A QoS aware Packet Scheduling Scheme for WiMAX

Mahasweta Sarkar and Harpreet Sachdeva

ABSTRACT

WiMAX is one of the most promising broadband wireless technologies today. The WiMAX standard-802.16-is designed to provide Quality of Service (QoS) to delay sensitive traffic like video and voice. However, the standard does not specify any particular scheduling algorithm to achieve the above objective. This paper proposes a novel packet scheduling scheme for providing QoS to downlink traffic i.e traffic transmission from the base station (BS) to the subscriber stations (SS) in WiMAX. The primary objective in designing the scheduling scheme is to provide each traffic type with appropriate resources such that it meets its average packet delay and average packet loss criteria. Extensive simulations have been performed to test the scheduling algorithm. Simulation results demonstrate that our scheduling scheme minimizes packet loss and almost always meets average packet delay constraint even in a heavily loaded network.

***Keywords* — WiMAX, scheduling, QoS, packet delay, packet loss**

1. INTRODUCTION

WiMAX has been designed to deliver QoS to traffic that requires it[1]. It achieves this by categorizing traffic into four different classes depending on their QoS requirement namely Unsolicited Grant Service (UGS), Real Time Polling Service (rtPS), Non Real Time Polling Service (nrtPS) and Best Effort (BE). Thus an application like video conferencing which constitutes of packets that are highly delay sensitive will be classified into a different traffic type (like UGS) that would deserve preferential treatment by the network than an application like web browsing which comprises of relatively delay insensitive data packets and can do with a best effort type service from the network – hence would be classified as BE traffic.

This work has been supported by the National Science Foundation grant award DUE-737048
Harpreet Sachdeva is with the Department of Electrical and Computer Engineering San Diego State University, San Diego
(e-mail: h_sachdevas@yahoo.com).
Mahasweta Sarkar is now with the Department of Electrical and Computer Engineering San Diego State University, San Diego, CA, USA
(e-mail: msarkar2@mail.sdsu.edu).

The IEEE standard for WiMAX -802.16 [1]- does not define a particular scheduling algorithm for packet transmission in either, the uplink (from SS to BS) or the downlink (from BS to SS) direction. Thus, developing QoS guaranteeing packet scheduling algorithms has been an active area of research in recent times [4][5].

The major objective of this paper is to develop an effective downlink packet-scheduling algorithm which provides QoS to different types of traffic according to their QoS requirement. The scheduling algorithm aims at minimizing packet loss and packet delay for all traffic types especially the ones that are sensitive to these performance metrics. This is achieved by implementing a two tier scheduling algorithm at the Base Station (BS). Incoming traffic at the BS is first classified by a traffic classifier into one of the four traffic categories as outlined in the 802.16 standard[1]. Each packet is buffered in a queue dedicated to its traffic class. The first tier of scheduling is in essence a resource allocation scheme which chops up the available bandwidth into four unequal portions, allocating each portion to a particular traffic class. For example, a 100 KB bandwidth maybe sliced up into four unequal chunks of 40KB, 30KB, 20 KB and 10 KB allocated to UGS, rtPS, nrtPS and BE traffic respectively. After this initial bandwidth allocation, the downlink packet scheduler residing at the Base Station (BS) schedules transmission of packets that are queued at the BS. Recall that there are four different queues pertaining to the four different traffic classes for each SS that the BS serves. The second tier of scheduling comes into play when a decision regarding the service policy of each particular traffic queue has to be called to order. The novelty of this scheme lies in the fact that the downlink packet scheduler follows four different service policies in serving the four different queues based on the QoS requirements of the packets buffered in each queue, namely "Fixed Bandwidth" service policy for UGS traffic; "Earliest Deadline First – Round Robin" service policy for rtPS traffic; "Weighted Round Robin" service policy for nrtPS traffic and "Fair Sharing" service policy for BE traffic. In addition, the scheduling algorithm ensures overall fairness by policing undue hogging of bandwidth by any traffic class. It is simple enough to implement and complies with the broad outlines stated in the IEEE 802.16 standard.

