International Journal of the Physical Sciences Vol. 6(16), pp. 4070-4081, 18 August, 2011
Available online at http://www.academicjournals.org/IJPS
DOI: 10.5897/IJPS10.329
ISSN 1992 - 1950 ©2011 Academic Journals

Full Length Research Paper

Providing QoS support through scheduling in WiMAX systems

K. A. Noordin[1]* and G. Markarian[2]

[1]Departmen of Electrical Engineering, Faculty of Engineering, University of Malaya, 50603 Kuala Lumpur, Malaysia.
[2]Department of Communication Systems, InfoLab21, Lancaster University Lancaster LA1 4WA, United Kingdom.

Accepted 21 April, 2011

The WiMAX system outlines support for quality of service (QoS) through several service classes differentiation. However, no specific scheduling has been defined to carry the task. In this paper, we propose a simple and standard-compliant scheduling algorithm for downlink and uplink connections. The proposed algorithm calculates and grants the needed resources in terms of slots based on the QoS requirements and the priority of the service classes. The simulation results show that the scheduling algorithm has fulfilled the QoS provisions of all service classes of WiMAX system in terms of delay and throughput requirements.

Key words: Scheduling, quality of service, WiMAX.

INTRODUCTION

The demand for multimedia traffic with various QoS requirements such as bandwidth and latency has been the main reason why the IEEE802.16 standard (IEEE Std 802.16-2004, 2004) and its derivative, known as WiMAX (WiMAX, 2001), provide support for QoS. However, there is no specific scheduling algorithm implemented by the standard to support various QoS service classes. Therefore, it is up to the vendors or service providers to implement their own scheduling algorithms. There are many scheduling algorithms which have been proposed in the literature and most of them are the modifications or enhancement of the algorithms used in the wired networks. Although the enhanced version of those algorithms such as fair scheduling (Lu et al., 1999) distributed fair scheduling (Vaidya et al., 2005), maxmin fair scheduling (Tassiulas and Sarkar, 2002) and weighted fair queuing (WFQ) (Demers et al., 1989) might work well in the wireless environment, it might not be the right candidate for WiMAX due to its specific features such as the request/grant mechanism, the definition of fixed frame length and the QoS provisions.

In this paper, a novel scheduling algorithm that guarantees the QoS of various service classes of WiMAX is proposed. The algorithm allocates the resources to each service classes in terms of slots. The number of slots needed is calculated based on the minimum and the maximum bandwidth requirements of each connection depending on its service class. The algorithm takes into account the polling interval for uplink scheduling and the packet size for the downlink scheduling in calculating the needed slots. Strict priority scheduling is employed for inter-class scheduling in which service class with higher QoS requirements is scheduled first. Each service class uses different priority scheme to determine the priority of a connection within a class, either using packet waiting time or minimum throughput required. The algorithm provides the following advantages: class-based prioritized scheduling to ensure QoS requirements are met and compliance with the standard since slots are used as the unit of allocation.

QoS in WiMAX

One aspect that distinguishes WiMAX from other broadband wireless access (BWA) systems is that it has been designed from the very beginning to have the capability to support QoS for heterogeneous traffic or service classes. QoS has different meanings to different end users, as much depends on the application and the

*Corresponding author. E-mail: kamarul@um.edu.my.

Table 1. QoS parameters of scheduling types.

Parameter	UGS	ertPS	rtPS	nrtPS	BE
Maximum sustained traffic rate	+	+	+	+	+
Maximum latency	+	+	+	-	-
Tolerated jitter	+	-	-	-	-
Request/transmission policy	+	+	+	+	+
Minimum reserved traffic rate	-	+	+	+	-
Traffic priority	-	-	-	+	+

use to which the end user is putting it. Usually a range of measurable performance parameters is employed from which those appropriate to a particular end user can be selected. The most commonly used parameters are bandwidth, latency and jitter.

Five scheduling types have been defined at MAC layer to support wide variety of applications with different QoS requirements, namely unsolicited grant service (UGS), enhanced real-time polling service (ertPS), real-time polling service (rtPS), non real-time polling service (nrtPS) and best effort (BE). The set of QoS parameters of these scheduling types are listed in Table 1. The table could be used to match or classify the applications or traffic types to a specific scheduling type based on their requirements. For instance, the VoIP application which has tight latency requirement and sensitive to large delay jitter should be assigned to the UGS scheduling type while real-time video streaming which requires a certain amount of bandwidth with tight delay to avoid quality degradation should be assigned to the rtPS type.

The QoS is implemented using the concept of service flow, defined as a unidirectional flow of packets with a particular set of QoS parameters which is identified by a service flow identifier (SFID). Service flows exist in both the uplink and downlink directions and may exist without actually being activated to carry traffic. Only the active service flows can forward packets and will be assigned with the 16-bit connection identifier (CID).

The concept of service flow along with the signaling mechanisms at the MAC level provides the rules for QoS implementation. However, the rules are still incomplete as certain aspects such as scheduling algorithm and admission control have been left out from being defined in the standard. There have been a number of proposals such as in (De Moraes and Maciel, 2006; GuoSong et al., 2002; Alavi et al., 2005; Wang and Markarian, 2004) that suggested either a new or an alternative QoS architecture to fill the gaps that have been left out by the existing architecture. Elements such as traffic classifier, traffic shaper, scheduler and admission control mechanism have been integrated in those architectures so as to provide full support of QoS in the system. However, in order to make the system standard compliant, this paper only considers scheduling algorithm which will be discussed and evaluated thus.

SCHEDULING SCHEMES AND RESOURCE ALLOCATIONS

Scheduling is crucial in ensuring the timely allocation of bandwidth and transmission opportunities to avoid traffic backlog and deadline which can lead to QoS violations. The amount of resources or bandwidth allocated to the scheduled traffic determines how well the QoS provisions are fulfilled so as to achieve fairness among all users or connections in a longer term. Since the scheduler works as a distributor to allocate the resources among the subscriber stations (SSs), therefore it can be said that scheduling and resource allocation should come together.

Classification of Schedulers

The schedulers can be divided into two main categories, the channel-unaware and the channel-aware schedulers as depicted in Figure 1. In the channel-unaware schedulers, the scheduling decisions are not affected by the channel conditions as the channel is normally assumed error-free. The channel-aware schedulers on the other hand do take into consideration the channel conditions in the scheduling decisions so as to exploit multi-user diversity. Furthermore improvements could be achieved by utilizing cross layer optimization techniques (Hui et al., 2007; Song and Li, 2005; WiMAGIC, 2008). In this paper however, the proposed algorithm falls on the first category of the schedulers and therefore the discussions afterwards will only focus on this type.

