Uplink Scheduling with Quality of Service in IEEE 802.16 Networks

Juliana Freitag and Nelson L. S. da Fonseca
Institute of Computing, State University of Campinas
Campinas 13089-971, São Paulo, Brazil
Email: {juliana,nfonseca}@ic.unicamp.br

***Abstract*—In order to support real-time and bandwidth demanding applications the IEEE 802.16 standard is expected to provide Quality of Service (QoS). Although the standard defines a QoS signaling framework and four service levels, scheduling mechanisms for this network are unspecified. In this paper, we propose a scheduling discipline for uplink traffic. Simulation results show that our scheme is capable to provide QoS. Moreover it shares fairly the resources among connections of the same service type.**

***Index Terms*—Wireless Networks, 802.16 Networks, Quality of Service, Scheduling.**

I. Introduction

To support a wide variety of multimedia applications, the IEEE 802.16 standard [1] defines four types of service flows, each with different QoS requirements. Each connection between the SS and the BS is associated to one service flow. The Unsolicited Grant Service (UGS) carries constant bit rate (CBR) flows of CBR-like applications such as Voice over IP. The real-time Polling Service (rtPS) is designed for applications with real-time requirements which generate variable size data packets periodically, such as MPEG video streams. QoS guarantees are given as bounded delay and assurance of minimum bandwidth. The non-real-time Polling Service (nrtPS) is adequate to better-than-best-effort services such as FTP services. Minimum bandwidth guarantees are also provided to nrtPS connections. The Best Effort service (BE) is used for best-effort traffic such as HTTP. More recently, a variation of the 802.16 standard, named 802.16e, was introduced to provide mobility to users. In the 802.16e standard a new service flow called extended real-time Polling Service (ertPS) was added. However, scheduling in 802.16e networks is out of the scope of the present paper.

To cope with diverse QoS requirements a signaling mechanism for information exchange between the base station (BS) and subscriber stations (SSs) was defined. This signaling mechanism allows the SSs to request bandwidth, i.e. transmission time slots, to the BS. Bandwidth allocation is provided on demand. When an SS has backlogged data, it sends a bandwidth request to the BS. The BS, in turn, allocates time slots to the SS based on both the bandwidth request and the QoS requirements of the requesting connection.

This research was sponsored by UOL (www.uol.com.br), through its UOL Bolsa Pesquisa program, process number 20060511022200a and by CNPq, process number 305076/2003-5.

Time slots are organized in frames. Each frame is divided in two parts: the downlink subframe, used by the BS to send data and control information to the SSs, and the uplink subframe, shared by all the SSs for data transmission. The granting of time slots to a connection depends on the service flow of that connection. UGS connections receive fixed size data grants periodically without explicit requests from the SSs. Periodic unicast request opportunities are provided to rtPS connections. They are also given to nrtPS connections, but using more spaced intervals than those given to rtPS connections. The nrtPS and the BE service flows share contention request opportunities.

Although the IEEE 802.16 standard defines the framework for QoS provisioning, the amount of bandwidth given to each service flow is determined by the scheduling mechanism, left to be defined by proprietary implementations. Different scheduling mechanisms [2], [3], [4], [8], [9] have been proposed in the literature, however, not all of them comply with the IEEE 802.16 standard. Moreover, most of them are based on scheduling policies, such as *Weighted Fair Queueing*, which are not so simple to implement.

In this paper, we introduce a BS uplink scheduling algorithm which allocates bandwidth to the SSs based on the QoS requirements of the connections. The proposed policy is fully standard-compliant and it can be easily implemented in the BSs. Results obtained via simulation, using the ns-2 simulator, show that the proposed scheme is able to support QoS guarantees as well as to allocate fairly bandwidth among flows of the same service type.

This paper is organized as follows. Section II presents the proposed uplink scheduler. Section III describes the simulation environment used to test the proposed scheduler. Section IV presents several simulation scenarios. Section V discusses related work. Finally, Section VI concludes the paper.

II. A Scheduling Mechanism

According to the IEEE 802.16 standard, the BS uplink scheduler provides grants (time slots) at periodic intervals to the UGS flows to send data. Periodic grants are also given to rtPS and to nrtPS flows to request bandwidth. Before satisfying bandwidth requests, the uplink scheduler must allocate resources to these periodic grants. Additionally, the uplink scheduler must guarantee that the delay and the bandwidth requirements of rtPS and nrtPS flows are met. The BS executes

1930-529X/07/$25.00 © 2007 IEEE
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2007 proceedings.

the uplink scheduler during each frame, and it broadcasts the schedule in the UL-MAP message in the downlink subframe.

