This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the WCNC 2007 proceedings.

How many traffic classes do we need in WiMAX?

Kamal Gakhar, Mounir Achir, and Annie Gravey
GET/ENST Bretagne, Department Of Computer Science,
CS83818, 29238 Brest Cedex 3, France
Email: firstname.lastname@enst-bretagne.fr

***Abstract*— This paper addresses the issue of multiservice support in IEEE 802.16 or WiMAX networks. The capacity for supporting multiple service classes is indeed important for any access technology where bandwidth is limited, which is the case for IEEE 802.16. The standard currently proposes four traffic classes, and specifies that for uplink traffic, the first one (UGS) receives periodic grants whereas the other three are served via polling. Supporting two different scheduling mechanisms may have a significant impact on the complexity of Network Interface Cards, and therefore on the CAPEX for WiMAX networks. Based on this analysis, the present work investigates whether a 802.16 network that only supports the 3 polling based classes is still capable of providing the QoS levels expected for all types of applications. Both the transfer plane QoS, in terms of latency and jitter, and the command plane QoS, in terms of blocking probability are assessed. In particular, a simple, multiservice call admission control (CAC) mechanism is proposed that significantly improves on a previously proposed CAC mechanism by favouring real time traffic over non real time traffic. The outcome of this study shows that it is indeed possible to support stringent QoS with only polling based traffic classes, and fairly simple traffic engineering mechanisms fully described in the paper.**

I. INTRODUCTION

Networks have evolved into mutiservice environments and it is inevitable to differentiate among various types of services unless resources are overprovisioned. For IP networks, the multiservice issue goes back to the early 90's when IntServ [1] was proposed. It was further readdressed by the proposition of DiffServ [2] which defines methods for marking traffic into different traffic classes. However, to minimize technical complexity and propose economically viable solutions, the question of implementing multiservice networks has also been around from the same era. To quote a question from [3]: "Is service priority useful in networks?". Indeed the resources needed to provide multiple differentiation do not come for free: command plane functions are made more complex (e.g. Call Acceptance Control complexity increases with the number of classes) and transfer plane functions are also made more complex, since scheduling should take account of various traffic classes. Thus for the sake of sytem's simplicity, it is desirable to minimize the number of traffic classes that are supported by the system.

In broadband wireless access (BWA) networks the multiservice challenge is even more severe because of classical limitations of radio environment which imposes restrictions on the number of users and the quality of network service that can be attributed to them. The question is, "How could we ensure that technically and economically the broadband wireless networks can be deployed and provide required QoS to the clients?". A more effective use of available bandwidth spectrum via dynamic resource reservation handles one aspect the issue [4]. Another facet is how to handle the given services in a system so as to incur minimum costs. Limiting the number of supported traffic classes seems to go in the right direction, especially for 802.16 networks as we show in this paper.

The present paper addresses the issue of identifying a minimal set of traffic classes for 802.16 or WiMAX networks (we use 802.16 and WiMAX terms interchangably). Section II describes the traffic classes specified in 802.16. The next Section investigates whether the more stringent polling class can support CBR real time traffic with a good QoS. Section IV addresses the issue of multiclass Connection Acceptance Control (CAC) for 802.16 networks; it analyzes the work in [5] and shows that with this CAC, real time traffic suffers from a significantly larger blocking probability than non real time traffic. We then propose an improvement on their work show that our mechanism is much simpler, and improves the efficiency of the CAC by blocking less real traffic connections. Section V concludes the paper.

II. WIMAX SERVICE CLASSIFICATION

How many traffic classes are needed to support economically the various traffic types generated by applications? Since Best Effort (BE) traffic does not have specific QoS requirements, the user generally does not specify a traffic profile for it. It is thus clear that Best Effort (BE) traffic should be handled differently from Committed Bandwidth (CB) traffic for which QoS assurances are requested, and traffic profiles are specified. If it were not the case, the QoS for CB traffic could be degraded by an uncontrolled amount of BE traffic. Furthermore, "real time" traffics which have stringent QoS requirements in terms of delay and jitter should be distinguished from "non real time" traffic which do not have these requirements. If it were not the case, no network policy could ensure that the delay related QoS parameters for "real time" traffics would be respected in the presence of "non real time" traffics (the alternative would be to offer "real time" QoS to all CB traffic which does not seem realistic).

