mCoSS: a multi-Constraints Scheduling Strategy for WiMAX Networks

Ikbal Chammakhi Msadaa
EURECOM
Sophia-Antipolis, France
msadaa@eurecom.fr

Fethi Filali
QU Wireless Innovations Center
Doha, Qatar
filali@quwic.com

Daniel Câmara
EURECOM
Sophia-Antipolis, France
camara@eurecom.fr

ABSTRACT

In this paper, we attempt to assemble the different pieces of the resource allocation puzzle of mobile WiMAX networks by addressing the main scheduling issues that are still open. We thus propose a novel multi-Constraints Scheduling Strategy (mCoSS) which maximizes the quality of service (QoS) degree of satisfaction for both real-time and non-real-time traffic in terms of delay and throughput. In the scheduling strategy presented in this paper, the access to the network is regulated via a traffic shaper which is inspired from the dual token bucket shaping mechanism. This technique allows traffic burstiness while bounding it. The modified dual token bucket mechanism is combined with a two-rounds scheduling algorithm reflecting the upper and lower bounds of service to be expected by each connection. The bandwidth request and grant policy adopted in these algorithms takes advantage of the different mechanisms proposed by the IEEE 802.16e standard. It adapts the choice of the appropriate technique to the service flow QoS constraints and the current availability of radio resources. Other concerns such as supporting the link adaptation capability and avoiding starvation of best effort traffic are also addressed in this solution. The performance of the proposed strategy is evaluated through simulation and compared to other scheduling solutions proposed in the literature. The obtained results show a nice tradeoff between fairness and efficiency with a high respect for the connections' QoS requirements.

Categories and Subject Descriptors

C.2 [**Computer Systems Organization**]: COMPUTER-COMMUNICATION NETWORKS; C.2.1 [**COMPUTER-COMMUNICATION NETWORKS**]: Network Architecture and Design—*Wireless communication*

General Terms

Algorithms, Performance

Keywords

mobile WiMAX, 802.16e, scheduling, shaping, dual token bucket, algorithms, QoS

1. INTRODUCTION

Over the past two decades, our daily lives have been reshaped by the fast development in the telecommunications environment. Broadband Internet and wireless ubiquity have become more than ever real needs in our modern lifestyle. Driven by this growing demand for high-speed broadband wireless services, Worldwide Interoperability for Microwave Access (WiMAX) technology has been developed. WiMAX is based on IEEE 802.16 standards and is the only mobile broadband technology currently in use. The technology is designed to support heterogeneous classes of services including data, voice and video. However, the IEEE 802.16 standard leaves unstandardized the resource management and scheduling mechanisms which are crucial components to guarantee QoS performance for these applications. In this paper, we tackle this problem and propose a multi-Constraints Scheduling Strategy (mCoSS) which maximizes the quality of service (QoS) degree of satisfaction for both real-time and non-real-time traffic in terms of delay and throughput. The proposed strategy is based on two main components (i) a traffic shaper that is inspired from the dual token bucket shaping mechanism. This mechanism allows traffic burstiness while protecting contract-conforming connections from misbehaving ones and (ii) a two-rounds scheduling algorithm which accommodates the needs of the different categories of applications.

The remainder of this paper is organized as follows. In Section 2, an overview of QoS support in IEEE 802.16 networks is given. Section 3 presents the related work. Section 4 explains the idea of the modified dual token bucket traffic shaping mechanism adopted in our strategy. In Section 5, we provide the details of the proposed two-rounds scheduling approach used by the mobile station (MS) and base station (BS) for DL and UL. The performance evaluation of mCoSS is given in Section 6 after describing the OFDMA-based WiMAX simulation model provided by QualNet. Section 7 concludes the paper by summarizing the main features supported by mCoSS and pointing out the main obtained results.

2. QOS SUPPORT IN WIMAX NETWORKS

The IEEE 802.16 standard defines a connection-oriented MAC protocol that is designed to accommodate a variety of

Q2SWinet'11, October 31–November 4, 2011, Miami, Florida, USA.
Copyright 2011 ACM 978-1-4503-0901-1/11/10 ...$10.00.

applications with different QoS requirements. Depending on the service to be tailored to each user application, a scheduling service is attributed to handle the flow. Based on that, a specific set of QoS parameters should be specified when creating a new service flow (as it is shown in Table 1). Uplink flows however are associated, in addition to a scheduling service, to one of these request/grant scheduling types: unsolicited grant service (UGS), real-time polling service (rtPS), extended real-time polling service (ertPS)—introduced by the IEEE 802.16e-2005 standard [1], non-real-time polling service (nrtPS), and best effort (BE). Each scheduling service is designed to meet the QoS requirements of a specific applications category. Except for UGS connections that receive the bandwidth in an unsolicited manner, the MS needs to inform the BS of its uplink requirements. To do so, a set of mechanisms such as polling, piggybacking, and bandwidth stealing is proposed by the IEEE 802.16 standard. It is worth mentioning that, whatever is the bandwidth request mechanism in use, bandwidth is always requested by an SS on a per-connection basis and addressed by the BS to the subscriber station (SS) as an aggregate of grants. Therefore, since the SS receives the allocated bandwidth as a whole in response to per-connection requests, it cannot know which request is honored. The SS can then use the grant either to send data, or to request bandwidth for any of its connections (bandwidth stealing), or even to send management messages.

