Hindawi Publishing Corporation
EURASIP Journal on Wireless Communications and Networking
Volume 2009, Article ID 512507, 14 pages
doi:10.1155/2009/512507

Research Article

Fairness and QoS Guarantees of WiMAX OFDMA Scheduling with Fuzzy Controls

Chao-Lieh Chen,[1] Jeng-Wei Lee,[2] Chi-Yuan Wu,[2] and Yau-Hwang Kuo[2]

[1] *Department of Electronic Engineering, National Kaohsiung First University of Science and Technology, Kaohsiung 811, Taiwan*
[2] *Department of Computer Science & Information Engineering, National Cheng Kung University, Tainan 701, Taiwan*

Correspondence should be addressed to Chao-Lieh Chen, frederic@ieee.org

Received 1 July 2008; Revised 8 December 2008; Accepted 29 December 2008

Recommended by Ekram Hossain

A fairness and QoS guaranteed scheduling approach with fuzzy controls (FQFCs) is proposed for WiMAX OFDMA systems. The controllers, respectively, adjust priority and transmission opportunity (TXOP) for each WiMAX connection according to QoS requirements and service classes. The FQFC provides intra- and interclass fairness guarantees by making connections within the same class achieve equal degree of QoS while at the same time making those without QoS requirements equally share the remaining resources. Even in dynamic environments such as mobile WiMAX networks with time-variant traffic specifications, the FQFC fairly guarantees delay, throughput, and jitter, which are seldom achieved at the same time by state-of-the-art solutions.

1. Introduction

IEEE 802.16 standard (WiMAX) [1, 2] is one of the most popular standards for fixed and mobile broadband wireless access systems to provide last mile access. Due to various users with diverse QoS requirements and wireless communication technologies, the resource scheduler plays an important role to provide fairness and QoS guarantees. As summarized in [3], a resource scheduler in wireless multimedia networks needs to possess the following features: efficient link utilization, delay bound, low implementation complexity, throughput, scalability, and fairness.

For WiMAX and OFDMA systems, various scheduling algorithms have been proposed for achieving QoS guarantees. For example, Liu et al. [4] proposed a priority-based scheduler which assigns each connection a priority updated according to QoS parameters and channel state and then assigns time slots to connections according to the order of priority values. The method has low implementation complexity because the scheduler simply updates the priority of each connection per frame and allocates time slots to the connection with the highest priority. However, it does not consider fairness and jitter issues which are important metrics for real-time applications. To maintain low implementation complexity under considering fairness and jitter, we use the priority-based scheduling scheme for initial priority assignment and afterward, the proposed a fairness and QoS guaranteed scheduling approach with fuzzy controls (FQFC) mechanism takes care of the scheduling job using fuzzy control approach. Many algorithms have been proposed to deal with the fairness problem, and can be briefly divided into two categories. The first category is to reduce the resource allocation problem into an optimization problem. Based on the optimization theory, for example, [5, 6] have good performance on spectrum efficiency and system utilization. They formulate the cross-layer optimization problem to maximize the average utility of all active users subject to certain constraints. However, in addition to implementation complexity, these methods still suffer some problems. To achieve optimal spectrum efficiency, the optimization approaches may, on the other hand, fail to provide QoS guarantees. Moreover, the relation between traffic specifications and network state is uncertain. Uncertainty and dynamics in mobile environment make exact modeling of objective function and constraints impossible when performing the optimization steps. In this paper, the FQFC adopts fuzzy control technique to deal with the modeling problem. The reason we use fuzzy control is to tackle uncertainty and dynamics in wireless communication environment. Among soft computing methods, inference based on probability theory is also widely used for modeling uncertainty and dynamics. However, the controller based on probability must rely on statistical observations to perform inference. Correctness of statistical information is based on the law of large number. In case that gathering large amount of statistic information in short time is difficult, it will be infeasible. Moreover, inference based on probability usually assumes some specific probability model for result of feedback observation to follow. As shown in [7], a single model usually fails to represent the behaviors of dynamic environments such as mobile wireless networks with sudden bursts or changes.

