Fair Scheduling for Real-Time Multimedia Support in IEEE 802.16 Wireless Access Networks

Yaser P. Fallah
Inst. of Transportation Studies (Depts EECS & CEE), University of California at Berkeley, USA
e-mail: yaserpf@eecs.berkeley.edu

Panos Nasiopoulos
Dept. of Electrical and Computer Eng.
University of British Columbia
Vancouver, Canada
e-mail: panosn@ece.ubc.ca

Raja Sengupta
Inst. of Transportation Studies (Dept of CEE), University of California at Berkeley, USA
e-mail: sengupta@ce.berkeley.edu

***Abstract* - Successful deployment of Broadband Wireless Access Networks such as WiMAX (IEEE 802.16) will be contingent on provisions for supporting multimedia traffic. In this paper, we review the quality of service features of access networks such as the 802.16 standard, and identify algorithms and schemes that are needed for supporting multimedia traffic in such networks. The 802.16 standard only specifies the features that should be implemented and leaves the design of a quality of service solution to developers. This includes the design of a mandatory scheduling framework. We present a comprehensive multimedia support framework based on the standard features. The framework specifies the architectures for the base station and the subscriber station, and contributes a number of algorithms for different service provisioning objectives. We use the concept of virtual packets to provide fair packet based centralized scheduling of uplink and downlink packets. The presented solution also provides algorithms for temporal and throughput fair scheduling in multirate physical layer of the 802.16 networks. An important part of the presented design is a multi-class fair scheduling scheme which is proposed for providing better delay performance for real time applications, while maintaining slightly longer term fairness.**

Keywords - WiMAX, QoS, Scheduling, SFQ, Multimedia, IEEE 802.16, MAC, TDMA.

I. Introduction

IEEE 802.16 is the recently approved standard for wireless metropolitan area networks (WMAN) [1][2]. This standard, whose industry-led equivalent is also referred to as WiMAX, will be a competitor to the other currently available broadband solutions such as DSL, cable, and IEEE 802.11 ([3]) in offering broadband access. Thus, it is necessary to develop solutions for supporting popular quality of service (QoS) demanding applications such as real time multimedia in WiMAX networks.

While QoS support has been considered during the development of 802.16, the standard only describes the required features and classes of QoS and does not define the design of a mandatory scheduler used for providing the required QoS. Such a design is left to vendors of WiMAX base station and subscriber stations. Devising scheduling algorithms for 802.16 WMANs is a challenging task for the following three reasons. First, 802.16 defines several classes of traffic with different types of QoS requirements (multi-class QoS); conventional fair scheduling or priority based scheduling cannot achieve both fair throughput guarantee and the desired differentiated delay performance at the same time. Second, stations may use different transmission rates at different times (multirate operation). And third, uplink and downlink traffic in 802.16, though separated using duplexing methods, ultimately share the same time resource (assuming that time division duplexing, TDD, mode is used in the MAC, as opposed to the less efficient and less flexible frequency division duplexing, FDD). Our proposed solution, in this article, addresses these issues and efficiently utilizes the available features of the 802.16 standard to provide fair guaranteed services for real-time applications.

In the past few years, several QoS solutions for 802.16 networks have been developed. However, to the best of our knowledge no model is provided for fair multi-class service provisioning and simultaneous scheduling of uplink and downlink packets in an 802.16 MAC. Hawa and Petr presented an architecture for scheduling uplink and downlink flows separately [4]. Cicconetti et al. have conducted a performance study to evaluate QoS provisioning available in the 802.16 MAC layer, operating in the FDD (frequency division multiplexing) mode [5].

In another work, Cho et al. proposed a new MAC architecture with a bandwidth allocation, as well as an admission control technique for better QoS support in 802.16 [6]. Gusak et al. carried out a comparison of two well-known scheduling schemes namely, WFQ and WRR when used in 802.16 MAC in terms of average packet delay [7]. They also proposed an algorithm for dynamic ratio adjustment of UL and DL duration within a frame. Although the existing solutions can enhance the performance of the 802.16 MAC in QoS provisioning, they do not address the fairness issues and the combined scheduling of uplink and downlink flows (necessary for the TDD mode). Moreover, the multirate operation of the physical layer (PHY) is not considered in these works.

In this paper, we present a detailed architecture and its supporting algorithms for providing fair multi-class services for uplink and downlink flows in an 802.16 MAC, considering the multirate operation. The proposed architecture also allows for other algorithms to be used as its components. We note that providing fairness is necessary in cases that customers expect to receive a fair service according to their contract. Fairness becomes especially important in the case of multirate operation of the PHY, due to the fact that one station's poor channel may degrade service quality for others.

In the next section some basic technical details of the MAC layer of the IEEE 802.16 standard are described. Section III presents a detailed description of the proposed architectural design followed by the performance evaluation of the model discussed in Section IV. Finally, we conclude the article in Section V.

II. Overview of The IEEE 802.16 Standard

The 802.16 MAC is connection oriented; each subscriber stations (SS) establishes a connection with the base station (BS) to deliver voice, video, or data using connections with different QoS requirements [1]. In contrast to the 802.11 standard [3], 802.16 achieves duplex operation through either FDD or TDD. The downlink channel (from BS to SS) is a broadcast channel, while the uplink channel (from SS to BS) is shared by SSs. Time in the uplink channel is usually slotted (mini-slots) and shared using time division multiple access (TDMA), whereas on the downlink channel BS uses a continuous time-division multiplexing (TDM) scheme. Other multiple access schemes such as orthogonal frequency division multiple access (OFDMA) have also been specified in later versions of the standard [2].

The duration of the downlink or uplink subframe in TDD mode is determined by the BS in a dynamic manner and is based on the amount of uplink or downlink load. The 802.16 MAC layer requires that the BS and SSs be synchronized at all times and transmit only in predetermined time slots or subcarriers. In TDMA mode, the uplink channel is divided into a sequence of mini-slots, which SSs can access in a synchronized manner controlled by the BS. If OFDMA is used, subcarriers are divided amongst SSs to allow simultaneous transmission in the same slots. The BS uses Uplink and Downlink maps (UL-MAP and DL-MAP) at the beginning of a subframe to indicate the assignment of mini-slots (e.g. Fig. 1) or subcarriers.