	Type of stream				
	real-time CBR	real-time VBR		delay-tolerant VBR	BE
e.g.	VoIP without VAD	VoIP with VAD	MPEG	FTP	Web
R^i_{max}	■	■	■	■	■
R^i_{min}	■	■	■	■	
L^i_{max}	■	■	■		
J^i	■	■			
$Prio$				■	■
UI^i_{gr} (UL)	■				
UI^i_{poll} (UL)		■			
$Service$ $Type$ (UL)	UGS	ertPS	rtPS	nrtPS	BE

Table 1: DL and UL service flows QoS parameters

3. RELATED WORK

The authors have surveyed in [2] scheduling solutions for WiMAX networks. This survey has shown two main tendencies for packet queuing-derived strategies: hierarchical scheduling structures and one-layer scheduling structures. We believe that, given the complexity of the scheduling decisions to be taken by BS (e.g. inter-SSs, inter- and intra-Service types) and SS schedulers, only a hierarchical structure can answer this problem. Through this same survey, it has been observed that in several works [3, 4, 5, 6], Fixed Priority is used as the inter-service types queuing strategy. However, while being simple, it is known that this policy, if not combined with a shaping mechanism, could lead to starvation of low priority service types (i.e. nrtPS and BE). Moreover, most of the hierarchical scheduling strategies proposed in the literature (e.g. [3, 7, 8]) propose a specific queuing discipline for each scheduling service type, which increases significantly the complexity of the proposed solution.

4. A MODIFIED DUAL-BUCKET SHAPING MECHANISM

In order to provide QoS for different types of flows, it is important to implement a traffic shaping mechanism to control the volume of traffic entering the network and to isolate well-behaving traffics from misbehaving ones. The two main traffic shapers implementations used in traffic engineering are: the leaky bucket and the token bucket. The leaky bucket provides a mechanism by which a flow is shaped to be sent to the network at a constant rate. The token bucket however, while providing rate control, allows the traffic to burst up to a configurable threshold. In order to accommodate the bursty characteristics of some categories of applications targeted by WiMAX, we choose the latter mechanism to model our traffic shaper. More specifically, we use the multiple-buckets variant of the token-bucket implementation. We associate each flow i with two buckets corresponding to the minimum reserved traffic rate R^i_{min}, and to the maximum sustained traffic rate R^i_{max}. These per-flow dual buckets reflect the lower and upper boundaries of the service to be provided for each flow. Each bucket has three components: a burst size, a mean rate and a time interval. Figure 1 represents the dual bucket structure associated to a service flow. The first bucket is characterized by:

- a mean rate, also called committed information rate (CIR), which specifies the amount of data that can be sent per time unit on average.
- a time interval T_c, also called the measurement interval; it specifies the time quantum in second per burst.
- a burst size, also called committed burst size (B_c); it corresponds to how much traffic can be sent per burst within a given measurement interval.

The three parameters are linked as follows: $CIR = \frac{B_c}{T_c}$. We set CIR to the minimum reserved traffic rate R^i_{min}, and T_c to a grant interval I^i_{gr} characterizing the i flow. For a real-time traffic i, this parameter corresponds to the maximum latency L^i_{max}. For non-real time flows, this parameter should not exceed the polling interval (for nrtPS) and might be set to a value that is a function of the mean transmitting interval of the flow. The introduction of this parameter is needed first to define the frequency of the allocations for each flow and because the standard does not specify the interval over which R^i_{min} and R^i_{max} are averaged. This first bucket reflects basically the service level agreement (SLA) a WiMAX system is committed to provide for a flow. Note that a BS or SS does not have to meet the latency service commitment (L_{max}) for service flows that exceed their minimum reserved rate [9].

The second bucket is used to make sure that the rate at which the traffic is transmitted stays within the allowed boundaries; i.e. it does not exceed R^i_{max}. As shown in Figure 1, the second bucket is defined through the following components: a mean rate called excess information rate

