Teixeira and Guardieiro *EURASIP Journal on Wireless Communications and Networking* 2011, **2011**:113
http://jwcn.eurasipjournals.com/content/2011/1/113

EURASIP Journal on
Wireless Communications and Networking
a SpringerOpen Journal

RESEARCH **Open Access**

A new and efficient adaptive scheduling packets for the uplink traffic in WiMAX networks

Marcio Andrey Teixeira* and Paulo Roberto Guardieiro

Abstract

In this article, an adaptive scheduling packets algorithm for the uplink traffic in WiMAX networks is proposed. The proposed algorithm is designed to be completely dynamic, mainly in networks that use various modulation and coding schemes (MCSs). Using a cross-layer approach and the states of the uplink virtual queues in the base station, it was defined a new deadlines-based scheme, aiming at limiting the maximum delay to the real-time applications. Moreover, a method which interacts with the polling mechanisms of the base station was developed. This method controls the periodicity of sending unicast polling to the real-time and non-real-time service classes, in accordance with the quality of service requirements of the applications. The proposed algorithm was evaluated by means of modeling and simulation in environments where various MCSs were used and also in an environment where only one type of MCS was used. The simulations showed satisfactory results in both environments.

1. Introduction

The WiMAX technology, based on the IEEE 802.16 standards, is a solution for fixed and mobile broadband wireless access (BWA) networks, aiming at providing support to a wide variety of multimedia applications, including real-time and non-real-time applications. As a broadband wireless technology, WiMAX has been developed with advantages such as high transmission rate and predefined quality of service (QoS) framework, enabling efficient and scalable networks for data, video, and voice. However, the IEEE 802.16 standards do not define the scheduling algorithm which guarantees the QoS required by the multimedia applications. The scheduling algorithm plays an important role in the provisioning of QoS for the different types of multimedia applications. New releases of the standards were published, such as IEEE 802.16m [1] and IEEE 802.16-2009 [2], in which changes were introduced in the MAC and PHY layers, but the scheduling algorithms have not been defined yet. Recent studies show that an efficient, fair, and robust scheduler for WiMAX is still an open research area [3-5]. The design of scheduling algorithms in WiMAX networks is specially challenging because the wireless communication channel is constantly varying. To make better use of the wireless link, the standard defines the use of adaptive modulation functions in the physical layer. However, a new issue emerges: how to make an efficient scheduling of the subscriber stations (SSs), located in different points away from the base station (BS), sending data in different burst profiles, in accordance with the modulation and coding schemes (MCSs) used for data transmission. This issue is important because the scheduler must guarantee the application's QoS requirements and allocates the resources in a fair and efficient way.

In this article, a new and efficient scheduling algorithm for uplink traffic in WiMAX networks is proposed. The proposed algorithm is applied directly to the uplink virtual queues in the BS and aims at supporting the real-time and non-real-time applications. Using a cross-layer approach and based on the earliest deadline first (EDF) scheduling, a new deadlines-based scheme for the real-time applications was defined. The deadlines are computed based on the information about the MCSs (physical layer-PHY), and the bandwidth request (BW-REQ) messages provided by the SSs. Thus, the proposed algorithm minimizes the effects on the QoS parameters resulting from variations on the signal-to-noise ratio (SNR). Moreover, based on the minimum bandwidth requirements of the real-time and non-real-time applications, a method that interacts with the polling mechanisms of BS was developed, aiming at guaranteeing the minimal bandwidth for those applications. This method

---

* Correspondence: marcio.andrey@ifsp.edu.br
Faculty of Electrical Engineering, Federal University of Uberlândia, Uberlândia, Minas Gerais 38400-902, Brazil

has the responsibility of making the balancing of the polling mechanism. The optimal way to request the bandwidth for a given QoS requirement is an open research area [6]. To the best of our knowledge, the proposed scheduling algorithm is the first based on the EDF algorithm that uses deadlines which are calculated according to the various MCSs along with the information about the bandwidth requests provided by the SSs, being appropriated to networks that use adaptive modulations. Moreover, the fact of adapting the polling interval to the bandwidth needs of rtPS and nrtPS connections is ignored in most previous studies, but it has being considered here.

The proposed scheduling algorithm has been evaluated by means of modeling and simulation. Experiments were performed considering environments where various MCSs were used and also environments where only one type of modulation was used. The simulations experiments have shown satisfactory results in both environments.

The remainder of this article is organized as follows: Section 2 presents an overview of IEEE 802.16 standards. Section 3 describes the proposed scheduling algorithm. Section 4 resumes the related works. Section 5 defines the network scenario and the main parameters used in the simulation. Section 6 shows the numerical results. Finally, Section 7 does the final considerations of this article.

2. An overview of IEEE 802.16 standards

The IEEE 802.16 is a set of telecommunication technology standards aimed at providing wireless access over long distances in a variety of ways [2]. The standards specify the PHY characteristics and also the medium access control (MAC) layer. The PHY defines the modulation schemes, the synchronization between the transmitter and the receiver, the multiplexing schemes among other characteristics whose scope is not part of this article. The MAC layer has mechanisms to provide QoS for the downlink and uplink traffics. The packets that cross the MAC layer are classified and associated with a service class. The service class defines a set of QoS parameters, such as delay, throughput, jitter, etc.

The IEEE 802.16 standard has some variants, where each one of them defines different features in the MAC and PHY layers. For example, the IEEE 802.16-2004 standard [7], also known as IEEE 802.16d, provides specifications for fixed BWA systems and addresses the first or last-mile connection in wireless metropolitan area networks. The IEEE 802.16-2005 standard [8] introduces the mobility support and defines a new service class named *extended real-time Polling Service* (ertPS). A new version of the standard is called IEEE 802.16m [9] started in 2007, where some advanced functions were included, mainly to meet 4G system requirements. In 2008, a new system profile called WiMAX Release 1.5 [2] was developed. To improve the received signal strength quality and extend the service of BS, the IEEE 802.16j-2009 [10] standard was published, in 2009, which specifies relay capabilities.

The IEEE 802.16 standards define five service classes: *Unsolicited Grant Service* (UGS), *extended real-time Polling Service* (ertPS), *real-time Polling Service* (rtPS), *non real-time Polling Service* (nrtPS), and *Best Effort service* (BE), in which each service class should be treated differently by the BS, aiming at supporting the coexisting of several multimedia applications, including real-time and non-real-time applications. The scheduling algorithm for the service classes is not defined by the IEEE 802.16 standards. The scheduling algorithm must guarantee the QoS for both multimedia applications (real-time and non-real-time), while efficiently utilizing the available bandwidth. The scheduling is implemented in the SS (uplink traffic) and in the BS (downlink and uplink traffic). However, in this study, it is being addressed the scheduling packets for the uplink traffic in the BS.