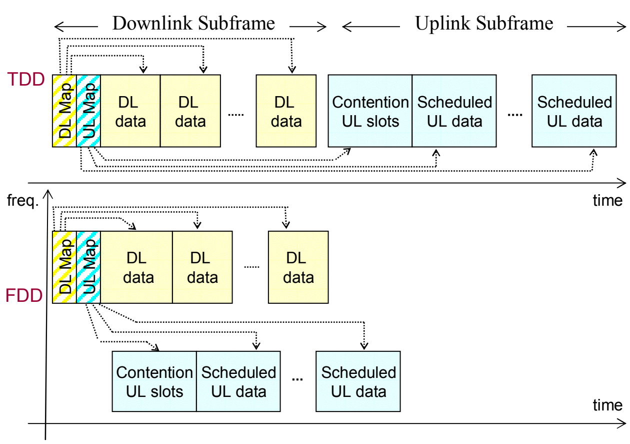

Fig. 1. IEEE 802.16 MAC Frames

Given that data transmission is always controlled by the BS, scheduling becomes one of the central tasks of the MAC. The new revision of the standard (802.16e) revises the description of QoS services offered and in particular clarifies the distinction between uplink and downlink flows [2]. Five types of uplink services are defined: Unsolicited Grant Service (UGS); Extended Real-Time Polling Service Real-Time Polling Service (rtPS); Non-Real-Time Polling Service (nrtPS) and Best Effort (BE) service. However, QoS descriptors used for specifying these uplink services are also used for downlink services, and the distinction is only in the modes of BW request for uplink flows.

Downlink packets can be directly scheduled by the BS in the downlink subframe, according to the QoS description of their connections. For uplink connections, each SS that needs to send data has to first request bandwidth from the BS. The BS will then assign uplink transmission opportunities to the SS in the next uplink subframes. Two modes of bandwidth (BW) requests are defined: contention based and polling based. In contention mode, SSs send BW requests during predetermined contention periods in the uplink frame. Collisions are possible for such requests. In polling based mode, BS assigns BW request opportunities to the stations, and SSs reply by sending BW requests. Other ways of contention free BW request are BW stealing (sending requests instead of data packets) or piggybacking.

When a UGS service is setup, the BS is responsible for assigning fixed size periodic data grants to the UGS flow; the SS is not allowed to explicitly request BW for an already established UGS flow. UGS service can be used for constant bit-rate (CBR) service flows such as T1/E1.

The ertPS service is similar to UGS, except for its use of dynamic and variable size data grants. In this case, the BS provides unicast grants of variable size in an unsolicited manner like in UGS, thus saving the latency of a bandwidth request. The SS indicates in piggybacked messages the requested size for data grants and the BS dynamically adjusts the grant size in the next cycle. ertPS can be efficiently used for VIOP flows with silence suppression.

For the rtPS and nrtPS services, the SSs can send BW requests in response to BW request opportunities assigned by the BS. The difference between these two modes is that the nrtPS flows receive few polling BW request opportunities during network congestion and are allowed to use contention requests, while the rtPS flows are polled regardless of the network load, and frequently enough to meet the delay requirements of the service flows. The rtPS flows are not allowed to use contention requests. The bandwidth assignment to rtPS flows is more flexible than UGS and can be used for VBR-like service flows such as video. The nrtPS can be used for providing better than best effort services for applications such as bandwidth-intensive file transfer. Although ertPS, rtPS and nrtPS flows require throughput or bitrate guarantee, ertPS and rtPS flows usually have additional requirements on delay. The delay performance of ertPS can be better due to the fact that no explicit BW request is needed.

Best Effort (BE) flows are allowed to request bandwidth only through piggybacking and contention mode.

The parameters describing a traffic flow and its QoS requirements in 802.16 are different for each class of traffic (except for ertPS and rtPS). Table I summarizes the minimum required information for each class. This difference between the type of QoS requirements indicates the need for scheduling algorithms that are able to provide different *classes* of quality of service, and not just different levels of QoS or service differentiation.

Table I. QoS PARAMETERS FOR DIFFERENT TRAFFIC CLASSES

Traffic Class / QoS Parameter	UGS	ertPS	rtPS	nrtPS	BE
Maximum Sustained Traffic Rate: R_{max}	x	x	x	x	x
Maximum latency: D_{max}		x	x		
Tolerated Jitter: J_{max}	x				
Minimum Reserved Traffic Rate: R_{min}	x (R_{max})	x	x	x	
Traffic Priority:				x	x
Request/Transmission Policy	x	x	x	x	x

There are several physical layer characteristics of the 802.16 standard that require special attention when devising a QoS solution. The 802.16 PHY operates in a multirate fashion. The multirate operation is possible using a user defined link adaptation mechanism that adjusts the modulation and coding schemes, thus the transmission rate, with the objective of maintaining certain reliability for the wireless link (e.g., a BER of 10^{-5}). For example, one set of rates provided by 802.16-OFDM is 6.9 13.8, 21.7, 27.5, 41.5, 55.3, and 62.2 Mbps. Any of these transmission rates can be used by each station or the BS, and for any transmission burst. The variation in transmission rate has a significant impact on the fairness and service guarantee in the MAC, and has to be considered in the scheduler design.

III. ARCHITECTURE FOR FAIR SCHEDULING IN 802.16 MAC

Traditional scheduling solutions cannot be directly employed in 802.16 networks, due to specific characteristics of the 802.16 MAC and PHY. These characteristics include the distributed multiple access environment with the possibility of dynamic adjustment of uplink/downlink durations, the multirate operation of PHY, and the multi-class definition of QoS requirements of connections. We propose a detailed QoS scheme that takes into account all these characteristics and aims at providing multi-class fair QoS in a centralized fashion. The prominent features of our solution are the following: (1) Combined Uplink/Downlink scheduling: our QoS framework uses *virtual packets* and combines the task of scheduling uplink and downlink flows into a central scheduling process that resides in the BS; (2) Service guarantee and scheduling different classes of traffic: the proposed framework uses a novel scheduling algorithm that allows long term service fairness while reducing the delay of time sensitive data; and (3) Multirate operation: the scheduling algorithm is designed to provide either temporal or throughput fairness.