The proposed uplink scheduler uses three queues, referred as low priority queue, intermediate queue and high priority queue. The scheduler serves the requests in strict priority order from the high priority queue to the low priority queue. The low priority queue stores the bandwidth requests of the BE service flow. The intermediate queue holds bandwidth requests sent by rtPS and by nrtPS connections. rtPS and nrtPS requests can migrate to the high priority queue to guarantee that their QoS requirements are met. Besides the requests migrated from the intermediate queue, the high priority queue stores periodic grants and unicast request opportunities that must be scheduled in the following frame. Fig. 1 shows the architecture of the proposed BS uplink scheduler.

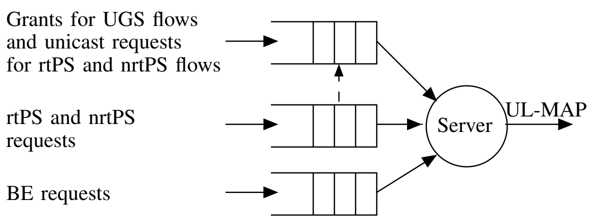

Fig. 1. Architecture of the proposed uplink scheduler

To guarantee bandwidth to UGS flows, data grants are periodically inserted into the high priority queue. The interval between grants to the UGS flows is informed by the SSs at the connection establishment time. Moreover, unicast request grants are also inserted into the high priority queue. Intervals between request opportunities to rtPS and to nrtPS flows are defined by the BS. To guarantee the maximum delay requirement, the BS assigns a deadline to each rtPS bandwidth request in the intermediate queue. Each time the scheduler is executed, the requests with deadline expiring in the frame following the next one migrate to the high priority queue. To calculate the deadline for migration, it would be necessary to know the arriving time of the packets at the SSs queues. Since the BS has no access to this information, the worst possible case is considered, which corresponds to the arrival at the queue immediately after the connection sent the last bandwidth request. Hence, the deadline of a request should be equal to the sum of the arriving time of the last request sent by the connection and its maximum delay requirement.

The scheduler guarantees the minimum bandwidth requirement of both rtPS and nrtPS connections over a window of duration T. Since these windows are non-overlapped and consecutive in the timeline, bandwidth is guaranteed in the long-run. Every time the scheduler is executed, it calculates a priority value to the requests in the intermediate queue considering the per connection minimum bandwidth requirement, amount of backlogged requests (in bytes), and amount of bandwidth received in the current window. The priority assigned to a request is inversely proportional to the amount of bandwidth received by the connection to which it belongs to. Low priority values are assigned to those requests sent by connections which has already received the minimum required bandwidth in the current window.

ALGORITHM *Scheduling*

1. insert, in the high priority queue, the periodic data grants and unicast request opportunities that must be scheduled in the next frame;
2. CheckDeadline;
3. CheckMinimumBandwidth;
4. schedule the requests in the high priority queue starting from the head of the queue;
5. if intermediate queue is empty and available_slots >0 then schedule the requests in the low priority queue starting from the head of the queue;

CheckDeadline:
6. for each request i in the intermediate queue do
7. if service[CID] == rtPS then
8. frame[i] = $\lfloor (deadline[i] - current\_time) \div frame\_duration \rfloor$;
9. if frame[i] == 1 then
10. if available_bytes $\geq$ BR[i]
11. migrate request i to high priority queue;
12. granted_BW[CID] = granted_BW[CID] + BR[i];
13. backlogged[CID] = backlogged[CID] $-$ BR[i];
14. available_bytes = available_bytes $-$ BR[i];

