An Improved IEEE 802.16 WiMAX Module for the NS-3 Simulator

M.A. Ismail*
ismail@sophia.inria.fr

G. Piro**, L.A. Grieco**
{g.piro,a.grieco}@poliba.it

T. Turletti*
turletti@sophia.inria.fr

(*) INRIA, Planète Project- Sophia Antipolis, France
(**) DEE - Politecnico di Bari, Italy

ABSTRACT

IEEE 802.16 WiMAX is a promising new wireless technology for providing broadband ubiquitous network access. As more and more researchers and industrials are interested in simulating such networks, a number of WiMAX simulators have been emerged in the networking community. One of the most recent WiMAX simulator is the one developed for ns-3. This module provides a standard compliant and well designed implementation of the standard and benefits from the major enhancements provided by ns-3 (compared to other network simulators) which has all the capabilities of becoming the leading network simulator in near future. However, this WiMAX module still lacks some important features which motivated this work. In this paper, we first provide a snapshot of existing WiMAX simulators available in the public domain, while highlighting their limitations. Then, we describe the new features and enhancements we have integrated within the ns-3 WiMAX module, and in particular: a realistic and scalable physical model, an IP packet classifier for the convergence sub-layer, efficient uplink and downlink schedulers, support for multicast traffic and pcap packet tracing functionality. The new design of the physical layer has improved the simulation time by several magnitude orders while still providing a realistic implementation of the standard. Furthermore, the IP classifier has enabled the simulation of an unlimited number of service flows per subscriber station, while the proposed schedulers improve the management of the QoS requirements for the different service flows.

Categories and Subject Descriptors

I.6.5 [**Simulation and Modeling**]: Model Development—*Modeling methodologies*; I.6.7 [**Simulation and Modeling**]: Simulation Support Systems—*Environments*

General Terms

Algorithm, Design, Performance, Verification

Keywords

IEEE 802.16, WiMAX, Performance, Verifications, Scheduling

1. INTRODUCTION

WiMAX (Worldwide Interoperability for Microwave Access) is an emerging technology for Broadband Wireless Access in metropolitan area networks, based on IEEE 802.16-2004/2005 standards [14, 15]. It was designed to provide high datarates (up to 100 Mbps), extended coverage (up to 50 Km) for fixed and mobile users, network scalability, security, and support of Quality of Services (QoS). It is currently one of the most promising global telecommunication systems. As a consequence, many modules for simulating its behavior have been proposed in recent years [25][8][13][23].

There are several WiMAX modules for ns-2 [20]. The Networks and Distributed Systems Laboratory (NDSL) proposed the first WiMAX module that supports, among other features, scheduling services and bandwidth management [4]. The National Institute of Standards and Technology (NIST) proposed a WiMAX module that provides a number of features including OFDM PHY layer and fragmentation and de-fragmentation [22]. The WiMAX Forum, together with the Rensselaer Polytechnic Institute (RPI) and the Washington University St. Louis (WUSTL), proposed a module that adds support for QoS scheduling, ARQ mechanism and OFDMA PHY layer [9]. Also, the Computer Networks Laboratory (CNL) developed a module that supports scheduling services and bandwidth management [3]. These modules present serious limitations as we will discuss in section2.

During the last year, two new WiMAX modules have been proposed for both ns-2 [25] and ns-3 [8]. Furthermore, other simulation platforms such as NCTUns [13] and Numbat [23] have been also developed to simulate WiMAX networks. Despite their important contributions, all these simulators present some limitations that will be highlighted in the next section.

In this work, we describe the enhancements we have added to the ns-3 WiMAX module [8] in order to overcome its main limitations. The most important features added are: (1) a realistic and scalable physical model; (2) an IP packet classifier for the convergence sub-layer; (3) efficient uplink and downlink schedulers; (4) support for multicast traffic; and (5) pcap packet tracing functionality. The new physical layer has improved the simulation time by several magnitude orders while still providing a realistic implementation of the standard. Furthermore, the IP classifier has enabled the simulation of an unlimited number of service flows per subscriber station, while the proposed schedulers improve the management of the QoS requirements for the different service flows.

The rest of the paper is organized as follows: Section 2 describes pros and cons of the most important WiMAX modules proposed so far. Section 3 highlights the new features we added to the ns-3 WiMAX module. In Section 4, future developments are described. Finally, Section 5 concludes the work.

2. WIMAX SIMULATORS

Herein we describe the principal differences among the main modules available in the public domain to simulate WiMAX networks: (i) the WINSE (WiMAX ns-2 Extension [25]) module, (ii) the WiMAX module for the ns-3 simulator [8], (iii) the NCTUns Simulation Tool [13] and (iv) the Numbat module for Omnet++ 4.0 [23]. To this aim, we will summarize the main features of the WiMAX technology, by highlighting, for each of them, how it is supported in the four considered simulation platforms.

A WiMAX network includes the BS (Base Station) and the associated SS (Subscriber Station) or MS (Mobile Subscriber). The standard provides specifications for both PMP (Point to Multipoint) and Mesh topologies. In PMP topology, the BS serves a number of SSs that are in the same broadcast range, like a cellular structure. In mesh topologies, instead, ad-hoc environments can be set-up. All the considered simulators support PMP topology, whereas only NCTUns implements mesh networks.

The IEEE 802.16 standard defines also MAC (Medium Access Control) and PHY (Physical) layers. The physical medium is divided into frames of fixed length and each frame is composed by DL (downlink) and UL (uplink) subframes for serving downlink and uplink traffic, respectively. Moreover, medium access is based on both TDMA (Time Division Multiple Access) and FDMA (Frequency Division Multiple Access) with TDD (Time Division Duplex) and FDD (Frequency Division Duplex) as multiplexing techniques [14]. Multiple PHY specifications are supported for LOS (Line of Sight) and NLOS (Non Line of Sight) operational environments in 10-66 GHz and 2-11 GHz frequency bands, respectively. To handle LOS communications, the standard defines the WirelessMAN-SC PHY, based on a single carrier modulation. For the NLOS case, the standard defines the WirelessMAN-SC, the WirelessMAN-OFDM, exploiting Orthogonal Frequency Division Multiplexing, and WirelessMAN-OFDMA, based on Orthogonal Frequency Division Multiple Access. The latter is able to improve multipath performance in NLOS environments [21]. The PHY layer supports also fixed and adaptive modulation techniques [14]. OFDM is implemented in all considered simulators, whereas OFDMA is only supported in WINSE and Numbat. Furthermore, only WINSE supports link adaptation by computing the effective SINR (Signal to Interference and Noise Ratio) from trace files generated with dynamic system simulators. It is worth to note that none of the considered simulators adopts a sophisticated approach to model the PHY layer, i.e., PHY is implemented as an abstraction of real PHY, ignoring any underlying PHY layer details.

