle state.

In the To-Send-Request state, either a unicast request or a contention request is sent when the request timer expires, and there is a transition to the Req-Sent state. When a Map arrives with a data grant for the service flow in the Req-Sent state, the contention resolution process stops, the FSM enters the To-Send state, and an event SendPacket occurs. If the Map has no data grant for the service flow and the T16 timer expired, the backoff window is increased by a factor of two and the FSM enters the Decision state and an event Packet occurs. In all other cases, there is no state transition.

The transition to Contention state with the occurrence of event ContentionOn happens when the service flow enters the contention process. In this case, a random number r within the backoff window is selected. If the number of slots allocated for contention is greater than r, the request timer is set to expire in the contention slot $(r + 1)$ and the FSM goes to the To-Send-Request state. Otherwise, a variable called *skipped* is set to $(r - number\ of\ contention\ slots)$ and the FSM goes to the Wait-For-Map state.

When the FSM goes to the Contention state and an event ContentionSlots occurs, it means that the contention process is ongoing and a contention request opportunity has been allocated to the service flow. In this case, if the number of slots allocated to contention is greater than *skipped*, the request timer is set to expire in the contention slot $(skipped + 1)$ and the FSM goes to the To-Send-Request state. Otherwise, *skipped* is set to $(skipped - number\ of\ contention\ slots)$ and there is a transition to the Wait-For-Map state.

In case an event ContentionBkoff occurs in the Contention state, and the maximum number of request retries has been reached, the packet stored in *current_pkt* is discarded and the contention process stops. If the queue is empty, the FSM goes to the Idle state; otherwise, a packet is dequeued and stored in *current_pkt*, there is a transition to the Decision state and an event Packet occurs. If the maximum number of request retries has not been reached, the machine executes the same actions defined for the event ContentionOn.

4 Validation

The simulation experiments presented in this section were designed to check the developed WiMAX module compliance to the IEEE 802.16 standard for the PMP topology and TDD duplexing mode. Specially, we check the division of time in frames, and the division of frames into downlink and uplink subframes. Moreover, we check whether or not grant allocation for the transmission of bandwidth requests and for the transmission of data packets follows

T16 is a timer defined in the IEEE 802.16 standard [1]. If an SS has sent a request in a contention request opportunity and no data grant has been given within T16, the SS shall consider the contention transmission lost.

the rules specified for the five types of service flow. Due to space limitation, we do not show results for the downlink service flows. We show results for the validation of the uplink service flows since their implementation is more complex than the downlink one given the bandwidth request mechanism.

The topology of the simulated network consisted of a BS with the SSs uniformly distributed around it. The frame duration was 1 ms and the capacity of the channel was 40 Mbps. *The scenarios were not intended to be representative of operational networks. The goal is to analyze the medium access mechanisms and the slots allocation for different offered loads, i.e., for both underloaded and overloaded conditions* [14]. We used CBR sources to simulate the traffic of the five types of service flow. This is important at this stage since it facilitates the analysis of results obtained [14]. More realistic traffic models should be considered for the evaluation of QoS mechanisms, such as admission control and scheduling [9].

4.1 Frames

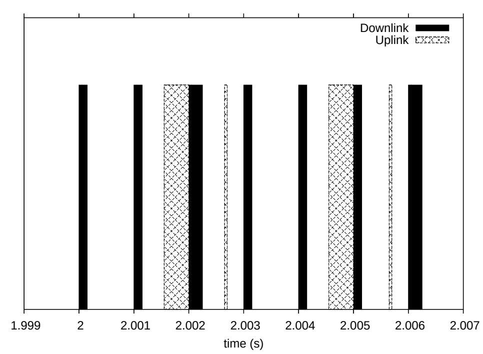

Figure 6: Frame division into downlink and uplink subframes

The implementation of the time frames is validated using a scenario with 2 SSs and 1 BS. One of the SSs has an uplink CBR flow and the other a downlink CBR flow, both with a data rate of 5 Mbps and mapped to the BE service. The duration of each subframe is 0.5 ms given that the frame duration is set to 1 ms.