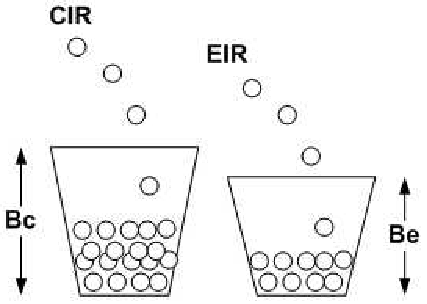


Figure 1: A Dual-Bucket Shaping Mechanism

(EIR), an excess burst size B_e, and a time interval T_e. In order to average the rate over the grant interval of the flow, we consider the same measurement interval. i.e. $T_e = T_c = I^i_{gr}$. More specifically, for a real-time flow i, $T_e = T_c = L^i_{max}$. B_e is configured in such a way that the maximum burst size does not exceed $R^i_{max} \times T_e$. In other words, $B_c + B_e = R^i_{max} \times T_e$ which implies that $B_e = EIR \times T_e = (R^i_{max} - R^i_{min}) \times T_e$. Note that when the capacity of the buckets B_c or B_e is reached, all the extra tokens are discarded. Using the configuration described above, if the buckets are empty at the beginning of the grant interval, the maximum burst size can be only reached at the end of the grant interval if no tokens are removed meanwhile. More specifically, if the packets are generated at R^i_{max} in a bursty way (still contract-conforming), they need to be delayed even if there are enough resources to transmit them since there are no enough tokens in the buckets. This configuration allows to smooth the traffic and to avoid bottlenecks at the next hop. Nevertheless, it might lead to a waste of resources. For more flexibility in resource management and in order to reach a better frame utilization rate, we choose to implement a modified version of the dual token bucket mechanism previously described. In this modified configuration, we keep the same values of the measurement intervals T_c and T_e, and burst sizes B_c and B_e. Nevertheless, we consider the buckets full at the beginning of the interval. This configuration, while bounding the burstiness to the allowed thresholds, allows it to occur at anytime during the grant interval. Note that for BE connections, the first bucket is empty since $CIR = R^i_{min} = 0$ and for UGS connections, the second bucket is empty since $R^i_{max} = R^i_{min}$ and $EIR = R^i_{max} - R^i_{min}$. Thus, the same settings remain applicable to all scheduling service types. The proposed traffic shaping is combined with a two-rounds scheduling algorithm. More details about the whole mechanism are provided in next section.

5. A TWO-ROUNDS SCHEDULING ALGORITHM

The scheduling framework we propose in this paper consists of three schedulers; two running at the BS: one for DL and one for UL and a scheduler running at the SS to redistribute the bandwidth allocated by the BS among the UL connections. Moreover, the UL schedulers (both at the BS and the SS) rely on a bandwidth request and grant process that allows the SS to transmit its non-UGS bandwidth needs to the BS which would decide the bandwidth grants accordingly. In this section, the three scheduling processes are described. At the beginning of each frame, the BS scheduler has to decide about the way of sharing the available bandwidth among active service flows. The scheduling process we propose consists of two scheduling rounds.

During the first round of the scheduling process, the objective is to honor the SLA by providing the minimum reserved traffic rate to non-BE active connections and by meeting the latency requirements of real-time services (UGS, ertPS, and rtPS). The frequency of these first allocations is set to the scheduling grant interval of the flow: I^i_{gr}. Referring to the dual token bucket mechanism described in the previous section, this first scheduling round is aimed at emptying the first token bucket of the flows whose grant interval expires in current frame interval. By proceeding this way, we avoid to schedule every single connection at each frame interval which decreases the overhead associated to a per-SS access. The algorithms corresponding to the implementation of this first round at the BS (DL and UL) and at the SS are provided in Algorithm 1, Algorithm 3, and Algorithm 2, respectively. The parameters considered in these algorithms are the following:

- $U = \{u1, u2, ..., uu\}$ the set of UGS SFs
- $E = \{e1, e2, ..., ee\}$ the set of ertPS SFs
- $R = \{r1, r2, ..., rr\}$ the set of rtPS SFs
- $N = \{n1, n2, ..., nn\}$ the set of nrtPS SFs
- $B = \{b1, b2, ..., bb\}$ the set of BE SFs
- T_f : time frame
- Gr^i_1 : the amount of bandwidth granted to connection i during the 1^{st} round of the scheduling process.
- Gr^i_2 : the amount of bandwidth granted to connection i during the 2^{nd} round of the scheduling process.
- Gr^i : the amount of bandwidth granted to connection i during the whole grant interval I^i_{gr}.
- R^i_{min} : The minimum reserved traffic rate for connection i
- R^i_{max} : the maximum sustained traffic rate for connection i
- L^i_{max} : the maximum tolerable latency for connection i
- I^i_{gr} : the grant interval for connection i
- N^i_q : the number of packets in connection i queue
- S^i_q : the size of connection i queue in bytes
- t_{cur} : current time
- t^i_{lgr} : time when connection i got its last grant

The connections participating to the first round of the scheduling process are considered in a strict priority order: UGS, ertPS, rtPS, and nrtPS. Only the amount of data conforming to the minimum rate i.e. equivalent to the number of tokens in the first bucket is scheduled after checking that