In order to handle packet retransmissions, the standard specifies an ARQ mechanism (including ARQ blocks, ARQ block rearrangement, ARQ feedback, ARQ windows, ARQ timers and ARQ discard) on top of the MAC layer. Only WINSE and NCTUns support ARQ.

Before providing a certain type of data service, the Base Station and the Subscriber Station establish an unidirectional logical link called *connection*. In this way, the MAC architecture can use the connection-oriented approach to provide accurate control over the air interface and to enable end-to-end QoS control. Two types of connections have been conceived: management ones, used to transmit and receive control messages (including Initial Ranging, Broadcast, Basic and Primary messages), and transport connections for data transmission. Regarding the possibility to discriminate packets belonging to multiple transport connections hosted on the same node, IP classifier has been only implemented in WINSE and NCTUns modules. In both simulators, the classification is based upon: (i) source address, (ii) destination address and (iii) flow ID.

Management connections can transmit different kind of management messages, the most important ones being DL-MAP (Downlink Map), UL-MAP (Uplink Map), DCD (Downlink Channel Descriptor), UCD (Uplink Channel Descriptor), RNG-REQ (Ranging Request), RNG-RSP (Ranging Response), DSA-REQ (Dynamic Service Addition Request), DSA-RSP (Dynamic Service Addition Response), DSA-ACK (Dynamic Service Addition Acknowledge) and REP-RSP. DL-MAP and UL-MAP messages are used to map the network access to the downlink and uplink subframes respectively. DCD and UCD messages include information about the channel as well as a set of DL and UL burst profiles, respectively. RNG-REQ and RNG-RSP messages are exchanged between BS and SS during the initial ranging process to setup network entry and to create management connections. REP-RSP message includes quality channel reports. Finally, DSA messages are used to create a service flow. WINSE implements only the following management messages: DL broadcast messages, DL-MAP, UL-MAP, DCD, UCD and REP-RSP. In addition, the ns-3 module [8] defines also: RNG-REQ, RNG-RSP, DSA-REQ, DSA-RSP, and DSA-ACK. Further messages have been defined in the Numbat module [23] such as: (i) SBC-REQ, SBC-RSP,PKM-REQ,PKM-RSP to handle network entry; (ii) SCN-REQ, SCN-RSP to manage scanning procedure; (iii) BSHO-REQ, MSHO-RSP, HO-IND to enable handover. No information on control messages adopted in NCTUns are available on the official documentation.

To provide QoS Management, the WiMAX standard introduces a centralized architecture exploiting three schedulers: two of them run at the BS to allocate uplink (UL) and downlink (DL) bandwidth, respectively; the third one runs at each SS and distributes among multiple connections the UL bandwidth assigned by the BS. In each frame, the BS schedulers allocate DL and UL bandwidth by taking into account the QoS needs of downlink connections and the SSs bandwidth requests, respectively. The bandwidth is requested on per connection basis but it is allocated on per SS basis. Four mechanisms have been defined to request uplink bandwidth: unsolicited bandwidth grants, piggyback bandwidth request, unicast polling and contention based procedures (including broadcast or multicast polling). Five scheduling services are defined: Unsolicited Grant Service (UGS), Real-time Polling Service (rtPS), Extended Real-time Polling Service (ertPS), Non-real-time Polling Service (nrtPS) and Best Effort (BE).It is important to note that the standard does not define any scheduling algorithm for these services, leaving, in this way, vendors free to implement their own solutions. The WINSE module supports all five scheduling services (UGS, ertPS, rtPS, nrtPS, BE) defined by the 802.16-2004 and 802.16e-2005 standards [14] [15]. Moreover, it adopts a simple PQ (Priority Queue) scheduling strategy for BS ans SS schedulers. The schedulers allocate first the bandwidth to management connections and then to transport connections, according to flow priorities (UGS > ertPS > rtPS > rtPS and BE). The bandwidth request mechanism is implemented with both standalone and piggyback approaches. The ns-3 module supports only four scheduling services (UGS, rtPS, nrtPS, BE) defined by the 802.16-2004 standard [14]. Each service flow is associated to exactly one scheduling service, and the QoS parameter set associated to a service flow defines the scheduling service it belongs to. Regarding BS uplink and SS schedulers, only a priority-based approach has been proposed in ns-3. Connections are served starting from those with the highest priority and ending to those with the lowest priority (UGS > rtPS > nrtPS > BE). Moreover, connections having the same priority are served on FCFS (First Come First Served) basis. Differently from the BS uplink scheduler, the BS downilink one assigns a higher priority to the broadcast, initial ranging and management connections, with respect to data connections (broadcast > initial ranging > basic > primary > UGS > rtPS > nrtPS > BE). The Numbat (NCTUns) module provides a very weak support to scenarios with QoS differentiation, because it supports only BE and UGS (BE) traffic classes.

Finally, handover support is simulated in all considered simulators with the exception of ns-3.

To summarize, the WINSE module, despite its important contribution, has some limitations. First of all, no sophisticated PHY layer has been developed and, furthermore, the module lacks of some worth WiMAX features such as service flows initialization, RNG-REQ and RNG-RSP messages, and primary management connection. Also the ns-3 WiMAX module presents some limitations. In fact, due to the lack of classifier, the module allows the simulation of only one service flows per SS. Moreover, no propagation and error models have been implemented and the BS is not able to update burst profile information dynamically. Furthermore, the proposed schedulers serve flows that have the same scheduling type on FCFS basis without considering that these flows could have different actual needs. Finally, the simple scheduler is not able to satisfy the expected QoS levels of all active data connections. NCTUns and Numbat, instead, provides a very weak support to QoS management and adopts a very simple PHY layer model.