There are three distinct scheduling processes in the WiMAX network, two at the base station (BS) and one at the SS (Chakchai et al., 2009). The first two are the DL-BS and UL-BS for the downlink and the uplink scheduling at the BS respectively while the third one is located at the SS for uplink after receiving grants from the UL-BS scheduler. The uplink scheduling (UL-BS) imposes more challenges since the BS makes scheduling decisions based on the bandwidth requests received from the SSs without having the actual information on the current queue status at the SSs and could only estimate them based on those requests. On the other hand, the downlink scheduling (DL-BS) is much easier since the BS

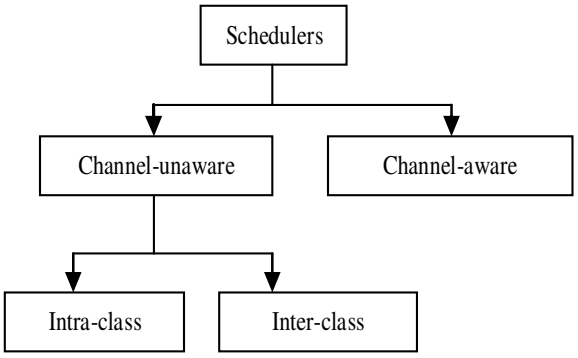

Figure 1. Classification of schedulers.

has the up-to-date information of the queue status of all downlink connections.

To allocate resource within the same QoS class, the intra-class scheduling is used. Variants of the round-robin based algorithm, weighted fair queuing (WFQ) and delay-based algorithm are employed for this purpose. The weighted round-robin based scheduling have been employed for uplink scheduling in Cicconetti et al. (2006), Alexander et al. (2006) and Sayenko et al. (2008) in which the weights are expressed in terms of queue length and packet delay or the number of slots. The deficit round-robin (DRR) has been used for the downlink scheduler and the SS scheduler in Cicconetti et al. (2006) since each head-of-line packet size is known and it is also suitable for variable sized packets.

When the packet size is unknown, the WFQ algorithm can be used. The resources are shared according to the weight of each queue and the weight can be based on bandwidth request (Naian et al., 2005) or the ratio of the connection average rate to the total average data rate (Wongthavarawat and Ganz, 2003a). The algorithm is normally used for non real-time traffic especially nrtPS service where minimum data rate guarantee is required (Sun et al., 2006; Wongthavarawat and Ganz, 2003b). The main disadvantage of this kind of algorithm is its complexity as it is based on the general processor sharing (GPS) scheme which requires the calculation of the virtual finish time for each queue.

When the delay bound is crucial especially for real-time service classes such UGS, ertPS and rtPS, the delay-based algorithm such as Earliest Deadline First has been used (Wongthavarawat and Ganz, 2003a; Jianfeng et al., 2005a). The largest weighted delay first (LWDF) (Stolyar and Ramanan, 2001) chooses the connection with the largest delay based on the HOL delay to avoid missing its deadline especially for real-time traffic which has strict delay requirements. When both real-time and non real-time traffic are present, the delay threshold priority queuing (DTPQ) method (Kim and Kang, 2005) suggested that the real-time traffic will be served only if its HOL delay exceeds a certain threshold. Using this technique the non real-time traffic will be guaranteed of its minimum data rate requirement and not be served only after the real-time traffic has been scheduled as in typical priority based scheduling scheme.

To schedule different service classes within a WiMAX network, priority-based inter-class scheduling mechanism has been applied. The real-time traffic normally will be assigned with higher priority than the non real-time time traffic. For instance, the priority order (from the highest to the lowest) will be UGS, ertPS, rtPS, nrtPS and BE (Jianfeng et al., 2005b; De Moraes and Maciel, 2005; Yan et al., 2008). Apart from that order, Jianfeng et al. (2005a) assigned the downlink connection with higher priority than the uplink connection within the same service class, so that the real-time traffic will always have higher priority than the non real-time traffic.

There might be a probability that the priority scheme will starve some connections of the lower service classes. The throughput of these connections could become lower if they miss their deadlines since the higher service classes will always be prioritized for transmission. To alleviate the problem, deficit fair priority queuing (DFPQ) with a counter was introduced to maintain the maximum allowable bandwidth for each service class (Jianfeng et al., 2005a,b). The counter decreases according to the size of the packets and when the counter falls to zero, the scheduler moves to another class.

Some scheduling issues

The various scheduling algorithms and mechanisms proposed and highlighted earlier have brought some issues in terms of implementation complexity. Each time when an SS joins or leaves the network, the scheduler needs to calculate configuration. Furthermore, as SSs send data, their request sizes change all the time. Hence, the scheduler at the BS should reassign slots for every 802.16 frame to achieve an accurate and fair resource allocation. It is important to note that since the scheduling interval is normally quite short, the scheduler does not have much time to make scheduling decisions. If we use the 5 ms frame duration for instance, the scheduler then needs to perform 200 scheduling decisions per second. The hierarchy of schedulers introduced in (Wongthavarawat and Ganz, 2003a) would be an example on how much computational overhead have been put on the schedulers since the different algorithms used in the schedulers themselves are already complex.

Another issue arises from the scheduling algorithms is whether they comply with the standard. For instance, the authors of (Shejwal and Parhar, 2007) have proposed a scheduling scheme based on a concept of service criticality (SC) in which a flow would receive the service through bandwidth allocation depending on degree of service criticality. However, the scheme suggested that the bandwidth request message sent to the BS consists of a tuple formed by the service critically index and the

required bandwidth which does not conform to the standard as it introduces new message format.

In inter-class scheduling on the other hand, the main issue is whether each service class should be considered separately, meaning having its own queue, or combined to reduce complexity. Sun et al. (2006) for example have divided the traffic queues into two; type one for UGS queue and unicast request grant for rtPS and nrtPS, and type two for rtPS, nrtPS and BE queues. Freitag and Da Fonseca, (2007) and Borin and Da Fonseca, (2008), the queues are divided into three categories, the low priority queue (BE requests), intermediate queue (rtPS and nrtPS requests) and high priority queue (grants for UGS and unicast request for rtPS and nrtPS). The intermediate queue will be moved to the highest priority queue if it approaching its deadline in the next frame interval.

PROPOSED SCHEDULING SCHEMES

Having discussed the scheduling issues above, it is therefore desirable to have a scheduling mechanism which is simple to implement and comply with the standard for practical reason. Therefore, we propose a simple algorithm that could be implemented for uplink and downlink scheduling at the BS as well as for the uplink scheduling at the SS. The algorithm is based on a strict priority scheduling in which the highest service class will be served first and improves our earlier work in (Noordin and Markarian, 2007). However, to prevent the lower service classes from being starved, the resource allocated to each connection will be granted based on some weights formed by the minimum and maximum bandwidth requirements. We propose a channel-unaware scheduler which does not take into account the channel conditions in making scheduling decision although we consider the modulation and coding scheme used in granting the bandwidth. Since WiMAX utilizes adaptive modulation and coding (AMC) in the PHY layer, the selected modulation and coding do reflect channel state information, therefore, the proposed algorithm also could be considered as the in-direct channel aware algorithm.