Algorithm 1: BS DL Scheduler: 1st round

```
Return: W the sum of connections weights to be used in the 2nd round
Begin
    $W \leftarrow 0$
    for $(i = 0; i < 5 ; i++)$ do
        for $(j = 0; j < N^i_{SF}; j++)$ do
            $Gr^j_1 \leftarrow 0$
            $w^j \leftarrow 0$
            if $(t_{cur} - t^j_{lgr} \geq I^j_{gr})$ then
                $tmp\_Gr^j_1 \leftarrow min\big(S^j_q, R^j_{min} \times I^j_{gr} - Gr^j\big)$
                $Gr^j_1 \leftarrow ovhd\_avail(tmp\_Gr^j_1, MCS(j))$
                $BW_r \leftarrow BW_r - Gr^j_1$
                $t^j_{lgr} \leftarrow t_{cur}$
                $w^j \leftarrow min\big(S^j_q, R^j_{max} \times I^j_{gr} - Gr^j\big) - Gr^j_1$
                $Gr^j \leftarrow 0$
                $W \leftarrow W + w^j$
            $W \leftarrow W + min(S^j_q, R^j_{max} \times I^j_{gr} - Gr^j)$
    return W
```

Algorithm 2: SS Scheduler: 1st round

```
Return: W the sum of connections weights to be used in the 2nd round
Begin
    $W \leftarrow 0$
    for $(i = 0; i < 5 ; i++)$ do
        for $(j = 0; j < N^i_{SF}; j++)$ do
            $Gr^j_1 \leftarrow 0$
            $w^j \leftarrow 0$
            if $(t_{cur} - t^j_{lgr} \geq I^j_{gr})$ then
                $tmp\_Gr^j_1 \leftarrow min(S^j_q, R^j_{min} \times I^j_{gr} - Gr^j)$
                $Gr^j_1 \leftarrow ovhd\_avail(tmp\_Gr^j_1)$
                $t^j_{lgr} \leftarrow t_{cur}$
                $w^j \leftarrow min\big(S^j_q, R^j_{max} \times I^j_{gr} - Gr^j\big) - Gr^j_1$
                $Gr^j \leftarrow 0$
                $W \leftarrow W + w^j$
            else if $\big((i \in R \; or \; i \in N) \; and \; (t_{cur} - t^j_{lgr} + T_f \geq I^j_{gr})\big)$ then
                if $(unicast\_BR\_Opp \geq 1)$ then
                    $send\_standalone\_BR$
                else if $(BWr \geq 6)$ then
                    /* bandwidth stealing */
                    $send\_standalone\_BR$
                else if $(N^0_{SF} \geq 1)$ then
                    $PM\_bit \leftarrow 1$
            $W \leftarrow W + min(S^j_q, R^j_{max} \times I^j_{gr} - Gr^j)$
    return W
```

Algorithm 3: BS UL Scheduler: 1st round

```
Return: W the sum of connections weights to be used in the 2nd round
Begin
    $W \leftarrow 0$
    for $(i = 0; i < 5 ; i++)$ do
        for $(j = 0; j < N^i_{SF}; j{+}{+})$ do
            $Gr^j_1 \leftarrow 0$
            if $(t_{cur} - t^j_{lgr} \geq I^j_{gr})$ then
                $tmp\_Gr^j_1 \leftarrow min(Req^j, R^j_{min} \times I^j_{gr}) - Gr^j$
                $Gr^j_1 \leftarrow ovhd\_avail(tmp\_Gr^j_1)$
                $BW_r \leftarrow BW_r - Gr^j_1$
                $t^j_{lgr} \leftarrow t_{cur}$
                $w^j \leftarrow min(Req^j_q, R^j_{max} \times I^j_{gr}) - Gr^j - Gr^j_1$
                $Gr^j \leftarrow 0$
                $W \leftarrow W + w^j$
            else if $\big((i \in R \; or \; i \in N) \; and \; (t_{cur} - t^j_{lgr} + T_f \geq I^j_{gr}) \; and \; ((N^0_{SF} == 0) \; or \; (N^0_{SF} > 0 \; and \; PM == 1))\big)$
            then
                $Unicast\_Poll$
            $W \leftarrow W + min(Req^j, R^j_{max} \times I^j_{gr} - Gr^j)$
    return W
```

Algorithm 4: BS DL Scheduler: 2nd round

```
Begin
    $W \leftarrow 0$
    for $(i = 0; i < 5 ; i++)$ do
        for $(j = 0; j < N^i_{SF}; j++)$ do
            $tmp\_Gr^j_2 \leftarrow \frac{w^j}{W} \times BW_r$
            $Gr^j_2 \leftarrow ovhd\_avail(tmp\_Gr^j_2)$
            $BW_r \leftarrow BW_r - Gr^j_2$
            $Gr^j \leftarrow Gr^j + Gr^j_2$
```

Algorithm 5: SS Scheduler: 2nd round

```
Begin
    $W \leftarrow 0$
    for $(i = 0;\ i < 5\ ;\ i++)$ do
        for $(j = 0;\ j < N^i_{SF};\ j++)$ do
            $Gr^j_2 \leftarrow 0$
            if $(w^j > 0)$ then
                $tmp\_Gr^j_2 \leftarrow \frac{w^j}{W} \times BW_r$
                $Gr^j_2 \leftarrow ovhd\_avail(tmp\_Gr^j_2)$
                $BW_r \leftarrow BW_r - Gr^j_2$
                $Gr^j \leftarrow Gr^j + Gr^j_2$
            if $(Gr^j_2 > 0$ and $S^j_q > 0)$ then
                if $(BW_r > 2)$ then
                    $Piggyback\_BR$
                else if $(Contention\_BR\_Opp)$ then
                    $send\_standalone\_BR$
```