In order to test the performance of the downlink scheduler, a C++ programming module has been written. Extensive simulations have been performed to test the scheduling algorithm. Results from the simulation shows, that the proposed downlink packet scheduling algorithm provides minimal packet loss and almost always meets the delay constraint of each traffic type with remarkable efficiency even when the network is overtly crowded.

The rest of the paper is organized as follows: Section 2 describes our downlink packet scheduling (DPS) algorithm, Section 3 describes the simulation set-up and also presents and discusses the results. We conclude the paper in Section 4.

2. SCHEDULING ALGORITHM

This paper focuses on designing a **D**ownlink **P**acket **S**cheduler (DPS) that supports Quality of Service (QoS) for different traffic types in WiMAX. DPS minimizes packet loss, meets packet delivery deadlines and achieves an application relevant fairness in resource allocation (bandwidth sharing) amongst all different traffic types.

2.1. Overview of the Scheduling Algorithm

Unlike other scheduling schemes [7]-[10]which focuses on a round robin like scheme between the different subscriber stations, serving one SS to the extent of emptying its buffer before moving on to the next, this paper takes a different approach towards scheduling. It suggests that the downlink scheduler serve packets to the SSs not by one SS at a time but by one traffic type at a time according to the traffic priority. This is because the UGS, rtPS and nrtPS traffic have specific requirements. For instance, UGS and rtPS traffic have strict packet deadlines and late packets that miss the deadline will be useless, but these two services can tolerate packet loss. However, for nrtPS traffic packet loss is not permitted but accommodates larger delays. DPS was designed to guarantee such customized service requirements. Thus DPS first transmits UGS packets followed by rtPS packets, followed by nrtPS packets and finally the BE packets. Note that, the scheduler has to now decide upon a service policy in serving packets belonging to a particular traffic type queued for the various SSs. Recall that the scheduler (which resides in the BS) buffers incoming packets for a particular SS. The scheduler has a traffic classifier which classifies incoming traffic into the 4 categories as discussed previously. To illustrate with a concrete example, let a BS be serving three SSs. Then the downlink packet scheduler that resides at the BS hosts incoming packets for these three SSs in three different queues. As per our scheduling scheme, each of these queues will be subdivided into four queues, each housing data packets belonging to a particular traffic type. Thus, in our example, the downlink scheduler will have twelve queues in total – three of which will house UGS traffic for SS1, SS2 and SS3 respectively; three of the remaining nine will house rtPS traffic for SS1, SS2 and SS3 respectively; three others will house nrtPS traffic for SS1, SS2 and SS3 respectively while the last three will house BE traffic for SS1, SS2 and SS3 respectively.

