Queue Based Scheduling for IEEE 802.16 Wireless Broadband

K. R. Raghu
Positioning and Wireless Technology Centre
Nanyang Technological University
Singapore
ragh0003@ntu.edu.sg

Sanjay K. Bose, Maode Ma
School of EEE,
Nanyang Technological University
Singapore
eskbose, emdma@ntu.edu.sg

Abstract— **IEEE 802.16 Wireless Broadband is a promising technology for providing last mile access. Quality of Service (QoS) is an important factor that has been addressed by the standard which defines four types of multimedia traffic classes. However resource allocation and scheduling between the traffic classes is vital in providing QoS which the standard has left undefined. In this paper we propose a queue based scheduling algorithm for real time and non real time traffic at the Medium Access Control (MAC) layer. The algorithm is based on resource sharing between real time and non real time traffic depending on their queue size and latency requirements. We have simulated various traffic scenarios by implementing the algorithm in QualNet WiMAX simulator and studied its performance.**

Keywords—**Broadband wireless networks, scheduling, IEEE 802.16, QoS, WiMAX.**

I. INTRODUCTION

IEEE 802.16 Wireless Metropolitan Area Network (MAN) air interface standard [1] technology is designed to provide a cost-effective last mile broadband access. It has provided extensive details for the Physical (PHY) and MAC layers. The standard aims at providing wireless broadband internet connectivity to residences and small offices supporting data rates of up to 70Mbps with the base station coverage of 2-10Kms. It provides mesh networking by interconnecting various access points and base stations. The network standard of [1] lays strong emphasis on quality of service (QoS) and defines different traffic classes with separate QoS specifications for various types of multimedia traffic. Scheduling and resource allocation are used to ensure QoS along with admission control and traffic policing. However, [1] leaves the details of design to the developers. This motivates the need for an effective scheduling mechanism to deliver QoS guarantees, especially to the real-time traffic.

The standard defines the following types of QoS services -

(i) Unsolicited Grant Service (UGS) is defined for supporting constant bit rate traffic, e.g. VoIP without silence suppression. Generally, a fixed bandwidth is allotted for this service so as to minimize delay without the need for bandwidth requests. The bandwidth required is negotiated during the connection set up.

(ii) Real-time Polling Service (rtPS) supports variable bit rate real-time traffic, e.g. VoIP with silence suppression or MPEG streams. Uplink bandwidth allocation is based on a polling scheme. The Base Station (BS) implicitly polls each subscriber station (SS) and the SS replies with a bandwidth request to obtain a grant for the messages that it can transmit. This scheme can guarantee QoS service to meet delay requirements.

(iii) Non Real-time Polling Service (nrtPS) supports variable bit rate traffic which is delay tolerant, e.g. FTP or HTTP traffic. Like rtPS, it also follows a polling approach for its bandwidth requests. Though it does not have a stringent QoS requirement a minimum data rate must be ensured to support the traffic under this scheme.

(iv) Best Effort (BE) supports data traffic which does not need any QoS provisioning. Its bandwidth allocation relies on a contention-based request and grant scheme.

All these traffic connections co-exist in a Subscriber Station and both the BS and the SS monitors all the traffic connections between them. The uplink bandwidth can be provided by the BS based on one of the two proposed schemes in [1]. Grant Per Connection (GPC) in which the bandwidth is allocated to each connection and Grant Per SS (GPSS) where the bandwidth is allocated to each SS as an aggregate of bandwidth for all the connections within the SS. The second method is found to be more efficient and scalable [2] [3]. As indicated earlier, the strict QoS requirements of real-time traffic has to be met. However, always providing a fixed bandwidth to real time traffic to minimize its delay is unfair to other services since rtPS traffic is also generally bursty. The non real-time services should be able to utilize the available bandwidth when the real-time traffic is inactive.