there is enough bandwidth to carry the corresponding payload and overhead. Note that at the BS side, since different flows may use different MCSs, a translation of Gr^i_1 in terms of time slots/OFDM symbols is needed to evaluate the remaining bandwidth BW_r (also considered in time slots in this case) (c.f. line 10 of Algorithm 1 and line 9 of Algorithm 3). After the first round of the scheduling process, a second round is triggered by the possible availability of extra bandwidth (remaining from the first phase). The objective of this second round is to share the remaining resources among the different connections. In this second round, the bandwidth allocation process is performed according to a simple weighted fair queuing strategy. The weight of each connection corresponds to the content of its queue while not exceeding the boundaries set by its two token buckets. After Gr^i_2 is decided, an amount of tokens—corresponding to the payload scheduled in the 2nd round—is removed from the first and then from the second bucket.

In this second phase of the scheduling process, the BE connections are given, proportionally, as much chance as other types of service flows to compete for available resources which decreases the risk of starvation for this category of traffic. The remaining needs of each non-UGS connection, i.e. the difference between the queue size and the allocated grants are then translated into bandwidth requests. The details of the proposed algorithm are provided for the BS (in DL) and the SS in Algorithm 4 and Algorithm 5, respectively.

Figure 2 illustrates the three possible configurations of the token buckets at the end of the grant interval for a given connection i, after performing the two scheduling rounds. Note that during the whole interval, the buckets are not refilled. In the first case, both buckets are empty which means that the connection has been scheduled at its maximum sustained traffic rate R^i_{max}. When only the first bucket is empty, this means that the connection has been scheduled at a rate $R^i; R^i_{min} <= R^i < R^i_{max}$. In other words, the scheduler has managed to meet at least the minimum requirements of the connection in terms of delay and throughput. The third case, shown in Figure 2, corresponds to the case where the first bucket is not completely empty i.e. $R^i < R^i_{min}$. This means that the available bandwidth was not enough to cover the needs of the connections participating to the 1st round of the scheduling process. In the two first cases, the two buckets associated to the considered connection are refilled with tokens and the grant interval is reset. In the last case however, the same buckets are maintained. Moreover, to reach R^i_{min}, the connections needs more bandwidth then what is reflected by the content of the first bucket. Therefore, at the beginning of the following frame $T_f \times R^i_{min}$ tokens from the second bucket are marked indicating that the threshold for the 1st round is not only set by the content of the first bucket but also with the marked tokens from the second one. The connection participates to the first round of the scheduling process as many times as needed, during the following time frames, till all the tokens of the first bucket and those marked in the second bucket are removed. It is only at that time that the two buckets associated to this connection are refilled and the grant interval is reset. This last case entails some latency for the considered flow. Nevertheless by shifting the corresponding grant interval, we decrease the chances that the same thing happens again (two or more heavy bursts coincide in the same time frame) especially if the burstiness occurs periodically.

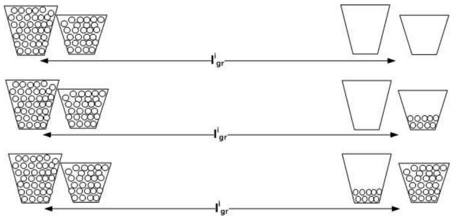


Figure 2: A Dual-Bucket Shaping Mechanism

As reported in Algorithms 3, 2, and 5, the bandwidth request and grant strategy we adopt in this paper adapts the choice of one or another of the available techniques to the considered service type and to the overhead entailed by the use of that technique. A comparison of the different bandwidth request and grant mechanisms is provided in [10].

6. PERFORMANCE ANALYSIS

To evaluate the performance of mCoSS, we have implemented the corresponding set of algorithms under QualNet 4.5 [11] which is the commercialized version of GloMoSim. mCoSS has been compared to Strict Priority (SP) and to a variant of the WFQ discipline. In this section we first give an overview of the features supported by the WiMAX simulation model proposed by QualNet. Then, we define the scenarios and simulation settings considered in the performance analysis before reporting and commenting the obtained results.

6.1 A WiMAX Simulation Model Under QualNet

QualNet 4.5 provides the Advanced Wireless Model Library which addresses both fixed and mobile WiMAX systems. The proposed simulation model is dedicated to OFDMA-based PMP networks operating in TDD mode. It supports the five service types UGS, ertPS, rtPS, nrtPS and BE and several types of bandwidth request mechanisms (polling-based, contention-based and CDMA-based). Most of the IEEE 802.16 management messages (DCD, UCD, UL-MAP, DL-MAP, DSx, etc.) are implemented and several features like the AMC, fragmentation, and packing are supported. Nevertheless, some bugs in the fragmentation mechanism (leak in the queues) have been noticed. We have fixed this bug by correcting the way the queue size is updated when a fragment or a whole packet is removed from the queue. Moreover, only CBR and VBR generators have been considered when mapping the QoS parameters from application to MAC level. We have extended this capability to Super-Application traffic generator which provides more flexibility in the flow configuration. The model provides also a basic admission control mechanism and a scheduling policy based on a variant of the WFQ strategy, which is different from the one we use in mCoSS. The WFQ variant implemented in QualNet calculates and assigns a finish time to each packet. In this calculation, WFQ uses the bit rate of the link, the number of queues, and the size of each packet in each of the queues. The WFQ scheduler then transmits the packet with the earliest finish time among all the queued packets. Thus, each time a packet is dequeued, the WFQ scheduler recomputes the finish time assigned to each packet which entails a high computational complexity and limits the scalability of the proposed approach.