Priority order

The proposed scheduler will schedule the service classes according to this priority order: UGS, ertPS, rtPS, nrtPS and BE. The idea of having this strict priority order is because the higher service class should be satisfied in terms of its requirements first before other service classes have their share of the resources. However, this strict priority order is different from the hierarchical structure in (Jianfeng et al., 2005b) where the downlink and uplink connections of a higher service class will have higher priority than the downlink and uplink connections of a lower service class. Here, since the downlink and uplink scheduling are two distinct processes and they are scheduled in separate subframe, the priority between downlink and uplink connections do not arise.

The connections within the same service class of ertPS and rtPS will be served according to their waiting time. The connection whose queue is approaching its deadline or its waiting time is approaching its maximum latency value will have higher priority than other connections. This will ensure that the connections do not exceed their latency requirements. For the uplink connections, since the BS has no knowledge when the packet at the SS arrived at its queue, the worst case is assumed in which the time corresponds to the arrival at the queue immediately after the connection sent the last bandwidth request to the BS (Freitag and da Fonseca, 2007). Therefore the waiting time is equal to the current time of the scheduling round minus the last request time. For nrtPS, since they can tolerate with delays, their priority will be based on how well their minimum bandwidth requirements over a predetermined time window have been fulfilled. The connection whose minimum bandwidth has not yet been satisfied will have a higher priority than other connections with satisfied minimum bandwidth requirement. The BE connections on the other hand will be served in a round robin manner since they have no specific requirements in terms of delay or minimum bandwidth.

Slots calculation

Having determined the priority of the connections, it is then required to allocate bandwidth to each connection in terms of number of slots. The bandwidth are allocated in such a way that all service classes will be served their minimum bandwidth requirements first after which the remaining bandwidth left will be distributed accordingly to achieve work-conserving behavior. The basic formula of the allocated bandwidth will be based on the work of (Alexander et al., 2006) which given as:

$$N_i = \frac{B_i}{S_i n_{fps}} \quad (1)$$

where N_i is the number of slots allocated to connection i, B_i is the required bandwidth of connection i, S_i is the slot size depending on MCS used and n_{fps} is the number of frames in one second. Therefore the idea is to allocate and to limit the number of slots given to each connection in each frame so as to satisfy the QoS requirements and not to starve lower priority connections. For UGS connections however, since the data is generated at a fixed interval and not necessarily at every frame, the slots are allocated every n_{UGS} frames according to the interval that has been negotiated during connection setup. Due to this reason we modified the basic formula in Equation 1 so that the number of slots allocated to each UGS connection i for every n_{UGS} frames can be formulated as:

$$N_i^{UGS} = \left\lceil \frac{B_i^{UGS}}{S_i n_{eff}} \right\rceil \quad (2)$$

Where B_i^{UGS} is the maximum sustained traffic rate and $n_{eff} = n_{fps}/n_{UGS}$ is the effective number of frames in a second during which the packet arrived at the queue, and the above equation can be used for both uplink and downlink directions of the ith UGS connection since this type of connection is unlikely to change its data rate. The number of slots is rounded up to the next integer value since the calculation might gives a floating point number whereas the slot should be regarded as having an integer value.

The typical application for ertPS connection is VoIP with silence suppression in which during the active phase, constant rate data is generated at a specific interval as in the UGS case, that is data is generated every n_{ertPS} frames. However, there exist a silent phase during which no data is generated. Therefore, the allocated slots would be calculated based on the request size for the uplink and the queue size for the downlink during the active phase. For the downlink, the number of slots is given by:

$$N_i^{ertPS} = \min\left(\left\lceil \frac{B_i^{\max(ertPS)}}{S_i n_{eff}} \right\rceil, \left\lceil \frac{Q_i^{ertPS}}{S_i} \right\rceil \right) \quad (3)$$

where Q_i^{ertPS} is the queue size, $B_i^{\max(ertPS)}$ is the maximum sustained traffic rate of the *i*th ertPS connection respectively and $n_{eff} = n_{fps}/n_{ertPS}$. For the uplink, since the ertPS will be polled at every t_{ertps} interval (in unit of frames) during which it requests for bandwidth, the number of slots in a frame can be formulated as:

$$N_i^{ertPS} = \begin{cases} 1, & R_i^{ertPS} = 0 \\ \min\left(\left\lceil \frac{B_i^{\max\ (ertPS\ )}}{S_i n_{eff}} \right\rceil, \left\lceil \frac{R_i^{ertPS}}{S_i} \right\rceil\right), & R_i^{ertPS} > 0 \end{cases} \quad (4)$$

where R_i^{ertPS} is the request size of the *i*th ertPS connection and $n_{eff} = n_{fps}/t_{ertPS}$. During the silent phase, one slot is allocated for the uplink connection so that it can request for bandwidth as soon the active phase starts. The rtPS connection type which generates variable bit rate traffic will be served at a regular interval. Since there are minimum reserved traffic rate and maximum sustained traffic rate for this kind of connection, the number of slots assigned should be related to these parameters. Therefore for downlink connection, the minimum and maximum number of slots are given as:

$$N_i^{rtPS\ (\min\ )} = \left\lceil \frac{B_i^{\min(\ rtPS\ )}}{S_i n_{fps}} \right\rceil \quad (5)$$

$$N_i^{rtPS(\max)} = \min\left(\left\lceil \frac{B_i^{\max(\ rtPS\ )}}{S_i n_{eff}} \right\rceil, \left\lceil \frac{Q_i^{rtPS}}{S_i} \right\rceil\right) \quad (6)$$

where $B_i^{\min(rtPS)}$ and $B_i^{\max(rtPS)}$ are the minimum and the maximum bandwidth requirements respectively of the rtPS connection. Here we use n_{eff} to find the maximum number of slots allocated in a frame and it is defined as:

$$n_{eff} = \begin{cases} \dfrac{B_i^{\max(\ rtPS\ )}}{p_{rtPS}}, & n_{fps} > \dfrac{B_i^{\max(\ rtPS\ )}}{p_{rtPS}} \\ \dfrac{\left(n_{fps}\right)^2}{B_i^{\max(\ rtPS\ )} \Big/ p_{rtPS}}, & n_{fps} < \dfrac{B_i^{\max(\ rtPS\ )}}{p_{rtPS}} \end{cases} \quad (7)$$