To achieve the required QoS, the MAC architecture of the BS and SS should be specified. We propose the architecture depicted in Fig. 2 as a comprehensive QoS solution. This architecture requires a number of algorithms for its components. To understand what components are needed, we consider the operation of the MAC from the QoS perspective.

Since the MAC is connection oriented, the first step for data transport is to setup a connection. At connection setup, a request is sent to the BS, declaring the traffic specifications of the stream that will use this connection. The *admission control* (AC) algorithm in the BS inspects the traffic specifications and admits or rejects the new connection (stream) based on a user defined policy. If the admitted flow is an uplink flow, the traffic specification is passed to a *virtual packet generator* (VPG) algorithm that generates virtual packets representing the actual uplink packets of the stations. The virtual packet generator also uses the uplink queue length information that is partly retrieved through unicast poll generation. Thus, a *poll generation* (PG) algorithm is required to schedule unicast request opportunities (URO) for uplink flows. Using virtual packets, it is possible to use a single scheduling algorithm for scheduling both uplink and downlink flows. This scheduler can use any conventional scheduling algorithm. We denote this as the Inner BS Scheduler (IBS). The operation of this scheduler has a significant effect on the overall performance of the system. We propose a novel multi-class fair queuing scheme (*MCFQ*) for the inner scheduler.

Actual or virtual packets that are served by the scheduler are passed to the frame builder module which constructs the MAC frame. Actual packets are aligned in a downlink subframe, while virtual packets are translated into uplink bandwidth grants called Transmission Opportunities (TO). UL and DL maps are then generated accordingly.

In the SS side, a simple scheduling algorithm that responds to bandwidth allocations by the BS is enough. This algorithm sends packets from the queues which receive a BW allocation from the BS, and if no such packets exist, it can serve other queues using a user defined priority scheme. In fact, the QoS architecture and algorithms at the BS are the major components required for QoS in 802.16 networks. The concepts and algorithms that are needed for enabling the BS architecture are described next.

A. Combined Downlink/Uplink Scheduling, VPG Algorithm

The scheduling discipline must consider both uplink and downlink traffic for scheduling at all times. Downlink packets are available in the BS buffers and can be directly scheduled, whereas uplink packets reside in the stations generating these packets and cannot be scheduled directly. However, the BS can use traffic specifications, indicated at connection setup time, to generate a local flow of "virtual packets" that represent the actual uplink packets. It is then possible to use a single scheduler to schedule these virtual packets, along with downlink packets. The concept of virtual packets has been used before in other works [14].

Virtual packets are generated using the specified patterns of uplink traffic flow (e.g., specified in dynamic service addition messages at service flow creation and activation time [1]) as well as the queue size information delivered through BW request messages. For example, for a voice call, a periodic flow of virtual packets similar to the real traffic is generated. When there is silence in the voice call (detected using polling messages), virtual packet generation is halted. When activity is detected through BW request messages, virtual packet generation resumes.

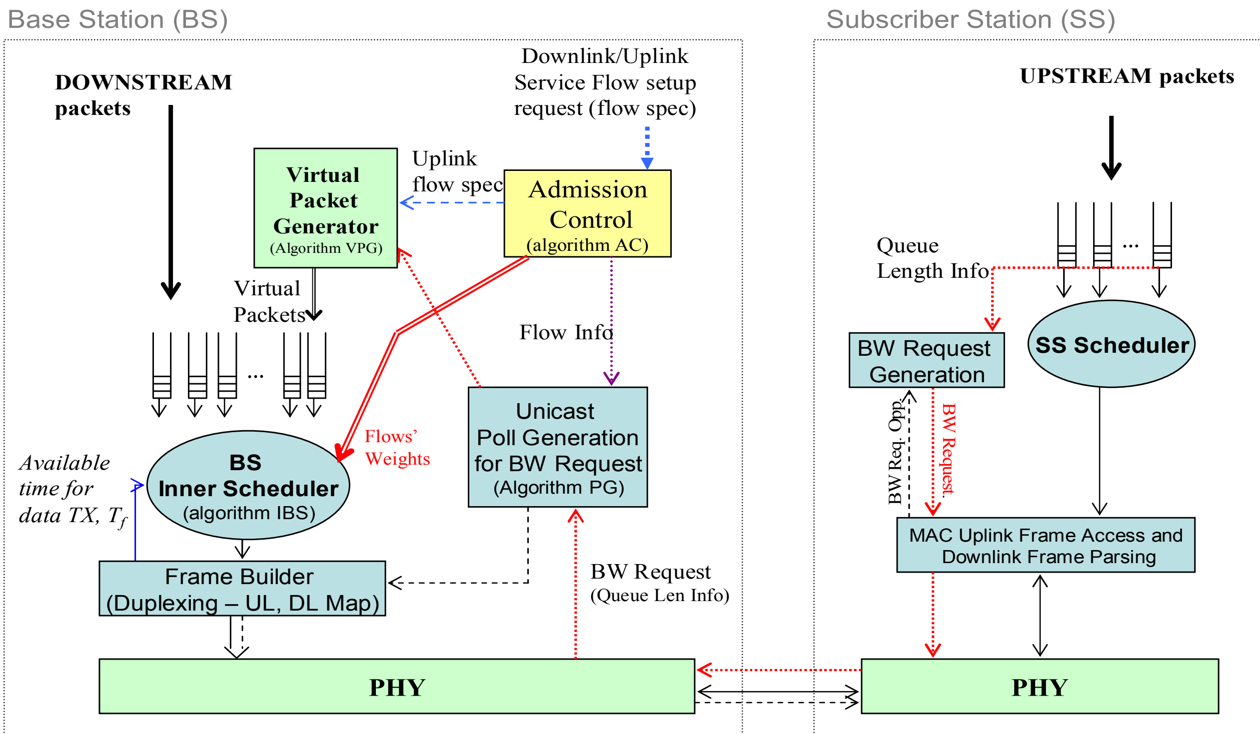

Fig. 2. Architecture of Fair Service Provisioning Mechanism for 802.16 MAC

The parameters used by VPG are taken from the minimum set of parameters that should be provided for a connection (see Table I). We make a practical assumption that for audio and video at least the following set of parameters is provided: maximum bitrate (R_{max}), average bitrate (R_{min}), burst size (b), and average packet size (p) or Service Interval (T_s). We use these parameters, in addition to queue length information, to generate virtual packets. The VPG algorithm relies on the polling algorithm which gathers queue size information (Q_i) from stations. For as long as $Q_i > 0$, the VPG algorithm generates virtual packets of length p, at intervals T_s, to achieve the average rate R_i , where $R_{max} \leq R_i \leq R_{min}$ is determined by the admission control scheme. If T_s is available, packet length can be calculated as $R_i.T_s$, whereas if p is available, T_s can be calculated as p/R_i.