Several bandwidth allocation and scheduling approaches have been proposed earlier. In [4] a scheduling algorithm is proposed to assign a fixed bandwidth for UGS, using Earliest Deadline First (EDF) technique for rtPS, Weighted Fair Queuing (WFQ) for nrtPS and equal distribution for BE. These algorithms have been improved by a proposal in [2] which evaluates the end-to-end delay by using a hybrid scheduling algorithm which is a combination of Earliest Due Date (EDD) for real time data traffic and Weighted Fair Queuing (WFQ) for non real time traffic. The scheduling algorithm in [5] ensures the targeted QoS requirement in a GPC system while [6] analyzes the performance of a stand alone Voice over Internet

1-4244-0983-7/07/$25.00 ©2007 IEEE

Protocol (VoIP) connection with silence suppression in scenarios where mobile stations only transmit voice traffic over the air interface. UGS scheduling scheme can take care of independent constant bit rate (CBR) voice traffic since it allots fixed amount of bandwidth in each frame. It would however result in wastage of bandwidth when used with (Variable Bit Rate) VBR sources. It is noted that [6] has also analyzed the suitability of rtPS scheme for VoIP. Since rtPS always uses a bandwidth request process for suitable size grants, it transports data more efficiently. However, this bandwidth request process adds MAC overhead and extra access delay. Since in most fixed broadband cases, an SS handles multiple classes of traffic from multiple devices, it requires an algorithm to prioritize real-time traffic flows over non real-time traffic and yet be fair. In [7], an adaptive queue-aware algorithm is proposed for uplink bandwidth allocation and rate control mechanisms in a SS for polling services in a GPSS system. By this bandwidth allocation scheme, the amount of bandwidth allocated for polling service can be adjusted dynamically according to the variations in traffic load, channel quality, and queue length at SSs so that the packet-level QoS performances such as protocol data unit (PDU) delay and PDU dropping probability can be maintained at the desired level. Here, rate control is also used to limit the transmission rate of the connections under polling service class so that the overall QoS performance can be controlled. However, the proposal in [7] treats real-time and non real-time services identically and also does not adequately exploit QoS factors like maximum latency in its scheduling. In [7], the real-time and non real-time traffic use the same queue and therefore the system cannot distinguish between the different QoS requirements of different traffic flows. Since the major difference between the two types of traffic is their tolerance to delay, it is important to separate them in different queues and explicitly incorporate the maximum latency specification of the real time traffic in the scheduling process itself.

The organization of the paper is as follows. Section II gives the IEEE 802.16 architecture and operation. Section III provides the description of the scheduling algorithm. Section IV describes the simulation scenarios and analyses the results. Section V concludes the paper.

II. IEEE 802.16 ARCHITECTURE

A broadband wireless access system includes at least one BS and many SSs, each of which can be identified by a unique 48-bit MAC address. The point to multipoint architecture is shown in Fig. 1. An SS supports multimedia traffic like VoIP, Video On Demand (VOD), MPEG streaming and non real time traffic such as HTTP and FTP coming from different devices. The BS controls the operation of the SSs. The TDMA/TDD MAC frame structure is defined in [1] where the frame is divided into uplink (UL) and downlink (DL) sub frames as shown in Fig. 2. The BS provides contention slots on the uplink for bandwidth requests by the SSs. At an SS, when a new MAC Protocol Data Unit (MPDU) arrives from the upper layer which does not belong to any of the currently existing flows, it is classified into one of the scheduling types and a new flow is created. Each flow will be identified by a Connection Identifier (CID) issued by the BS during the flow set up.

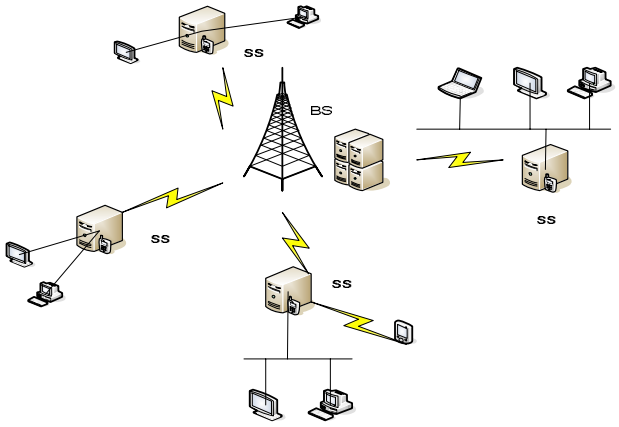

Figure 1. Wireless Broadband Access Network

The BS can control the frame size as well as the duration of the sub frames. This enables it to have adaptive coding and modulation schemes. The frame contains both broadcast and unicast information in the DL sub frame. Each sub frame has a number of slots and the BS assigns the uplink slots dynamically to the SSs. At the start of each downlink frame, the BS informs each SS when their respective data burst is scheduled on the downlink by the DL-MAP. It also specifies the time slot and duration for which each SS can transmit on the uplink through the UL-MAP.