The uplink scheduling is more complex than downlink scheduling. In the downlink scheduling, the BS has complete knowledge of the queue status and the BS is the only one that transmits during the downlink subframe. The data packets are broadcasted to all SSs and an SS only picks up the packets destined to it. In the uplink scheduling, the input queues are located in the SSs and are hence separated from the BS. So, the BS does not have any information about the arrived time of packets in the SSs queues. Moreover, the uplink medium access is based on request/grant mechanisms. The SSs need to send bandwidth request messages to the BS, which then decides how many slots are granted to each subsequent uplink subframes.

The standard defines two main request/grant mechanisms: unicast polling and contention-based polling. The unicast polling is the mechanism by which the BS allocates bandwidth to each SS to send its BW-REQ messages. The BS performs the polling periodically. After this, the SSs can send its BW-REQ messages as a standalone message in response to a poll from the BS or it can be piggybacked in data packets. The contention-based polling allows the SSs to send their bandwidth requests to the BS without being polled. The SSs send BW-REQ messages during the contention period. If multiple request messages are transmitted at the same time, collisions may be occurred. There are other mechanisms that the SSs can use to request uplink bandwidth such as multicast polling, channel quality indicator channel, etc. Depending on the QoS and traffic parameters associated with a service, one or more of these mechanisms may be used by the SSs [11]. A comparison of these mechanisms is presented in [6].

Having received the BW-REQ messages sent by SSs, the BS must decide, through the scheduling algorithm, how many slots are provided to each SS in the subsequent uplink subframe. Moreover, it is necessary to consider the overhead caused by the use of polling mechanisms, and to make a balancing of these mechanisms. There are two main reasons for this. First, maximize the channel utilization. To maximize the channel utilization, it is needed to minimize the overhead caused by polling mechanisms. Second, minimize the scheduling delay. This parameter depends on the polling mechanisms adopted by the scheduler, since it corresponds to the interval time when the bandwidth is requested and when it is allocated. Thus, it is needed to use an adaptive polling adjustment scheme to meet the constraints of delay-sensitive applications and to maximize the channel utilization. The optimization of the polling mechanisms is still an open research topic [3].

3. Proposed scheduling algorithm

The WiMAX networks are designed with an MCS method that can alter the modulation and coding rates of a connection based on the state of the wireless link [12]. The standard defines a framework on how to use different MCSs. However, similar to the scheduling, the standard does not define the link adaptation algorithm. Thus, basing on a cross-layer approach, the proposed algorithm was developed to be completely dynamic and predictive, once it is used the MCS method information in the scheduling. The algorithm is applied directly to the uplink virtual queues in the BS and aims at supporting the real-time and non-real-time applications. For this purpose, it was defined a new deadlines-based scheme for the real-time applications, a method for managing the unicast polling mechanism and a module to monitor the BS resources, named *QoSMonitoring*. This module has all information about the resources existent in the BS, and makes an estimative of the delay and throughput of the service classes. This estimative is used along with thresholds defined for the QoS parameters of each service class. The proposed algorithm was developed to work with the five service classes, but in the this study, we analyze the performance of the proposed algorithm with only four service classes: UGS, rtPS, nrtPS, and BE. In future studies, the ertPS service class will be analyzed, when we will include mobility scenarios. Figure 1 shows the proposed scheduling architecture defined in this study. As it can be seen from Figure 1, the proposed scheduling architecture includes: the uplink virtual queues, the BS scheduler module and two new components: the *QoSMonitoring* module and the *Type of* MCS module. Both modules provide information which is used in the scheduling of the service classes. The description of these modules, and also, the description of the proposed scheduling algorithm are made below.

3.1 The UGS scheduling

In accordance with the IEEE 802.16 standards, the UGS service receives unsolicited bandwidth to avoid excessive delay and has higher transmission priority among the other services. Since the resource allocation for the UGS service is made, the scheduling algorithm distributes the remaining resources for the rtPS, nrtPS, and BE services. Once the UGS resources are allocated as specified by

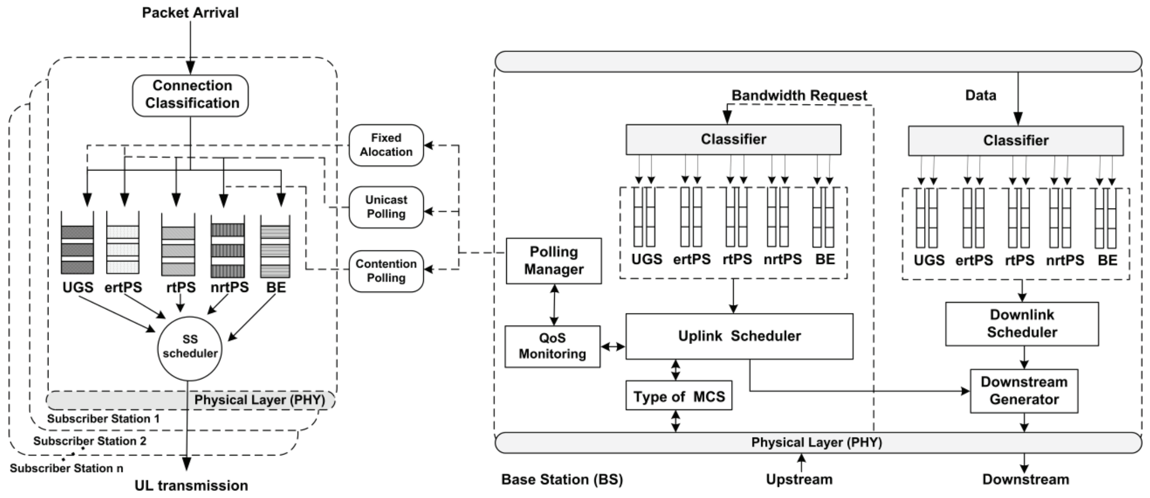

Figure 1 Proposed scheduling architecture.

the standard, the main focuses of the proposed algorithm are the rtPS, nrtPS, and BE service classes.