WiMAX is a BWA connection oriented technology that has been adopted from IEEE 802.16 standard. The downlink is operated by broadcast and the uplink by Time Division Multiplexing (TDM). The dynamic TDM for the uplink traffic is controlled by the Base Station (BS) that allocates grants to the Subscriber Stations (SS) on the basis of their requirements. The superframe in 802.16 allows uplink and downlink demands to be fulfilled. Traffic handling on the uplink is controlled by BS which allocates grants to each SS, according to some policy and the SSs in the system operate in TDM mode. The 802.16 currently proposes 4 traffic classes, and specifies that for uplink traffic, the first one (UGS) is intended for CBR traffic. This class receives periodic grants whereas the other three classes are served via polling. These 3 classes are rtPS (real time polling service), nrtPS (non real time polling service) and BE. All classes except BE are intended for CB traffic; both UGS and rtPS are intended to support real-time services with stringent delay requirements.

As we can refer, the 802.16 standard proposes that two different types of uplink traffic scheduling can be implemented (periodic and polling based). Supporting two different scheduling mechanisms may have a significant impact on the complexity of Network Interface Cards, and therefore on the capital expenditure (CAPEX) for Wimax networks. Based on this analysis, the present work investigates whether a 802.16 network that only supports the 3 polling based classes is still capable of providing the QoS levels expected for all types of applications. Our interest here lies in identifying policies ensuring that different traffic classes receive the required QoS.

However, 802.16 standard specifies neither the BS scheduling algorithm (called here “Dynamic Bandwidth Assignment”, or DBA) nor the scheduling policy used by the SS. This allows vendors to differentiate their offers. Furthermore, as usual, Connection Access Control (CAC) policy is not mandated by the standard, but is optional, leaving the operators to implement their own policies. In order to address the problem of the optimal number of traffic classes, we had to design scheduling policies to be analyzed and to identify an efficient CAC. The study of the transfer phase is proposed in Section III, and the definition of a CAC is addressed in Section IV.

III. QoS DELIVERED IN THE TRANSFER PHASE

In this Section, we address whether in a non-congested system the stringent requirements set by the 802.16 standard for UGS traffic can be met by the rtPS traffic class. We assume that the system is not congested for UGS, rtPS, and nrtPS traffic due to the CAC implementation (see Section IV). We focus our analysis on the uplink transfer since the downlink is controlled directly by the BS which can enforce a central scheduling policy. The behaviour of uplink transfer is controlled by grant allocation mechanisms and it is obviously very complex. We have, therefore, chosen to perform this study by simulation in order to handle realistic traffic types.

A. Packet scheduling on the uplink in WiMAX networks

We propose here both a simple DBA and a scheduling policy to be implemented in the SS.

As explained above, the DBA has to share the transmission opportunities in each superframe among all SS. It should not starve any SS while favouring CB traffic over BE traffic (*It is not about starving a traffic class over another but a SS over another*). The simple DBA shown in fig. 1 ensures that UGS traffic always gets its grants, that rtPS traffic is served before nrtPS traffic, and that BE traffic is served only if the CB traffic does not need the grants. Obviously, it may happen that in some superframes, there are not enough resources to satisfy all demands. If the unsatisfied demands are only from BE connections, this is not an issue. On the other hand, if unsatisfied demands are from rtPS or nrtPS traffic, this has an impact on committed QoS and the CAC should ensure that this degradation is acceptable. We address this particular point in Section IV.