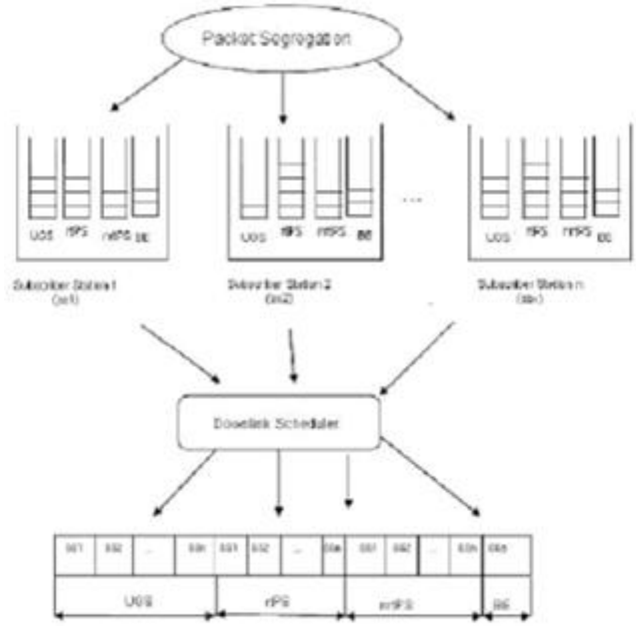

Figure 1 illustrates this design concept. To support all types of service flows (UGS, rtPS, nrtPS and BE), the DPS uses a combination of strict priority service discipline, earliest deadline first Round Robin (EDF-RR)[2] and weighted round robin (WRR) [3]. We will consider the following general approach in our scheduling. Since the real time applications are delay sensitive, we will assume that the requirements of real-time applications (UGS and rtPS) need to be met first while the packets of non real time data can be deferred. Thus we propose that, given the requirements of SSs for all the classes, the BS will first try to satisfy the needs of the UGS applications in the downstream first followed by the rtPS, nrtPS and finally the BE traffic types. The scheduling algorithm is outlined below.

2.2. Working Principle of the Scheduler

Bandwidth allocation per flow follows strict priority, from highest to lowest: UGS, rtPS, nrtPS and BE. One disadvantage of the strict priority service discipline is that higher priority connections can starve the bandwidth of lower priority connections. To overcome this problem, we include the traffic policing module in each SS which forces the connection's bandwidth demand to stay within its traffic contract. This will prevent the higher priority connections from using bandwidth more than their allocation. 802.16 allocates fixed bandwidth (fixed time duration) to UGS connections based on their bandwidth requirement. DPS abides by that philosophy as well. For rtPS connections we apply Earliest Deadline First-Round Robin (EDF-RR) service discipline. Thus packets with earliest deadline will be scheduled first in a round robin fashion. We use EDF-RR for rtPS packets as they are delay sensitive and hence needs to be served according to their deadline. We use EDF-RR instead of EDF alone because using EDF alone could starve some sessions leading to certain sessions missing their packet deadline. We apply weighted round robin (WRR) service discipline to nrtPS service flow. We schedule nrtPS packets in a round robin fashion prioritizing the sessions based on their weights. The weights for each session are calculated based on the session's packet loss constraint. The remaining bandwidth is equally distributed amongst each of the BE connections serving the BE queue in a First-in-First-out (FIFO) order.

3. SIMULATION SET-UP AND RESULTS

The performance of the DPS algorithm is evaluated via simulations. The simulations were performed on the Unix written in the C++ programming language.

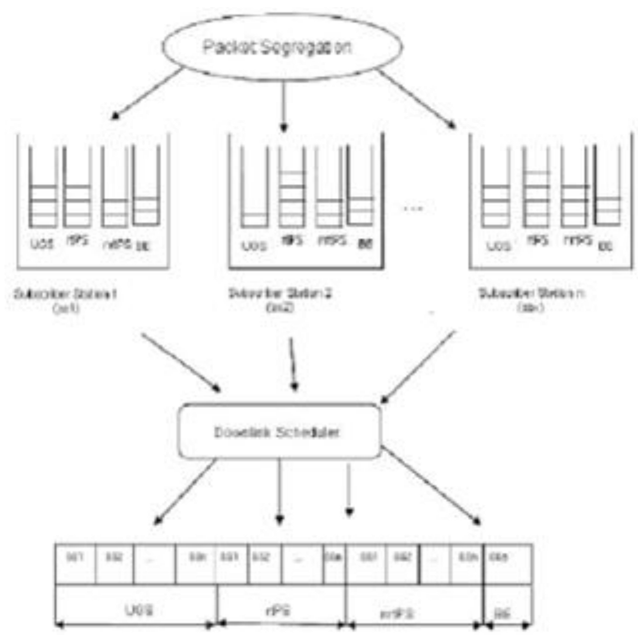

Figure 1: Functionality of the Packet Classifier and the Scheduler [6]