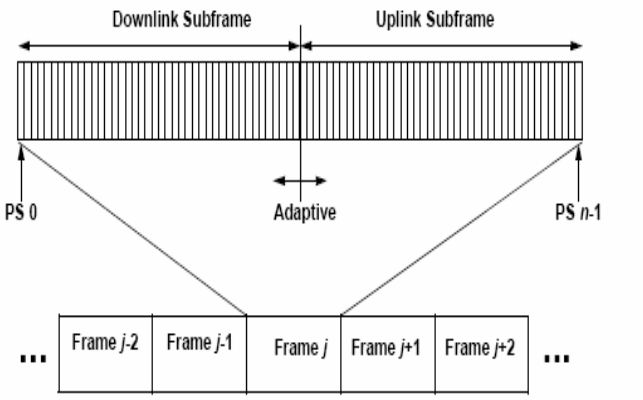

Figure 2. Time Division Duplexing in IEEE 802.16 Frame

In the unicast polling scheme, the BS periodically polls every SS which can send its bandwidth request for polling service traffic during the slot allotted to it. The BS schedules each SS based on the aggregate bandwidth required by all the flows within the SS in a GPSS system. The SS can send additional bandwidth requests in an incremental or aggregate fashion against specific CIDs when it is polled again.

III. SCHEDULING ALGORITHM

The objective of the proposed scheduling algorithm is to efficiently allocate the bandwidth granted by the BS to the SSs such that the QoS requirement of the real-time traffic can be met while at the same time providing a fair share of the bandwidth to non-real time traffic. We consider only the UGS, rtPS and nrtPS traffic classes without including Best Effort

service. Each traffic flow in a SS can be mapped to one of the three (UGS, rtPS, nrtPS) traffic classes as defined in [1]. In each uplink frame, the BS allocates certain number of time slots to each SS depending upon the bandwidth request for each flow managed by the SS. For every SS, the aggregate bandwidth granted by the BS is given by

$$B_{grt}^{tot} = \sum_{j=1}^{k} B_j^{ugs} + \sum_{j=1}^{l} B_j^{rt} + \sum_{j=1}^{m} B_j^{nr} \quad (1)$$

Where *k, l, m* are the number of UGS, rtPS, nrtPS flows and B_j^{ugs}, B_j^{rt}, B_j^{nr} are the individual bandwidth request of each flow. The BS allots a part of the uplink frame (burst duration D_i) to each SS_i ($i \in (1, 2, \ldots, n)$) proportional to the aggregate bandwidth grant and indicates this in the UL_MAP of the downlink frame. For each SS, given B_w as the uplink bandwidth, the total uplink frame-length available (in bits) is

$$F_{tot} = D_i * B_w \quad (2)$$

The SS in turn schedules each of the service flows. Since the UGS flows are CBR flows, they are scheduled based on their maximum sustained rate as

$$F_{ugs} = \sum_{j=1}^{k} B_j^{ugs} * (C_j^t - C_j^{t-1}) \quad (3)$$

Where C_j^t is the current time and C_j^{t-1} is the last bandwidth allocation time of the UGS flow *j*. This generally takes a fixed chunk of bandwidth as the frame size is fixed in our case. The remaining uplink frame length F_{poll} that is allotted for the polling service will be

$$F_{poll} = F_{tot} - F_{ugs} \quad (4)$$

The real-time and non real-time traffic are serviced based on the number of MPDU's in their respective flows at the beginning of the uplink frame. We define a parameter α as the ratio of the maximum time a rtPS or nrtPS MPDU can wait in the queue (i.e. *max_mpdu_delay*) to the maximum latency specification of the real-time flows.