6.2 Performance evaluation

Channel Frequency	3.5 GHz
Channel bandwidth	10 MHz
FFT size	2048
Cyclic prefix gain	8
Propagation pathloss model	Two-ray
BS antenna Tx power	33 dBm (= 2 W)
BS antenna height	32 m
BS antenna gain	15 dBi
MS antenna Tx power	23 dBm (= 200 mW)
MS antenna height	1.5 m
MS antenna gain	-1 dBi
Type of antenna	omnidirectional
Frame duration	10 ms
DL subframe duration	5 ms

Table 2: Simulation settings

In this section, we consider the parameters settings reported in Table 2. We consider a DL/UL ratio of 1:1 from a total frame size of 10 ms. A simple two-ray pathloss propagation model has been used and no shadowing or fading has been considered to offer a "simple" environment for the comparison of the different algorithms.

In the following scenarios, we consider an audio stream of 30 mns configured as an UL rtPS connection. The audio frame size is set to 1600 bytes and the number of frames per second follows a uniform distribution between 10 and 25 fps (frame/second). The QoS parameters of the considered stream are configured as follows: R^i_{min} = 128 kbps, R^i_{max} = 320 kbps and I^i_{gr} = 100 ms.

Scenario 1: mCoSS Shaping Capability.

In this scenario, we propose to test the shaping capability of our multi-Constraints Scheduling Strategy (mCoSS). Therefore, we place two MSs at the same distance from the BS and we configure an audio stream for each MS as mentioned before: R^i_{min} = 128 kbps, R^i_{max} = 320 kbps and I^i_{gr} = 100 ms. While MS1 respects these boundaries, MS2 transmits the audio stream at a much higher rate varying from 640 kbps to 1.28 Mbps. More than 30 experiments have been run to validate the shaping capability of our algorithm and to compare it to the WFQ and SP algorithms implemented in QualNet. Figure 3 plots the transmission (Tx) and reception (Rx) rates of both the misbehaving and the well-behaving traffics for the three algorithms: mCoSS, WFQ, and SP. The Tx rate represents the rate at which the application is generated at the MS while the Rx rate is the reception rate at the BS. We can see from Figure 3 that for the well-behaving traffic sent by MS1, the three algorithms have almost equal performance in terms of throughput. For the misbehaving traffic however, both SP and WFQ allow it to reach more than 800 kbps while mCoSS forces the traffic to stay within the set boundaries: the reception rate at the BS does not exceed 315 kbps. Tables 3 and 4 report the obtained E2E delay and jitter for both traffics using the different scheduling algorithms. As a consequence of the shaping policy adopted by mCoSS, the misbehaving traffic generated by MS2 is penalized (in comparison to SP and WFQ) in terms of E2E delay since packets exceeding R^i_{max} are delayed and possibly dropped if their number exceed the buffers capacity. On the other hand, the E2E delay of well-behaving traffic is halved compared to WFQ and SP. With both WFQ and SP, the two traffics experience comparable E2E delays; the misbehaving traffic gets even a shorter average jitter than the well-behaving traffic.

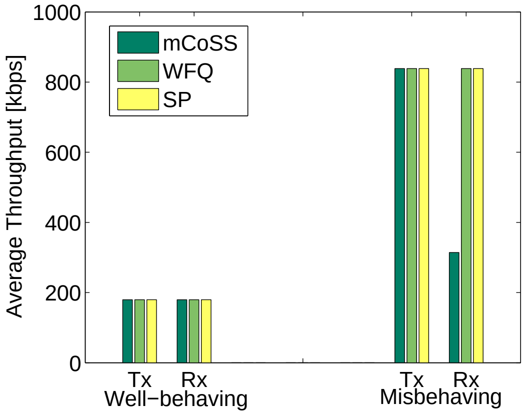


Figure 3: mCoSS Shaping Capability

From the obtained results, we can see that mCoSS is capable of forcing a traffic to stay within the allowed thresholds and isolating a well-behaving traffic from a misbehaving one. The absence of shaping in WFQ and SP has affected the performance of the first traffic and could even have a much worse effect if the second traffic had been generated at a rate which overloads the whole network.

Scenario 2: Fairness and QoS Degree of Satisfaction.