3.1. Simulation Set-up

The downlink bandwidth is assumed to be 10 MB which is divided into 3, 4, 2 and 1MB respectively for UGS, rtPS, nrtPS and BE traffic. Time is divided into discrete units called slots. Packets for a SS arrive at the BS only at the beginning of each time slot. They can be served in the same slot they arrived in or during a later slot. Thus the BS has the provision of queuing data packets for the SSs. In the BS there are four main queues each corresponding to a traffic type. When a packet arrives at the BS, the traffic classifier first classifies the traffic into one of the 4 categories mentioned above, after which it puts the packet in its respective queue for the particular destination SS. Data transmission from the queues is handled by an entity called the "scheduler" which schedules packet transmissions based on the DPS algorithm. Note that the scheduler resides in the BS and runs the DPS algorithm. Each packet going to a particular SS has a specific QoS requirement in terms of packet delay and packet loss. We aim to have our DPS schedule packet transmission such that the QoS requirement of each and every packet is met. We simulated three different traffic scenarios. In each of the scenarios, we varied the packet generation rate of each traffic type and studied the effect of our DPS on average packet loss and average packet delay for each traffic type. The table below summarizes the simulation set-up for the three different scenarios. For example, in the first scenario, for the amount of traffic generated, we have a bandwidth requirement of 3 MB for the UGS traffic, 4.1 MB for rtPS traffic and 2.9 MB for nrtPS traffic. In Scenario 3, the application actually demands more bandwidth than what the network can provide (10MB). It is a challenge for the DPS to still be able to provide QoS to the overloaded network. Since BE traffic is delay tolerant, we do not study the effect of DPS on BE traffic. The appropriate performance metric for BE traffic would be to measure its throughput. Since we focus on packet delay and packet loss in this work, we ignore BE traffic performance criteria. However, it is of interest to us to study it in our future work. It is to be noted that in our simulation we assume a buffer size that can accommodate twice the size of data that can be sent in a 10 millisecond (ms) frame. Thus in our simulation we assume a buffer size of 600 bytes for the UGS traffic, 800 bytes for rtPS traffic, and 420 bytes for nrtPS traffic.

Table 1: Simulation parameters for 3 traffic scenarios

Traffic Type	BW required in Scenario 1	BW required in Scenario 2	BW required in Scenario 3	BW allocated to this traffic type
UGS	3 Mbps	3.3 Mbps	4 Mbps	3 Mbps
rtPS	3.9 Mbps	4.2 Mbps	5 Mbps	4 Mbps
nrtPS	2.1 Mbps	2.3 Mbps	3 Mbps	2 Mbps

3.2. Discussion of the Results

We focus on two performance metric - average packet loss and average packet delay. We vary the packet generation rate of each traffic type which puts a different demand on the bandwidth requirement by each of these traffic types. A higher packet generation rate translates to more number of packets which extrapolates to a higher bandwidth requirement. Figure 2 shows the percentage of packets that do not meet the delay deadline. Once again, our scheduler efficiently schedules packet such that most packet deadlines are met. On an average, the highest percentage of packet not making their delay deadline is 0.43 for UGS traffic and 0.31 for rtPS traffic, which is negligible considering the tremendous load on the network. On the other hand, in a moderately crowded network only 0.04 and 0.01% of the UGS and rtPS packets respectively cannot meet their deadlines.

The point worth noticing is that since rtPS traffic is less delay tolerant than UGS traffic, the scheduler schedules packet transmission in such a way that more of rtPS packets meet their deadlines rather than UGS packets which are more delay tolerant, under every load condition. The same argument also holds true for nrtPS traffic. In times of scarce bandwidth, the scheduler provides for the rtPS traffic more aggressively than the nrtPS traffic since the later kind of traffic is more delay tolerant than the former. This goes on to validate the efficiency of our scheduler.