Figure 6 shows the traffic transmitted in a period of the simulation. The traffic includes UL-MAP messages, bandwidth requests, and data packets. Note that in the first half of each frame (each mark in the x axis represents the beginning of a new frame), transmission happens on the downlink direction, while in the second half of the frame, transmission occurs in the uplink direction. Moreover, the time elapsed between two downlink transmissions is always 1 ms, indicating that the frames duration is in conformance with the configured value.

These results indicate that the WiMAX module is compliant to the framed format defined by the IEEE 802.16 for the TDD mode.

4.2 UGS uplink service

When a user creates a UGS connection in the WiMAX module, he or she shall set the grant interval. To verify whether or not an uplink UGS connection receives periodic grants for data transmission, we simulate a network with 1 BS and 2 SSs. One of the SSs has an uplink UGS flow which transmission starts at time 0.5 s and the grant interval is set to 15 ms. Traffic is generated by a CBR source with a data rate of 500 Kbps. The other SS has an uplink BE flow which transmission starts at time 1.0 s. Traffic is also generated by a CBR source with data rate of 2 Mbps.

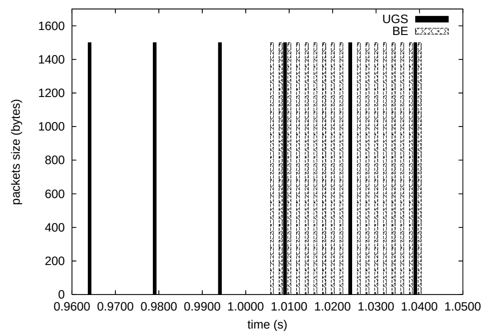

Figure 7: Validation of the uplink UGS service flow.

Figure 7 shows the transmission of data packets in an interval of the whole simulation. In spite of the entrance of the BE flow, the BS allocates data grants to the UGS flow within the defined interval.

The grant allocation for the uplink UGS service was also tested using a scenario in which the channel is overloaded by 10 BE flows, each generating 4 Mbps CBR traffic. Results for this experiment are not presented here given the space limitation. These results indicate that the grant allocation for the UGS service is not affected by the high load of a low priority service.

4.3 rtPS uplink service

The simulation scenario to test the rtPS service consists of 1 BS and 1 SS. The SS has an uplink rtPS flow with packets generated at a CBR rate of 1 Mbps. The interval for unicast polling is set to 15 ms. This interval is defined by the user in the network configuration script.

Figure 8 illustrates the transmission of bandwidth requests and data packets. It can be noted that grants for bandwidth request transmission are allocated within the defined interval. Moreover, it can be seen that after one frame the SS receives a grant to send data packets. This delay of a frame between the request transmission and the data grant allocation occurs because the grant allocation announced by the UL-MAP is not for the uplink subframe of the current frame, but for the one of the next frame.

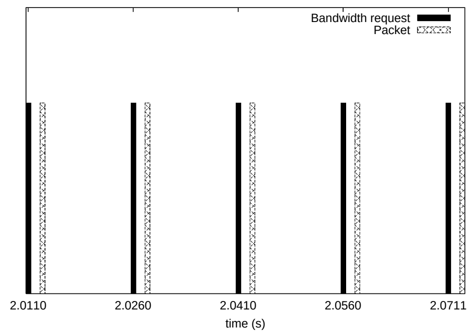

Figure 8: Validation of the uplink rtPS service flow.

4.4 ertPS uplink service

The simulation scenario to test the ertPS service consists of 1 BS and 1 SS. The SS has an ertPS uplink connection with an 1 Mbps CBR flow. In order to check the grant/request mechanism for this service we vary the packet size. The initial packet size is 200 bytes. At time 5 s the packet size changes to 500 bytes, and at time 5.06 s the packet size is changed back to the initial value.