To conclude, Tab.1 reports the main features of the modules analyzed in this section.

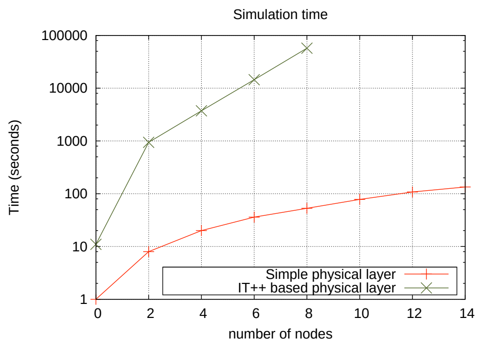

Figure 1: Time needed to execute a WiMAX scenario of 5 second while varying the number of SS.

3. NEW FEATURES ADDED TO THE NS-3 WIMAX MODULE

To overcome the main limitations of the existing ns-3 WiMAX module described in the previous section, we have developed new features and we have enhanced some existing ones. The most important features we have developed are: (1) A scalable and realistic physical and channel model; (2) a packet classifier for the IP convergence sublayer; (3) efficient uplink and downlink schedulers; (4) support for Multicast and Broadcast Service (MBS); and (5) pcap packet tracing functionality. Herein we describe in details the design and implementation of theses new features.

3.1 The physical and channel model

In [8] two different versions of physical layer are proposed. The first one is very basic, it simply forwards bursts received by the MAC layer ignoring any underlying physical layer details. The second one is a more complete implementation. It uses an external OFDM module [6] which implements the fundamental operations of the OFDM specification using the IT++ library [16]. So why do we need another physical layer? To answer this question we have to analyze the two previous physical implementations in depth. The first one is very simplistic, it does not take into account the specificity and constraints related to the WiMAX physical layer and thus any simulation performed using this implementation is not realistic and may lead to erroneous results. Conversely, the second implementation is very complete, and all the OFDM operations are implemented. However it is hardly usable for scalability reason. The IT++ library consumes a huge amount of CPU resources to perform encoding, decoding and modulation resulting to very long simulation time. To illustrate the problem, we have developed a simplistic PMP WiMAX scenario in which half of the subscriber stations sends UDP packets at 1Mbps and the other half receives them during 10 seconds (simulation time). In Figure 1, we draw the (real) time needed to simulate this scenario while varying the number of subscriber stations. The program was compiled in optimized mode (waf -d optimized) using g++-4.2 and has been executed on a Linux machine with a CPU of 2GHz simple core and 2GB of RAM.

As shown in Figure 1, the time needed to execute this sce-

PHY and MAC features	WINSE	ns-3 WiMAX module	Numbat	NCTUns
PHY				
OFDM	x	x	x	x
OFDMA	x		x	
Link Adaptation	x			
Topology				
PMP	x	x	x	x
MESH				x
MAC				
ARQ	x		x	
Initial Ranging	x	x	x	x
Service Flow Initialization		x	x	x
Management Connection	x	x	x	x
Transport Initialization	x	x	x	x
ertPS connection	x			
UGS	x	x	x	
rtPS and nrtPS connections	x	x		
BE connection	x	x	x	x
IP Packet Classi↓er	x			x

Table 1: Main features of the considered WiMAX simulators.

nario using the IT++ based physical layer is more than 10 hours when the number of SS is equal to 8. In comparison the simple physical layer the same simulation takes about 3 minutes.

We chose the Wireless MAN OFDM PHY specifications as the more relevant for implementation as it is the schema chosen by the WiMAX Forum [10]. This specification is designed for non-light-of-sight (NLOS) including fixed and mobile broadband wireless access.

The proposed model uses a 256 FFT processor, with 192 data subcarriers. It supports all the seven modulation and coding schemes specified by Wireless MAN-OFDM. It is composed of two parts: the channel model and the physical model.

3.1.1 Channel model

The channel model we propose is implemented by the class $SimpleOFDMWimaxChannel$ which extends the wimaxchannel class. The channel entity has a private structure named *m_phyList* which handles all the physical devices connected to it. When a physical device sends a packet (FEC Block) to the channel, the channel handles the packet, and then for each physical device connected to it, it calculates the propagation delay, the path loss [1] according to a given propagation model and eventually forwards the packet to the receiver device. The principal functions of this class are described by algorithm 1.

The channel class uses the method $GetDistanceFrom()$ to calculate the distance between two physical entities according to their 3D coordinates. The delay is computed as $delay = \frac{distance}{C}$, where C is the speed of the light.

We have implemented three different propagation models for the path loss (L) calculation: FRIIS, LOG-DISTANCE and COST-Hata-Model.

[1]Path loss (or path attenuation) is the reduction in power density (attenuation) of an electromagnetic wave as it propagates through space.

Algorithm 1 Channel model algorithm

```
INPUT:
  $FecBloc$, $TxDevice$ , $TxPower$, $ModulationType$
00: BEGIN
01:  FOR each $RxDevice$ IN $m\_phyList$ DO
02:    IF $RxDevice$ <> $TxDevice$ THEN
03:      compute $Distance$ ($3RxDevice$, $TxDevice$)
04:      compute $Delay$ ($distance$)
05:      compute $PathLoss$ ($distance$, $PropagationModel$,
                     $RxDevice$, $TxDevice$)
06:      schedule Receive ($Delay$, $RxDevice$, $TxPower$,
                     $ModulationType$, $PathLoss$,
                     $FecBloc$)
07:    END IF
08:  END FOR
09: END
```

Friis propagation. The Friis transmission equation is well known in telecommunications engineering, and gives the power received by one antenna under idealized conditions given another antenna some distance away transmitting a known amount of power [12]. This simple form applies only under ideal conditions. But ideal conditions are almost never achieved in ordinary terrestrial communications, due to obstructions, reflections from buildings, and most importantly reflections from the ground.