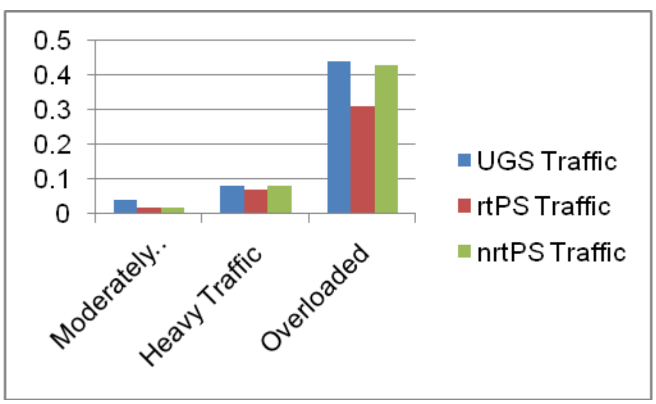

Figure 2: Average percentage of packets that cannot meet the delay deadline for UGS, rtPS and nrtPS traffic under various load conditions

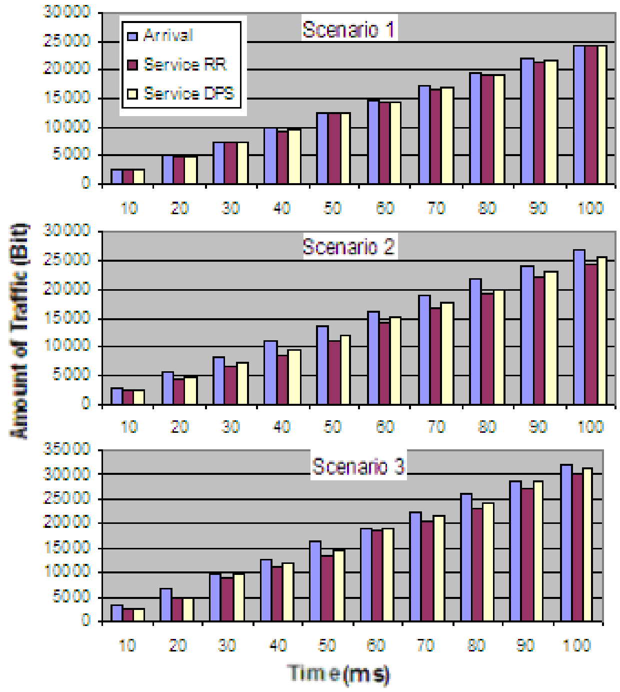

Figure 3: UGS traffic generated and served under three different network load - moderately heavy [Scenario 1], heavy[Scenario 2] and overloaded[Scenario 3]

Figures 3, 4 and 5 show the packet loss that the system undergoes under various traffic loads for UGS, rtPS and nrtPS traffic. The X-axis in these figures represents time while the Y-axis represents amount of traffic in bits. The graphs in Figures 3, 4 and 5 shows the amount of UGS, rtPS and nrtPS traffic generated at a particular point in time versus that which is transmitted. When the height of the “arrival” bar and the “service” bar are equal, it indicates that there were no packet losses. As seen from the graphs, there is almost no packet loss in Scenario 1 and minimal packet loss in Scenario 2 and Scenario 3 for all three traffic categories for our DPS scheme. That is because, in the latter scenarios, the network is heavily loaded. The bandwidth requirement in the last scenario even surpasses the bandwidth available in the network. Thus our DPS shows remarkable performance in meeting packet delay constraint and minimizing packet loss even when the network is overloaded. When we compare our DPS scheme to that of the widely used Round Robin (RR) scheme, we observe that the RR scheme always has a higher packet loss especially when the load on the network is high (Scenarios 2 and 3 for all different traffic types). In the Round Robin scheme the scheduler simply serves the packets by transmitting one packet from UGS, rtPS, nrtPS and BE queues in a round robin manner disregarding the service requirement of the packet. Thus in the Round Robin scheme we observe more packets missing their deadlines. Also no intelligent decision is made on the amount of appropriate bandwidth which should be allocated to each of the packet types. Our DPS algorithm improves upon the RR scheme on both these counts. The comparative results in Figures 3, 4 and 5 validates the sanctity of our DPS scheme over the RR scheme.