- `Step 1 : UGS grants are prenegotiated and are fixed within a superframe before the transmission starts.`
- `Step 2 : Then the BS allocates grants to the rtPS traffic in the following manner: if the sum of all demands for rtPS is smaller than available allocation then all demands are satisfied otherwise the BS serves the demands as per their proportion to the total demand.`
- `Step 3 : If there are remaining grants, the BS then handles the nrtPS demands in the same manner.`
- `Step 4 : BE traffic gets grants only if there are remaining grants after serving the nrtPS demands.`

Fig. 1. Basic DBA Algorithm

The QoS delivered to traffic flows is also impacted by the scheduling policy implemented in the SS. To address this issue, we have tried to identify a policy as simple as possible. Within a class, the packets are served by FIFO (first-in, first-out) policy. Now, regarding UGS traffic, we assume here that the UGS traffic uses its grants, and that grants unused by UGS traffic are lost (this is compliant with the 802.16). UGS traffic that did not have grants wait in the SS till the next superframe. Regarding the three polling based classes, we assume that a simple priority based scheduling is implemented: rtPS traffic is served first, then nrtPS traffic and lastly BE traffic if there is no more rtPS or nrtPS traffic to be served.

B. Simulating IEEE802.16/WiMAX transfer phase

We briefly describe here the modifications made to discrete event simulator ns-2 [6], [7] for our work. At the time of first version of this paper, to the best of our knowledge, there was no detailed published implementation of MAC layer of IEEE 802.16 for ns-2. Thus we needed to include a new procedure for simulating the TDMA mechanisms of 802.16 MAC layer.

We do not simulate physical layer conditions. We also assume that there are no losses in the system due to wireless conditions. We then create a new buffer management technique, modifying the **DropTail** implementation in ns-2, which we call ***BlocQ***. This allows the simulation of packet based TDM, either static or dynamic, in the upstream direction. ***BlocQ*** allows to modify buffer implementation of packets in a queue at run time while keeping the original functionality of DropTail. For the sake of simplicity, we have chosen not to explicitly simulate control (polling and grant) messages. Instead,

TABLE I
TRAFFIC CATEGORIES AND PARAMETERS

Super Frame	10 ms
Uplink Time	50% of Super Frame
Channel	40 Mbps
UGS	VoIP, Poisson Traffic on UDP, packetSize - 240 bytes, rate - 80 Kbps, 10% of system traffic
rtPS	VoIP and Video, Exp-On/Off Traffic on UDP, packetSize - 240 bytes, mean rate - 390 Kbps, 40% of system traffic
nrtPS	large data files (Exp-On/Off Traffic on TCP - packetSize - 512 bytes) and streaming download (Exp-On/Off Traffic on TCP - packetSize - 240 bytes) - about 40% of system traffic
BE	FTP over TCP

we have implemented two procedures, one for evaluating the requirements of SSs by counting the number of buffered bytes, the second for implementing the procedure described in fig. 1. Therefore, the transmission windows in the next polling cycle for each SS are computed using the requirements evaluated by measuring the buffer size of each traffic class, and a "priority based" DBA.

C. Delay related QoS offered to UGS and rtPS traffic

We analyse here whether it is possible to treat UGS and rtPS services together. Tab. I shows the basic parameters used in the simulation studies. We consider two SSs that are polled each at the end of their respective allocations. We assume that each SS sends various types of traffics. We have chosen to represent UGS traffic by a Poisson process, which models the superposition of several CBR connections (UGS is specified for CBR traffic). Both rtPS and nrtPS traffic are modelled as on-off traffic with exponentially distributed on and off periods (in other simulations, not shown here, we have also considered Pareto distributed periods for nrtPS traffic, with no obvious impact). In terms of traffic load, few hundred thousands of packets were simulated with confidence intervals observed within limits.

Fig. 2 shows the histogram of the delays for UGS and rtPS traffic types in the system (with rtPS traffic having higher peaks), whose values differ significantly. The latency can be modelled as the mean delay, and the jitter as the difference between an upper and a lower quantile of the delay. For real time traffic, it is usually necessary to implement a so-called "**playout buffer**" in order to deliver correctly the successive packets to a codec. The network induced delay can then be approximated by what is called here "effective delay" i.e. the sum of the latency and the jitter as defined above.