3.2 The rtPS, nrtPS, and BE scheduling

As described above, the uplink scheduling is made based on the BW-REQ messages sent by SSs. The rtPS service uses unicast polling mechanism and receives from BS periodical grants to send its BW-REQ messages. The nrtPS service can use contention request opportunities or unicast request polling [6]. However, the nrtPS connections are polled on a regular interval to assure a minimum bandwidth. The interval that BS polls the nrtPS connections is defined dynamically by the proposed algorithm. The BE service uses contention base polling to send its BW-REQ messages.

The BS should reserve part of the bandwidth for the polling processes. In addition, the scheduler must guarantee the requirements of limited maximum delay for the rtPS service and the minimal bandwidth for the rtPS and nrtPS services. If there are resources left, it is assigned to the BE service, since this service does not have any QoS requirements.

3.2.1 Ensuring limited maximum delay for rtPS service

In uplink scheduling, the BS maintains a virtual queue for each active uplink connection and updates such virtual queues based on the received BW-REQ messages. The rtPS scheduler guarantees the limited maximum delay for the rtPS service through the use of a new deadlines-based scheme defined in this study. The scheduler assigns a deadline for each rtPS connection. The deadlines calculation is made using the following parameters: the information about the MCSs used for the sending packets between the SS and the BS; the information about the BW-REQ messages sent by the SSs and the information about the polling delay of the rtPS connections. In the rtPs service, the virtual queues are updated within a polling interval. However, a large number of SSs brings a long polling delay [13]. The rtPS scheduler takes into account the polling delay to really guarantee the limited maximum delay.

The proposed scheduler is characterized as being completely dynamic, because of the nature of the parameters used in the deadlines calculation. Suppose that M_i represents the ith BW-REQ message of an rtPS connection in the BS, Equation 1 is used to calculate the ith deadline value.

$$\text{deadline}_i = TT_i + PD_i, \forall i \mid Mi \tag{1}$$

The description of the parameters used on the deadlines calculation is made as follows:

- TT_i: transmission time calculated for each rtPS connection. This calculation is made based on the modulation techniques used in the PHY and based on the size of bandwidth requests. The TT_i parameter is calculated in accordance with the expression (2):

$$TT_i = \left(\frac{\text{BWrequest\_size} \times 8}{\text{bpsymbol}}\right) \times \text{symbol\_time} \tag{2}$$

where BWrequest_size is the amount of bytes requested by the SSs to uplink transmission. This information is obtained from BW-REQ messages sent by SSs; bpsymbol is the amount of bits/symbol used in the transmission. This former parameter is dependent on the MCS used; and symbol_time is the OFDM or OFDMA symbol duration time.

- PD_i: polling delay corresponds to the interval time when the bandwidth is requested and when it is allocated [14]. This parameter is dependent on the number of rtPS connections. When there are few rtPS connections at the network, the polling delay is low but, when the number of rtPS connections increases, the polling delay also increases, being considered in the deadlines calculation.

Once calculated the deadlines, the proposed algorithm organizes the rtPS connections by the lowest deadline. Thus, the scheduler defines the transmission order of the rtPS connections, which is included in the UL-MAP message. The UL-MAP message is sent to the SSs by the downlink channel in each frame. The proposed scheduling algorithm is shown in Figure 2.

The rtPS service is designed to support variable-rate services. Therefore, the scheduling algorithm should guarantee a limit value for the delay and a minimum bandwidth to provide QoS. The algorithm calculates a deadline for each rtPS connection (line 10) as defined in expression (1). After this, it sorts the rtPS connections by the lowest deadline (lines 13-19). Thus, it is possible to minimize the delay existent at the access network by the use of various MCSs. Moreover, it is needed to verify whether the deadlines of the rtPS connections will not expire in the next frame. In this case, it was defined a parameter named L_f. The L_f parameter represents the length of the frame (in terms of time), and is used to verify if the calculated deadlines will not expire (lines 11-12). Thus, it is possible to drop, previously, the BW-REQ messages whose deadlines will not be met.

3.2.2 Ensuring minimal bandwidth for rtPS and nrtPS services

The scheduler ensures the minimal bandwidth for rtPS and nrtPS services in accordance with the minimum bandwidth requirement per connection, the amount of bytes received in a current period, and the amount of backlogged requests (in bytes).

The minimum bandwidth is defined by the *MinimumReservedTrafficRate* variable, which expresses the minimal data rate value in bps and is used as a threshold. In

Proposed *Scheduling Algorithm*

```
1: Verifies the bandwidth messages request at the BS queue;
2: begin
3:    for BWrequest i at the BS queue do
4:      begin
5:        if (BWrequest[i] = rtPS) then
6:          begin
7:            BW[i] += BWrequest[i].length;
8:              if (connections_numbers > 1) then
9:                begin
10:                 deadline[i] = Deadline_Calculation(BWrequest[i].length);
11:                 if (deadline[i] > $L_f$) then
12:                     drop BWrequest[i];
13:                 if (deadline[i] < deadline[i+1]) then
14:                     begin
15:                       tmp = deadline[i+1];
16:                       deadline[i+1] = deadline[i];
17:                       deadline[i] = tmp;
18:                     end;
19:                     UL-MAP = request order by the deadlines;
20:               end;
21:                 QoSMonitoring(i);
22:         end; //rtPS
23:       if (BWrequest[i] = nrtPS) then
24:           begin
25:             BW[i] += request[i].length;
26:                 QoSMonitoring(i);
27:               end; //nrtPS
28:     end;
29:    returns UL-MAP;
30: end;
31: QoSMonitoring(cid);
32: begin
33:   if (cid = rtPS) then
34:     begin
35:       estimate_delay = (estimate_delay * 0.9) + (sample_delay * 0.1);
36:         if (BW[cid] < MinimumReservedTrafficRate) then
37:           Adjust_periodicity_polling(cid);
38:     end; //rtPS
39:   if (cid = nrtPS) then
40:       begin
41          if (BW[cid] < MinimumReservedTrafficRate) then
42:           Adjust_periodicity_polling(cid);
43:       end; //nrtPS
44: end; //QoSMonitoring
45:  Adjust_periodicity_polling(cid)
46: begin
47:   if (cid = rtPS) then
48:     begin
49:       if (estimate_delay < rtPS_threshold) then
50:           polling_interval_rtPS -= α;
51:        else
52:           polling_interval_rtPS  = current;
53:     end; //rtPS
54:   if (cid = nrtPS) then
55:       begin
56:         if (estimate_delay < rtPS_threshold) then
57:           polling_interval_nrtPS -= α;
58:          else
59:           polling_interval_nrtPS += α;
60:        end; //nrtPS
61: end; //Adjust_periodicity_polling
```