Log distance propagation. The log-distance path loss model is a radio propagation model that predicts the path loss a signal encounters inside a building or densely populated areas over distance. It is formally expressed as:$L = L_0 + 10nlog_{10}(\frac{d}{d_0}) + X$, where n is the pathloss exponent, L is the total path loss, L_0 is the path loss at the reference distance d_0, d is the length of the path, and X is a random variable with zero mean, reflecting the attenuation caused by shadowing. In case of no shadowing, this variable is 0.

This model is valid only for frequencies close to 1900 MHz, for a receiver with omnidirectinal antennas at a height of two meters and BS antenna heights between 10 meters and 80 meters.

COST-Hata-Model. The COST-Hata-Model is the most often cited of the COST231 models[1], also called the Hata Model PCS Extension. It is a radio propagation model that extends the Hata Model (which in turn is based on the Okumura Model) to cover a more elaborated range of frequencies. It is formulated as:

$$L = 46.3 + 33.9log_{10}(f) - 13.82log_{10}(h_B) - \\ a(h_R) + (44.9 - 6.55log_{10}(h_B))log_{10}(D) + C \quad (1)$$

Where L is the total path loss, f is the transmission frequency, h_B is the base station antenna effective height [2], d is the length of the path, h_R is the subscriber station antenna effective height, C is the correction factor equal to $0dB$ for medium cities and suburban areas, and $3dB$ for metropolitan areas. $a(h_R)$ is the subscriber station antenna height correction factor as described in the Hata Model for Urban Areas. It is computed as $a(h_R) = (11log_{10}f - 0.7)h_R - 1.56log_{10}f - 0.8$

This model is mainly applied in the 1.5-2GHz carrier frequency, 30-300m Base Station height, 1-10m subscriber station height and 1-20km distance. The WiMAX Forum recommends using COST231 model for system simulations and network planning of macro-cellular systems in both urban and suburban areas for mobility applications. The WiMAX Forum also recommends adding a 10dB fade margin to the path loss to account for shadowing.

3.1.2 Physical model

The physical layer performs two main operations: (i) It receives a burst from a channel and forwards it to the MAC layer, (ii) it receives a burst from the MAC layer and transmits it on the channel.

Transmission process. A burst is a set of WiMAX MAC PDUs. At the sending process, a burst is converted into bitstreams and then splitted into smaller FEC blocks which are then sent to the channel with a power equal P_{tx}.

Reception Process. The reception process includes the following operations:

1. Receive a FEC block from the channel
2. Calculate the noise level N using formula [24]: $N = -114 + N_F + 10log_{10}(BW)$, where: N_F is the noise figure of the receiver [3], and BW is the bandwidth.
3. Estimate the signal to noise ratio (SNR) with the following formula: $SNR_{rx} = P_{tx} - L - N$.
4. Determine if a FEC block can be correctly decoded.
5. Concatenate received FEC blocks to reconstruct the original burst.
6. Forward the burst to the upper layer.

Two processes have been developed to evaluate if a FEC block can be correctly received or not. The two processes use pre-generated trace files on this purpose. The trace files are generated by the OFDM simulator (described later). A class named $SNRToBlockErrorRateManager$ handles a repository containing seven trace files (one for each modulation and coding scheme). A repository is specific for a particular channel model. A trace file is made of 4 columns. The first column provides the SNR value (1), whereas the other columns give respectively the block error rate (2), the block error rate knowing that the previous block was correct (P_{GB}) (3) and block error rate knowing that the previous block was erroneous $(1 - P_{BG})$(4). These trace files are loaded into memory by the $SNRToBlockErrorRateManager$ entity at the beginning of the simulation.

The first process uses the first and second columns to determine if a FEC block is correctly received. When the physical layer receives a packet with an SNR equal to SNR_{rx}, it asks the $SNRToBlockErrorRateManager$ to return the corresponding block error rate $BlER$. A random number $RAND$ between 0 and 1 is then generated. If $RAND$ is greater than $BlER$, then the block is correctly received, otherwise the block is considered erroneous and is ignored.

The second process uses the first, the third and the fourth columns of the trace files. It maintains a Markov chain with two states "good" and "bad" (Gilbert model). In the good state, the block is correctly received, whereas in the "bad" state the block is considered erroneous and is ignored. When the physical layer receives a packet with SNR equal to SNR_{rx}, it asks the $SNRToBlockErrorRateManager$ to return the corresponding values of P_{GB} and P_{BG}. A random number $RAND$ between 0 and 1 is then generated. If the current state is "good" and $RAND$ is lower than P_{BG} the state is changed to "bad" otherwise the state remains "good". If the current state is "bad" and $RAND$ is lower than P_{GB} the state is changed to "good" otherwise the state remains "bad".

3.1.3 OFDM Simulator

To generate trace files giving the block error rate for each modulation and coding scheme at a given SNR value, we have simulated WiMAX OFDM. This simulation was developed in $C++$ and uses an external mathematics and signal

[2]In telecommunication, the term effective height refers to the height of the center of radiation of an antenna above the effective ground level.

[3]Noise figure is a measure of degradation of the signal to noise ratio (SNR), caused by components in the RF signal chain. It is a number by which the performance of a radio receiver can be specified.

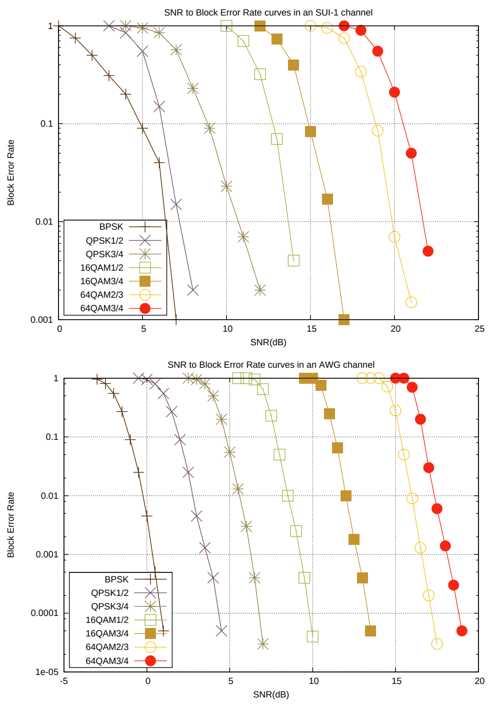

Figure 2: SNR to Block Error Rate curves in a SUI-1 and AWGN channels.

processing library $IT++$. This simulator is developed according to [14] and includes: a random block generator, a Reed Solomon (RS) coder, a convolutional coder, an interleaver, a 256 FFT-based OFDM modulator, a multi-path channel simulator and an equalizer. The multipath channel is simulated using the *TDL_channel* class of the $IT++$ library. Several channel profiles are already implemented such as: ITU channel models, COST 207 channel models and COST 259 channel models.