where p_{rtPS} is the packet size of the rtPS connection. By using the packet size in the above equation the maximum number of slots allocated will be in multiple of complete packet size so that a whole packet can be transmitted in one burst of a frame instead of being fragmented into smaller size and transmitted in different burst in multiple frames (if the granted slots in a frame are much smaller than the size of a packet). For the uplink rtPS connection, since it is polled every t_{rtPS} interval (in unit of frames), the minimum and the maximum number of slots in a frame can be described as:

$$N_i^{rtPS(\min)} = \begin{cases} 1, & R_i^{rtPS} = 0 \\ \left\lceil \dfrac{B_i^{\min(rtpPS)}}{S_i n_{fps}} \right\rceil, & R_i^{rtPS} > 0 \end{cases} \quad (8)$$

$$N_i^{rtPS(\max)} = \min\left(\left\lceil \frac{B_i^{\max(rtPS)}}{S_i n_{eff}} \right\rceil, \left\lceil \frac{R_i^{rtPS}}{S_i} \right\rceil\right) \quad (9)$$

where $n_{eff} = n_{fps}/t_{rtPS}$. One slot is allocated to the rtPS connection if the request size is zero so that it can use it to request for more bandwidth when needed since rtPS is prohibited from using contention for bandwidth request and only polled once every t_{rtPS} interval. The nrtPS connections normally serve data intensive application like FTP which requires minimum bandwidth guarantee. The slots allocated to the downlink connections of nrtPS follow that of the rtPS, hence:

$$N_i^{\min(nrtPS)} = \left\lceil \frac{B_i^{\min(nrtPS)}}{S_i n_{fps}} \right\rceil \quad (10)$$

$$N_i^{\max(nrtPS)} = \min\left(\left\lceil \frac{B_i^{\max(nrtPS)}}{S_i n_{eff}} \right\rceil, \left\lceil \frac{Q_i^{nrtPS}}{S_i} \right\rceil\right) \quad (11)$$

where as in rtPS case, n_{eff} is defined as:

$$n_{eff} = \begin{cases} \dfrac{B_i^{\max(nrtPS)}}{p_{nrtPS}}, & n_{fps} > \dfrac{B_i^{\max(nrtPS)}}{p_{nrtPS}} \\ \dfrac{\left(n_{fps}\right)^2}{B_i^{\max(nrtPS)} \Big/ p_{nrtPS}}, & n_{fps} < \dfrac{B_i^{\max(nrtPS)}}{p_{nrtPS}} \end{cases} \quad (12)$$

For the uplink nrtPS connections, they will be polled less regularly than the rtPS connections but they can participate in contention to request for more bandwidth. Therefore, we formulate the number of slots allocated in each frame as follows:

$$N_i^{nrtPS(\min)} = \begin{cases} 0, & R_i^{nrtPS} = 0 \\ \left\lceil \dfrac{B_i^{\min(nrtpPS)}}{S_i n_{fps}} \right\rceil, & R_i^{nrtPS} > 0 \end{cases} \quad (13)$$

$$N_i^{\max(nrtPS)} = \min\left(\left\lceil \frac{B_i^{\max(nrtPS)}}{S_i n_{eff}} \right\rceil, \left\lceil \frac{R_i^{nrtPS}}{S_i} \right\rceil\right) \quad (14)$$

where $n_{eff} = n_{fps}/t_{nrtPS}$ and t_{nrtPS} is the polling interval. There is no need to reserve one slot to the uplink connection when the request size is zero since it can request for more bandwidth using the contention method. For BE connections, there will be no minimum number of slots allocated since there is no minimum bandwidth requirement for this kind of traffic. However, to achieve some fairness among all BE connections, the allocated slots would

depend on the ratio of the rate of a connection to the total rate of all BE connections. Therefore the maximum number of slots allocated in each frame for the downlink is given by:

$$N_i^{\max(BE)} = \min\left(\left\lceil \frac{B_i^{\max(BE)} S_{avail}}{\sum_{j=1}^{J} B_j^{BE}} \right\rceil, \left\lceil \frac{Q_i^{BE}}{S_i} \right\rceil\right) \quad (15)$$

where S_{avail} is the available remaining slots and J is the total number of BE connections. The uplink connection can request more bandwidth using contention as in nrtPS case apart from using the unicast bandwidth request through polling, therefore the number of slots allocated would be:

$$N_i^{\max(BE)} = \min\left(\left\lceil \frac{B_i^{\max(BE)} S_{avail}}{\sum_{j=1}^{J} B_j^{BE}} \right\rceil, \left\lceil \frac{R_i^{BE}}{S_i} \right\rceil\right) \quad (16)$$

The allocation of the slots according to the above formula will try to satisfy the minimum requirements of all connections first after which the remaining available slots in a frame will be allocated firstly to the rtPS connections, followed by nrtPS and BE connections. The priority of the allocation in each of these service classes follows the same method as has been described earlier for these service classes. However, the total slots granted to each connection after allocating the additional slots should not exceed the maximum number of slots described in the above formula. The UGS and ertPS connections are unlikely to change its data rates and therefore will not be granted additional resources from the remaining slots.

The above algorithm can be summarised as follows:

1. Sort the priority is each service class using *Sort_Priority* except for UGS and BE;
2. Allocate minimum slots to the connections using *Grant_minimum_Slots* in this order: UGS, ertPS, rtPS, nrtPS;
3. If there are free slots available after step 2, allocate them to each rtPS, nrtPS and BE connections (in this order) using this formula:

$$N_i^{add(C)} = N_i^{\max(C)} - N_i^{\min(C)};$$

$$C \in \{rtPS, nrtPS, BE\}, N_i^{\min(BE)} = 0.$$

such that the maximum number of slots in each frame are not exceeded.

Sort_Priority:

4. For each ertPS connection i
5. Waiting_time[i] = Current_time – Queue_time[i];
6. Sort Waiting_time[ ] in decreasing order
7. Repeat steps 4-6 for rtPS connections.
8. For each nrtPS connection i
9. Granted_bandwidth[i] = Granted_bandwidth[i] + Allocated_slots[i];
10. if Minimum_bandwidth[i] > Granted_bandwidth[i]
11. Priority[i] = Minimum_bandwidth[i] – Granted_bandwidth[i];
12. else
13. Priority[i] = 0;

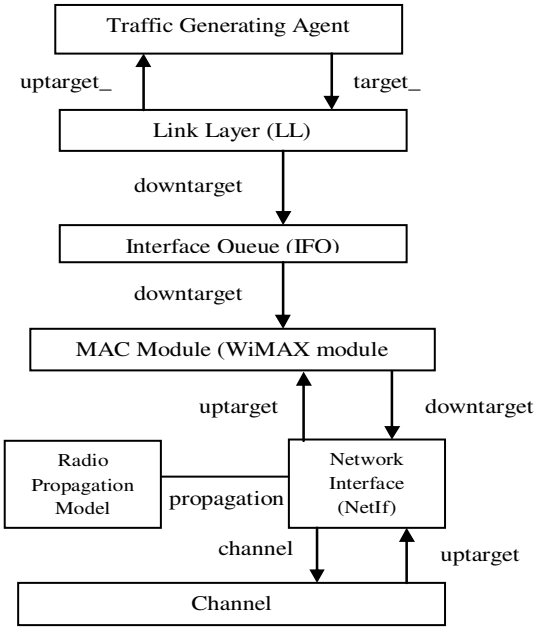

Figure 2. NS2 architecture and WiMAX module.