4. CONCLUSIONS

In this paper a novel downlink packet scheduling (DPS) algorithm is presented which provides QoS for downlink traffic in WiMAX. The primary objective in designing the scheduling scheme is to provide each traffic type with appropriate resource such that it meets its performance criteria. The novelty of our scheme lies in the fact that unlike most scheduling algorithms, DPS schedules traffic transmission not per SS but per traffic type. Such a scheduling scheme delivers the required QoS of each traffic type. We verify the efficiency of our algorithm via extensive simulations under severely trying traffic loads. We have carried out the simulations in the Unix environment using the C++ programming language. We have simulated three different load conditions – moderately heavy load, heavy load and an overloaded network. In a moderately heavily loaded network, the packet generation rate of each traffic type is high enough such that it almost consumes the entire bandwidth quota of the network for all traffic types. In the heavily loaded and overloaded network, the packet generation rate of each traffic type is slightly higher and much higher respectively, than the bandwidth availability of the network. Simulation results show that the proposed DPS algorithm provides minimal packet loss and almost always meets the delay constraint of each traffic type even under heavily loaded network condition. In addition, DPS provides relevant fairness in resource allocation to the different traffic types.

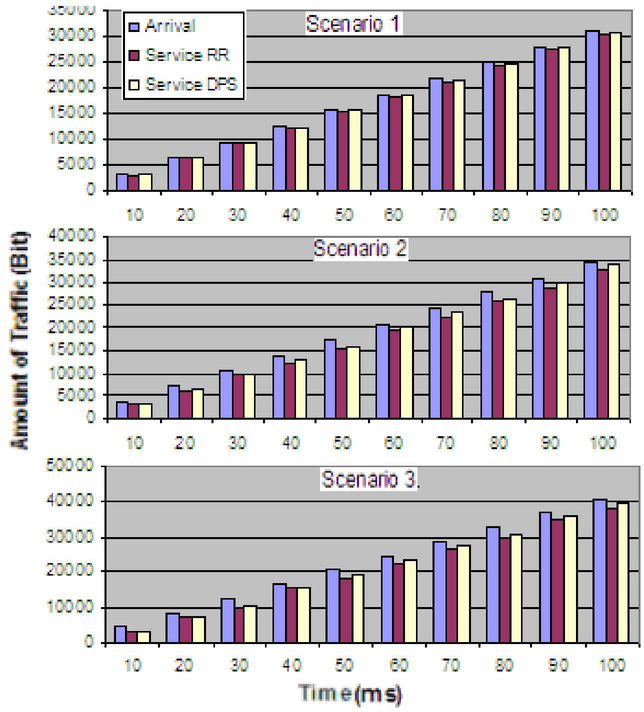

Figure 4: Traffic generated and served for rtPS traffic for moderately heavy[Scenario 1], heavy[Scenario 2] and overloaded [Scenario 3]network load conditions

Such a scheduling scheme delivers the required QoS of each traffic type. We verify the efficiency of our algorithm via extensive simulations under severely trying traffic loads. We have carried out the simulations in the Unix environment using the C++ programming language. We have simulated three different load conditions – moderately heavy load, heavy load and an overloaded network. In a moderately heavily loaded network, the packet generation rate of each traffic type is high enough such that it almost consumes the entire bandwidth quota of the network for all traffic types. In the heavily loaded and overloaded network, the packet generation rate of each traffic type is slightly higher and much higher respectively, than the bandwidth availability of the network. Simulation results show that the proposed DPS algorithm provides minimal packet loss and almost always meets the delay constraint of each traffic type even under heavily loaded network condition. In addition, DPS provides relevant fairness in resource allocation to the different traffic types.