$$\alpha = \frac{\max\_mpdu\_delay}{\max\_latency\_of\_rtPS\_flow} \quad (5)$$

It should be noted that, for convenience, we define that MPDU's belonging to a nrtPS flow can wait in the queue for *max_mpdu_delay* seconds, even though, in practice, they may not have any maximum latency defined. The *max_mpdu_delay* depends on the buffer length of the queues at all the SSs. The parameter α can be considered as a design parameter to control the QoS given to real-time and non real-time services and can be varied to obtain the desired delays for real-time and non real-time traffic flows. If N_j^{rt} is the number of MPDU's in the real-time queue *j* at an SS at the start of the uplink frame, then the total number of MPDU's in all the real-time queues at the SS at this instant is

$$N^{rt} = \sum_{j=1}^{l} N_j^{rt} \quad (6)$$

If N_j^{nr} is the number of MPDU's in the non real-time queue *j* in the SS at the start of the uplink frame, then the total number of MPDU's for all non real-time traffic flows in the SS at that instant would be

$$N^{nr} = \sum_{j=1}^{m} N_j^{nr} \quad (7)$$

The real-time traffic flows are allotted a frame time of

$$F_{tot}^{rt} = F_{poll} * \frac{N^{rt} * \alpha}{(N^{rt} * \alpha + N^{nr})} \quad (8)$$

The non real-time traffic flows are allotted a frame time of

$$F_{tot}^{nr} = F_{poll} * \frac{N^{nr}}{(N^{rt} * \alpha + N^{nr})} = F_{poll} - F_{tot}^{rt} \quad (9)$$

This would make the bandwidth allocation depend on both the traffic in the queues as well as the latency requirement. To decrease delay of rtPS flows we may use a larger α so that more bandwidth gets allotted to real time flows.

The individual real time flows $v \in (1, 2, 3\ldots, l)$ are allotted

$$F_v^{rt} = F_{tot}^{rt} * \frac{N_v^{rt}}{\sum_{j=1}^{l} N_j^{rt}} \qquad v \in (1, 2, 3\ldots, l) \quad (10)$$

as their respective frame lengths and the individual non-real-time flows $w \in (1, 2, 3\ldots, m)$ are allotted frame-lengths

$$F_w^{nr} = F_{tot}^{nr} * \frac{N_w^{nr}}{\sum_{s=1}^{m} N_s^{nr}} \qquad w \in (1, 2, 3\ldots, m) \quad (11)$$

This process is repeated at the beginning of every uplink by every SS. The advantage of this algorithm is that the bandwidth provided to the real-time traffic is based on its latency and queue size so that its average end-to-end delay can be reduced. At the same time the nrtPS traffic can obtain additional bandwidth when the real-time traffic is less i.e. in inactive burst periods. At the SS, it would also take care of traffic fluctuations between the bandwidth request periods. Since the bandwidth request/grant will take some time to be sent by SSs to the BS along with scheduling time at the BS, the rtPS traffic can actually vary within this period. But this algorithm would not be affected because the granted bandwidth is always redistributed as per the QoS requirements of real-time and non real-time traffic at SSs.

IV. SIMULATION AND RESULTS

The proposed queue based scheduling approach has been simulated using the QualNet WiMAX simulator. The algorithm was implemented at the MAC layer and the end-to-end delay is obtained for different traffic intensities. We simulated a WiMAX PMP scenario with six SS and one BS as shown in the Fig. 3. Each SS supports three in-coming (UGS, rtPS, nrtPS) and three outgoing (UGS, rtPS, nrtPS) flows. We used CBR traffic to emulate UGS flows and Poisson traffic to emulate

rtPS and nrtPS flows. The total bandwidth is 30 Mbps. The TDMA frame size is set as 10ms and divided equally between up-link and down-link. The number of slots (bytes) allocated to polling service during a frame is taken as the product of the average data rate and the frame time. Table I gives the traffic parameters used for simulations in Fig. 4.