The second category is a utility-based scheduler. A utility function is a measure of relative satisfaction from users' requirements. The schemes in [8, 9] apply utility functions to maximize the total utility of all connections. Utility-based optimization approaches did guarantee QoS of some connections but also starve others. On the other hand, some approaches such as [10, 11] propose utility-fair bandwidth adaptation schemes for multiclass traffic in wireless networks. Rather than achieving resource fairness, the bandwidth adaptation schemes make sure that all connections can obtain similar utility values to achieve the so-called utility fairness. These schemes are effective in both achieving utility fairness and increasing network resource utilization. However, the utility-fair schemes may fail to provide QoS guarantees since it does not consider the priority of the connections.

In this paper, our objective is to provide efficient control for both QoS and fairness guarantees of WiMAX OFDMA scheduling. For QoS guarantees, we address the problem of head-of-line (HOL) delay and jitter control for real time applications and throughput control for nonreal-time applications. The FQFC scheduler assigns each connection a priority and TXOP, and adjusts them according to channel quality, QoS requirements, and service classes. Due to uncertainty and dynamics of the environment, it is difficult to find out the mapping between priority and QoS requirements. For fuzzy inference, it is the simplest way to model a complex system when there is few and uncertain information available. In the field of controller design, fuzzy controller is one of the most popular approaches. Moreover, fuzzy control has been widely used in researches on communication networks such as [7, 12–17]. However, there are few articles talking about using fuzzy control for WiMAX. In this paper, the FQFC model is developed for WiMAX OFDMA systems and is proved that both fairness and QoS are guaranteed. Other fuzzy control methods [14–17] may be proved to achieve certain degree of QoS. However, fairness is seldom assured in these and state-of-the-art approaches. Then, we define two types of fairness including intraclass and interclass fairness. To achieve intraclass fairness, we set up a reference goal to each connection according to the QoS requirements, and make the connections achieve the goal by priority scheduling and TXOP allocation. If each connection can achieve its QoS requirement, intraclass fairness is guaranteed. For achieving interclass fairness, the FQFC does not allocate superfluous resources out of what required. Compared to state-of-the-art methods, connections of high-priority classes release more resources to lower priority ones. Thus, the FQFC makes the connections without QoS requirements evenly share the remaining resources. Based on the priority scheduling and TXOP allocation methods, the FQFC provides both intraclass and interclass fairness with QoS guarantees and featuring low implementation complexity.

This paper is organized as follows. In Section 2, we introduce background including network configuration, MAC QoS, PHY resource allocation, and fairness descriptions. Section 3 describes the FQFC mechanism and depicts the design of the fuzzy controllers for each service class. In Section 4, we investigate the mechanism performance of QoS and fairness through simulations. Finally, we conclude the paper in Section 5.

2. Background

2.1. Network Configuration. WiMAX specifies two communication modes which form different topologies—point-to-multipoint (PMP) and mesh modes. In PMP mode, a base station (BS) centrally allocates downlink (from BS to SS) and uplink (from SS to BS) resources to subscriber stations (SSs). All SSs are only allowed to communicate with a BS. In mesh mode, SS can act as a router to assist its neighbor to relay data. In the 802.16 standard, this mode is optional and is not discussed in this paper. Hence, we focus on proposing a downlink scheduling algorithm to provide QoS guarantees in PMP mode.

IEEE 802.16 WiMAX PHY adopts the orthogonal frequency-division multiple access (OFDMA) technology based on OFDM modulation. The OFDMA technology allows multiple users transmitting packets at the same OFDMA symbol via different subchannels, such that wireless resources are utilized ultimately.

2.2. Scheduling Services in MAC Layer. IEEE 802.16 MAC protocol is connection-oriented; each connection is assigned a connection ID (CID) and a single scheduling service determined by a set of QoS parameters. Four scheduling services in the 802.16 standard are supported: unsolicited grant service (UGS), real-time polling service (rtPS), nonreal-time polling service (nrtPS), and best effort (BE). The UGS supports real-time constant bitrate data streams, such as voice over IP (VoIP) without silence suppression. The QoS parameters of UGS service are minimum reserved traffic rate, the tolerated jitter, maximum latency, and request/transmission policy. The rtPS supports real-time variable-rate data streams, such as MPEG video or VoIP with silence suppression. The QoS parameters of rtPS are maximum latency, request/transmission policy, minimum reserved traffic rate, and traffic priority. The nrtPS supports delay-tolerant variable-rate data streams, such as FTP. The QoS parameters of nrtPS are minimum reserved traffic rate, request/transmission policy, and traffic priority. The

BE supports best-effort data streams. The QoS parameter is request/transmission policy. In IEEE 802.16e [2], an additional service class called extended real-time polling service (ertPS) has superior efficiency than both UGS and rtPS. It supports real-time variable-rate data streams, such as VoIP with silence suppression. The QoS parameters of ertPS are minimum reserved traffic rate, maximum latency, request/transmission policy, and the tolerated jitter. Hence, considering the QoS requirements of the four class services, we calculate the reference goal as traffic specification (TSPEC) according to these QoS parameters.