Grant_minimum_Slots:

14. For each UGS connection in sorted queue
15. Use (2) for downlink and uplink
16. For each ertPS connection in sorted queue
17. Use (3) for downlink and (4) for uplink
18. For each rtPS connection in sorted queue
19. Use (5) for downlink and (8) for uplink
20. For each nrtPS connection in sorted queue
21. Use (10) for downlink and (13) for uplink
22. For each BE connection in sorted queue
23. Use (15) for downlink and (16) for uplink.

SIMULATIONS

To evaluate the effectiveness of the scheduling algorithm, a software package was developed using NS2 simulator and few simulation runs were conducted. Figure 2 depicts the architecture of NS2 in which the module for WiMAX has been added. A simple network topology which consist of one base station and a few subscriber stations is considered. The physical layer parameters described in Table 2 are used with the downlink to uplink TDD split ratio is set 1:1. In order to evaluate the scheduler at the BS, each subscriber station is allowed to have one type of traffic only. The traffic models with the bandwidth requirements shown in Table 3 are used. Table 4 on the other hand describes the slot size in bytes for each modulation and coding schemes of OFDMA PHY.

The UGS service will received data grants for every 20 ms (5 frames) since the BS must allocate data grants to the UGS connections at intervals equal to the UGS application packet generation rate (802.16e-2005 and 802.16-2004/Cor 1-2005, 2006). The interval between unicast request opportunities of the rtPS service is 20 ms whereas the intervals of the nrtPS and the BE services are 0.5 s. For rtPS service, the delay requirement is 100 ms and for nrtPS the delay is 1s. The BE service is chosen to have a constant traffic at a rate of 512 Kbps so that its impact on other higher priority service classes can be observed. Since the admission control mechanism is not a subject of interest here, only

Table 2. Physical layer parameters.

Parameter	Value
Frame length	5 ms
Symbol duration	100.84 µs
No. of OFDM symbols	49
No. of uplink subchannels	35
No. of downlink subchannels	30

Table 3. The traffic settings for the simulation.

Traffic type	Packet size (bytes)	Application	Minimum bandwidth (bits/s)	Maximum bandwidth (bits/s)
UGS	200	CBR/UDP	64000	64000
ertPS	200	CBR/UDP	0	64000
rtPS	300	VBR/UDP	256000	512000
nrtPS	1040	FTP/TCP	256000	512000
BE	200	CBR/UDP	-	-

Table 4. The slot size for OFDMA PHY.

Modulation	Channel coding	Slot size (bytes)
64-QAM	3/4	27
64-QAM	2/3	24
16-QAM	3/4	18
16-QAM	1/2	12
QPSK	3/4	9
QPSK	1/2	6

a simple mechanism is employed in which a connection is admitted only if there is sufficient bandwidth available after all existing connections have been served. Ideal channel conditions are assumed in the simulations due to the channel-unaware approach taken in the scheduling algorithm proposed.

RESULTS AND DISCUSSION

We conduct several simulation scenarios to evaluate the proposed scheduling algorithm. The first three scenarios evaluate the effectiveness of the scheduler in allocating resources without compromising on the QoS provisions in terms of delay and minimum bandwidth requirements to the connections with various service classes. For these scenarios, we assume all connections will be using 64-QAM 3/4 modulation scheme. To check whether delay intolerant service classes like UGS, ertPS and rtPS connections are satisfied in terms of their delay requirements, the delay plots will be used. For delay tolerant services like nrtPS and BE classes, their throughput will be plotted to check whether they have received their minimum bandwidth guarantee and not being deprived of the bandwidth, respectively. The last scenario was conducted to evaluate the scheduler when the connections use different modulation schemes. Connection using more robust modulation scheme such as 16-QAM will have to be allocated more slots than connection with less robust modulation scheme such as 64-QAM for the same amount of request size since the former slot size is less than the latter. Therefore the scheduler must ensure enough slots are allocated to the available connections with different priorities in order to satisfy their QoS requirements as much as possible.

Scenario 1

In the first scenario, the scheduler is evaluated on its effectiveness on ensuring QoS requirements of various service classes especially the impact on the QoS of lower priority service classes when the highest priority service class that is (UGS) load is increased. For this purpose,

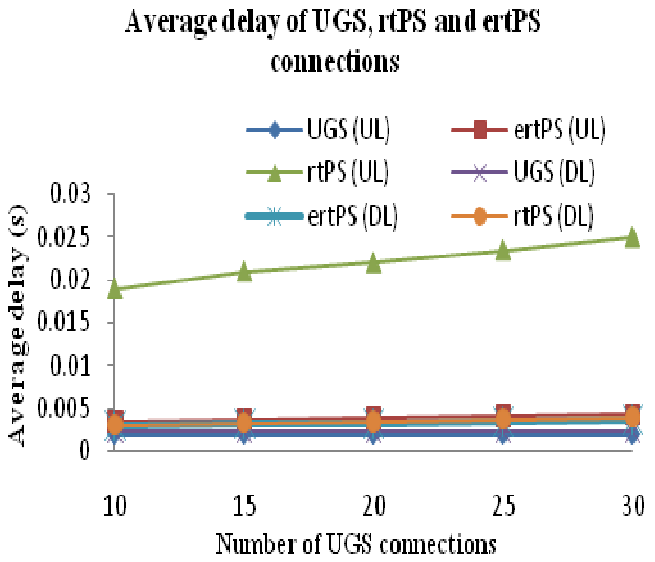

Figure 3. Average delay of delay-sensitive service classes on increasing UGS load.