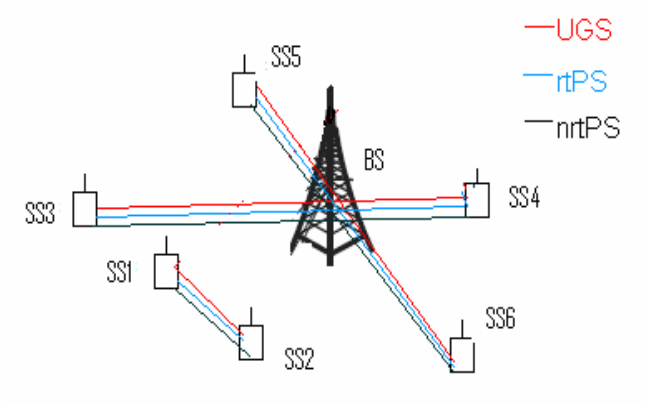

Figure 3. WiMAX PMP Simulation Set up

In a UGS traffic flow, an MPDU is of fixed size (300 bytes). A traffic flow for polling service is a Poisson process and MPDU size is exponentially distributed with mean of 1000 bytes.

TABLE I. TRAFFIC PARAMETERS FOR SIMULATION

Traffic Intensity	Data Rate of UGS (CBR)	Average Data Rate of rtPS (Poisson arrival)	Average Data Rate of nrtPS (Poisson arrival)
0.33	128Kbps	512Kbps	256Kbps
0.45	128Kbps	700Kbps	350Kbps
0.51	128Kbps	800Kbps	400Kbps
0.64	128Kbps	1.0Mbps	500Kbps
0.77	128Kbps	1.2Mbps	600Kbps
0.83	128Kbps	1.3Mbps	650Kbps
0.89	128Kbps	1.4Mbps	700Kbps

The simulations are run for 3, 5, 10 and 15 minutes and the average end-to-end delays are computed on different traffic intensities. The average end-to-end delays for UGS, rtPS, nrtPS traffic are compared in Fig. 4. Here UGS traffic is CBR traffic and is serviced at its maximum sustained rate without any bandwidth constraint hence it has the minimum end to end delay. For scheduling the polling services in this case, $\alpha = 2$ has been set. The real-time flow has a higher bit-rate, twice that of non real-time flow, and hence causes more occupancy in the queue. By our proposed queue-aware scheduling algorithm, servicing of queues is based on their buffer state and the real-time flow is effectively scaled up by $\alpha = 2$, the real-time flow experiences much lower delay than the non real-time flow, with average delay almost similar to UGS flows. We have also simulated a scenario for traffic conditions where nrtPS rate is twice that of rtPS. The parameters used are indicated in Table II. Fig. 5 compares the UGS, rtPS and nrtPS delays for this scenario. It can be seen that rtPS delay has increased over UGS delay (which is roughly similar for both scenarios), and that the nrtPS delay is lower.

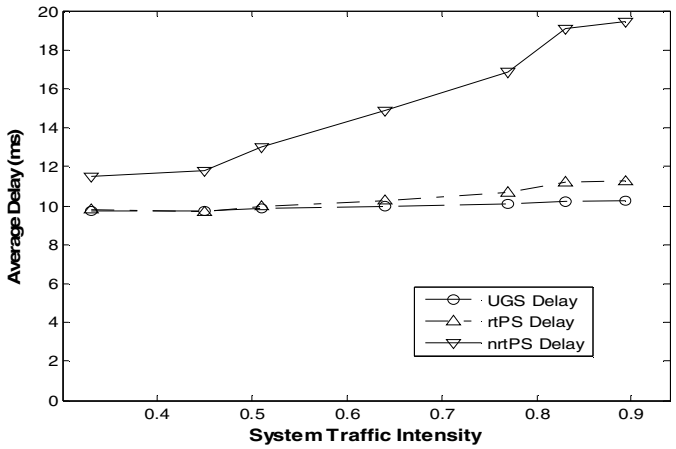

Figure 4. Delay Comparison of rtPS, nrtPS and UGS

TABLE II. TRAFFIC PARAMETERS FOR SIMULATION

Traffic Intensity	Data Rate of UGS (CBR)	Average Data Rate of rtPS (Poisson arrival)	Average Data Rate of nrtPS (Poisson arrival)
0.33	128Kbps	256Kbps	512Kbps
0.45	128Kbps	350Kbps	700Kbps
0.51	128Kbps	400Kbps	800Kbps
0.64	128Kbps	500Kbps	1.0Mbps
0.77	128Kbps	600Kbps	1.2Mbps
0.83	128Kbps	650Kbps	1.3Mbps
0.89	128Kbps	700Kbps	1.4Mbps