CheckMinimumBandwidth:
15. For each connection CID of type rtPS or nrtPS do
16. backlogged_tmp[CID] = backlogged[CID];
17. granted_BW_tmp[CID] = granted_BW[CID];
18. for each request i in the intermediate queue do
19. if BWmin[CID] $\leq$ granted_BW_tmp[CID] then
20. priority[i] = 0;
21. else
22. priority[i] = backlogged_tmp[CID] $-$ (granted_BW_tmp[CID] $-$ BWmin[CID]);
23. granted_BW_tmp[CID] = granted_BW_tmp[CID] + BR[i];
24. backlogged_tmp[CID] = backlogged_tmp[CID] $-$ BR[i];
25. sort the intermediate queue;
26. For each request i in the intermediate queue do
27. if available_bytes $\geq$ BR[i] then
28. migrate request to the high priority queue;
29. granted_BW[CID] = granted_BW[CID] + BR[i];
30. backlogged[CID] = backlogged[CID] $-$ BR[i];
31. available_bytes = available_bytes $-$ BR[i];

In the Algorithm *Scheduling*, the proposed discipline is presented. After inserting the periodic grants in the high priority queue, the algorithm checks which requests in the intermediate queue should be scheduled in the following frame and the scheduler migrates the requests to the high priority queue (Lines 2 and 3). In Line 4, the scheduler serves the high priority queue. The number of slots needed to serve the requests in the high priority queue is always less or equal to the number of available slots in the uplink subframe. In Line 5, the scheduler allocates bandwidth to the requests in the low priority queue when both the intermediate queue is empty and there are available slots in the uplink frame.

The procedure *CheckDeadline* verifies for each rtPS request whether it should be migrated to the high priority queue or not. The conditions for migration are: the request deadline expires in the frame following the next one, and the amount of bandwidth requested (BR[i]) is less than or equal to the amount of available bytes[1] in the next uplink frame. In case of migration, the algorithm updates the amount of bandwidth allocated to the corresponding connection in the current window (BW_granted[CID] [2]), the amount of backlogged requests sent by the corresponding connection (backlogged[CID]), and the amount of available bytes in the next uplink frame. These variables are used in the following step of the algorithm to guarantee the minimum bandwidth requirement.

The Procedure *CheckMinimumBandwidth* first calculates a priority value for each request in the intermediate queue (lines 15-24). Then, it sorts the intermediate queue according to the priority values (line 25). Finally, while there is available bandwidth, the scheduler migrates the requests to the high priority queue and updates the variables BW_granted[CID], backlogged[CID], and the amount of available bytes in the following uplink frame (lines 26-31).

The proposed mechanism is based on the IEEE 802.16 standard, the service level ertPS (extended real time Polling Service) specified in the IEEE 802.16e standard is not considered. The ertPS service uses a grant mechanism similar to the one used to support UGS connections. The difference is that periodic allocated grants can be used to send bandwidth requests to inform the required grant size. The proposed scheduler can be easily extended to support this service by allocating bandwidth to the ertPS connections in the same way it allocates bandwidth to UGS connections.

III. Simulation Experiments

To assess the effectiveness of the proposed scheme, we carried out discrete event simulation experiments using a specially designed ns-2 module for IEEE 802.16 networks.

The purpose of this study is to explore the scheduler efficacy. Its performance is evaluated considering ideal channel conditions. Extensions of the scheduling mechanism and of the ns-2 module to account for special characteristics of wireless environments are currently under development.

The topology of the simulated network consisted of a BS wire-attached to a fixed node through a 100 Mbps link with a 2 ms delay. The BS was located at the center of a 250x250 meter area, and the SSs were uniformly distributed around it. The frame duration was 5 ms and the capacity of the channel was 40 Mbps, assuming a 1:1 downlink-to-uplink TDD split.

For all simulated scenarios, it is assumed the presence of an admission control mechanism, so that results are not affected by excessive number of connections in the network. In order to prevent that packet scheduling in the SSs interfere with the evaluation of the scheduling mechanism at the BS, each SS has only one traffic flow. We consider four types of traffic: voice, video, FTP and WEB, which are associated to UGS, rtPS, nrtPS, and BE services, respectively.

The voice model is an "on/off" one with duration of periods exponentially distributed with a mean of 1.2 s and 1.8 s for the "on" and for the "off" periods, respectively. During "on" periods, packets of 66 bytes are generated every 20 ms [5]. Video traffic is generated by real MPEG traces [7]. The WEB traffic is modeled by a hybrid Lognormal/Pareto distribution. The body of the distribution corresponding to an area of 0.88 is modeled as a Lognormal distribution with a mean of 7247 bytes, and the tail is modeled as a Pareto distribution with a mean of 10558 bytes [6]. FTP traffic is generated using an exponential distribution with a mean of 512 KBytes.