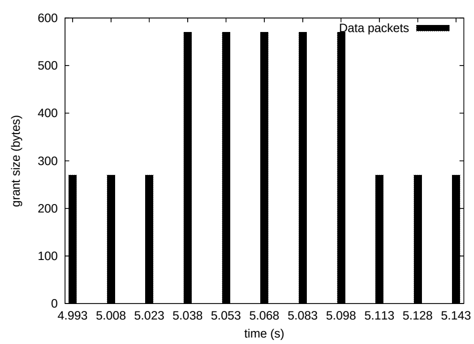

Figure 9: Validation of the uplink ertPS service flow.

Figure 9 shows the grant sizes allocated to the ertPS connection in a period of the simulation experiment. In the beginning of the simulation, when the packet size is 200 bytes, the BS allocates a grant of 270 bytes for the payload and for the overhead transmission. When the packet size changes to 500 bytes, the ertPS connection sends a bandwidth request at time 5.023, and then the BS allocates a grant of 570 bytes. In the interval [5.038, 5.083], the connection sends 500-byte packets. At time 5.098, the connection sends a 200-byte packet with a piggybacked bandwidth request to reduce the grant size. It can be noted that data grants are allocated periodically in the interval configured to 15 ms.

4.5 nrtPS uplink service

The scenario used to test the uplink resource allocation for the nrtPS service has 1 BS and 2 SSs. One SS has an nrtPS uplink connection with an 1 Mbps CBR flow. The interval for the unicast polling is set to 15 ms. The other SS has a BE uplink connection with a 40 Mbps CBR flow.

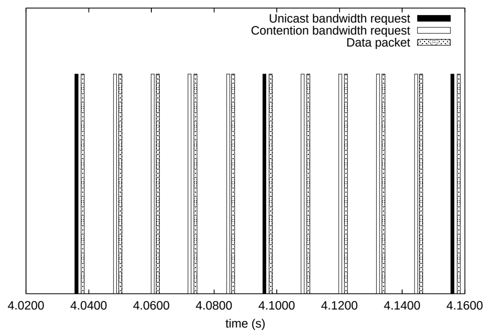

Figure 10: Validation of the uplink nrtPS service flow.

Figure 10 shows bandwidth requests and data packets sent by the nrtPS flow. Although the network is overloaded by BE traffic, the nrtPS flow is able to send bandwidth requests using unicast request opportunities and contention request opportunities as well as data packets. Some unicast request opportunities are not used (not shown) since at the time they were allocated the service flow had no packets to send.

4.6 Uplink BE service flow

The experiment to test the uplink resource allocation for the BE service checks whether a BE flow is able to send bandwidth requests and data packets using a scenario with flows with higher priority. The simulated network consists of 1 BS and 5 SSs, each one with an uplink connection. Two SSs have UGS connections, two SSs have rtPS connections and the other one has a BE connection. Traffic is generated by CBR applications with a data rate of 2 Mbps for the UGS and rtPS services and 1 Mbps for the BE service.

Figure 11 shows bandwidth requests and data packets sent by the BE flows. Even in the presence of higher priority traffic, the BS allocates grants to the BE service to send bandwidth requests and data packets.

5 Conclusion

This paper presents the design and validation of a module based on the IEEE 802.16 standard for the *ns-2* simulator. The developed module implements the five service flow types with their bandwidth request/grant mechanisms. Moreover, it allows users to configure the QoS requirements for applications with different demands.

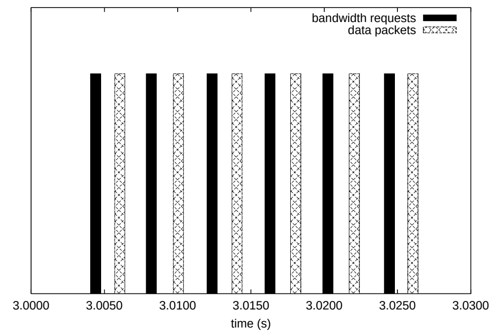

Figure 11: Validation of the uplink BE service flow.