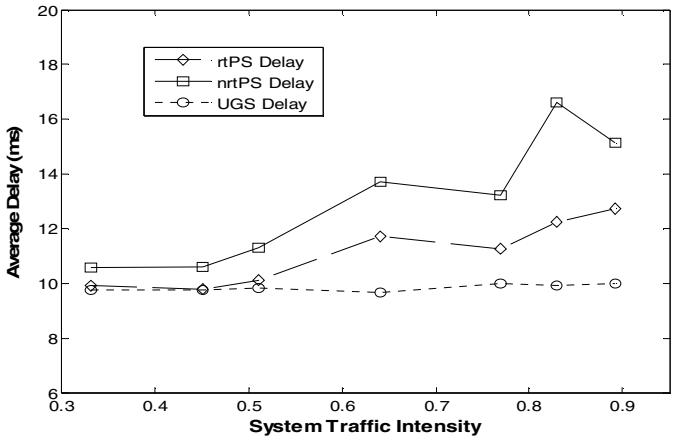

Figure 5. Delay Comparison of rtPS, nrtPS and UGS for data rate of nrtPS twice that of rtPS.

The comparison of rtPS and nrtPS delays for the two scenarios is illustrated in Fig. 6. The increase in rtPS delay can be attributed to the occupancy of the rtPS queue which has fallen compared to the previous scenario and that of the nrtPS queue has increased causing more bandwidth to be allotted to non real time queue. However, the parameter α =2 ensures that rtPS gets higher bandwidth allocation and keeps the rtPS delay under control. The effect of the design parameter α is illustrated in Fig. 7.

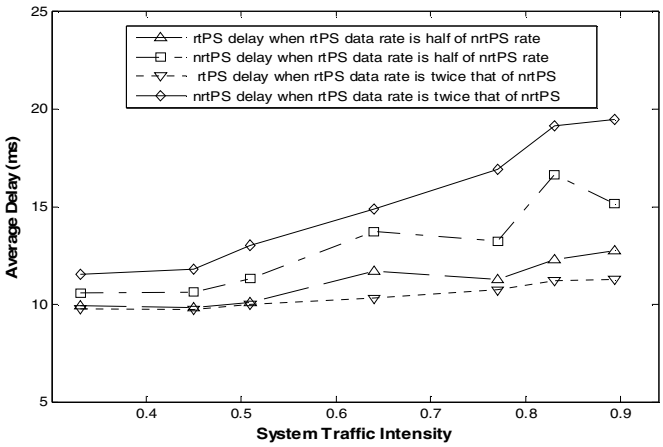

Figure 6. Delay Comparison of rtPS, nrtPS with α =2 for different traffic rates.

It compares α values of 2 and 4 for the scenario where nrtPS data rate is twice that of rtPS. It can be seen that for increased α value the rtPS delay is considerably reduced while nrtPS delay is increased. This is because increasing α increases the bandwidth available for rtPS and decreases bandwidth allocated to nrtPS according to (8) and (9).