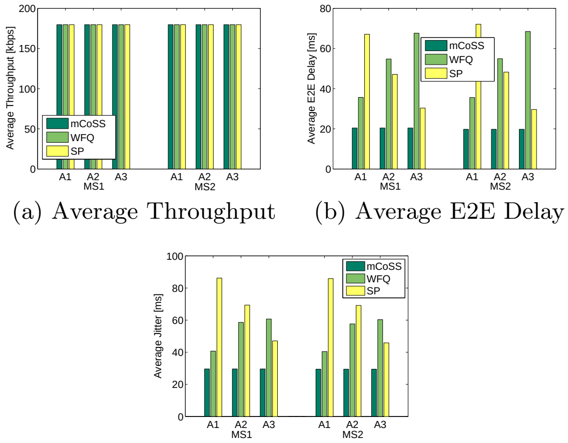
In this second scenario, we consider the same MSs having each three audio streams with the same configuration. Through this scenario, we aim at evaluating, in same channel and traffic conditions, the performance of our scheduling

	MS1	MS2
	Well-behaving	Misbehaving
mCoSS	0.255	13.6
WFQ	0.57	0.53
SP	0.57	0.53

Table 3: mCoSS Shaping Capability: E2E Delay (sec)

	MS1	MS2
	Well-behaving	Misbehaving
mCoSS	22	80
WFQ	69	27.7
SP	69	27.7

Table 4: mCoSS Shaping Capability: Jitter (ms)



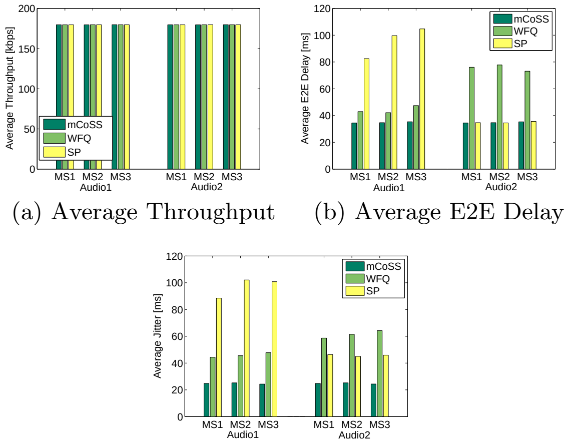
(a) Average Throughput (b) Average E2E Delay

(c) Average Jitter

Figure 4: 2 MSs with 3 Audio streams each

algorithm in terms of inter-MSs and inter-SFs fairness and to compare the QoS degree of satisfaction of the six connections using the three algorithms. Figure 4(a) plots the obtained average throughput of the 1st, 2nd, and 3rd audio streams (A1, A2, and A3) of MS1 and MS2. As long as the throughput is concerned, the three algorithms offer the same level of performance. The average end-to-end (E2E) delay and jitter, however, experience a less stable behavior from one algorithm to another as we can see from Figures 4(b) and 4(c), respectively. With WFQ, the E2E delay varies from 35 to 67 ms from one service flow/MS to another. The same behavior is noticed for SP for which the E2E average delays vary from 30 to 72 ms. mCoSS on the other hand provides lower and much more stable results for the six flows for both E2E delay (around 20 ms) and jitter (less than 30 ms).

Considering throughput, delay and jitter, mCoSS, in comparison with SP and WFQ, provides the best and most stable performance among SFs which results in a better inter-SFs and inter-MSs fairness.



(a) Average Throughput (b) Average E2E Delay

(c) Average Jitter

Figure 5: 3 MSs with 2 Audio streams each

Scenario 3: AMC support.

Through this last scenario, we aim at validating the capability of mCoSS to adapt the allocated bandwidth to the channel conditions of the MS; a capability that is already supported in QualNet implementation of WFQ and SP. Since the objective is to test the AMC capability of mCoSS, we consider 3 MSs placed at three different positions from the BS: at 1km, 2 km, and 3 km away from the BS. These three distances correspond to three SNR levels matching UIUC 1 (QPSK 1/2), UIUC 4 (16-QAM 3/4) and UIUC 7 (64-QAM 3/4). We configure two audio streams at each MS with the same settings previously specified. As we can see from Figure 5(a), like for the previous scenario, the three algorithms have almost equivalent performance for the throughput. However, the difference in E2E delay (plotted in Figure 5(b)) between Audio 1 and Audio 2, with SP, is more noticeable than in the second scenario. Indeed it varies for MS3 for example from 35 ms to more than 100 ms which exceeds the maximum latency of the service flow. The same behavior is observed for average jitter in Figure 5(c).

For mCoSS, the increase of the number of MSs and the use of different MCSs had almost no effect on the performance of the algorithm. The same stability of results is observed in this scenario which confirms the fairness of the algorithm.

The performance evaluation of mCoSS presented in this paper is by no means comprehensive, yet it shows and validates some of the key features supported by the proposed strategy. More simulation scenarios though—involving more service types—need to be considered.