Given a channel profile, for each modulation and coding scheme, we run several simulations while varying the SNR. The simulator outputs the values of $BlER$, P_{GB} and P_{BG}.

Figure 2 shows the obtained SNR to BLER curves for all the coding and modulation scheme of WiMAX in an AWGN and and SUI-1 channels.

Note that these trace files could be generated by other WiMAX OFDM simulators or ideally by performing experiments in real environments.

3.2 IP packet classifier for the IP Convergence Sub-layer

The IEEE 802.16 convergence sublayer (CS) is the component of the MAC that is responsible for mapping between the MAC service and the internal connection oriented service of the MAC CPS (Common Part Sublayer), through classification and encapsulation. The classifier implements a set of matching criteria applied to each packet entering CS layer. All data packets for the upper layer are associated with a unique service flow ID (SFID) or CID. The classifier classifies packets regarding their SFID and appends them to the appropriate queue. The scheduler then retrieves packets from the queues and transmits them on the link at the appropriate time slots as defined by the UL or DL map messages.

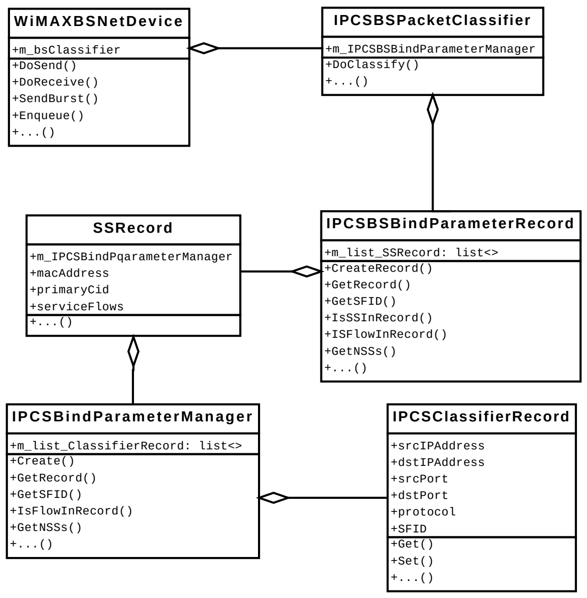

Figure 3: Class diagram of the BS classifier

The WiMAX simulator presented in [8] lacks a full implementation of classifier. Due to this limitation, the module supports only one service flow per subscriber station. To overcome this limitation, we developed two IP CS packet classifiers: The first classifier, for the SSs, operates on the uplink flows, whereas the second one, for BSs, operates on the downlink flows.

The SS classifier is used to map incoming packets from the IP layer to appropriate connections based on a set of criterion. It maintains a list of packet matching rules which associates to each IP flows a service flow ID (SFID). When a packet is received from the upper layer, the classifier extracts the source IP address, the destination IP address, the source port, the destination port and the protocol from the headers. Then it checks if there is a record in the packet matching rule list corresponding to the current processed packet. If such record is found, the corresponding SFID is extracted and the packet is appended to the queue of the service flow. Otherwise, the packet is appended to the queue of the default service flow defined by the simulation scenario.

The classifier of the BS maintains a list containing the registered SSs. Each record in this list contains another list of packets matching criteria. When a packet is received from the upper layer, The BS classifier extracts the destination IP address and checks it against the IP address of the registered SSs. If no corresponding SS is found the packet is released. Otherwise, the remaining parameters of the IP flows are extracted from the headers of the packet, and the same operations described for the SS classifier are performed.The class diagram of the BS classifier is represented in figure 3.

3.3 Uplink and Downlink schedulers

As described in Section 1, the standard defines three schedulers without imposing to vendors any specific scheduling algorithm. As a consequence, a lot of research is currently ongoing on this topic and a number of algorithms has been proposed. [26] provides an extensive survey of recent scheduling algorithms that can be applied to WiMAX networks. These algorithms can be classified into two categories: channel-unaware schedulers and channel-aware-schedulers.

The channel-unaware ones, which do not take into account channel conditions, include Round Robin (RR), Deficit Round Robin (DRR), Weighted Deficit Round Robin (WDRR), Weighted Fair Queueing (WQF), Priority Queue (PQ), Deficit Fair Priority Queue (DFPQ), Earliest Deadline First (EDF) [28], Largest Weighted Delay First (LWDF) [27], and Delay Threshold Priority Queueing (DTPQ) [18].

The channel-aware schedulers include Proportional Fair (PF), Maximum C/I (known as also Maximum Throughput), Modified Largest Weighted Delay First (M-LWDF) [2], Link-Adaptive Largest Weighted Throughput (LA-LWT) [17].

Recent works have compared the performance of many well-known scheduling strategies, applied to WiMAX networks (see [19], [5], [7]). They have demonstrated that there is not a single scheduling scheme that provides superior performance with respect to the other ones and that the development of an efficient, fair and robust scheduler for WiMAX is still an open research area.

To provide an important support to research activities on scheduling, we have implemented pure virtual uplink and downlink scheduler classes (called *UplinkScheduler* and *DownlinkScheduler* respectively) that offer a basic implementation of some methods common to all scheduling strategies. In this way, it is simple to add a generic scheduling algorithm, by: (i) creating a new scheduling class, (ii) inheriting methods from the virtual class and (iii) defining methods, in which are implemented scheduling functions, according to the scheduling design. Finally, to demonstrate the flexibility of the new module, we have enhanced the priority-based scheduling schemes already proposed in [8] and we have implemented the new UL scheduler proposed in [11].