With each generated virtual packet, the projected queue length Q_i is reduced by the size of the virtual packet. Q_i is increased by the amount of incremental data declared in each queue length update from the stations. This process ensures that virtual packets are only generated for the amount of available data at the stations. Since data generation rate may be variable in the station, the algorithm maintains a budget parameter, g_i, to keep track of the unused service, and assigns it to the station when it has more data. This mechanism ensures uplink flows receive a fair share of the service. The budget parameter is limited to the allowed burst size of the flow b_i in order to enforce a leaky bucket shaped traffic. This algorithm is described below.

```
------------------------- Each loop runs in a separate thread.
Loop QueueLenUpdate:
        If queue_length update is received: (inc_Q, i)
             Qi = Qi +  inc_Q
End loop
--------------------------------------------------------------------
gi = 0; // gi is the credit for queue i
Ri = getRateAC(); //AC determines guaranteed rate: Rmax≤Ri≤Rmin
Loop VPG:
        gi = gi + Ri x Ts ;    //or +VPi_length = p, with Ts = p/Ri;
        if (gi > bi )      gi = bi ;
        if (Qi > 0) {
            if (gi > Qi)   // BW grant should be limited to Qi
                 VPi_length = Qi ;
            else // BW grant is limited to the available budget gi
                   VPi_length =gi ;
            Generate_and_send_VP (i, VPi_length);
            gi = gi – VPi_length;
            Qi = Qi – VPi_length; // decrement to count for the
                     //data which will be sent in the uplink subframe
            Wait Ts ;
        }else
            Wait Ts ;
End loop;
```

Since uplink packet sizes may vary and be different from the virtual packet length, stations should fragment their packets (or zero pad) to fit them in the assigned TO. BE data requests are directly translated to virtual packets of the requested size and no periodic VP generation is needed.

B. Poll Generation Algorithm:

The unicast poll generation scheme is a simple yet important part of the QoS architecture, and is needed for rtPS and nrtPS classes. We propose an algorithm as follows: for rtPS class, if a piggybacked unicast request for a queue has arrived during the previous frame, the BS does not send a URO to the corresponding station. If no piggybacked request arrives, the BS generates UROs with a period equal to the service interval (SI) of the traffic. Since rtPS traffic (e.g., video) is usually periodic, the service interval is known. SI can also be calculated as a fraction of the maximum latency. It must be noted that for rtPS class, the 802.16 standard permits the BS to issue UROs even if previous requests are still unfulfilled.

For nrtPS, traffic generation is usually not periodic. For this reason, the interval between URO assignments is implementation dependent. Also, if a previous request of an nrtPS flow is unfulfilled, new UROs are not generated. The standard suggests the interval between UROs (after the last BW assignment indicating no data at the station) to be in the order of 1 second or less. When the network is lightly loaded, we reduce this period (T_i) in order to reduce the response time; while it is increased to around 1 second when network population grows. Therefore, for nrtPS flows with no unfulfilled requests, T_i is calculated as follows:

$$T_i=\min(1\text{sec},\theta * N_{nrtPS});$$

where θ is a user defined parameter (e.g., 100msec) and N_{nrtPS} is the number of nrtPS flows.

C. Fair Scheduling Algorithm

The architecture presented in this paper allows for use of any conventional scheduling algorithm as the central scheduler. For example, algorithms such as weighted round robin (WRR) or weighted fair queuing (WFQ) can be used [11]. Considering that 802.16 supports different classes of QoS (i.e., UGS, ertPS, rtPS and nrtPS for uplink, and similar parameterized flows for downlink) with different requirements, the conventional fair scheduling algorithms cannot efficiently provide delay and throughput guarantees and maintain fairness at the same time. To address these issues we propose a new type of scheduling algorithm.

QoS is usually specified in terms of guaranteed bandwidth (throughput) and bounded delay. This is true for ertPS and rtPS class, i.e., for video and audio flows. However, for the nrtPS class and non real time applications, only guaranteed bandwidth is required and large latency and jitter are tolerated. Fairness in assigning service is also an implicit QoS requirement for flows which are promised the same level of service by the network. For example, users expect their service levels be maintained while the traffic of other users exceeds its originally declared level.

Bandwidth (throughput) guarantee is achieved through scheduling and admission control. For the scheduler part, we need to ensure that each flow gets a weighted share of the channel capacity. This is usually achieved using fair algorithms such as WFQ. Using these algorithms, the delay performance of each flow is strongly dependent on the weight (rate) assigned to it. On the other hand, nrtPS flows do not have strict delay requirements, and the above algorithms treat both nrtPS and rtPS classes the same way. This is an undesirable property of the conventional fair algorithms.

Better scheduling schemes can be devised to use the fact that nrtPS class does not require delay guarantees. We briefly describe such an algorithm in this section. The proposed algorithm provides short term fairness amongst flows of the same class, and long term fairness amongst different classes of traffic. It also tries to reduce the delay bound for rtPS flows, at the cost of increasing the delay for nrtPS flows.