Figure 2 Proposed scheduling algorithm.

each frame, the algorithm stores the amount of bandwidth received for each connection (lines 7 and 25), and verifies, using the *QoSMonitoring()* module, if the minimum bandwidth of the rtPS and nrtPS services is within the predefined limits (lines 36 and 41). If so, the BS maintains the initial configuration. This means that the priority of the connections does not change. However, if not, the BS will execute the *Adjust_periodicity_polling()* method, aiming at keeping the minimum bandwidth. This method will increase the priority of the connection among the other connections at the same service class and will decrease the polling interval of the connection. It is important to see (lines 56-59) that the polling interval of the nrtPS connections will be decreased only if the average delay of the rtPS connections is within predefined limits. Otherwise, only the priority of the connection will be changed. The estimated delay is calculated by *QoSMonitoring()* module (line 35). As it can be seen (lines 23-27), the nrtPS scheduling algorithm is simpler than rtPS scheduling algorithm, because this service class does not have temporal restriction.

3.2.3 The dynamic polling management

The BS performs the polling to the SSs periodically. After this, the SSs send its bandwidth requests using the BW-REQ messages, which can be sent as a standalone message in response to a poll from the BS, or it may be piggybacked in data packets. The IEEE 802.16 standards define what service class can use unicast or contention-based polling, but does not define an efficient mechanism to do it. It is necessary to make a balancing of the polling mechanisms because, the more resources are allocated for the polling, the fewer resources are left to the transmission data. To make better use of the polling and to maximize the throughput at the network, the BS, using the *QoSMonitoring()* module, monitors the amount of resources allocated for each service class. The resource assigned to the service classes is represented in the system according to the following classifications: R_{UGS}, R_{rtPS}, R_{nrtPS}, and R_{BE}. The total amount of resources is represented by R_{total}. Making the ratio among these values, for example, using (R_{rtPS}/R_{total}) it is possible to determine the percentage of the resource allocated by the specific service class. Therefore, the *QoSMonitoring()* module verifies if the minimum bandwidth is being guaranteed only comparing such percentage with the predetermined threshold. Moreover, the monitoring module makes an estimative of the delay for the rtPS service, and then, makes the balancing of the unicast polling. As it can be seen in the algorithm (line 35), we use an exponential weighted moving average (EWMA) to estimate the average delay. Figure 3 shows an example of the sample delay and of the estimate delay versus simulation time obtained from the *QoSMonitoring()* module for the rtPS service class [15].

If the average delay of the rtPS class is within predefined limits, it is possible to increase the polling interval to the nrtPS service (line 56-59), making a better distribution of the available resources at the network. Thus, it is possible to control the periodicity of sending

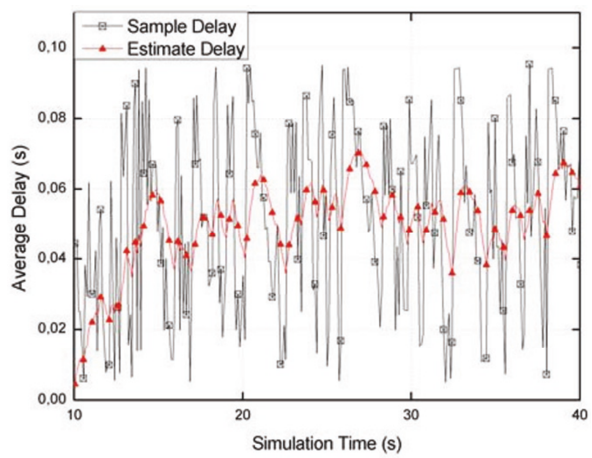

Figure 3 **Estimated delay obtained from the QoSMonitoring module in simulation time**.

unicast polling, in accordance with the QoS requirements of the applications. The symbol "α" in the algorithm represents the polling interval value that will be increased or decreased, depending on the available resources at the networks.

4. Related works

In [16], it is proposed a scheduling algorithm for the rtPS service. This algorithm identifies the SS which has low quality of transmission, and depending on this, the SS is removed temporarily from the scheduler list. In our proposal, a scheduling list of SSs is made based on the deadlines, giving to all SSs opportunity for transmission.

In [17], it was proposed an adaptive packets scheduling algorithm. According to the backlogged traffic, the MCS, and the QoS requirements of the applications, the algorithm allocates the bandwidth in adaptive way for each service class. However, it was not defined the used polling mechanism, being this a very important question to be considered. In this study, it was defined a method that interacts with the polling mechanism of BS, and makes a balancing of unicast polling to the rtPS and nrtPS services.

Gidlund and Wang [18] propose a scheduling algorithm that is a combination of the legacy scheduling algorithms EDF and WFQ, for the uplink traffic. The EDF scheduling is used to control the delay bound for the real-time applications and the WFQ scheduling is used to guarantee minimal bandwidth for the non-real-time-applications. In this study, the algorithm is based on the EDF scheduling, where deadlines are defined to guarantee the delay bound for the real-time applications. The minimal bandwidth is guaranteed through the control of the periodicity of unicast polling for the real-time and non-real-time applications. Thus, our algorithm is less complex than the one described in [18].

In [19], it was defined an analytical technique for obtaining an optimal polling interval. Using this polling interval, the BS should poll the SSs to ensure that the delay requirements of traffic are met. The authors also devised an opportunistic deficit round robin (O-DRR) scheme that schedule the sessions by taking into account the variations in the wireless channel and the delay constraints of multicast traffic. However, the utilization of the O-DRR scheduler introduces an extra overhead on the scheduling, because it is necessary to maintain a quantum size and a deficit count for each SS.

In the uplink scheduling, the BS makes the allocation decisions based on the bandwidth requests from the SS and the associated QoS parameters. Thus, it is important to take into account the polling mechanisms and also the scheduling mechanisms to guarantee the QoS for the applications. However, most of the previous studies [20-23] take into account only the scheduling mechanisms. The scheduling algorithm proposed in this study differs from these previous studies mainly because it interacts with the polling mechanisms aiming at adjusting the interval polling dynamically, and then to guarantee the QoS requirements to the real-time and non-real-time applications.