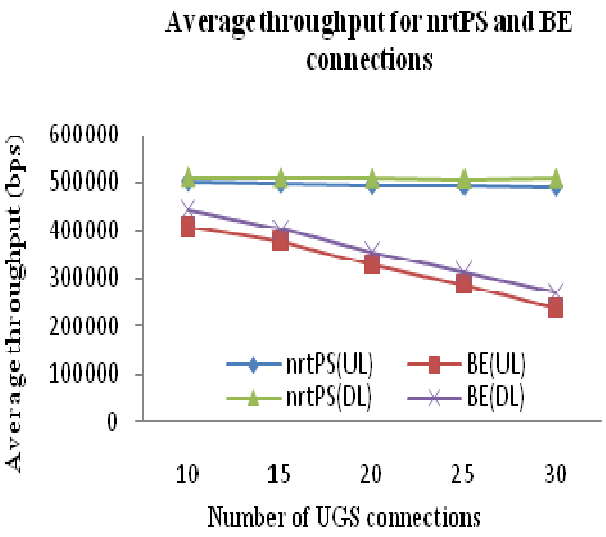

Figure 4. Average throughput for delay-tolerant service classes under increasing UGS load.

the simulated scenario includes one BS and 65 SSs in which there are 10 ertPS connections, 5 rtPS connections, 10 nrtPS connections, 10 BE connections, and the number of active UGS connections are varied from 15 to 30. The above connections are simulated for both uplink and downlink directions. From Figure 3 we can see that the average delay for UGS in both downlink and uplink cases is constant which means that it is not affected by the load increase. This should be the case as the UGS is guaranteed to be provided with data grants at fixed interval by the scheduler. The delay for ertPS cases also shows quite a constant value although it is a little bit higher than the UGS case since the former has less stringent delay requirements than the latter. The delay of uplink rtPS shows some slight increment with the UGS load increase but it is still lower than the required one. The downlink rtPS however, shows quite a constant and lower value compared to the uplink case. This is because for the downlink, there is no bandwidth request involved and the scheduler provides the required bandwidth to the connection whenever there are packets in the queue. In the uplink, the connection only receives periodic data grants at a certain interval, hence the higher average delay but still does not exceed its requirement.

The throughput for the nrtPS and BE connections for both downlink and uplink is shown in Figure 4 where it can be seen that the throughput of the nrtPS connections hovers around 500 Kbps and does not seem affected much by the increase of the UGS load. The nrtPS connections are simulated by the FTP application over the TCP protocol which tries to send as much data as possible and since there are enough resources available, the throughput near to the maximum sustained rate value as seen in the graph can be achieved. The throughput of the uplink nrtPS connections is slightly lower than that of the downlink since in the uplink the connections have to participate in contention to request for bandwidth and collisions may occur.

The throughput of the BE connections on the other hand shows a notable decrement as the UGS load increases for both downlink and uplink cases. This is because the BE service has no minimum bandwidth requirement and is given the remaining resources after all other higher priority classes have been served. As in nrtPS case, the uplink BE connections have lower throughput than the downlink since collisions might have occurred during contention bandwidth request.

Scenario 2

In the second scenario, the impact of the load increase of the rtPS service on the performance of other service classes is evaluated. The rtPS service is normally assigned to real-time video traffic which has a variable bit rate and can be quite bursty at times. Therefore the scheduler must ensure that the rtPS service is ensured its QoS requirements without compromising the QoS level of the other service classes, especially the classes with stringent delay requirements like UGS and ertPS. So for this scenario, there will be one BS with 53 SSs in which there are 15 UGS connections, 10 ertPS connections, 10 nrtPS connections, 10 BE connections, and the number of rtPS connections is varied from 1 to 8. As in previous scenario, the connections exist for both downlink and uplink directions. It can be seen from Figure 5 that the delay of UGS connections almost constant for both downlink and uplink as the rtPS load increases. The ertPS connections also have quite a constant delay and

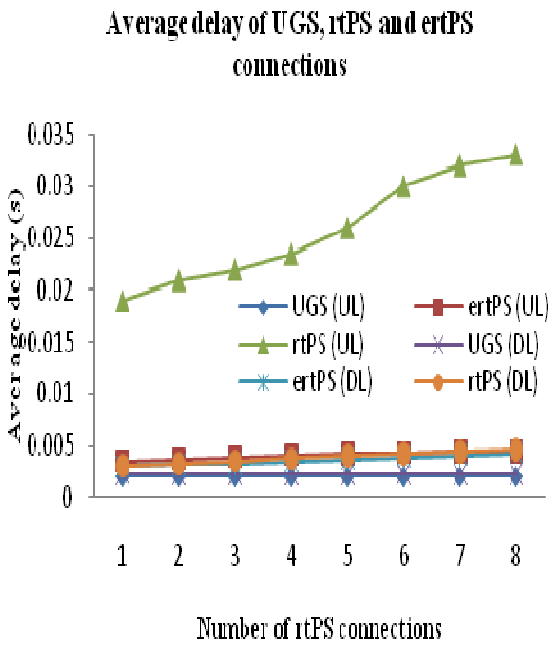

Figure 5. Average delay of delay-sensitive service classes on rtPS load increase.

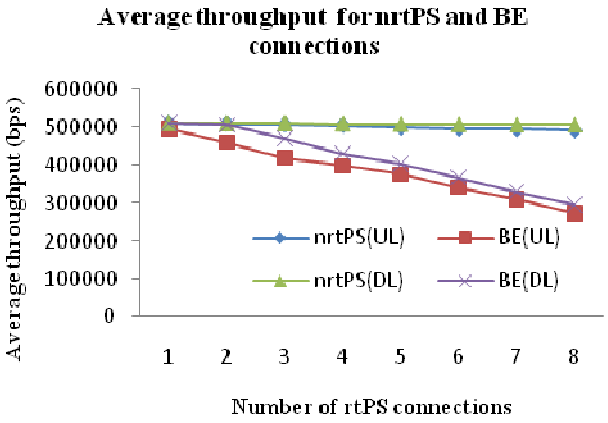

Figure 6. Average throughput of delay-tolerant service classes on rtPS load increase.

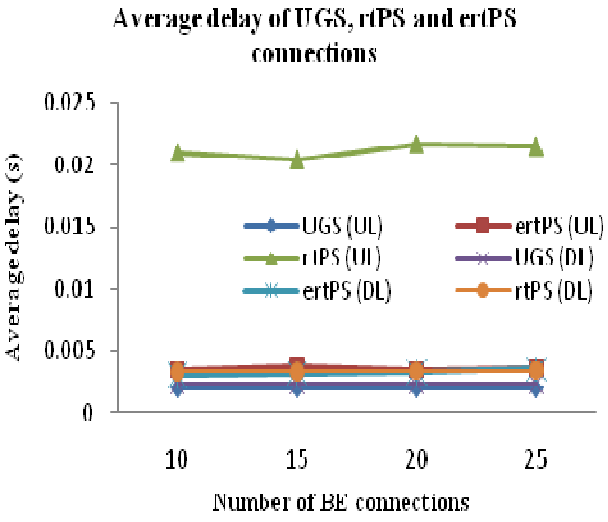

Figure 7. Average delay of delay-sensitive service classes on BE load increase.

are not affected much by the increase of rtPS load. The uplink rtPS connections however show an increase delay when the number of rtPS connections are increased. This is because as the load increases and the rtPS connections request for bandwidth, there is a possibility that the resources for that particular scheduling round are not sufficient to serve all connections and therefore some rtPS connections have to wait for the next round of bandwidth request, hence, the increased delay.