2.3. Resource Allocations in PHY Layer. IEEE 802.16 OFDMA system defines two types of subcarrier permutations, distributed subcarrier permutation and adjacent subcarrier permutation. The former permutation type includes partially and fully used subcarriers (PUSC and FUSC) which are pseudo-randomly selected and grouped into subchannels, while the later includes adaptive modulation and coding (AMC), and only adjacent subcarriers are clustered to form subchannels. Dispersing noise and interference in fast changing environment, the PUSC and FUSC modes are suitable for mobile networks. For AMC mode, the BS allocates appropriate subchannels for connections with larger SNR to enhance system performance, and it is suitable for fixed or low mobility environment. To support mobile WiMAX, the FQFC scheduling and allocation are based on distributed subcarrier permutation.

In OFDMA, the basic allocation unit is a slot that composes of one subchannel along with an OFDMA symbol, such that the resource allocation becomes a two-dimensional problem. By using the distributed subcarrier permutation, all subchannels are the so-called *equally adequate* for all SSs [18], and our resource allocation is based on Raster algorithm [18], in which the frame is filled row by row, from left to right and from top to bottom, and efficiently reduces the burst numbers.

2.4. Fairness. In wireless networks, the fairness definition is not straightforward. As described in [19], a fair resource allocation usually does not produce equal connection data rate because the diverse connections also suffer from diverse channel conditions, network states, and dynamics. The dynamics result from mobility and time-variant traffic specifications (TSPECs). Moreover, WiMAX needs to provide QoS guarantees for four classes of scheduling services. Therefore, for fairness, it is necessary to consider QoS guarantees for different class connections. We define two types of fairness described as follows.

(i) Intraclass fairness: the connections within the same class achieve equal degree of QoS.

(ii) Interclass fairness: the connections with QoS requirements achieve exactly their demands, and those without QoS requirements equally share the remaining resources.

Hence, our objective is to achieve both intraclass and interclass fairness.

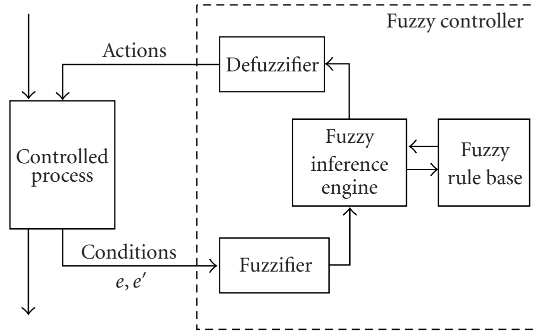

Figure 1: A general architecture of a fuzzy controller.

2.5. Fuzzy Controller. Classical controller requires modeling of the physical reality. This is significant in control problems; however, it is difficult or even impossible to construct precise mathematical models. The difficulty may result from time-variant system behaviors, dynamics, and uncertainty in mobile wireless communication environment. Fuzzy controllers perform well under these circumstances. A general fuzzy controller consists of four components: a fuzzifier, a fuzzy rule base, a fuzzy inference engine, and a defuzzifier. The interconnections among these components and the controlled process are shown in Figure 1. The fuzzifier maps crisp input into appropriate fuzzy sets to express uncertainties. The fuzzy inference engine uses the fuzzified measurements to evaluate the fuzzy implication results. Finally, the defuzzifier deals with confliction of fuzzy implications and transforms the fuzzy implication results back to the crisp output. Two conditions are usually monitored by the controller: error e and the derivative of the error e'. With e and e', the fuzzy controller issues control actions.

3. Design of the Proposed Scheduling Mechanism

In this section, we describe the scheduling mechanism for multiple connections with various QoS requirements. The FQFC scheduler assigns two variables with fuzzy inference values for each connection with CID i, that is, the priority P_i and the maximum number of packets TXOP_i that connection i can transmit in a frame duration. The FQFC scheduler first initializes the two variables based on the characteristics of connections and adjusts them, respectively, by two fuzzy controllers to adapt to the dynamics of system. As shown in Figure 2, the priority controller adjusts P_i according to channel quality, QoS requirements, and service classes. With the priority, the FQFC decides the transmission order of connections. The TXOP controller adapts TXOP_i according to transmission rate and the queue length difference between two contiguous transmissions of the MAC layer.