The interval between data grants for the UGS service is 20 ms since the BS, under optimal conditions, must allocate to the UGS connections data grants at intervals equal to the UGS application packet generation rate [1]. The interval between unicast request opportunities of the rtPS service is 20 ms and the interval of the nrtPS service 1 s.

For rtPS service, the delay requirement is 100 ms and each connection has its own minimum bandwidth requirement which varies according to the mean rate of the transmitted video. The nrtPS service has minimum bandwidth requirement of 200Kbps, and the BE service does not have any QoS requirement.

IV. Numerical Results

This section presents results for four simulation experiments. The purpose of these experiments is to investigate the scheduler capability to provide QoS under different offered load scenarios as well as the capability of providing fair bandwidth sharing among competing flows in the same service level.

For the UGS service, we plotted the delay in order to check whether they receive grants at a fixed interval as required by this service. For the rtPS service, we also analyze the connections delay to check whether the maximum delay requirement is guaranteed or not. The delay is measured as the time interval between the arrival of the packet at the SS queue and its reception at the BS. For the nrtPS service, we plot the connections throughput, since this service should be given minimum bandwidth guarantees. We also plot the throughput for the BE connections to check whether they suffer from starvation.

Each simulation was run ten times with different seeds to generate a 95% confidence interval using the replication method. Figures show the mean obtained value and the 95% confidence interval.

A. Experiment 1

The first experiment verifies whether the BS is able to allocate bandwidth to connections in the same service level in a fair way, regardless the number of connections in the network. The simulated network has one BS and the number of SSs varies between 10 and 30. Each SS has one nrtPS connection that generates FTP traffic with rate of 600 Kbps.

[1]The amount of available bytes in a frame is equal to the amount of available slots multiplied by the number of bytes that can be transmitted in one slot.

[2]CID stands for connection identifier. The IEEE 802.16 standard specifies a 16-bit identifier for each connection.

1930-529X/07/$25.00 © 2007 IEEE
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2007 proceedings.

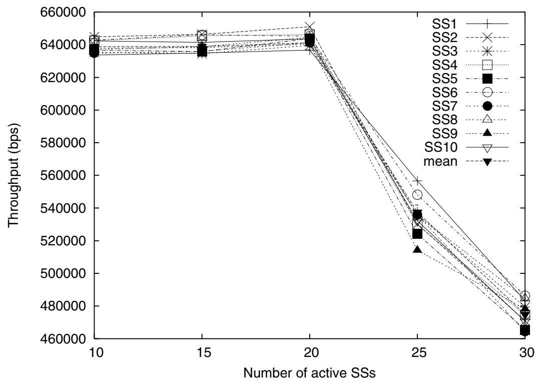

Fig. 2. Throughput of each SS

For better visualization, Fig. 2 shows the throughput of ten SSs. As can be seen, the scheduler at the BS allocates resources fairly among active nrtPS connections. The coefficients of variation of the throughput for the scenarios with 10, 15, and 20 active SSs are 0.01, whereas for the scenarios with 25 and 30 active SSs they are equal to 0.02. The number of slots received at a certain period is proportional to the number of active connections. When there are up to 20 active SSs, the network is not overloaded and the SSs have higher throughput. When the number of active SSs is higher than 20, each connection receives fewer number of slots when compared to the previous situation. When there are 25 SSs in the network, the throughput of SS1 and SS9 differ by 45 Kbps. This small difference is due to the high number of nrtPS connections contending for the shared slots reserved for bandwidth request. Repeated collisions impose extra delays to the transmission of bandwidth requests and, consequently, they reduce the throughput of the SSs in collision.

B. Experiment 2

The aim of experiment 2 is to investigate whether or not the increase of the UGS traffic load degrades the QoS level of services with lower priority. For this purpose, the simulated scenario includes one BS and 81 SSs. In the experiment, there are 6 rtPS connections, 20 nrtPS connections, 20 BE connections, and the number of active UGS connections varies from 15 to 35.