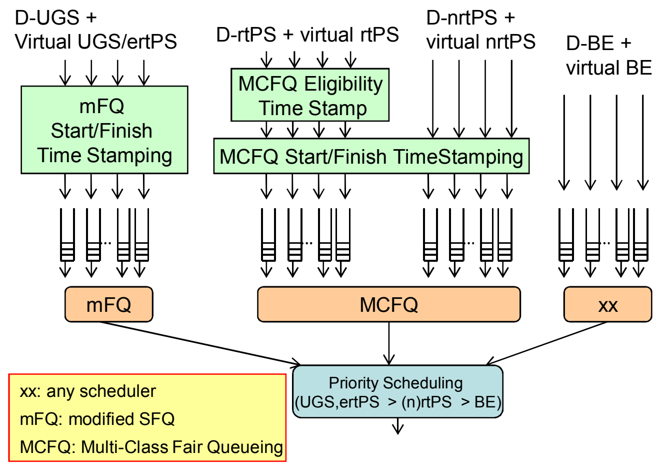

Fig. 3. Inner Scheduler Architecture

Since downlink connections can use any traffic description, including the descriptions specified for the five uplink services, we can classify downlink flows into the same five service types. However, ertPS class uses the same descriptors as rtPS (Table I) and is not needed for the downlink direction. Therefore, we assume 4 service types for the downlink direction, identified as: D-UGS, D-rtPS, D-nrtPS and D-BE. We note that in the centralized scheduler, uplink and downlink services of the same type are grouped together; therefore we drop the D- prefix where there is no distinction. Uplink ertPS class is grouped with UGS due to its requirement for unsolicited UTO grant.

The architecture of the proposed scheduler is presented in Fig. 3. This design uses 2 layers of scheduling. Three fair scheduling algorithms are used for scheduling flows of the same type for classes of combined uplink UGS and ertPS (which have unsolicited UTO grants) and downlink UGS like flows, combined downlink or uplink rtPS and nrtPS flows, and downlink or uplink BE. A priority scheduler is used in the second layer to serve the packet that is selected by the fair scheduling algorithms. The priority scheduler gives the highest priority to UGS/ertPS/D-UGS (by definition, generated UGS/ertPS virtual packets should be immediately served). Combined downlink or uplink rtPS/nrtPS is given higher priority than BE. We emphasize that fairness is provided amongst flows of the same class (e.g., amongst combined downlink or uplink rtPS/nrtPS flows, or amongst combined D-UGS/UGS/ertPS).

The BE traffic is only served when no traffic of other QoS classes exists. Avoiding starvation for BE requires AC and

VPG modules limiting the amount of both uplink traffic assigned to the QoS classes. Uplink traffic shaping is simply achieved by the VPG generating conforming virtual flows; downlink traffic shaping requires a traffic shaper in the BS. Since no QoS guarantee is needed within the BE class, any scheduler can be used amongst BE flows; we use WRR to provide long term fairness amongst BE flows.

UGS flows are scheduled using a fair queuing algorithm. We propose to use a modified Start-time Fair Queuing (SFQ) [13]. The modification we propose is to change the SFQ algorithm to achieve either temporal or throughput fairness, when a multirate PHY is considered. This modification is discussed later in this section.

Since rtPS and nrtPS classes require BW guarantee, we combine these two classes and present a novel multi-class fair queueing algorithm (MCFQ) to provide throughput or BW guarantee for both classes, while providing better delay performance for the rtPS (both uplink and downlink) class. We note that most QoS demanding applications will use these two classes. The MCFQ algorithm is described below.

MCFQ Algorithm

MCFQ is based on the architecture shown in Fig. 3. In MCFQ, similar to SFQ, packets are tagged with start and finish time tags, denoted here as (S_i^k , F_i^k) for packet k of flow i, as follows:

$$S_i^k = \max(F_i^{k-1}, V(t)) \qquad (1)$$
$$F_i^k = S_i^k + L_i^k / \varphi_i$$

where, $V(t)$, is a virtual time function that tracks the overall progress of the system, φ_i are weights assigned to each flow, and L_i^k is the length of the packet.

Additionally, rtPS packets are time stamped with an eligibility time, which is designed to allow a burst of size b_i to be served for rtPS flows ahead of the usual SFQ order, as is explained later. The eligibility time is defined as:

$$E_i^k = S_i^k - b_i / \varphi_i \qquad (2)$$

The eligibility is determined by comparing this value to the virtual time of the system. Thus a packet is deemed eligible at time t if $E_i^k \le V(t)$, and ineligible otherwise. Virtual time is maintained to track the progress of the system, and can be used to simulate a generalized processor sharing (GPS) fluid server (similarly done in WFQ or SFQ). MCFQ adjusts virtual time differently from SFQ.

In MCFQ, in each round of service, the scheduler serves the eligible rtPS packet with the smallest start time; if no eligible rtPS packet is found, any packet with the smallest start time tag is served. The virtual time is set to the start-time tag of the head of line packet with the smallest start time:

$$V(t) = \min\{S_j^k \mid k : \textit{HoL packet of queue } j \ \& \ j \in \textit{all queues}\} \qquad (3)$$

Note that this packet might be an nrtPS packet and not actually in transmission yet; this step makes the algorithm behavior very different from other GPS based schemes.

This choice of virtual time simulates the behavior of SFQ in the long run, although packets are not served in SFQ order in the short term. This operation ensures that the scheduler is fair in the long term and does guarantee bitrate, while at the same time the rtPS flows get much better delay performance compared to when a single class scheduler (e.g., SFQ) is used. Setting the E_i^k according to (2) has the effect that a burst of rtPS, less than b_i in length and arriving at time t, will be served before all nrtPS flows waiting at this time; the reason is that the packets k in the burst will have $E_i^k = S_i^k - b_i / \varphi_i < V(t)$ and are eligible before all other nrtPS packets until $V(t)$ increases with the serving of the next nrtPS or ineligible rtPS flow with higher start time value. As a result, the packets in the burst will only have to compete with other eligible rtPS flows. Packets that arrive immediately following this burst, will not be eligible, since their start time stamp will be ahead of $V(t)$ with more than b_i / φ_i, making them ineligible for service with the current burst of the same flow. The effect is a natural shaping of the flow which will result in long term fairness amongst rtPS and nrtPS flows. A rigorous study of the properties of the MCFQ algorithm follows similar arguments as in [13][15] and will be presented in our upcoming publications.

D. Multirate operation of the PHY and Fairness of Scheduling

The multirate operation of the PHY significantly affects the fairness of the service provided to flows [10][15]. Packets that are transmitted using low transmission rates take longer time on the channel, and lower the overall capacity of the system, negatively affecting other flows. This situation may be deemed unfair if we define fairness in terms of the service time. This type of fairness is also called *temporal fairness* for which the goal is to provide the stations (flows) with the same (weighted) amount of service time regardless of their PHY transmission rate. In contrast to temporal fairness, we can also define fairness in terms of the throughput that is assigned to stations (flows) in a given duration of time (usually long-term); this is called *throughput fairness*. In this case, fairness is achieved if all stations achieve the same (weighted) throughput in a given duration of time [15]. The scheduling process can be designed to provide either temporal or throughput fairness, based on the overall performance objective.