5. Modeling and simulations

The simulation aims at studying the properties of the proposed scheduling algorithm and analyzing their characteristics in a network that has a variety of burst profiles (various MCSs) and in a network that has only one burst profile (one MCS). The proposed scheduling algorithm has been evaluated using NS-2 [24,25]. Figure 4 shows the simulation scenario.

The simulation scenario consists of a BS and several SSs distributed around the BS in a random mode. As it can be seen from Figure 4, the BS coverage area was divided into R_n regions where the value n represents the

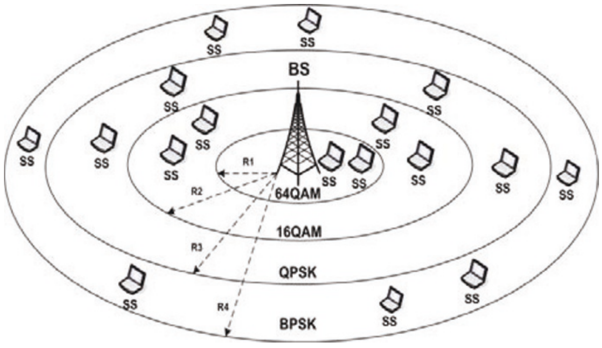

Figure 4 Simulation scenario.

number of regions into the coverage area. The SSs are grouped into R_n regions and each one has the same MCS. The BS executes the mechanism of link adaptation comparing the values of SNR received from the SSs with thresholds, and selecting the MCS that will be used for the sending packets. The calculation methodology used to define the coverage area of the BS and also to define the division of R_n regions is the same used in [26]. To determine the path-loss between BS and SS, the model specified in [27] has been used. This model is proposed for planning WiMAX networks at 3.5 GHz. The calculation methodology used to define the system capacity is the same defined in [25], assuming 40% of the system capacity for downlink and 60% for uplink. Table 1 shows the main parameters used in the simulation.

The sources of traffic used in the simulation were voice, video, Web, and file transfer, which were mapped, respectively, by the service classes: UGS, rtPS, BE, and nrtPS. The voice traffic was modeled by means of an on/off source. During the "on" periods, packets of 66 bytes were generated every 20 ms, following the exponential distribution. The video traffic was modeled by a traffic source that generates, regularly, packets in different sizes, simulating the MPEG traffic. The web traffic was modeled by a hybrid Lognormal/Pareto distribution. The body of the distribution corresponding to an area of 0.88 was modeled as a Lognormal distribution with a mean of 7,247 bytes, and the tail was modeled as a Pareto distribution with a mean of 10,558 bytes [23]. The file transfer traffic was generated using a source with exponential distribution and average packets size of 512 kb. In all the simulations runs, we estimated the 95% confidence interval of each performance measure.

6. Numerical results

6.1 Experiment 1

The first experiment verifies the performance of the proposed algorithm in an environment with several transmissions using one type of modulation and also in an environment with several transmissions using various types of MCSs. Thus, it is possible to analyze the deadlines-based scheme in both environments. The simulated network includes one BS and 30 SSs with one rtPS connection per SS. The MCSs were used in accordance with the distance of the SS from BS. The transmission rate varies from 200 to 800 kbps per rtPS connection, and the number of active SSs varies from 5 to 30. In this experiment, the link was saturated at approximately 65%, and the use of the control admission calls has been considered. This experiment was performed with the EDF scheduling algorithm and with the proposed scheduling algorithm. In this way, it is possible to compare the performance of our deadlines-based scheme with a scheduling algorithm that is also based on deadlines. Traditionally, the EDF selects among queued packets, those with the lowest deadlines. The packets that remain more time in the queue will have higher priority, because their deadlines will expire in the next frame. Since the BS does not have any information about arrival time of packets in the SS input queue, it was considered the arrival time of the BW-REQ messages in the BS queue to calculate the EDF deadlines. Moreover, the proposed algorithm uses an adaptive polling mechanism and the EDF scheduling uses a traditional polling mechanism with fixed polling interval, where the interval polling was set to 40 ms. Figure 5 shows the average delay, where only one MCS was used (64QAM 3/4).

As it can be seen from Figure 5, the proposed algorithm is more efficient than EDF. The difference of the average delay between the proposed algorithm and the EDF is low. This shows that the results for the proposed algorithm are similar to the original EDF. In this case, the deadlines of the proposed algorithm were calculated

Table 1 Main parameters used on the simulation

Parameters	Values
Frequency Operation	3.5 GHz
Frequency band	5 MHz
Sampling factor	144/125
Antenna height (SS)	1.5 m
Antenna height (BS)	60 m
Transmit antenna gain	1
Received antenna gain	1
System loss factor	1
Frame duration	20 ms
Cyclic prefix	0.25
Simulation time	100 s

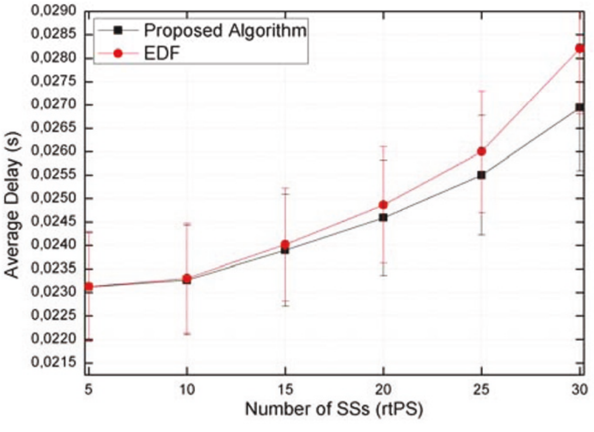

Figure 5 Average delay versus number of SSs with rtPS connections. Only one MCS was used by the SSs for the sending packets.

using only the information about the BW-REQ messages and the queue delay, once it was used only one MCS at the access network. Figure 6 shows the average delay in a scenario where various MCSs were used.

It is possible to see in Figure 6 that the increase of the average delay is more significant when the EDF scheduling was used. This happened because there are several SSs in the access network using different MCSs for the sending packets. This information was used in the deadlines calculation of the proposed scheduling algorithm, what did not happen in the deadlines calculation of the EDF scheduling. The EDF algorithm organizes the uplink subframe in accordance with the lowest deadline, and does not consider the different burst profiles existent in the access network. On the other hand, the proposed algorithm organizes the uplink subframe into bursts with different profiles in an efficient way. Thus, the use of the deadlines-based scheme defined in this study really reduces the average delay. The proposed algorithm is appropriate in both environments, especially when various MCSs are used. This is an open research issue in the access networks that use adaptive modulation.