We first observe that the minimum delay for UGS packets is the radio delay (which is negligible). This is because the allocations are periodically given to UGS traffic, and that some packets get served immediatly in the superframe. We see that the majority of UGS traffic gets served in 1 superframe with the maximum delays goes up to about 2 superframes. For rtPS

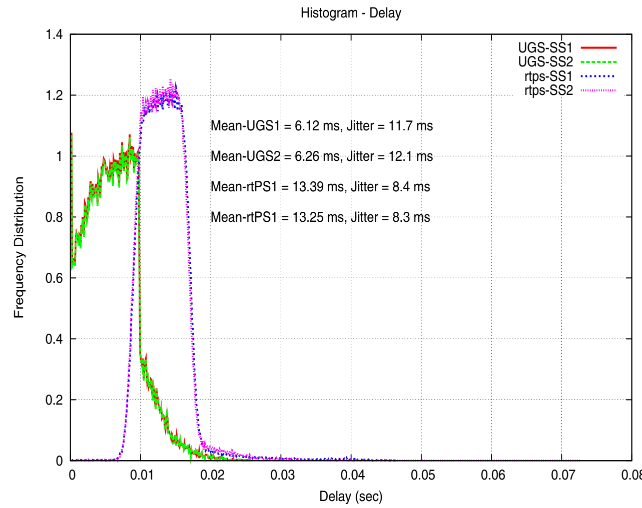

Fig. 2. Delay distribution of UGS and rtPS traffic

class, however, the packets should wait at least more than half of a superframe size. The majority of rtPS packets have to wait between 1 to 2 superframes before they are served. The jitter for rtPS traffic is smaller than for UGS traffic although the rtPS traffic is more bursty than the UGS traffic because the polling mechanism ensures that resources are allocated when requested, and not periodically.

An interesting observation is that the effective delay for the two classes only differ by a few milliseconds. This fact encouraged us to check whether the rtPS traffic class could offer UGS traffic a QoS similar to the one offered by the UGS class.

Fig. 3 shows the histograms of the delays for both UGS traffic and rtPS traffic supported by the single rtPS class (with rtPS traffic having higher peaks). Other traffics share the network but are of no interest in this analysis and thus are not shown here. In this case there is no prenegotiated allocation for UGS traffic in the system. The allocations for all service types are done as dicussed in the algorithm (see fig 1) except $Step1$). It is easily seen that the delay distributions are very similar; moreover, the latency and jitter performance for all traffics is the same as compared to the one for rtPS traffic when the UGS class is used. In this case, the rtPS class is able to efficiently use the resources that were previously allocated to the UGS class.

This shows that UGS traffic can indeed be supported by the rtPS traffic class, which means that, from a transfer performance point of view, it is not necessary to implement specific UGS mechanisms.

IV. CALL ADMISSION CONTROL

In this section we first present a short state of the art on call admission control (CAC) mechanisms for 802.16 networks, followed by the analysis of the CAC proposed in [5] which is

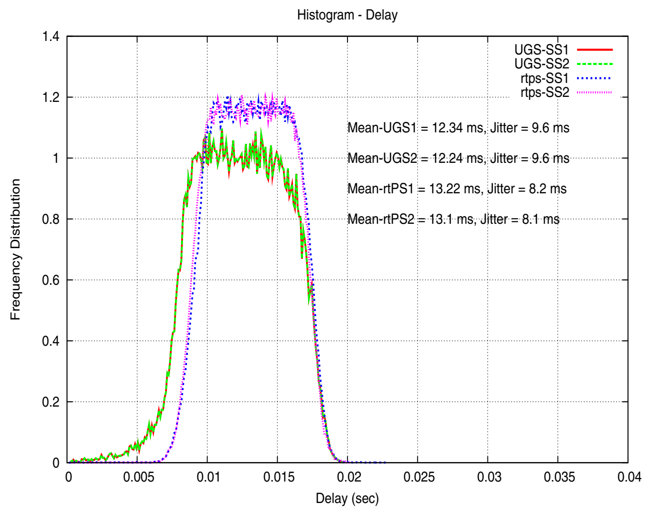

Fig. 3. Delay distribution of UGS and rtPS traffic as Only **rtPS** Class

the basis of our own proposition presented at the end of the section. Moreover, there have been relatively few papers on CAC for 802.16 networks.