3.3.1 The enhanced BS Uplink scheduler

A new BS uplink scheduler has been implemented by the *UplinkSchedulerRtps* class. It works very similarly to the simple scheduler proposed in [8] except for the rtPS connections. Its goals are (1) to serve all active rtPS connections every uplink subframe and (2) to assign to each rtPS connection an uplink burst that is able to fully satisfy the bandwidth request, subject to the constraint on the total UL bandwidth. The scheduler is based on a priority-based approach. The priority is enforced by serving active connections from the highest to the lowest priority (UGS $>$ rtPS $>$ nrtPS $>$ BE).

For rtPS connections, the BS allocates periodically request opportunities, which are exploited by SSs to send bandwidth requests. Bandwidth requests are computed by SSs to fully transmit their pending rtPS data. Knowing bandwidth requests, the BS allocates the UL according to the following two steps:

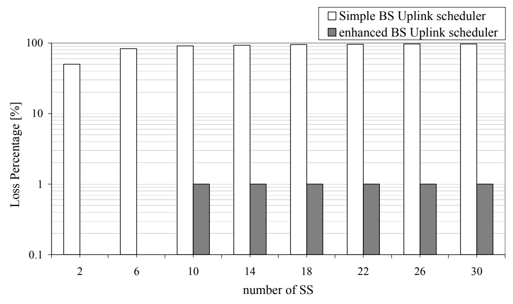

Figure 4: Packet Loss Ratio vs. number of SSs.

`Step 1: UL allocation.`
The BS virtually allocates to each rtPS connection an uplink burst that is able to fully satisfy the requested bandwidth. At the end of this step the scheduler compares the sum of bandwidth requests with respect to the real UL bandwidth.

Let $ReqBW(i)$ and aBW be the bandwidth virtually assigned to the $i-th$ rtPS connection and the uplink bandwidth available for rtPS connections, respectively.

A channel saturation occurs iff $\sum_{i=1}^{N} ReqBW(i) > aBW$, where N is the number of active rtPS connections.

If a channel saturation occurs the virtual bandwidth assignment previously computed is scaled down by Δ, so that the capacity constraint $\sum_{i=1}^{N}(ReqBW(i) * \Delta) \leq aBW$ is satisfied, where $\Delta = \frac{aBW}{\sum_{i=1}^{N} ReqBW(i)}$.

`Step 2: Creation of Uplink Bursts and Update of the` *`RequestedBandwidth`* `Value.`
During the final step, the uplink scheduler creates uplink bursts and then stores for each rtPS connection the difference between requested and allocated bandwidths. These values will be considered as default bandwidth requests for the next uplink sub-frame.

To compare the enhanced BS Uplink scheduler to the simple one, we have considered a scenario with only rtPS service flows, implemented by udp client/server applications. CBR traffic sources sending one 1024 Bytes sized packet every 0.12 s have been used. More precisely, in this scenario each SS hosts a client application and the BS hosts all the server applications. To describe the improvements achieved by this scheduler with respect to the simple one, Figs. 4-6 show the packet loss ratio, the percentage of served SS, and packet latencies as a function of the number of SSs.

It is worth to note that, unlike the simple BS uplink scheduler, the enhanced BS Uplink scheduler is able to serve all active SSs, while providing latencies smaller than 30 ms. Furthermore, the enhanced scheduler is also able to lower packet loss ratio of more than one order of magnitude with respect to the simple one.

3.3.2 The Migration-based Quality of Service uplink scheduler

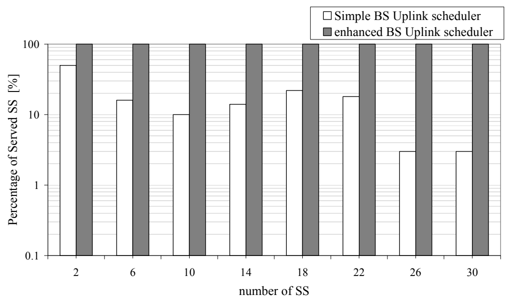

Figure 5: Ratio of served SSs vs number of SSs.

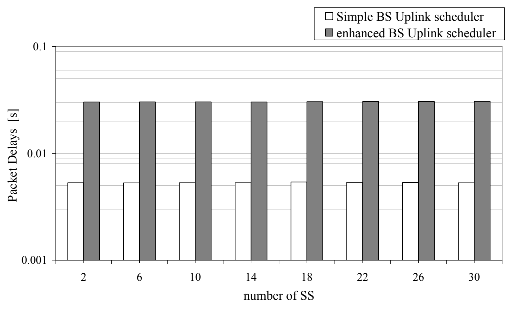

Figure 6: Packet latency vs. number of SSs.

A MBQS (Migration-based Quality of Service) uplink scheduler, proposed in [11], has been implemented [4] for the uplink scheduler into *UplinkSchedulerMBQoS* class.

This scheduler uses three queues: the low priority queue, the intermediate queue and the high priority queue. It serves the requests in strict priority order.

To guarantee the QoS expected by the UGS connections, UGS data grants are periodically inserted into the high priority queue. Moreover, the high priority queue holds also the unicast request opportunities that must be scheduled in the following frame.

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.

In particular, the BS assigns a deadline to each rtPS bandwidth request in the intermediate queue. So, each time the scheduler is executed, the bandwidth requests having deadlines that would expire in the following frame migrate to the high priority queue.

Moreover, the scheduler is able to guarantee a minimum bandwidth requirement for both rtPS and nrtPS connections over a time window of fixed duration. For that purpose, the scheduler assigns, every time it is executed, a priority value that is inversely proportional to the amount of bandwidth received by each connection. After this task, the rtPS and nrtPS connections are sorted and served according to their priorities.

Finally, the low priority queue is used to store the bandwidth requests for providing a BE service.