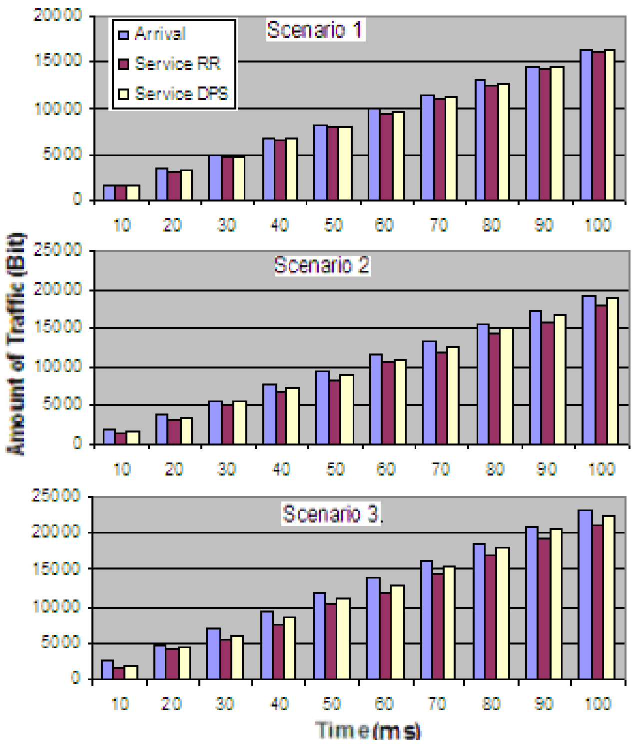

Figure 5: Traffic generated and served for nrtPS traffic for moderately heavy[Scenario 1], heavy[Scenario 2] and overloaded [Scenario 3]network load conditions

REFERENCES

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

[2] Deming Liu and Yann-Hang Lee, *"An Efficient Scheduling Discipline for Packet Switching Networks Using Earliest Deadline First Round Robin"*. Proceedings of the IEEE 12th International Conference on Computer Communications and Networks (ICCCN), October 2003.

[3] M. Katevenis, S. Sidiropoulos, C. Courcoubetis, *"Weighted Round-Robin Cell Multiplexing in a General-Purpose ATM Switch Chip"*, IEEE Journal on Selected Areas in Communications, vol. 9, no. 8, October 1991, pp. 1265-1279

[4] Kitti Wongthavarawat and Aura Ganz, *"Packet Scheduling for QoS Support in IEEE 802.16 Broadband Wireless Access Systems"*, in Wiley Interscience International Journal of Communication Systems, Volume 16, Issue 1 pp 81-96, February 2003.

[5] H. Mohammad, A.Najah and K.Issa, *"Uplink and Downlink Scheduling for Point to Multipoint WiMAX Networks",* Wiley Interscience Journal of Wireless Communication and Mobile Computing Jones, October 2008

[6] J.Thaliath, M.M Joy, E.Priya John, D.Das, *"Service class downlink scheduling in WiMAX",* in Proceedings of 3rd International Conference on Communication Systems Software

[7] O.Yang and J.Lu, *"Call Admission Control and Scheduling Schemes with QoS Support for Real-time Video Applications in IEEE 802.16 Networks"*, Journal of Multimedia, vol.1, no.2, May 2006.

[8] K.Vinay, N.Sreenivasulu, D.Jayaram and D.Das, *"Performance evaluation of end-to-end delay by hybrid scheduling algorithm for QoS in IEEE 802.16 network"*, Proceedings of IEEE International Conference on Wireless and Optical Communication Networks, April 2006.

[9] M.Settembre, M.Puleri, S.Garritano, P.Testa, R.Albanese, M.Mancini and V.Lo Curto, *"Performance analysis of an efficient packet-based IEEE 802.16 MAC supporting adaptive modulation and coding"*, Proceedings of IEEE International Symposium on Computer Networks, pp.11-16, June 2006.

[10] J.Lin and H.Sirisena, *"Quality of Service Scheduling in IEEE 802.16 Broadband Wireless Networks",* Proceedings of First International Conference on Industrial and Information *Systems*, pp.396-401, August 2006.