The 802.16 networks have two characteristics which are seldom associated: the transfer mode is connection oriented but relies on polling to implement multiplexing in the uplink (see Section III). This is why a straightforward application of existing multiclass CACs may not be appropriate.

The work in reference [8] analytically models a scheme which provides highest priority to UGS traffic, and maximizes the bandwidth utilization by bandwidth borrowing and degradation modeling. Non-UGS requests are accepted only if the bandwidth is still available. Different traffic types are modelled by Poisson traffic. The QoS architecture considered in [9] proposed probability based analytical models for uplink bandwidth allocation scheduling and channel utilization. Once again, only Poisson traffic is considered. The work in [10] studies an algorithm for adaptive frame partitioning for data and voice flows but assumes a single flow per SS. The CAC policy in [11] follows simple criterion of accepting a connection if there is bandwidth left. Though a differentiation is made for given number of connections of a given class, the only segregation is between BE and "the rest" of traffic type. It proposes Deficit Fair Priority Queue (DFPQ) in order to make sure that BE traffic gets served. The approach of having dynamic CAC to address packet-level QoS while having constraints on connection-level QoS has been considered in [12] but it did not include traffic differentiation when applying CAC. The above references either use simplistic assumptions regarding traffic types, or do not support all specified 802.16 traffic classes. However, the work presented in [5] proposes a multiclass CAC, that explicitly takes into account the traffic profiles for UGS, rtPS and nrtPS as specified by the 802.16 standard. This CAC is analyzed in the following Section.

A. Performance of a multiclass CAC for 802.16 networks

Let us first define the terminology used in the remainder of the paper:

- f : superframe duration
- $m_i * f$: maximum delay requirement for connection i
- C_{up} : total capacity allocated for uplink transmission
- C_{UGS} : current capacity allocated to UGS connections
- C_{rtPS} : current capacity allocated to rtPS connections
- C_{nrtPS} : current capacity allocated to nrtPS connections
- r_i : leaky bucket rate for connection i
- b_i : leaky bucket size of connection i

The admission or the rejection of a new connection in [5] is conditioned by the bandwidth availability and the maximum delay requirement for rtPS connections. The admission control policy for a new connection is given by the following conditions:

- **nrtPS connection**
  $r_i < C_{up} - C_{UGS} - C_{rtPS} - C_{nrtPS}$
- **UGS connection**
  $r_i < C_{up} - C_{UGS} - C_{rtPS} - C_{nrtPS}$
  $\forall\, k,\, b_k + f * r_k < m_k * f * (r_k / C_{rtPS}) * (C_{up} - C_{UGS} - r_i)$
- **rtPS connection**
  $r_i < C_{up} - C_{UGS} - C_{rtPS} - C_{nrtPS}$
  $\forall\, k,\, b_k + f * r_k < m_k * f * (r_k / (C_{rtPS} + r_i)) * (C_{up} - C_{UGS})$
  $b_i + f * r_i < m_i * f * (r_i / C_{rtPS} + r_i) * (C_{up} - C_{UGS})$

For all traffic types, the first condition just checks bandwidth availability. Since it is assumed that nrtPS traffic has less priority than UGS and rtPS, this condition is sufficient for a new nrtPS connection.

The second condition (the "delay" condition) for a UGS connection is about checking that the maximum delay condition shall still be respected for each currently active *rtPS* connection if the new UGS connection is accepted. Indeed, the 802.16 standard states that the network should be able to guarantee a maximum delay for each rtPS connection. We have similar conditions for a new rtPS connection.