[4]This work has been done during the Google Summer of Code'09. Google Summer of Code is a global program that offers student developers stipends to write code for various open source software projects. More details can be found at: http://socghop.appspot.com/

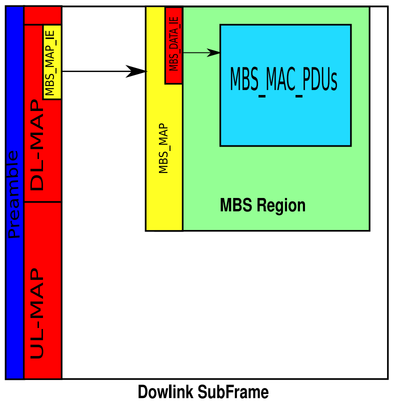

Figure 7: DL sub-frame structure with MBS zone.

3.3.3 *The enhanced BS Downlink scheduler*

The Downlink scheduler, implemented by the *DownlinkSchedulerRtps* class, is very similar to the UL scheduler described in Section. 3.3.1.

However, in this case the highest priority is assigned to the broadcast, initial ranging and management connections (broadcast $>$ initial ranging $>$ basic $>$ primary $>$ UGS $>$ rtPS $>$ nrtPS $>$ BE).

Scheduling decisions are made in two consecutive steps. During *step*1, the scheduler: (i) virtually assigns to all rtPS connections a downlink burst that is able to fully drain the MAC queue and (ii) handles channel saturations as done for the uplink scheduler. During *step*2, the scheduler creates downlink bursts and updates MAC queue lengths according to the bandwidth assigned to each rtPS connection.

3.4 Adding multicast support

In the proposed module, the MBS service can be supported by either constructing a separate MBS OFDM zone in the downlink frame along with unicast service or the whole OFDMA downlink frame can be dedicated to MBS. One MBS_MAP_IE descriptor is included in DL_MAP for each MBS zone. The MBS_MAP_IE specifies the location and the physical configuration of the MBS zone. The location is defined using the OFDM symbol offset parameter. Each MBS zone contains one MBS map located at the first sub-channel of the first OFDM symbol. An MBS map contains multiple MAP_DATA_IEs. A MAP_DATA_IE specifies the location, the physical configuration and the CID of an MBS burst. An MBS burst consists of a number of MAC PDUs (Figure 7).

The BS establishes a downlink multicast service by creating a connection with each SS to be associated with the service. Any available traffic CID value may be used for the service (i.e., there are no dedicated CIDs for multicast transport connections). To ensure proper multicast operation, the CID used for the service is the same for all SSs on the same channel that participate in the connection. The SSs need not to be aware that the connection is a multicast or broadcast transport connection. The data transmitted on the connection with the given CID is received and processed by the MAC of each involved SS. Thus, each multicast SDU is transmitted only once. Since a multicast transport connection is associated with a service flow, it is associated with the QoS and traffic parameters for that service flow.

3.5 Other Enhancements

In addition to the features described in previous sections, some other features have been developed such as:

- Pcap packet tracing: To analyze the WiMAX traffic and check the correctness of the generated 802.16 MAC PDU format, we have implemented a high level API (in the wimax-helper class) to enable the tracing function from the simulation script. The tracing function generates one pcap file per net-device. The generated files can be analyzed using wireshark or tcpdump after installing the required plugins to decode WiMAX PDU and MAC2MAC headers.
- Improvement of the logging functionality: The proposed model uses the ns-3 logging module to provide a selectable, multi-level approach to message logging. Logging can be disabled completely, enabled on a component-by-component basis, or enabled globally; and it provides selectable verbosity levels. Each level can be requested singly or cumulatively; and logging can be set up using a shell environment variable (*NS_LOG*) or by logging system function call
- Improvement of the memory management: the memory management has been highly improved making the module using less memory and avoiding memory leaks. Valgrind tools have been used for this purpose. Figure 9 reports the memory usage of two PMP WiMAX scenarios while varying the number of nodes. The first scenario is the one described in Section 3.1 while the second one simulates a multicast transmission from BS to SSs at a rate of 4Mbps.
- Enhancing the scalability in term of CPU usage: in addition to the new physical layer, we have optimized several parts of the code to reduce the CPU usage while still providing the same functionalities. Figure 9 reports the time required to execute the two WiMAX scenarios described previously while varying the number of nodes. The first scenario which took more than 10 hours to be executed with the previous version of the module takes now less than 12 seconds on the same Linux 2GHz CPU and 2GByte RAM machine!

4. CONCLUSION AND FUTURE WORK

This paper presents detailed design and implementation of new features to enhance the existing WiMAX module of the ns-3 simulator. The proposed features includes a realistic and scalable physical/channel model, an IP packet classifier, sophisticated uplink and downlink schedulers, support for multicast traffic and pcap packet tracing functionality. We hope that these features will make easier and more realistic the evaluation and design of WiMAX systems.

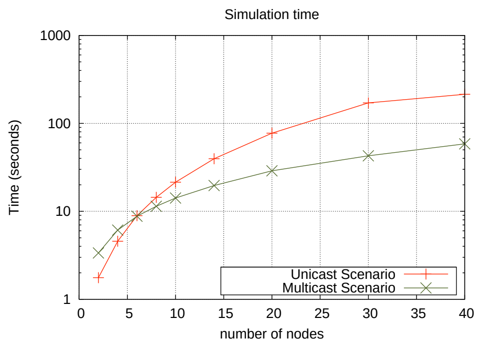

Figure 8: Simulation time vs. number of nodes.

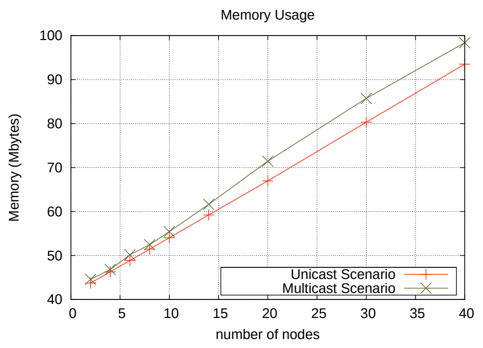

Figure 9: Memory required vs. number of nodes.

The new proposed module[5] has been ported to the latest ns-3 development version and is planned to be merged in version 3.7 of ns-3 expected in end of 2009.

Our WiMAX module still have some limitations. The fragmentation and de-fragmentation of MAC PDUs is not supported. The optional feature of packet header suppression is not implemented too. However work to support these two functionalities is currently under way.