Experiments were designed to evaluate the implementation of main IEEE 802.16 MAC functionalities. Results indicate that the BS is able to manage medium access both in the downlink and in the uplink directions, as well as to allocate grants for the transmission of bandwidth request and data according to the established for the five service flow types in the standard. Additionally, it can be concluded that the SSs are able to send bandwidth requests and data packets in the grants allocated by the BS and announced through the UL-MAP message.

We believe that the WiMAX module can benefit research on 802.16 networks, specially those on bandwidth allocation and QoS provision. Currently, we are developing an OFDM channel which includes varying wireless link capacity and the location-dependent channel state. Future work will focus on admission control and scheduling algorithms that can deal with the link variability.

References

[1] IEEE Standard 802.16-2004, Part 16. *Air Interface for Fixed Broadband Wireless Access Systems*, 2004.

[2] Cable Television Labs Inc. *Data Over Cable Service Interface Specifications - Radio Frequency Interface Specification.* SP-RFIv2.0.

[3] *DOCSIS Research Project*. http://www.cs.clemson.edu/~jmarty/docsis.html, 2005.

[4] J. Chen *et all. Design and Implementation of WiMAX Module for ns-2 Simulator.* In Proceedings of the Workshop on ns-2: the IP network, paper n^o 5, 2006.

[5] J. Chen, W. Jiao, H. Wang, *A Service Flow Management Strategy for IEEE 802.16 Broadband Wireless Access Systems in TDD Mode.* In Proceedings of the IEEE International Conference on Communications, pp. 3422-3426, 2005.

[6] J. Freitag, and N. L. S. Fonseca, *Uplink Scheduling with Quality of Service in IEEE 802.16 Networks.* In Proceedings of the IEEE Global

Telecommunications Conference - GLOBECOM, pp. 2503-2508, 2007. DOI `http://dx.doi.org/10.1109/GLOCOM.2007.476`

[7] J. F. Borin, and N. L. S. Fonseca, *Scheduler for IEEE 802.16 networks.* IEEE Communications Letters, vol. 12, no. 4, pp. 274-276, 2008. DOI `http://dx.doi.org/10.1109/LCOMM.2008.072110`

[8] J.F. Borin, and N.L.S. Fonseca, *Simulator for WiMAX networks.* Simulation Modeling Practice and Theory, vol. 16, no. 7, pp. 817-833, 2008. DOI `http://dx.doi.org/10.1016/j.simpat.2008.05.002`

[9] J. F. Borin, and N. L. S. Fonseca, *Uplink Scheduler and Admission Control for the IEEE 802.16 Standard.* In Proceedings of the IEEE Global Telecommunications Conference - GLOBECOM, pp. 1-6, 2009. DOI `http://dx.doi.org/10.1109/GLOCOM.2009.5425779`

[10] J. Sun, Y. Yau, and H. Zhu, *Quality of Service Scheduling for 802.16 Broadband Wireless Access Systems.* In Proceedings of the IEEE 63rd Vehicular Technology Conference, pp. 1221-1225, 2006.

[11] K. Wongthavarawat e A. Ganz, *IEEE 802.16 based last mile broadband wireless military networks with Quality of Service support.* In Proceedings of the IEEE MILCOM'03, pp. 779-784, 2003.

[12] N. Shrivastav, *A network simulator model of the DOCSIS protocol and a solution to the bandwidth-hog problem in the cable networks.* Master Dissertation. North Carolina State University, EUA, 2003

[13] *OPNET.* http://www.opnet.com/WiMAX/index.html, 2007.

[14] R. Jain. *The Art of Computer Systems Performance Analysis.* John Wiley & Sons, Inc., 1991.

[15] R. Rouil. *The Network Simulator ns-2 NIST add-on - IEEE 802.16 model (MAC+PHY).* Available at http://www.antd.nist.gov/seamlessandsecure, 2007.

[16] *The Network Simulator - ns-2.* http://www.isi.edu/nsnam/ns/, 2002.