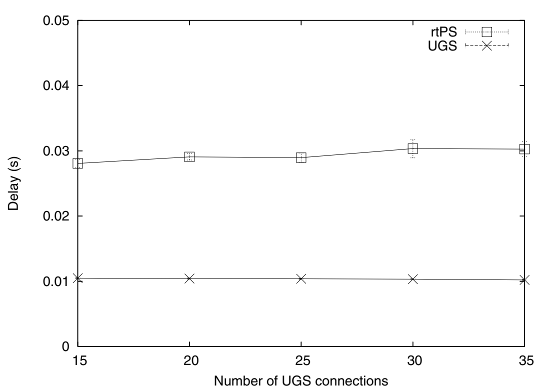

Fig. 3. Delay of UGS and rtPS connections

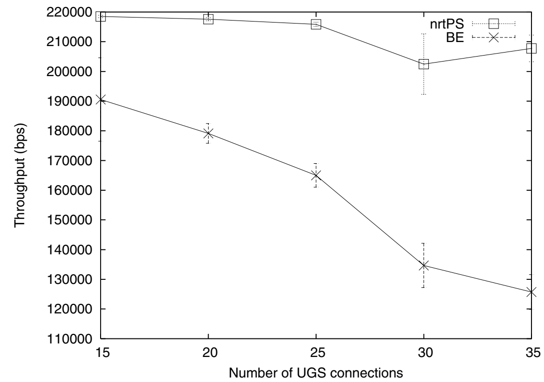

Fig. 4. Throughput of nrtPS and BE connections

Fig. 3 shows the delay of the UGS and of the rtPS connections. The delay of the UGS connections was not affected by the load increase, which shows that the scheduler is able to provide data grants at fixed intervals as required by this service. The delay of the rtPS connections presented very little oscillations as the number of UGS connections increased. Moreover, the delay values were considerable lower than the required one.

As can be seen in Fig. 4, the throughput of the nrtPS connections decreased slightly as the UGS load increased. Nonetheless, all the nrtPS connections had the minimum bandwidth requirement guaranteed. On a different trend, the throughput of the BE connections decreased sharply, which was expected due to the load increase of a higher priority service flow.

C. Experiment 3

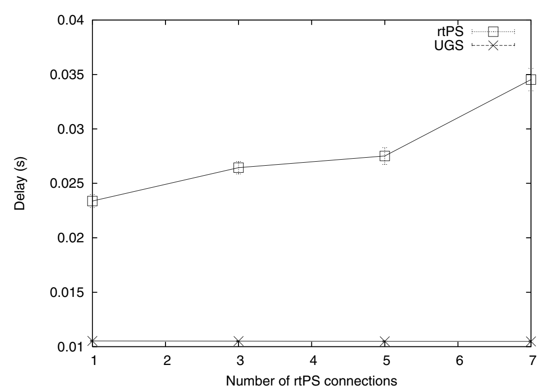

Fig. 5. Delay of UGS and rtPS connections

Experiment 3 verifies the impact of the load increase of the rtPS service on the performance of other service classes. The rtPS service carries video traffic which is quite bursty, and the peak loads must be handled by the scheduler without penalizing the QoS level of the other service classes. As a consequence, the variation of the number of rtPS connections has stronger impact on the offered load and on the traffic dynamics than the variation of the UGS load pursued in the Experiment 2.

1930-529X/07/$25.00 © 2007 IEEE
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2007 proceedings.

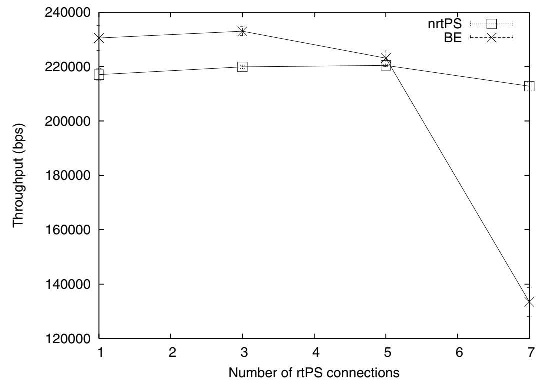

Fig. 6. Throughput of nrtPS and BE connections

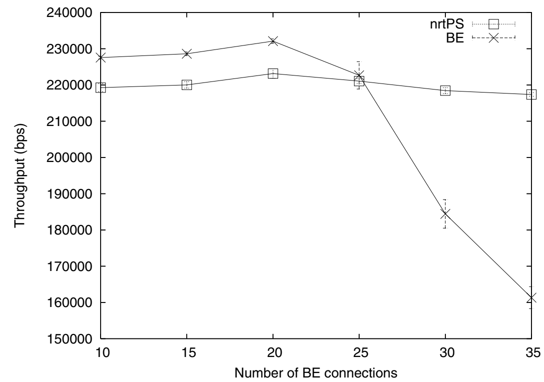

Fig. 8. Throughput of nrtPS and BE connections

The scenario for this experiment consists of one BS and 62 SSs. There are 15 UGS connections, 20 nrtPS connections, 20 BE connections, and the number of active rtPS connections varies from 1 to 7.