For a given k, the delay condition ensures that every bit arriving in a given superframe can be transmitted in less than m_i frames. Indeed, the maximum bits received in one superframe is limited by $b_i + r_i * f$ according to the leaky bucket specification; moreover, the rate factor takes into account the fact than UGS has higher priority than rtPS, and that rtPS has higher priority than nrtPS, which is translated into the scheduling policy as shown in Section III.

We note upfront that accepting a new UGS or rtPS connection involves as many tests as the number of rtPS connections already active. This is obviously less than practical since this number may be arbitrarily large. We address this particular point in the following paragraphs by proposing a global condition to check maximum delay performance.

Let us first assess the behaviour of the CAC proposed in [5]. In this study, we compute the blocking probabilities versus

TABLE II
SIMULATION PARAMETER VALUES

Parameters	Values
C_{uplink}	67 $Mbps$
r_i	0 - 2 $Mbps$
f	1 ms
m_{min}	10 ms
m_{max}	50 ms
b_i	0 - 30000 $bits$
ρ	$30 - 70$ $erlang$
ρ_{UGS}	$\rho/3$
ρ_{rtPS}	$\rho/3$
ρ_{nrtPS}	$\rho/3$

offered load ρ and versus the traffic profile parameters (rate r_i and bucket size b_i) respectively. This study is performed by simulation : new connections appear according to a Poisson Process and their duration is exponentially distributed. We also assume, rather arbitrarily, that bandwidth demands are identical for all traffic classes although the leaky bucket size is used in the CAC only for rtPS traffic. Tab. II presents the simulation parameters. The tolerated delay of an rtPS connection is taken randomly between m_{min} and m_{max}. As in [5], we assume that the frame duration is $1ms$ whereas we had previously considered a $10ms$ frame duration. This is due to the evolution of the 802.16 standard, and should not impact the qualitative findings of our study.

The following figures illustrate the impact of traffic parameters on the blocking probability. Fig. 4 shows the blocking probabilities versus the leaky bucket size (b) for $r = 1Mbit/s$. The figure shows clearly the influence of the burstiness (represented by b) : when b increases, the blocking probabilities for UGS and rtPS traffic sharply increases while the blocking probability for nrtPS traffic decreases. This should obviously be avoided: indeed, although the conditions for higher priority traffic may be more stringent, it is not desirable that low priority traffic starves high priority traffic!

Fig. 5 now shows the blocking probability as a function of the offered traffic load (ρ). We note that the blocking probability for rtPS traffic is a bit higher than the one for UGS, and is of several orders of magnitude larger than the one for nrtPS traffic. This is a bad behaviour as it indicates that nrtPS traffic is affecting both UGS and rtPS traffic in the network. This observation led us to propose a novel call admission control mechanism that permits a better sharing of resources between traffic classes.

B. A novel CAC for 802.16 networks

We propose here a modified version of the previous CAC. Our approach modifies the second condition of traffic acceptance compared to what has been discussed for UGS and rtPS traffic in [5]. Our first objective is to substitute a global condition to the set of individual conditions, one for each rtPS connection. Our second objective is to limit the unfairness of the CAC regarding UGS and rtPS traffic.

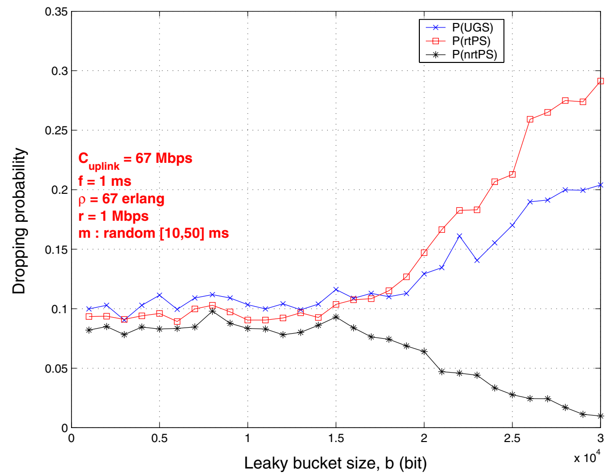

Fig. 4. blocking probability versus leaky bucket size b.