A detailed analysis of algorithms for this purpose, are out of the scope of this paper and can be found in [15]; however for the purpose of providing a complete picture for the scheduling solution, we present the modifications required for fairness provision. To provide an insight, we take the SFQ example and then extend it to MCFQ. For throughput fairness, one can simply use the SFQ algorithm. This algorithm is inherently throughput-fair. For temporal fairness, SFQ can be modified to Start time Temporal Fair Queuing (STFQ) algorithm, by modifying the finish time from (1) to the following:

$$F_i^k = S_i^k + L_i^k / (\varphi_i \cdot C_i^k) \qquad (4)$$

where C_i^k is the PHY transmission rate at which packet k of flow i is transmitted.

The MCFQ algorithm presented in the previous section is by design throughput fair. To convert it to a temporal fair algorithm, we convert the bit-based service tracking parameters of the algorithm to time-based components. For example, similar to STFQ, the finish time equation changes to (4), and the eligibility time equation changes to the following:

$$E_i^k = S_i^k - b_i / (\varphi_i . C_i^k) \quad (5)$$

These changes are enough for converting the algorithm to a temporal fair algorithm, since the progress of each queue, in terms of the received service, is only tracked in the finish time and eligibility time equations.

Although multirate operation provides reliable links most of the time, occasional packet loss is still possible. The lost packets are retransmitted in the MAC. In the case of throughput fair scheduling, the packet transmission is simply retried, and the algorithm remains throughput fair. For temporal fair case, the time stamps of the retransmitted packet is recalculated as if the packet has just arrived to the queue head of line. This readjustment is performed easily for SFQ, since time stamping packets can be done for head of line packets only [14].

E. Admission Control

Any admission control algorithm may be used in the architecture in Fig. 2, without need for modifying architectural design. For example, algorithms based on bitrate reservation may be used. Such algorithms reserve resources to guarantee the minimum bitrate (equal to the average bitrate of the flow), the peak rate (maximum sustained rate), or a value in between, determined based on stochastic guarantees. These algorithms have already been extensively discussed in research literature and are not repeated here [9].

However, we point out that the weight assignment for flows is done in the admission control module. Moreover, the admission control module should be aware of the decision to use temporal or throughput fairness, and act accordingly.

IV. PERFORMANCE EVALUATION

To demonstrate the effectiveness of the proposed solutions we conducted a series of simulation experiments. These experiments highlight the abilities of the proposed solution in providing the following services: 1) fair service guarantee for uplink/downlink flows, 2) fair guaranteed services in multirate networks, 3) providing preferential delay performance for real time (e.g, rtPS) flows, while guaranteeing fair service for nrtPS as well. We implemented a detailed discrete event simulator in C for the 802.16 MAC in TDD mode, and observed the delay and throughput performance. We assumed a multirate PHY (802.16 OFDM). In the first experiment we setup a network with 10 uplink and 10 downlink flows (all of rtPS type). Packets of uniformly distributed size in [500,1500] were generated with exponential inter-arrival time of 5.3 msec (Poisson arrival process). The number of bits served for each flow and the delay performance of the flows are depicted in Fig. 4. The slope of the curve for the number of served bits gives the throughput. It is seen that both type of flows achieve the same throughput. It is also seen that the uplink flows incur more delay, which is expected since uplink packets are only served after a report of their arrival is received by the BS. These performance curves demonstrate that the scheme based on virtual packets is indeed able to guarantee bitrate, despite the fact that virtual packets are generated almost periodically and have a size different from uplink packets.

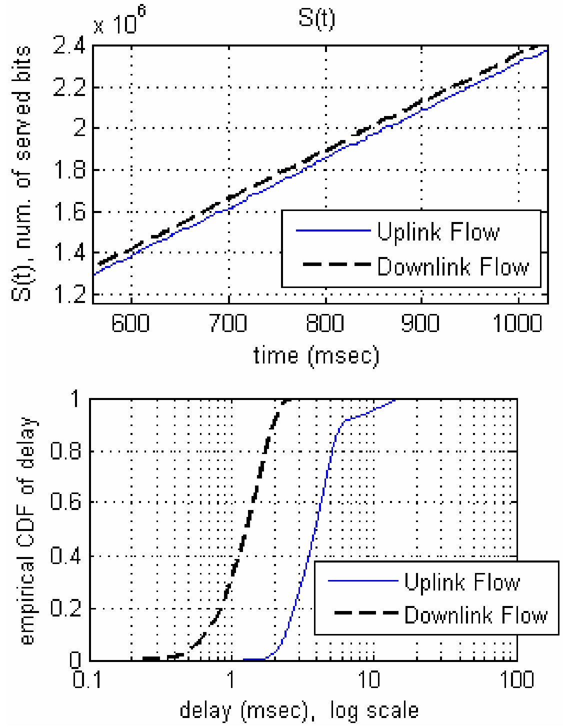

Fig. 4. Throughput and Delay performance, uplink vs. downlink

In the second experiment we evaluated the throughput in a multirate environment for the same flows as in the previous example. Fig. 5 shows the served bits curve for both temporal fair (STFQ) and throughput fair (SFQ) algorithms. STA1, STA2 and STA3 were transmitting at PHY speeds of 64, 32, and 21.6 Mbps, respectively.

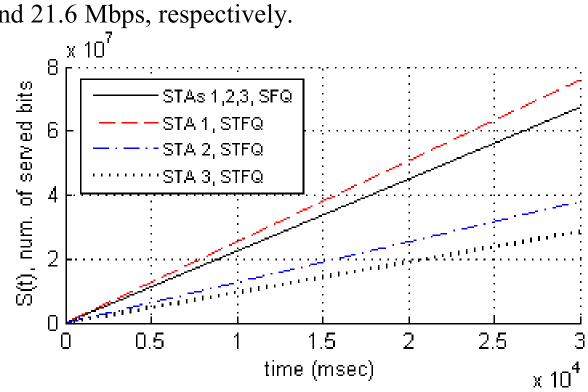

Fig. 5. Throughput Fairness vs. Temporal Fairness (PHY rates of Stations 1,2 and 3: 65, 32.5, 24.4 Mbps)

It is seen in Fig. 5, that using a temporal fair scheduler, each flow gets a throughput proportional to its PHY bitrate. With a throughput fair algorithm, all flows get the same throughput, which is less than the maximum throughput a station would get in the temporal fair case. This reduction in throughput is due to the fact that the reduced capacity is now shared amongst all stations, and not just the stations with lower transmission rates.