3.1. Controller Design for ertPS & rtPS. Unlike the UGS class having the highest priority that constant bandwidth can be

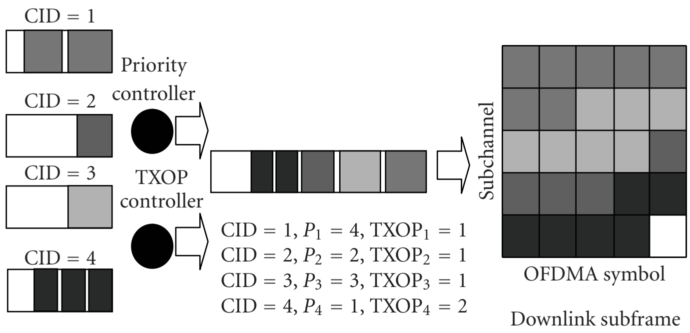

Figure 2: The proposed scheduling mechanism.

achieved by allocating fixed number of slots [1], the two service classes, rtPS and ertPS, that both support the real-time variable bit-rate data streaming require efficient and effective control to achieve QoS guarantees and fairness. A real-time connection of these two classes usually has two QoS specifications, maximum allowable latency (deadline) and jitter. The FQFC control for real-time connections comprises three steps: (1) set up goal delay and tolerable range, (2) adjust priority according to recent HOL delay, and (3) adjust TXOP according to the jitter requirement. The main idea is that the FQFC maintains the delay and jitter of each connection below the delay and jitter goals, respectively. First, we set the goal delay and the tolerable range. Figure 3 shows the control mechanism for real-time connections. Goal delay controller determines goal delay and tolerable range bounded by the lower and upper bounds. Then, the priority controller decides transmission order of connections, and the TXOP controller decides the number of transmitted packets to maintain the jitter. We describe the design of the three controllers as follows.

3.1.1. Goal Delay Controller. The purpose of the goal delay controller is to control delay and jitter within a tolerable range. If the delay exceeds the tolerable range, the FQFC increases the priority. If the delay is below the tolerable range, the FQFC decreases the priority. As shown in Figure 2, to avoid packet dropping, the goal delay is below the deadline. Since system load and transmission rate affect HOL delay obviously, we use them to decide the goal delay. Due to uncertainty and that the TSPEC changes rapidly in mobile WiMAX environment, we cannot use exact formulation to represent the goal delay. Therefore, we divide the delay space into three parts and use fuzzy sets S (small), M (medium), and L (large) to represent these three parts, respectively. Then, we decide which part that the goal delay belongs to according to the system load and transmission rate. The goal delay controller selects a goal delay and sets its upper and lower bounds to form a tolerable range for control. We denote $g_i(t)$, $g_i^{\text{up}}(t)$, and $g_i^{\text{low}}(t)$ as the goal delay of connection i in the tth frame and its upper and lower bounds, respectively. The goal delay controller uses triangular and trapezoidal membership functions as shown in Figure 4. The fuzzy input variables are system load (SL) and transmission

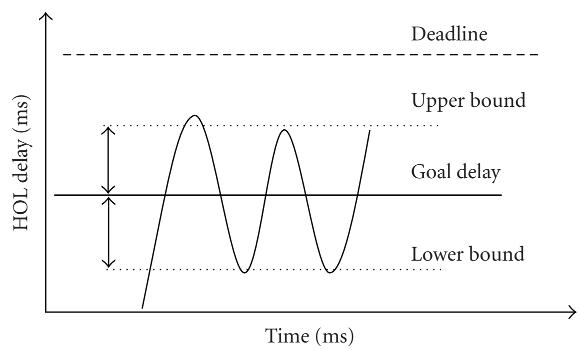

Figure 3: Control mechanism for real-time connection.

rate (TR), and the output function is the goal delay (GD). The fuzzy sets of SL, TR, and GD are defined as follows:

$$T(\text{SL}) = \{\text{Low}, \text{Medium}, \text{High}\} = \{\text{L}, \text{M}, \text{H}\},$$
$$T(\text{TR}) = \{\text{Fast}, \text{Medium}, \text{Slow}\} = \{\text{F}, \text{M}, \text{S}\},$$
$$T(\text{GD}) = \{\text{Small}, \text{Medium}, \text{Large}\} = \{\text{S}, \text{M}, \text{L}\}.$$

According to system load, the controller decides the goal delay $g_i^{\text{load}}(t)$ by the following fuzzy rules.