7. CONCLUSION

Most of the hierarchical scheduling strategies proposed in the literature and described in [2] (such as [3, 7, 8]) propose a specific queuing discipline for each scheduling service type, which increases significantly the complexity of the proposed scheduling policy. Unlike those approaches, the multi-Constraints Scheduling Strategy (mCoSS) proposed in this paper is designed to be applicable to all service types. Based on a modified dual token bucket traffic shaping mechanism used for all the scheduling service types, mCoSS allies the genericity of the approach to the specificity of the configuration since the dual bucket mechanism is configured on a per-flow basis.

This shaping mechanism is combined with a two-rounds

scheduling strategy which reflects (i) at the first round, the minimum data rates and latency requirements the BS and MS schedulers are committed to provide and (ii) at the second round, the efficiency and fairness of the resources management. In this second round, the remaining bandwidth is shared using a simple WFQ strategy; the allocations should nevertheless remain within the thresholds set by the dual token bucket shaping mechanism.
The bandwidth request and grant mechanism adopted in mCoSS is designed to make a tradeoff between increasing the accuracy of the bandwidth needs perception at the BS and decreasing the overhead associated to frequent unicast polling. Indeed the proposed strategy alternates between bandwidth stealing, piggybacking, unicast, broadcast and group polling, and the use of PM bit based on the considered scheduling service type and the available resources. The proposed mCoSS has been implemented under QualNet 4.5 simulator and compared to Strict Priority (SP) and to a variant of WFQ discipline. The preliminary results reported in this paper validate and confirm the shaping, fairness and AMC support capability of the proposed mCoSS. They also show that, compared to SP and WFQ, mCoSS provides better and more stable end-to-end-delay and jitter performances. More simulations need though to be carried out to check and validate other aspects of the proposed scheduling strategy.

A further extension of this work would be to associate this scheduling solution with an admission control policy (AC) like the one proposed in [12]. This AC policy presents the advantage of using a token bucket mechanism which regulates the number of accepted rtPS and nrtPS connections in order to be able to accept more BE connections. A similar approach [13] consists in applying a buffering mechanism for connections that cannot be accepted due to a lack of resources. The buffered connections could then initiate a backoff period to attempt a new reservation of resources, when available. Both mechanisms have proven their efficiency to increase the number of accepted BE connections without degrading the overall performance of the network.

8. REFERENCES

[1] IEEE Std 802.16e 2005. IEEE Standard for Local and metropolitan area networks Part 16: Air Interface for Fixed and Mobile BWA Systems-Amendment 2: Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands and Corrigendum 1. 2005.

[2] Ikbal Chammakhi Msadaa, Daniel Câmara, and Fethi Filali. Scheduling and CAC in IEEE 802.16 fixed BWNs : a Comprehensive Survey and Taxonomy. *"IEEE Communications Surveys & Tutorials"*, 12(4):459–487, 2010.

[3] Kitti Wongthavarawat and Aura Ganz. Packet scheduling for QoS support in IEEE 802.16 broadband wireless access systems. *International Journal of Communication Systems*, 16(1):81–96, Feb. 2003.

[4] Kitti Wongthavarawat and Aura Ganz. IEEE 802.16 based last mile broadband wireless military networks with quality of service support. *IEEE Military Communications Conference, 2003. MILCOM 2003*, 2(1):779–784, Oct. 2003.

[5] Naian Liu, Xiaohui Li, Changxing Pei, and Bo Yang. Delay Character of a Novel Architecture for IEEE 802.16 Systems. In *Sixth International Conference on Parallel and Distributed Computing, Applications and Technologies, 2005. PDCAT 2005*, pages 293–296, Dec. 2005.

[6] 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. In *International Symposium on Computer Networks, 2006*, pages 11–16, Jun. 2006.

[7] R. Perumalraja, J.J.J. Roy, and S. Radha. Multimedia Supported Uplink Scheduling for IEEE 802.16d OFDMA Network. In *Annual India Conference, 2006*, pages 1–5, Sept. 2006.

[8] Maode Ma, Jinchang Lu, S.K. Bose, and Boon Chong Ng. A three-tier framework and scheduling to support QoS service in WiMAX. In *6th International Conference on Information, Communications & Signal Processing*, pages 1–5, Dec. 2007.

[9] IEEE 802.16-2009. IEEE Standard for Local and metropolitan area networks Part 16: Air Interface for Broadband Wireless Access Systems. May 2009.

[10] So-In Chakchai and R. Jain and A.-K. Tamimi. Scheduling in IEEE 802.16 e mobile WiMAX networks: key issues and a survey. *IEEE Journal on Selected Areas in Communications*, 27(2):156–171, Feb. 2009.

[11] Scalable Network Technologies. Qualnet 4.5, March 2008. http://www.scalable-networks.com/products/qualnet/.

[12] Lynda Mokdad and Jalel Ben Othman. Admission control mechanism and performance analysis based on stochastic automata networks formalism. *Journal of Parallel and Distributed Computing archive*, 71(4):594–602, April 2011.

[13] Lynda Mokdad and Jalel Ben Othman. Stochastic automata networks for modelling scheduling scheme in WiMAX networks. *International Journal of Interconnection Networks (JOIN)*, 10(4):481–495, 2009.