Fig. 5 shows the delay of the UGS and of the rtPS connections. Again, the UGS connections had constant delay under all the offered loads. In spite of rtPS load increase, the delay values did not surpass the required one.

Fig. 6 plots the throughput of the nrtPS and of the BE connections. When there are up to 5 active rtPS connections, the BE traffic has higher throughput than the nrtPS connections. This result indicates that if the resources reserved to service flows with higher priority than the BE service flows are not being fully used, they are utilized by the BE traffic, avoiding bandwidth waste. Note that all the nrtPS traffic was served since each nrtPS connection generated around 200 Kbps. When the number of rtPS connections was greater than 5, the resources were used to satisfy the higher priority traffic demands.

D. Experiment 4

In this experiment we verify whether the increase of the BE traffic load influences or not the QoS level of services which has higher priority. The simulated scenario has one BS and 70 SSs. There are 15 UGS connections, 5 rtPS connections, 15 nrtPS connections, and the number of active BE connections varies from 10 to 35.

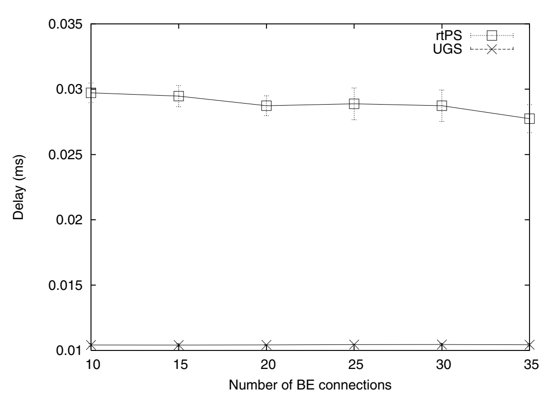

Fig. 7. Delay of UGS and rtPS connections

UGS connections experienced constant delays for all the simulated loads, and the delay of the rtPS connections did not show significant variations (Fig. 7).

Fig. 8 shows the throughput of the nrtPS and of the BE connections. The throughput of the nrtPS connections oscillates very little, and the scheduler was able to provide the required minimum bandwidth. The BE service had higher throughput than the nrtPS service when the number of active BE connections was small, since the scheduler allocates the slots not used by the service flows with higher priority to the BE service given its work-conserving behavior. When the number of active BE connections is high, the throughput decreases since each connection receives fewer slots.

V. Related Work

The scheme proposed in this paper differs from previous work [3], [4], [9] by supporting both minimum bandwidth and maximum delay requirements. These requirements were also considered in [2] and in [8], but at cost of adding an extra module in the BS to compute scheduling deadlines. These works introduce complex scheduling schemes, composed of hierarchies of schedulers, such as Earliest Deadline First (EDF), Deficit Round Robin (DRR), Weighted Fair Queueing (WFQ), and Worst-case Weighted Fair Queueing (W^2FQ). Such hierarchy introduces overhead in scheduling which is pursued on a time scale of a frame. For instance, in OFDM-based systems it is possible to exist 400 frames per second [1], which requires fast schedulers. Therefore, simpler solutions than those are desirable.

Other previous works [10], [11] introduce scheduling policies only for real-time traffic.

The policy introduced here supports the four service types defined by the standard, and it uses a simple approach based on three priority queues, which facilitates operation and low hardware overhead. Moreover, in [3] and in [8] the rtPS flows have absolutely higher priority over nrtPS flows in resource allocation. When the real-time load is high, the non-real-time packets can suffer a long delay, even when the deadline of real-time packets are loose. If the nrtPS services use TCP as the transport layer protocol, the excessively long delay may activates the slow-start procedure and, consequently, degrade the system performance [13]. The proposed scheduler does not

1930-529X/07/$25.00 © 2007 IEEE
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2007 proceedings.

give higher priority to the nrtPS flows unconditionally. Instead, only the real-time requests that are close to the deadline are given high priority.