(R1) If system load is L, then $g_i^{\text{load}}(t)$ is S.

(R2) If system load is M, then $g_i^{\text{load}}(t)$ is M.

(R3) If system load is H, then $g_i^{\text{load}}(t)$ is L.

The following controller uses normalized data rate with respect to the transmission rate in the highest modulation mode to decide the goal delay $g_i^{\text{TX}}(t)$.

(R1) If transmission rate is F, then $g_i^{\text{TX}}(t)$ is S.

(R2) If transmission rate is M, then $g_i^{\text{TX}}(t)$ is M.

(R3) If transmission rate is S, then $g_i^{\text{TX}}(t)$ is L.

Using Mandamni implication and the centroid defuzzifier, we obtain the outputs, $g_i^{\text{load}}(t)$ and $g_i^{\text{TX}}(t)$. Considering system load and transmission rate, the final goal delay is

$$g_i(t) = g_i^{\text{load}}(t) \times w_1 + g_i^{\text{TX}}(t) \times w_2, \tag{1}$$

where w_1 and w_2 are the weighting factors of system load and transmission rate, respectively.

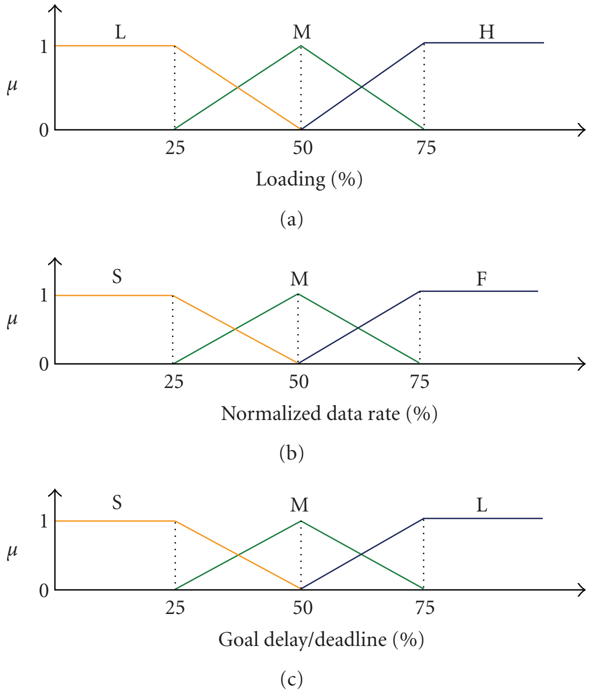

Figure 4: Membership functions of fuzzy sets for goal delay.

With goal delay $g_i(t)$ and required jitter $j_i(t)$, we define $g_i^{\text{up}}(t)$ and $g_i^{\text{low}}(t)$ as the upper and lower bounds of the tolerable range, where $g_i^{\text{up}}(t) = g_i(t) + j_i(t)/2$ and $g_i^{\text{low}}(t) = g_i(t) - j_i(t)/2$.

3.1.2. Priority Controller for Real Time Services. Figure 5 shows the control system including the priority controller, the WiMAX system plant, and the delay observer. The delay observer detects the HOL delay $d_i(t)$. Then, the priority controller compares it with the delay requirement $g_i(t)$, and adjusts priority $P_i(t)$. If $e_i(t) = d_i(t) - g_i(t)$ is around zero, the control system is stabilized around the requirement.