Since the scheduler schedules the rtPS connections with longer waiting time after receiving bandwidth requests from those connections, the rtPS connection which is not scheduled in the previous round will be likely scheduled in the next round and the delay therefore would not exceed the required ones. The downlink rtPS connections have a smaller delay than the uplink since they do not have to wait to be polled to be served. They will be served as soon as there are packets in the queue and the resources are available.

The throughput of the nrtPS connections in Figure 6 hovers around the 500 Kbps value as in previous scenario due to the nature of the FTP over TCP protocol which tries to send data as much as possible. The BE throughput however shows a declining trend as the rtPS load increases since the latter have higher priority than the former and more resources are allocated to them. Therefore the remaining bandwidth left for the BE connections reduces as the rtPS load increases. The throughput of the uplink BE connections is also slightly lower than the downlink due to the same reason as in the previous scenario.

Scenario 3

The effect of increasing the number of connections of lower priority service class i.e. BE towards the QoS level of higher priority service classes is investigated in this scenario. Therefore for this purpose, the simulation includes one BS and 65 SSs which consists of 15 UGS connections, 10 ertPS connections, 5 rtPS connections, 10 nrtPS connections, and a number of BE connections which varies from 10 to 25 . The connections are simulated for both downlink and uplink directions as in previous scenarios.

As shown in Figure 7, the delay of UGS is quite constant even the number of BE connections increases. The delay of ertPS follows the same trend and is not affected much by the increase of BE loads. The same

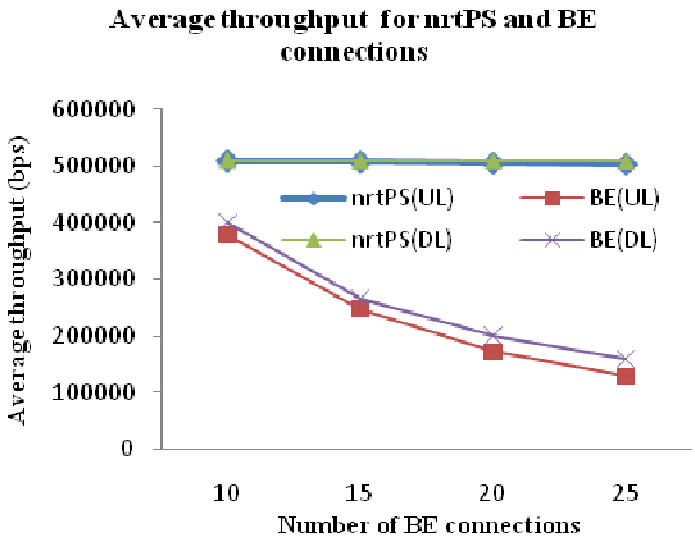

Figure 8. Average throughput of delay-tolerant service classes on rtPS load increase.

Table 5. The bandwidth requirements settings for Scenario 4.

Traffic type	Packet size (bytes)	Application	Minimum bandwidth (Kbits/s)	Maximum bandwidth (Kbits/s)
UGS	200	CBR/UDP	64	64
rtPS	300	VBR/UDP	512	1024
nrtPS	1040	FTP/TCP	768	1024
BE	200	CBR/UDP	-	-

goes to the rtPS which also shows a slight fluctuation on its delay. Since all these connections have higher priority than the BE service, the scheduler will ensure that these connections are served first before the BE connections get their share of the remaining bandwidth. Therefore, the increase in number of BE connections does not affect their strict delay requirements.

The average throughput of nrtPS connections for both downlink and uplink directions in Figure 8 do not vary much from the previous scenarios when the BE load increases as the resources are sufficient for such connections to send data as much as possible. However the throughput of the BE connections declined as the number of BE connections increases since the resources left after all other higher service classes have been serviced had to be shared among them. The uplink BE connections also shows a lower throughput than the downlink since they have to contend for bandwidth and collisions might have occurred. Therefore it can be said that the scheduler has fulfilled the QoS requirements of all service classes in by serving them in priority order so that the higher priority service classes are not affected by the increment of the number of lower service classes.

Scenario 4

In all previous scenarios, the 64-QAM 3/4 modulation and coding scheme was employed in the simulations. Since different modulation and coding schemes carries different amount of bits per symbol which translates into the number of slots required, the scheduler must ensures that enough slots are provided for each type of connections. In this scenario, the scheduler is evaluated whether it can satisfy QoS requirements of connections of different service classes when they employ different modulation coding schemes. However, only downlink connections are simulated in this scenario since the effect would be the same for the uplink. The simulation includes one BS and 8 SSs in which there are 2 connections each for UGS, rtPS, nrtPS and BE connection respectively. The bandwidth requirements for each service class in this simulation have been changed and are described in Table 5. The BE connections is set to generate a constant bit rate traffic at the rate of 2048000 bps. The modulation and coding scheme employed for each SS is described by the legend in Figure 8.

From Figure 9, it can be seen that both UGS connections are having constant throughput throughout the simulation duration. Having a more robust modulation scheme (such as QPSK 3/4) requires more slots to achieve the same rate as in less robust modulation scheme (such as 64 QAM 3/4). This is also true for rtPS connection where the minimum and maximum bandwidth requirements are guaranteed by the scheduler in which the scheduler has allocated sufficient slots to each of them. The nrtPS connections have also being allocated sufficient slots by the scheduler in which their throughput is approximately identical to their maximum sustained rate. The BE connections shares the remaining available slots and from the graph it is obvious that BE_2 have higher throughput than BE_1 since the former was using the less robust modulation and coding scheme than the latter. From the graph it can also be seen that the throughput of BE connections is influenced by rtPS connections in which the throughput of the former increases when the throughput of the latter decreases. This can be explained by the fact that the scheduler has allocated remaining slots not used by higher service classes to the lower service class.

CONCLUSION

In this paper, a scheduling algorithm for both downlink and uplink connections in a WiMAX network has been presented. The proposed algorithm makes scheduling decision based on the priority of the service classes involved and allocates resources in terms of needed slots. The calculation of the slots depends on the bandwidth requirements of each connection and the algorithm ensures that the granted resources do not exceed the maximum requirement of the each connection to prevent lower service classes from being starved. The proposed algorithm also complies with the standard as it does not introduce any new signaling mechanism. The results obtained show that the algorithm has fulfilled the QoS provisions of all service classes of WiMAX network in terms of delay and throughput requirements.