6.2 Experiment 2

The second experiment aims at analyzing the behavior of the average delay in an environment where various MCSs were used for the sending packets. However, in this case, this analysis was performed by the MCSs used in the coverage area by the BS. The characteristics of the traffic were the same as used in the previous experiment. Figure 7 shows the average delay by modulation area.

As it can be seen from Figure 7, there is a little difference in the average delay among the modulation areas. This means that the proposed algorithm distributes the resources in a fair and efficient way to the modulation areas. The biggest difference happened when the QPSK (1/2) was used. In this case, due to the QPSK modulation, it was used more resources for the sending packets, influencing directly the average delay of this coverage area. However, this did not harm the other coverage area, keeping the average delay within the specified standards [1,8].

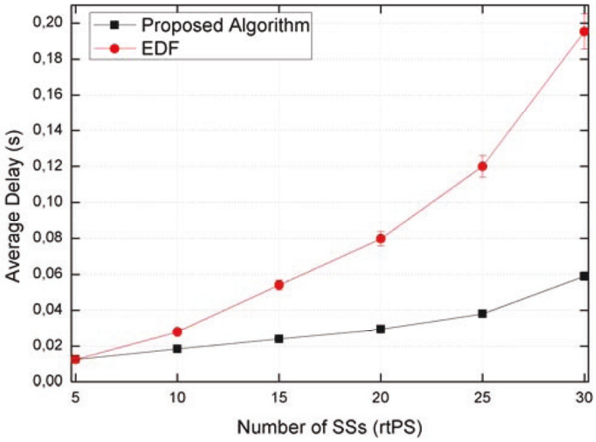

Figure 6 Average delay versus number of SSs with rtPS connections. Three different modulations were used: (64QAM, 16QAM, and QPSK).

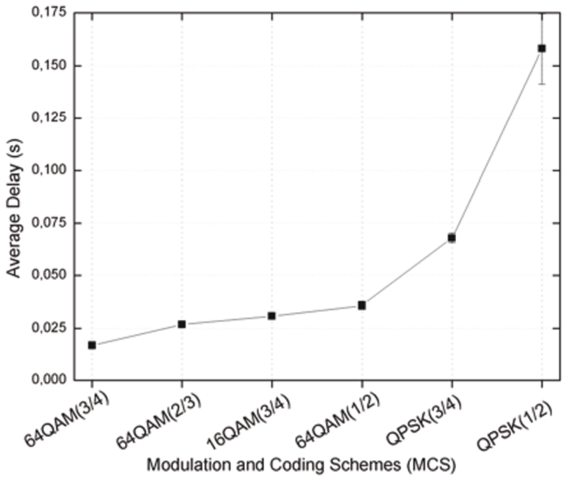

Figure 7 Average delay by modulation area.

6.3 Experiment 3

The aim of Experiment 3 is to investigate the behavior of rtPS and UGS services in accordance with the increase of the rtPS traffic load. For this purpose, the simulated scenario includes one BS, 15 SSs with one UGS connections per SS. In this experiment, each UGS connection generates Constant Bit Rate (CBR) traffic with a rate of 134 kbps. 25 SSs with one rtPS connection per SS that varies from 5 to 25 active SSs. The rtPS transmission rate varies from 120 to 260 kbps per rtPS connection, 10 SSs with one nrtPS connection per SS and 10 SSs with one BE connection per SS. The nrtPS and BE services were used as background traffic. The MCSs used in this experiment were 64QAM(3/4, 2/3), 16QAM(3/4). The MCSs were distributed to SSs by BS through method of link adaptation. It was defined a threshold of 100 ms for the rtPS average delay. Figure 8 shows the throughput of the rtPS and UGS services.

We can see from Figure 8 that the increase of the rtPS load traffic did not interfere in the UGS service. The throughput of the UGS service remained constant as defined by the standard. The throughput of the rtPS service also had a satisfactory result. The difference between the rtPS load traffic and the rtPS throughput was low.

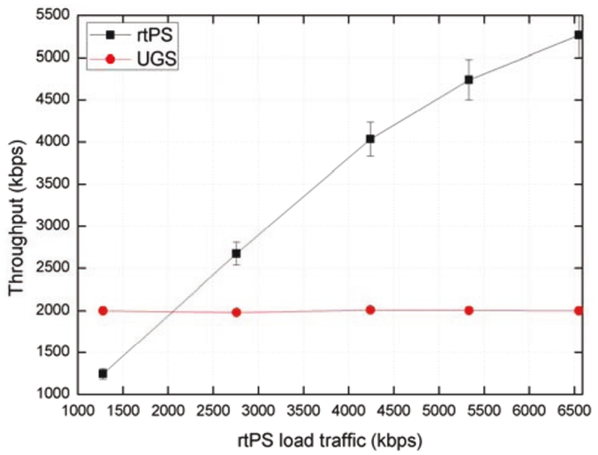

Figure 8 UGS and rtPS throughput versus rtPS load traffic.

Figure 9 shows the average delay of the rtPS and UGS services. However, in this case, the average delay of the rtPS service is also compared with EDF algorithm.

As it can be seen from Figure 9, the average delay of the rtPS service presented an increment when the rtPS load traffic increased. However, with the proposed algorithm, the average delay values remained lower than the threshold. The same did not happen when we used the EDF algorithm. The average delay of the UGS service was not affected by the rtPS load increase. This happened because the scheduler is able to provide data grants at fixed intervals as required by this service.

6.4 Experiment 4

The Experiment 4 verifies the impact of the load increase of the rtPS service on the performance of the nrtPS service. Thus, it is possible to analyze whether the proposed algorithm is able to guarantee the minimal bandwidth to nrtPS service. The simulated network has one BS, 15 SSs with one nrtPS connection per SS and 25 SSs with one rtPS service per SS. The number of active rtPS connections varies from 5 to 25. Each nrtPS connection generates FTP traffic with rate of 300 kbps, and the minimum reserved traffic rate defined to each nrtPS connection is of 30 kbps. The experiment was executed with the proposed algorithm and with the algorithms: WRR and RR. Figure 10 shows the throughput of the nrtPS connections.