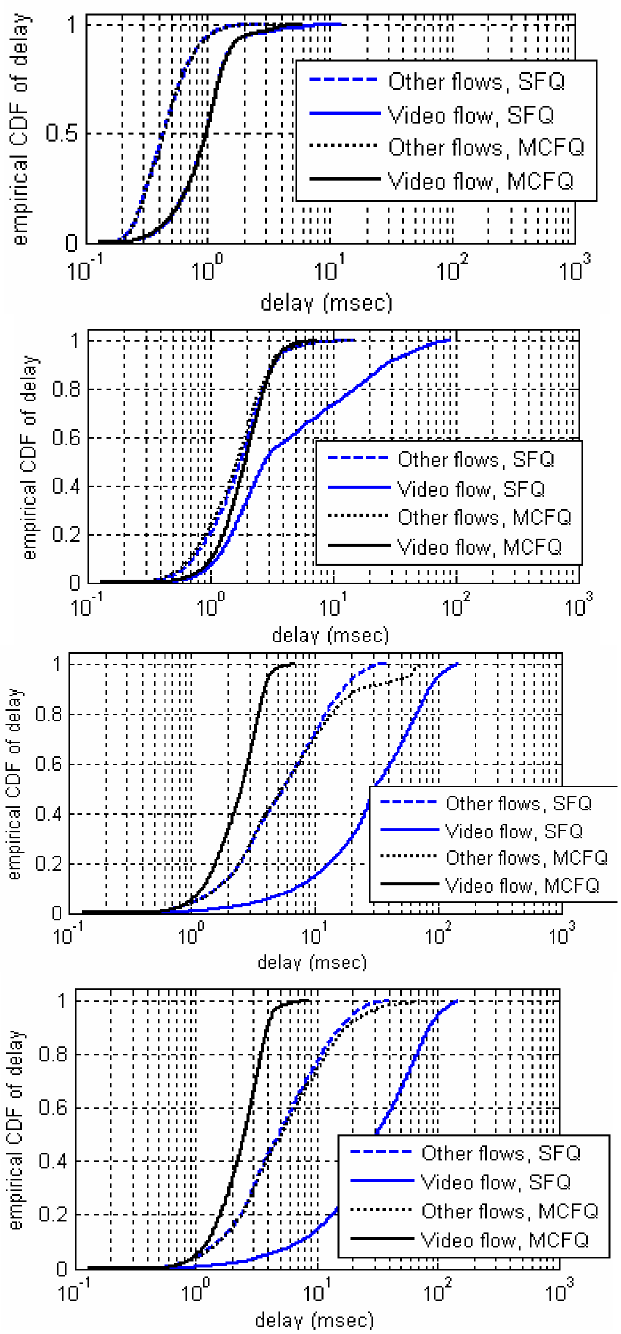

Fig. 6 Delay performance for different load levels, MCFQ vs. SFQ, from top (50%, 75%,95%,100%)

In the last experiment, we observed the delay performance of the multi-class fair scheduler proposed in this paper. We compared the results with those obtained by other fair algorithms such as SFQ. For this experiment we considered a "foreman" H.264 video flow (2Mbps) being served as an rtPS traffic in a network loaded with 19 other nrtPS background traffic. We set one nrtPS flow to 2Mbps to compare it with video; the other flows had a variable load in order to observe the effect of different levels of load on MCFQ and SFQ delay performance. All flows had equal weight (or reserved throughput). Packet sizes for nrtPS flows were set as in the previous example, and a 20KB burst size was used for rtPS flows in MCFQ.

In conventional fair scheduling, all flows are assigned weights in a way that they all receive a fair guaranteed throughput. The results are demonstrated in Fig. 6. Since the video flow has a higher burst size than the other flows, it incurs a larger delay when conventional fair scheduling algorithms like SFQ are used; this is despite the fact that all flows received their requested share of throughput or BW (shown in Fig. 7) in long term. When network is not heavily loaded (50% in Fig. 6) all flows incur small delay. When network gets close to its full capacity (75% and above), MCFQ demonstrates a significant better delay performance for rtPS flows, while still maintaining long term fairness and throughput guarantee for all. The better delay performance for rtPS flow, in MCFQ, comes at the cost of delay increase for nrtPS flows, which is perfectly acceptable since nrtPS flows only require bitrate or BW guarantee. The bottom plot in Fig. 7 shows how rtPS flow in MCFQ receives better service (jump in served bits curve at 1 sec intervals where I-frames are sent), but has to give up the extra service in long run (lower slope and the resulting sawtooth effect on curve).

In Fig. 8 and Fig. 9 we plotted the delay instances of each packet for the 100% load scenario, to see in detail how each packet is incurring delay and that the flows queues are stable (the other nrtPS queues will not be stable since their load is selected to saturate the network to 100%). Fig. 6, Fig. 8 and Fig. 9 all show that the maximum incurred delay for rtPS flows is greatly reduced if MCFQ is used.