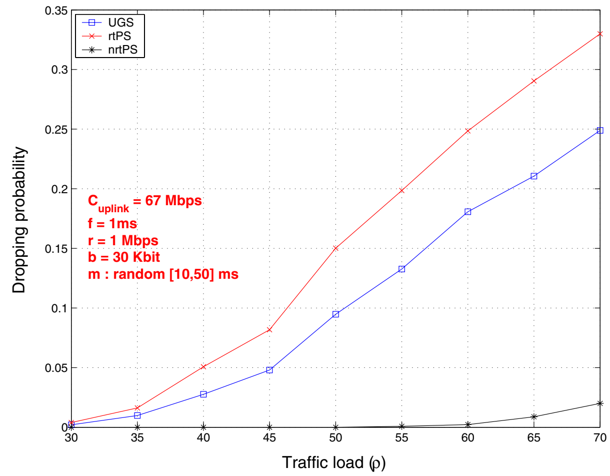

Fig. 5. The blocking probability versus the traffic load ρ.

The following equations define the new admission control conditions for each traffic class.

- **nrtPS**
  $r_i < C_{up} - C_{UGS} - C_{rtPS} - C_{nrtPS}$
- **UGS**
  $r_i < C_{up} - C_{UGS} - C_{rtPS} - C_{nrtPS}$
  $\frac{\sum_k (b_k + f.r_{k,rtPS}) + \sum_k (f.r_{k,UGS}) + f.r_{i,UGS}}{C_{uplink}} < f.min(m_k)$
- **rtPS**
  $r_i < C_{up} - C_{UGS} - C_{rtPS} - C_{nrtPS}$
  $\frac{b_i + f.r_{i,rtPS} + \sum_k (b_k + f.r_{k,rtPS}) + \sum_k (f.r_{k,UGS})}{C_{uplink}} < f.min(m_k)$

The delay condition for a UGS connection consists in stating that all the traffic that arrives during a given frame (which is given by $\sum_k (b_k + f.r_{k,rtPS}) + \sum_k (f.r_{k,UGS}) + f.r_{i,UGS}$) should experience a delay which is smaller than the most stringent bound $f.min(m_k)$ currently negotiated for rtPS connections. This is done by assuming that UGS and rtPS connections share the total available bandwidth C_{up} if requested, nrtPS traffic being served with less priority. The

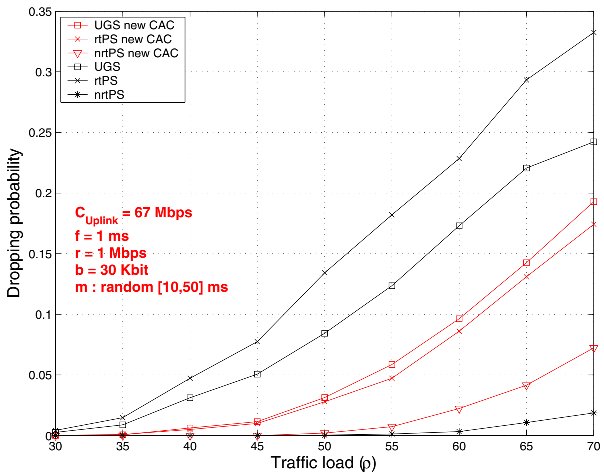

Fig. 6. The blocking probability versus the traffic load ρ (New CAC).

second equation for a new rtPS connection is similar. This means that the CAC does not take into account that UGS traffic has higher priority than rtPS traffic, leaving the scheduling to the DBA and the intra-SS scheduling mechanism. As desired, a number of individual conditions are thus replaced by a single one, which fulfils our first objective.

Fig. 6 should be compared with Fig. 5 as it shows the same simulation scenario for the novel CAC. The results shown there are very interesting : first of all, we see that the blocking probabilities for UGS and rtPS traffics are significantly better with this new CAC, while the blocking probability for nrtPS traffic increases. This fulfills our second objective. Our CAC policy does not consider implicit scheduling like the work in [5]. We consider a minimum delay for rtPS traffic which results into "delay" condition for all traffic types which means that eventually we can schedule different traffic classes as we would like.