IEEE 802.16e amendment introduces several concepts related to power management. An SS with active connections negotiates with the BS to temporarily turn off its connections to the air interface for a predetermined amount of time and goes into a sleep mode. We plan to implement this functionality for the next release of the module as power saving is crucial for mobile wireless stations.

5. ACKNOWLEDGMENTS

Part of this work was funded by the COST Action IC0703 *Data Traffic Monitoring and Analysis (TMA): theory, techniques, tools and applications for the future networks.*

The authors would like to thank Mariem Abdelmoula for contributing to the design and development of the IP packet

[5]The code is available at http://code.nsnam.org/iamine/ns-3-wimax-release/

classifier during a 4 months internship at INRIA Sophia Antipolis.

6. REFERENCES

[1] Digital mobile radiotowards future generation systems, cost 231 final report. Technical report, Coperation européenne dans le domaine de la recherche Scientifique et Technique.

[2] 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.*, pages 150–154, 2001.

[3] J. F. Borina and N. L. da Fonseca. Wimax module for the ns-2 simulator. In *International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'07), IEEE*, September 2007.

[4] J. Chen, C. Wang, F. Tsai, C. Chang, S. Liu, J. Guo, J. S. W. Lien, and C. Hung. Design and implementation of wimax module for ns-2 simulator. In *In 1st International Conference on Performance Evaluation Methodologies and Tools (VALUETOOLS'06), ACM*, October 2006.

[5] C. Cicconetti, A. Erta, L. Lenzini, and E. Mingozzi. Performance evaluation of the ieee 802.16 mac for qos support. pages 26–38, 2008.

[6] F. Cristiani. Simulation of wimax physical layer. Technical report, Università degli Studi di Napoli Federico II, December 2007.

[7] P. Dhrona, N. A. A. Ali, and H. S. Hassanein. A performance study of scheduling algorithms in point-to-multipoint wimax networks. In *LCN*, pages 843–850, 2008.

[8] J. Farooq and T. Turletti. An ieee 802.16 wimax module for the ns-3 simulator. In *Simutools '09: Proceedings of the 2nd International Conference on Simulation Tools and Techniques*, pages 1–11, ICST, Brussels, Belgium, Belgium, 2009. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[9] W. Forum. Wimax forum system level simulator ns-2 mac+phy add-on for wimax (ieee 802.16). Technical report, July 2008.

[10] W. Forum. Wimax forum. available at `http://www.wimaxforum.org`, 2009.

[11] J. Freitag and N. da Fonseca. Uplink scheduling with quality of service in ieee 802.16 networks. In *Global Telecommunications Conference (GLOBECOM '07), IEEE*, pages 2503–2508, November 2007.

[12] H. T. Friis. Introduction to radio and radio antennas. *Spectrum, IEEE*, 8(4):55–61, April 1971.

[13] S.-M. Huang, Y.-C. Sung, S.-Y. Wang, and Y.-B. Lin. Nctuns simulation tool for wimax modeling. In *WICON '07: Proceedings of the 3rd international conference on Wireless internet*, pages 1–6, ICST, Brussels, Belgium, Belgium, 2007. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

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

[15] IEEE Std. 802.16e-2005. *IEEE Standard for Local and Metropolitan Area Networks - Part 16: Air Interface for Fixed Broadband Wireless Access Systems - Amendment 2: Physical and Medium Access Control Layer for Combined Fixed and Mobile Operation in Licensed Band.* IEEE 802.16e-2005, February 2005.

[16] IT++. It++ library of mathematical, signal processing and communication routines. 2008. available at `http://itpp.sourceforge.net/`, 2009.

[17] Y. J. and S. C. Liew. Link-adaptive largest-weighted-throughput packet scheduling for real-time traffics in wireless ofdm networks. In *IEEE Global Telecommunications Conf., St. Louis, MO*, pages 5–9, 2005.

[18] D. H. Kim and C. G. Kang. Delay threshold-based priority queueing packet scheduling for integrated services in mobile broadband wireless access system. In *IEEE Int. Conf. High Performance Computing and Communications, Kemer-Antalya, Turkey*, pages 305–314, 2005.

[19] J. Lakkakorpi, A. Sayenko, and J. Moilanen. Comparison of different scheduling algorithms for wimax base station: Deficit round-robin vs. proportional fair vs. weighted deficit round-robin. In *IEEE Wireless Communication and Networking Conference (WCNC)*, pages 1991–1996. IEEE, March 2008.

[20] NS-2. Network simulator. available at `http://www.isi.edu/nsnam/ns/`, 2009.

[21] L. Nuaymi. *WiMAX: Technology for Broadband Wireless Access.* Wiley, 1 edition, 2007.

[22] N. I. of Standards and Technology. The network simulator ns-2 nist add-on - ieee 802.16 model (mac+phy). Technical report, June 2007.

[23] Omnet++. numbat - new ubiquitous mobility basic analysis tools: Wimax, dhcpv6 implementation in omnet++. available at `http://klub.com.pl/numbat/`.

[24] S. Saunders and A. Aragón-Zavala. *Antennas and Propagation for Wireless Communications.* Wiley, March 2007.

[25] A. Sayenko, O. Alanen, H. Martikainen, V. Tykhomyrov, O. Puchko, and T. Hämäläinen. Winse: Wimax ns-2 extension. In *Simutools '09: Proceedings of the 2nd International Conference on Simulation Tools and Techniques*, pages 1–10, ICST, Brussels, Belgium, Belgium, 2009. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[26] C. So-In, R. Jain, and A.-K. Tamimi. Scheduling in ieee 802.16e mobile wimax networks: Key issues and a survey. In *IEEE Journal on Special Areas in Communications (JSAC), Vol. t, No. 2.* IEEE, February 2009.

[27] A. L. Stolyar and K. Ramanan. Largest weighted delay first scheduling: Large deviations and optimality. In *Annals of Applied Probability*, pages 1–48, 2001.

[28] H. Zhang. Service disciplines for guaranteed performance service in packet-switching networks. In *Proceedings of the IEEE*, 1995.