The schedulers proposed in [2], [4], and [9] can interrupt the transmission of rtPS flows to start the transmission of nrtPS flows even when there are pending requests in the rtPS queue. This imply that whether in the event of deadline expiration or non-existence of available bandwidth, the corresponding packets at the SS can be discarded. The proposed scheduler serves first the rtPS requests whose deadline is close, and, then, it serves the remaining rtPS and nrtPS requests according to the minimum bandwidth requirement of the requesting connections which eliminates the mentioned drawback of the schedulers presented in [2], [4], and [9].

VI. Conclusions

In this paper, an uplink scheduling mechanism for IEEE 802.16 networks was introduced. The proposed solution supports the four service levels specified by the standard and considers their QoS requirements for scheduling decisions. Furthermore, it uses a simple approach based on three priority queues and the scheduling algorithm does not require extensive calculations. In fact, the complexity of the proposed mechanism is $O(k + rlogr)$, where k is number of slots in the uplink subframe, and r is the number of rtPS and nrtPS bandwidth requests in the intermediate queue.

Simulations experiments show the efficacy of the proposed scheme. If there are connections of different service levels in the network, then the scheduler allocates enough slots for each connection so that the QoS requirements are supported. Moreover, the slots not used by the high priority services are shared among the BE connections avoiding bandwidth waste.

References

[1] *IEEE Standard for Local and Metropolitan Area Networks - Part 16: Air Interface for Fixed Broadband Wireless Access Systems*. IEEE Std., Rev. IEEE Std802.16-2004, 2004.

[2] J. Chen, W. Jiao, and H. Wang, *A Service Flow Management Strategy for IEEE 802.16 Broadband Wireless Access Systems in TDD Mode*. In Proceedings of the IEEE International Conference on Communications, pp. 3422-3426, 2005.

[3] G. Chu, D. Wang, and S. Mei, *A QoS Architecture for the MAC Protocol of IEEE 802.16 BWA System*. In Proceedings of the IEEE International Conference on Communications, Circuits and Systems and West Sino Expositions, pp. 435-439, 2002.

[4] M. Hawa and D. W. Petr, *Quality of Service Scheduling in Cable and Broadband Wireless Access Systems*. In Proceedings of the 10th IEEE International Workshop on Quality of Service, pp. 247-255, 2002.

[5] P. Brady, *A model for generating On-Off speech patterns in Two-Way conversations*. Bell System Technical Journal, vol. 48, pp. 2445-2472, 1969.

[6] P. Barford et all, *Changes in web client access patterns: Characteristics and caching implications*. Technical Report 1998-023, Boston University, 1998

[7] P. Seeling, M. Reisslein, and B. Kulapala, *Network Performance Evaluation Using Frame Size and Quality Traces of Single-Layer and Two-Layer Video: A Tutorial*. IEEE Communications Surveys and Tutorials, vol. 6, no. 2, pp. 58-78, 2004.

[8] K. Wongthavarawat and A. Ganz, *IEEE 802.16 based last mile broadband wireless military networks with Quality of Service support*. In Proceedings of the IEEE MILCOM'03, pp. 779-784, 2003.

[9] J. Sun, Y. Yau, and H. Zhu, *Quality of Service Scheduling for 802.16 Broadband Wireless Access Systems*. In Proceedings of the IEEE 63rd Vehicular Technology Conference, pp. 1221-1225.

[10] H. Lee, T. Kwon, and D. Cho, *An Enhanced Uplink Scheduling Algorithm Based on Voice Activity for VoIP Services in IEEE 802.16d/e System*. IEEE Communications Letters, vol. 9, no. 8, pp. 691-693.

[11] O. Yang, and J. Lu, *A new scheduling and CAC scheme for real-time video application in fixed wireless networks*. In Proceedings of the 3rd IEEE CCNC, pp. 303-307.

[12] *The Network Simulator - ns-2*. http://www.isi.edu/nsnam/ns/, 2002.

[13] W. S. Jeon, D. G. Jeong, and B. Kim, *Design of packet transmission scheduler for high speed downlink access system*. IEEE Transactions on Wireless Communication, vol. 3, no. 5, pp. 1789-1801, 2004.