REFERENCES

Alavi HS, Mojdeh M, Yazdani N (2005). A Quality of Service Architecture for IEEE 802.16 Standards. In the Proceedings of Asia-Pacific Conf. Comm., 249-253.

Alexander S, Olli A, Juha K, Timo H (2006). Ensuring the QoS requirements in 802.16 scheduling. In the Proceedings of the 9th ACM international symposium on Modeling analysis and simulation of wireless and mobile syst., 108-117.

Borin JF, da Fonseca NLS (2008). Scheduler for IEEE 802.16 networks. IEEE Comm. Lett.,12(4): 274-276.

Chakchai SI, Jain R, Tamimi AK (2009). Scheduling in IEEE 802.16e mobile WiMAX networks: key issues and a survey. IEEE J. Selected Areas in Comm., 27(2): 156-171.

Cicconetti C, Lenzini L, Mingozzi E, Eklund C (2006). Quality of service support in IEEE 802.16 networks. IEEE Network. 20(2): 50-55.

Demers A, Keshav S, Shenker S (1989). Analysis and simulation of a fair queueing algorithm. In the Proceedings of Symposium proceedings on Communications architectures and protocols. 1-12.

De Moraes LFM, Maciel Jr. PD (2005). Analysis and evaluation of a new MAC protocol for broadband wireless access. In the Proceedings of Int. Conf. Wireless Networks, Comm. Mobile Comput., 107-112.

De Moraes LFM, Maciel, Jr PD. (2006). An alternative QoS architecture for the IEEE 802.16 standard. In the Proceedings of 2006 ACM CoNEXT conference.

Freitag J, da Fonseca NLS (2007). Uplink Scheduling with Quality of Service in IEEE 802.16 Networks. In the Proceedings of IEEE Global Telecomm. Confer., 2503-2508.

GuoSong C, Deng W, Shunliang M (2002). A QoS architecture for the MAC protocol of IEEE 802.16 BWA system. In the Proceedings of Communications, Circuits and Systems and West Sino Expositions. 435-439.

Hui DSW, Lau VKN, Wong Hing L (2007). Cross-Layer Design for OFDMA Wireless Systems With Heterogeneous Delay Requirements. IEEE Transactions on Wireless Comm., 6(8): 2872-2880.

IEEE Std 802.16-2004 (2004). IEEE Standard for Local and Metropolitan Area Networks Part 16: Air Interface for Fixed Broadband Wireless Access Systems. Revision of IEEE Std 802.16-2001.

IEEE Std 802.16e-2005 and IEEE Std 802.16-2004/Cor 1-2005 (2006). IEEE Standard for Local and metropolitan area networks Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems Amendment 2: Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands and Corrigendum 1. (Amendment and Corrigendum to IEEE Std 802.16-2004). 1-822.

Jianfeng C, Wenhua J, Qian G (2005a). An integrated QoS control architecture for IEEE 802.16 broadband wireless access systems. In the Proceedings of IEEE Global Telecomm. Conf.,

Jianfeng C, Wenhua J, Hongxi W (2005b). A service flow management strategy for IEEE 802.16 broadband wireless access systems in TDD mode. In the Proceedings IEEE Int. Conf. Comm., 3422-3426.

Kim DH, Kang CG (2005). Delay Threshold-Based Priority Queueing Packet Scheduling for Integrated Services in Mobile Broadband Wireless Access System. In the Proceedings of IEEE Int. Conf. High Performance Comput. Comm., 305-314.

Lu S, Bharghavan V, Srikant R (1999). Fair scheduling in wireless packet networks. IEEE/ACM Trans. Networking. 7: 473-489.

Naian L, Xiaohui L, Changxing P, Bo Y (2005). Delay Character of a Novel Architecture for IEEE 802.16 Systems. In the Proceedings of Sixth International Conference on Parallel and Distributed Comput. Appl. Technol., 293-296.

Noordin KA, Markarian G (2007). Uplink Scheduling for Quality of Service Support in IEEE 802.16 Systems. In the Proceedings of The Ninth International Symposium on Comm. Theory Appl.,

Sayenko S, Alanen O, Hamaainen T (2008). Scheduling solution for the IEEE 802.16 base station. Int. J. Comput. Telecomm. Networking. 96-115.

Shejwal A, Parhar A (2007). Service Criticality Based Scheduling for IEEE 802.16 WirelessMAN. In the Proceedings of The 2nd International Wireless Broadband and Ultra Wideband Comm., 12-12.

Song G, Li Y (2005). Utility-based resource allocation and scheduling in OFDM-based wireless broadband networks. IEEE Comm. Magazine. 43(12): 127-134.

Stolyar AL, Ramanan K (2001). Largest Weighted Delay First Scheduling: Large Deviations and Optimality. Annals of Appl. Probability. 11: 1-48.

Sun J, Yanling Y, Hongfei Z (2006). Quality of Service Scheduling for 802.16 Broadband Wireless Access Systems. In the Proceedings of IEEE 63rd Vehicular Technol. Conf., 2006. VTC 2006-Spring. 1221-1225.

Tassiulas L, Sarkar S (2002). Maxmin fair scheduling in wireless networks. In the Proceedings of IEEE Comput. Comm. Conf., 763-772.

Vaidya NH, Bahl P, Gupta S (2005). Distributed fair scheduling in a Wireless LAN. IEEE Trans. Mobile Comput., 4: 616-629.

Wang P, Markarian G (2004). Traffic Management Architecture for IEEE

802.16 System. In the Proceedings of The Seventh International Symposium on Wireless Personal Multimedia Comm.,

WiMAX Forum (2001). Available: http://www.wimaxforum.org/home/

Wongthavarawat K, Ganz A (2003a). Packet scheduling for QoS support in IEEE 802.16 broadband wireless access systems. Int. J.Comm. Syst., 16: 81-96.

Wongthavarawat K, Ganz A (2003b). IEEE 802.16 based last mile broadband wireless military networks with quality of service support. In the Proceedings of IEEE Military Comm. Conf., MILCOM 2003. 779-784.

Worldwide Interoperability Microwave Broadband Access System for Next Generation Wireless Communications (WiMAGIC) (2008). Available: http://www.wimagic.eu/

Yan W, Chan S, Zukerman M, Harris RJ (2008). Priority-Based fair Scheduling for Multimedia WiMAX Uplink Traffic. In the Proceedings of IEEE Int.Conf. Comm., 301-305.