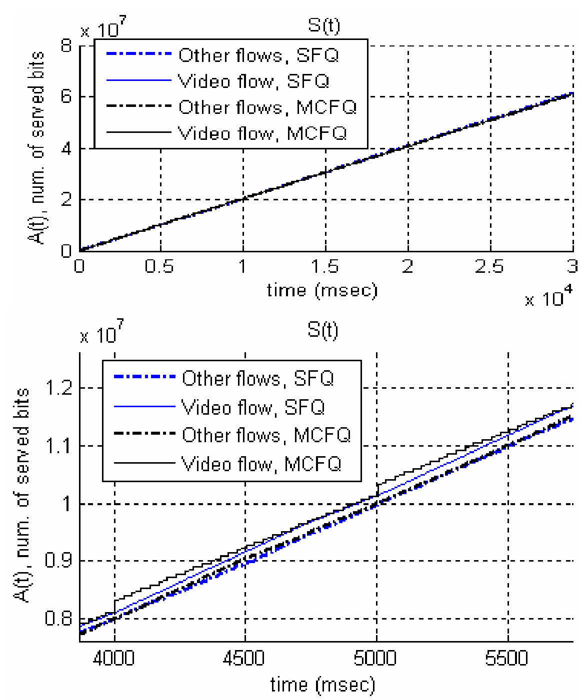

Fig. 7 Served bits for rtPS and nrtPS flows, using MCFQ and SFQ algorithms; top plot (simulation time view), bottom plot (zoomed in)

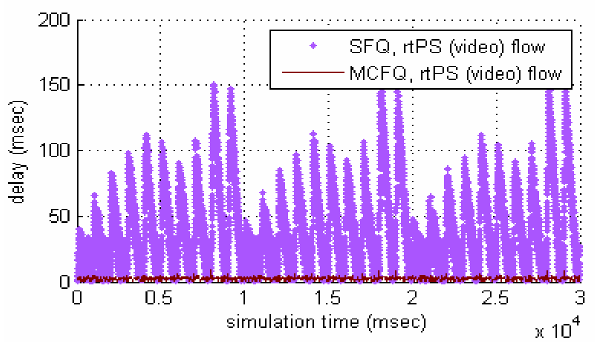

Fig. 8 Packet delay for rtPS video flow, MCFQ vs. SFQ; (background load 100%)

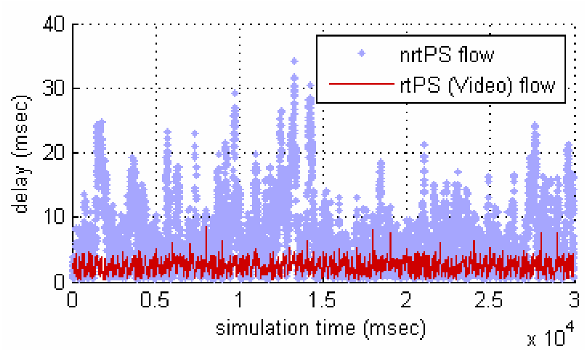

Fig. 9 MCFQ delay instances for one simulation run, nrtPS flows vs. rtPS (video) flow; (load 82%)

V. CONCLUSION

Although the 802.16 standard provides features for QoS provisioning, it intentionally does not mandate or provide a complete QoS solution. Such a solution requires many mechanisms and algorithms that are left to be developed by vendors. In this paper, we presented a comprehensive architecture for QoS provisioning in 802.16 networks. We identified several mechanisms in this architecture that must be implemented before a final QoS solution is available. The main contributions of this paper are the algorithms that are developed as enabling components of the presented architecture.

The multi-class fair scheduling algorithm that has been presented here is able to support different classes, as well as, levels of QoS. We have also presented measures to adjust the scheduling algorithm for temporal or throughput fairness provision in multirate networks. The architectures and algorithms presented in this paper provide a good example for constructing a QoS enabled WiMAX access network. Nevertheless, these algorithms can also be used in other access networks with similar structural requirements (uplink and downlink difference, multirate PHY or different classes of traffic).

REFERENCES

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

[2] IEEE Standard for Local and Metropolitan Area Networks Part 16: Air Interface for Fixed Broadband Wireless Access Systems. Amendment 2: Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands. ANSI/IEEE Std 802.16e-2005

[3] Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications. ANSI/IEEE Std 802.11: 1999 (E) Part 11, ISO/IEC 8802-11, 1999.

[4] M. Hawa and D. W. Petr, "Quality of Service scheduling in cable and broadband wireless accesss systems", *Tenth IEEE International Workshop on Quality of Service*, 15-17 May 2002 Page(s):247 – 255.

[5] C. Cicconetti, A. Erta, L. Lenzini, and E. Mingozzi, "Performance Evaluation of the IEEE 802.16 MAC for QoS Support", *IEEE Transactions on Mobile Computing*, Vol. 6, No. 1, January 2007.

[6] D.-H. Cho, J.-H. Song, M.-S. Kim, and K.-J. Han, "Performance Analysis of the IEEE 802.16 Wireless Metropolitan Area Network," *Proc. First Int'l Conf. Distributed Frameworks for Multimedia Applications (DFMA '05)*, pp. 130-137, Feb. 2005.

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

[8] IEEE Standard 802.11e/ Amendment 8, "Medium Access Control (MAC) Quality of Service (QoS) Enhancements," July 2005

[9] M. Ghaderi and R. Boutaba, "Call admission control for voice/data integration in broadband wireless networks", *IEEE Transactions on Mobile Computing*, Vol. 5, No. 3, March 2006.

[10] Y. Yuan, D. Gu, W. Arbaugh, J. Zhang, " Achieving Fair Scheduling Over Error-Prone Channels in Multirate WLANs ",Wireless Networks, Communications and Mobile Computing, 2005 International Conference on Volume 1, 13-16 June 2005 pp. 698-703

[11] A. Parekh, R. Gallager, "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single-Node Case," IEEE/ACM Trans. on Networking, vol.1, no. 3, pp. 344-357, June 93.

[12] J.C.R. Bennett,Z. Hui," WF^2Q: Worst-Case Fair Weighted Fair Queueing", INFOCOM '96. Fifteenth Annual Joint Conference of the IEEE Computer Societies. Networking the Next Generation. Proceedings IEEE

[13] P. Goyal; H.M. Vin; C. Haichen, "Start-Time Fair Queueing: A Scheduling Algorithm For Integrated Services Packet Switching Networks", Networking, IEEE/ACM Trans. on, Vol 5, Issue 5, Oct. 1997 pp. 690 - 704

[14] Y. Pourmohammadi Fallah, H. Alnuweiri "Hybrid Polling and Contention Access Scheduling in IEEE 802.11e WLANs", *Journal of Parallel and Dist. Comp., Elsevier*, Vol 67, Issue 2, Feb. 2007, pp. 242-256.

[15] Y. Pourmohammadi Fallah, H. Alnuweiri, "Analysis of Temporal and Throughput Fair Scheduling in Multi-Rate IEEE 802.11e WLANs", *Computer Networks*, *Elsevier*, Vol. 52 , Issue 16 , November 2008 ,pp. 3169-3183