As it can be seen from Figure 10, the throughput of the nrtPS connections decreased as the rtPS load increased. This behavior was expected due to a load increase of a service class with higher priority. However, the proposed algorithm shows better performance than WRR and RR algorithms. The proposed algorithm interacts with the polling mechanism and adjusts the unicast polling interval dynamically. Thus, in accordance with the available resources, the SSs receive more grants to request more bandwidth, and the nrtPS service can get more bandwidth. On the other hand, the algorithms WRR and RR do not interact with the polling mechanism, and they use a fixed polling interval, receiving less bandwidth than the proposed algorithm.

6.5 Experiment 5

The Experiment 5 analyzes how the proposed scheduler distributes the resources for the non-real time applications. In this case, it is verified whether the increase of the nrtPS traffic load influences or not on the BE service class. The simulated scenario has one BS and 20 SSs with one BE connection per SS, 30 SSs with one nrtPS connection per SS that varies from 5 to 30 SSs active. It was used in this experiment 5 SSs with one UGS connection per SS, and 5 SSs with one rtPS connection per SS as background traffic. Figure 11 shows the

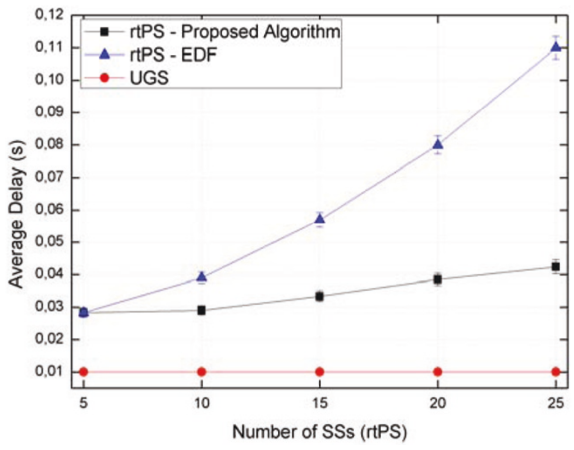

Figure 9 Average delay of UGS and rtPS connections.

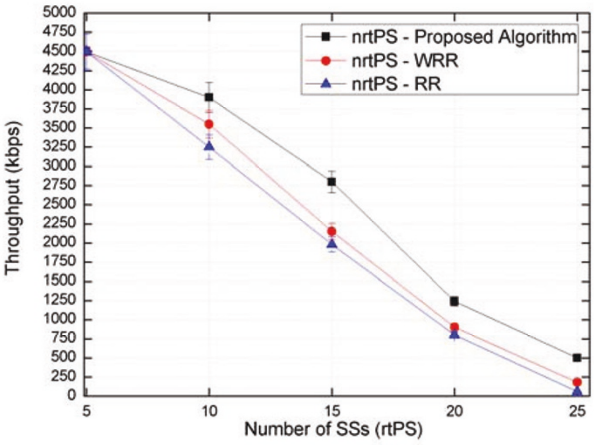

Figure 10 nrtPS throughput versus rtPS load traffic.

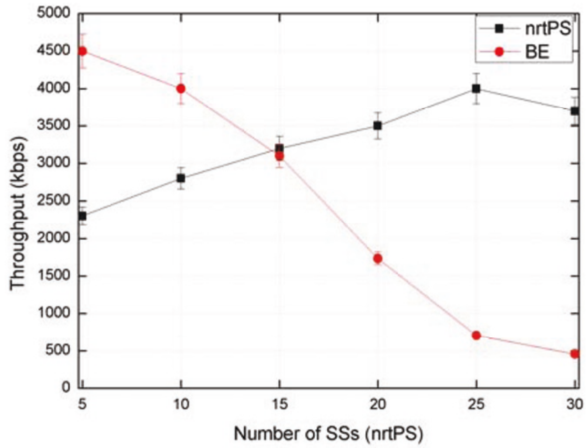

Figure 11 Throughput of nrtPS and BE connections.

throughput of the nrtPS and of the BE services. As we can see, the BE service has higher throughput than nrtPS when the number of active SSs with nrtPS connection was small, since the scheduler allocates the resource (slots) not used by SSs with higher priority (e.g., UGS, rtPS, and nrtPS). When the number of SSs with nrtPS connections increases, the scheduler adjusts the unicast polling interval and distributes the existent resources among nrtPS connections. The throughput of BE decreases, since each BE connection receives fewer resources (slots).

Figure 12 shows the average delay of the rtPS and nrtPS services. As it can be seen from Figure 12, the average delay of the rtPS service was not affected by the load increase of the nrtPS service, which shows that the adaptive polling can help the scheduler to make a balance between the delay constraints of the rtPS and the throughput requirements of the nrtPS.

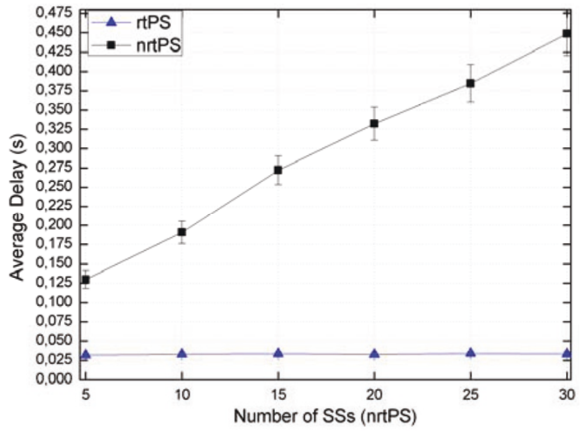

Figure 12 Average delay of rtPS and nrtPS connections.

The average delay of the nrtPS increased as the nrtPS load increased. However, the important issue of the nrtPS service is to guarantee the bandwidth requirements.

7. Conclusion

In this article, an efficient scheduling algorithm and a new adaptive polling scheme for uplink traffic in WiMAX networks were proposed. The algorithm uses a new deadlines-based scheme defined to the real-time applications and uses a cross-layer approach. The deadlines calculation is made using the information about the MCSs in the PHY and the information about the BW-REQ messages sent by the SSs to the BS. Moreover, the algorithm interacts with the polling mechanisms of BS to control the periodicity of sending unicast polling to the rtPS and nrtPS service classes. Thus, the interval polling is adjusted dynamically.

The behavior of the proposed algorithm was analyzed in an environment where various MCSs were used and also in an environment where only one MCS was used. Simulations reveal that the proposed algorithm is efficient in both environments, minimizing the average delay according to the MCSs used in the PHY. This algorithm also interacts with the polling mechanism, adapting the polling interval, and guaranteeing the minimal bandwidth to the real-time and non-real-time applications.