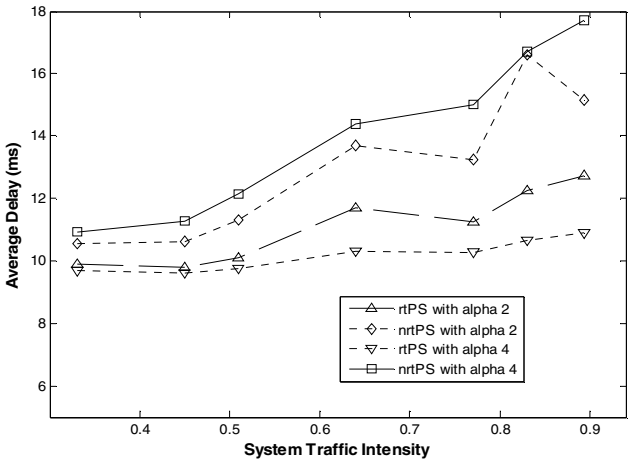

Figure 7. Comparison of rtPS, nrtPS Delays for α = 2, 4

From the above results, it might seem that increasing α would invariably result in decreasing the rtPS delay. However, this is not entirely true because the bandwidth assignment also depends on the relative queue sizes. Fig. 8 compares the end to end delay with α =4 for two different traffic rates. It is clear that when a rtPS flow has a higher data rate than that of a nrtPS flow, increasing the value of α would not serve any purpose except increasing the nrtPS flow's delay. Hence the relative rates of the rtPS and nrtPS flows must be taken into consideration while choosing the value of α . If rtPS flows at the SS have higher data rates compared to its nrtPS flows, then α can be small. But if the nrtPS flows dominate, then α must be chosen higher so as to keep the real-time traffic delay within bounds.

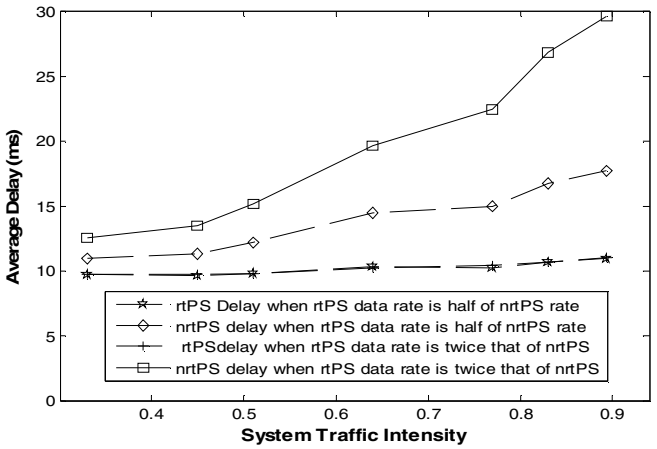

Figure 8. Delay Comparison of rtPS, nrtPS with α =4 for different traffic rates.

V. CONCLUSION

In this paper, we have proposed a new packet scheduling algorithm to be implemented at the MAC layer of IEEE 802.16 wireless broadband PMP networks. The proposed scheduling scheme can provide a better resource allocation among real-time and non real-time traffic streams within the polling service framework. The algorithm provides QoS guarantees taking into consideration of the queue sizes of different traffic and also their latency requirements. It can provide higher bandwidth to real-time traffic to reduce their delays while providing excess resources to non real-time traffic. The performance of the algorithm has been proved to be good enough to meet the QoS requirements of real-time traffic and provide reasonable transmission service to non real-time traffic by extensive simulations.

REFERENCES

[1] IEEE Standard for Local and Metropolitan Area Networks Part 16: Air Interface for Fixed Broadband Wireless Access Systems," IEEE Std. 802.16-2004 (Revision of IEEE Std 802.16-2001), pp. 1-857.

[2] 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," Proc. of IFIP Intl. Conf. Wireless and Optical Communications Networks, 2006.

[3] H. S. Alavi, M. Mojdeh, and N. Yazdani, "A Quality of Service Architecture for IEEE 802.16 Standards," Proc. of Asia-Pacific Conf. Communications, Australia, 2005, pp. 249- 253.

[4] K. Wongthavarawat and A. Ganz, "Packet Scheduling for QoS support in IEEE 802.16 broadband wireless access systems," Intl. J. Communication Systems, vol. 16, Issue 1, Feb 2003, pp. 81-96.

[5] A. Sayenko, O. Alanen, J. Karhula, and T. Hämäläinen, "Ensuring the QoS requirements in 802.16 scheduling," Proc. the 9th ACM Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems, Malaga, 2006, pp.108- 117.

[6] L. Howon, K. Taesoo, C. Dong-Ho, L. Geunhwi, and C. Yong, "Performance Analysis of Scheduling Algorithms for VoIP Services in IEEE 802.16e Systems," Proc. of Vehicular Technology Conf. 2006, pp. 1231- 1235.

[7] D. Niyato and E. Hossain, "Queue-aware uplink bandwidth allocation and rate control for polling service in IEEE 802.16 broadband wireless networks," IEEE Trans. on Mobile Computing, vol. 5, Issue 6, June 2006, pp. 668-679.