Another important feature should also be noted: since the delay conditions for UGS and rtPS traffic are now very similar, the blocking probability performance for the two traffic classes are now very close. This implies that supporting UGS traffic within the rtPS class is not a problem and shall not impact on the blocking performance for either rtPS or UGS traffic: a CBR traffic has a low bucket size, which means that its impact on the delay condition is going to be negligible.

V. Conclusion

In this work we have discussed the multiservice nature of a WiMAX network. We have based our study on the four traffic classes as specified in the 802.16 standard.

We have shown that neither transfer plane QoS delay parameters (see Section III) nor command plane blocking parameters (see Section IV) are greatly impacted by supporting UGS traffic within the rtPS traffic class. This leads to our proposal of offering only three traffic classes instead of four.

This may have a significant impact on the cost of Network Interface Cards for network elements in WiMAX networks since dealing with two separate scheduling mechanisms (periodic for UGS and polling based for the other classes) greatly increases the complexity of the design.

The paper also proposes very simple mechanisms, both in the command plane (CAC) and in the transfer plane (DBA and intra-SS scheduling) and presents preliminary results showing that that these mechanisms can indeed provide WiMAX network with a robust multiclass support. This implies that packet scheduling can have a very small impact on the cost of WiMAX cards. Furthermore, we have shown that a simple multiservice CAC can support real time, CB data service and BE traffic with little complexity. It is up to the network operator to specify traffic profiles that ensure a good utilization of the link.

References

[1] R. Branden, D. Clark, and S. Shenker, "Integrated Services in the Internet Architecture: an Overview," RFC 1633, Jun 1994.

[2] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An Architecture for Differentiated Services," RFC 2475, Dec 1998.

[3] S. Bajaj, L. Breslau, and S. Shenker, "Is Service Priority Useful in Networks?" in *ACM SIGMETRICS 98*, 1998, pp. 66–77.

[4] K. Gakhar, M. Achir, and A. Gravey, "Dynamic Resource Reservation in IEEE 802.16 Broadband Wireless Networks," in *Fourteenth IEEE International Workshop on Quality of Service (IWQoS 2006)*, Jun 2006, pp. 141–148.

[5] K. Wongthavarawat and A. Ganz, "Packet Scheduling for QoS support in IEEE 802.16 Broadband Wireless Access Systems," *International Journal of Communication Systems*, vol. 16, no. 1, pp. 81–96, 2003.

[6] E. Altman and T. Jiménez, *NS simulator for beginners*, Universidad de Los Andes, Mérida, 2003.

[7] K. Fall and K. Varadhan, *The ns Manual*, Aug 2006.

[8] H. Wang, W. Li, and D. P. Agrawal, "Dynamic Admission Control and QoS for 802.16 Wireless MAN," in *Wireless Telecommunications Symposium*, 2005, pp. 60–66.

[9] D.-H. Cho, J.-H. Song, M.-S. Kim, and K.-J. Han, "Performance Analysis of the IEEE 802.16 Wireless Metropolitan Area Network," in *First International Conference on Distributed Frameworks for Multimedia Applications*, 2005.

[10] O. Gusak, N. Oliver, and K. Sohraby, "Performance Evaluation of the 802.16 Medium Access Control Layer," *Lecture Notes in Computer Science - Proceedings of ISCIS*, vol. 3280, pp. 228–237, 2004.

[11] J. Chen, W. Jiao, and H. Wang, "A Service Flow Management Strategy for IEEE 802.16 Broadband Wireless Access Systems in TDD mode," in *IEEE International Conference on Communications*, vol. 5, May 2005, pp. 3422–3426.

[12] L. Huang and C. C. J. Kuo, "Dynamic Call Admisson Control with Joint Connection-level and Packet-level QoS Support in Wireless Multimedia Networks," in *IEEE Symposium on Real-Time and Embedded Technologies and Applications*, May 2004.