In future study, the proposed scheduling algorithm will be evaluated in mobile environments (including ertPS).

Competing interests

The authors declare that they have no competing interests.

Received: 1 February 2011 Accepted: 27 September 2011
Published: 27 September 2011

References

1. A Bacioccola, C Cicconetti, C Eklund, L Lenzini, Z Li, E Mingozzi, IEEE 802.16: history, status and future trends. Comput Commun. **33**, 113–123 (2010). doi:10.1016/j.comcom.2009.11.003
2. IEEE 802.16j-2009. *IEEE Standard for Local and Metropolitan Area Networks - Part 16: Air Interface for Fixed Broadband Wireless Access Systems* (May 2009)
3. C So-in, R Jain, A Tamimi, Scheduling in IEEE 802.16e mobile WiMAX networks: key issues and a survey. IEEE J Sel Areas Commun. **27**(2), 156–171 (2009)
4. P Dhrona, NA Abu, HS Hassanein, A performance study of scheduling algorithms in point-to-multipoint WiMAX networks. Comput Commun. **32**, 511–521 (2009). doi:10.1016/j.comcom.2008.09.015
5. ST Cheng, MT Hsieh, BF Chen, Fairness-based scheduling algorithm for time division duplex mode IEEE 802.16 broadband wireless access systems. IET Commun. **4**, 1065–1072 (2010). doi:10.1049/iet-com.2009.0083
6. D Chuck, KY Chen, JM Chang, A comprehensive analysis of bandwidth request mechanisms in IEEE802.16 networks. IEEE Trans Veh Technol. **59**(4), 2046–2056 (2010)
7. IEEE standard for local and metropolitan area networks - Part 16: air interface for fixed broadband wireless access systems. IEEE Std., Rev. IEEE Std 802.16-2004 (2004)

8. IEEE 802.16e-2005, IEEE standard for local and metropolitan area networks - Part 16: air interface for mobile broadband wireless access systems. IEEE Std., Rev. IEEE Std 802.16-2005 (2005)
9. K Etemad, Overview of mobile WiMAX technology and evolution. IEEE Commun Mag. **46**(10), 31–40 (2008)
10. IEEE 802.16 Relay Task Group, 11/2010, http://wirelessman.org/relay/
11. J Lakkakorpi, A Sayenko, Uplink VoIP delays in IEEE 802.16e using differents ertPS resumption mechanisms, in *Third International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*, pp. 157–162 (December 2009)
12. MS Kuran, G Gur, T Tugcu, F Alagoz, Appliactions of the cross-layer paradigm for improving the performance of WiMAX. IEEE Wirel Commun. **17**, 86–95 (2010)
13. B Chang, C Chou, Adaptive polling algorithm for reducing polling delay and increasing utilization for high density subscribers in WiMAX wireless networks, in *10th IEEE Singapore International Conference on Communication systems (ICCS)*, pp. 1–5 (February 2007)
14. IC Msadaa, D Camara, F Filali, Scheduling and CAC in IEEE 802.16 fixed BWNs: a comprehensive survey and taxonomy. IEEE Commun Surv Tutor. **12**(4), 459–487 (2010)
15. MA Teixeira, PR Guardieiro, A predictive scheduling algorithm for the uplink traffic in IEEE 802.16 networks, in *the 12th IEEE International Conference on Advanced Communication Technology (ICACT)*. **1**, 651–656 (February 2010)
16. CF Ball, F Treml, X Gaube, EA Klein, Performance analysis of temporary removal scheduling applied to mobile WiMAX scenarios in tight frequency reuse, in *the 16th Annual IEEE International Symposium on Personal Indoor and Mobile Radio Communications, PIMRC'2005, Berlin*. **2**, 888–898 (September 2005)
17. M Settembre, M Puleri, S Garritano, P Testa, R Albanese, M Mancini, V Lo Curto, Performance analysis of an efficient packet-based IEEE 802.16 MAC supporting adaptive modulation and coding, in *International Symposium on Computer Networks*, pp. 11–16 (July 2006)
18. M Gidlund, G Wang, Uplink scheduling algorithms for QoS support in broadband wireless access networks. J Commun. **4**, 133–142 (2009)
19. KH Rath, B Abhijeet, V Sharma, An opportunistic uplink scheduling scheme to achieve bandwidth fairness and delay for multiclass traffic in Wi-Max (IEEE802.16) broadband wireless networks, in *Global Telecommunications Conference*. IEEE Globecom, pp. 1–5 (April 2006)
20. K Wongthavarawat, A Ganz, Packet scheduling for QoS support in IEEE 802.16 broadband wireless access systems. Int J Commun Syst. **16**, 81–96 (2003). doi:10.1002/dac.581
21. J Lakkakorpi, A Sayenko, J Moilanen, Comparison of different scheduling algorithms for WiMAX base station: deficit round-robin vs. proportional fair vs. weighted deficit round-robin, in *Wireless Communications and Networking Conference, 2008. WCNC 2008*. IEEE, pp. 1991–1996 (April 2008)
22. A Lera, A Molinaro, S Pizzi, Channel-aware scheduling for QoS and fairness provisioning in IEEE 802.16/WiMAX broadband wireless access systems. IEEE Netw Mag. **21**, 34–41 (2007)
23. JF Borin, NLS Fonseca, Scheduler for IEEE 802.16 networks. IEEE Commun Lett. **12**, 274–276 (2008)
24. The Network Simulator-ns2, http://nsnam.isi.edu/nsnam/index.php/
25. B Aymen, N Loutfi, WiMAX capacity estimations and simulation results, in *IEEE 67th Vehicular Technology Conference, Singapore, VTC Spring 2008*, pp. 1741–1745 (May 2008)
26. C Tarhini, T Chahed, On capacity of OFDMA-based IEEE802.16 WiMAX including Adaptive Modulation and Coding (AMC) and inter-cell interference. in *LANMAN'2007*, Princeton, NJ (June 2007)
27. GL Stuber, *Principles of Mobile Communication*, 2nd edn. (Kluwer, Norwell, MA, 2001)

doi:10.1186/1687-1499-2011-113
Cite this article as: Teixeira and Guardieiro: A new and efficient adaptive scheduling packets for the uplink traffic in WiMAX networks. *EURASIP Journal on Wireless Communications and Networking* 2011 **2011**:113.