In our design, we denote negative, zero, and positive forces with fuzzy singletons S, M, and L. The control actions of these singletons at the conclusion parts of fuzzy rules are as follows:

$$\begin{aligned}
\text{S: } & P_i(t) = P_i(t-1) - \delta_i(t), \\
\text{M: } & P_i(t) = P_i(t-1), \\
\text{L: } & P_i(t) = P_i(t-1) + \delta_i(t),
\end{aligned}$$

where $\delta_i(t)$ is the priority influence of connection i in the tth frame. The priority controller must confirm that the HOL delay will not exceed the deadline. Hence, it adapts $\delta_i(t)$ according to the time duration between goal delay and the deadline. Let D_i be the deadline, ΔD_i be the guard time before the deadline, P_{rtPS} be the maximum priority of real-time connections, and t_{frame} be the frame duration. Then, we have

$$\delta_i(t) = \frac{P_{\text{rtPS}}}{(D_i - \Delta D_i - g_i(t))/t_{\text{frame}}}. \tag{2}$$

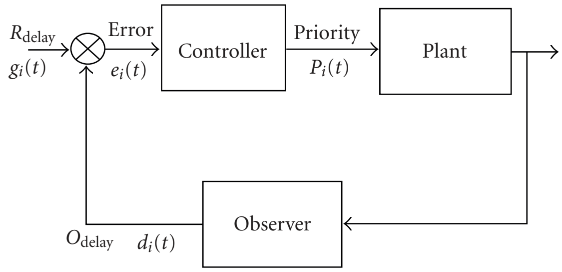

Figure 5: The block diagram of the control system for HOL delay.

As we can see in (2), when the goal delay is closer to the deadline, the adaptation force of the priority is larger. We depict the priority initialization and controlled direction as follows.

(a) *Priority Setting.* When the connection is in an initial stage or the HOL delay is below $g_i^{\text{low}}(t)$, the priority controller assigns the connection a priority according to channel quality, QoS requirement, and service classes. For a real-time connection i, the priority $P_i(t)$ in the tth frame is assigned by (3) which was proposed in [4]:

$$P_i(t) = \begin{cases} P_{\text{rtPS}} \times \dfrac{r_i(t)}{R_i^{\max}} \times \dfrac{1}{F_i(t)}, & \text{if } F_i(t) \geq 1,\ r_i(t) \neq 0, \\ P_{\text{rtPS}}, & \text{if } F_i(t) < 1,\ r_i(t) \neq 0, \\ 0, & \text{if } r_i(t) = 0, \end{cases} \tag{3}$$

where P_{rtPS} is the maximum priority of real-time connections, $R_i^{\max}$ is the data rate of connection i in the highest modulation mode, and $r_i(t)$ is the data rate of connection i in the tth frame. $r_i(t)/R_i^{\max}$ is the normalized data rate and the connection with high received SNR results in higher priority. $F_i(t)$ is the delay requirement indicator:

$$F_i(t) = D_i - \Delta D_i - d_i(t) + 1, \tag{4}$$

where D_i is the deadline, ΔD_i is the guard time before the deadline, and $d_i(t)$ denotes the HOL delay. If $F_i(t) \geqq 1$, the larger $F_i(t)$ denotes the higher satisfaction of delay requirement, which causes lower priority. If $F_i(t) < 1$, the HOL delay has been over the guard time of deadline. The connection should get resources immediately to avoid packet losses. Hence, the priority is set as P_{rtPS}. When $r_i(t)$ is zero, the connection i is under deep fading and should not be scheduled.

(b) *Priority Controller.* Let the controller action be the priority $P_i(t)$. One of the input $e_i(t)$ is the difference between the actual value of the observed HOL delay $d_i(t)$ and the desired value $g_i(t)$, that is, $e_i(t) = d_i(t) - g_i(t)$. The universe of $e_i(t)$ is $[-g_i(t), D_i(t) - g_i(t)]$. The variable $e_i(t)$ has three linguistic values N, E, and P which represent fuzzy concepts "Negative," "Equal," and "Positive," respectively. The fuzzy sets N, E, and P are characterized by the membership functions shown in Figure 6.

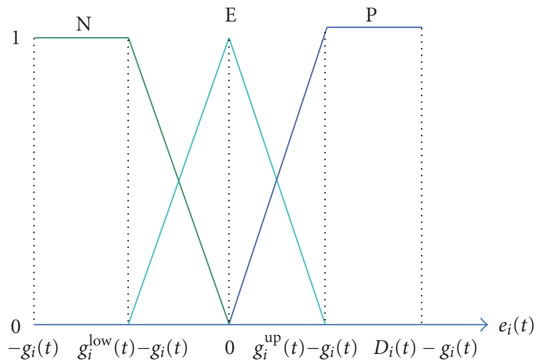

Figure 6: Membership functions of linguistic values for $e_i(t)$.

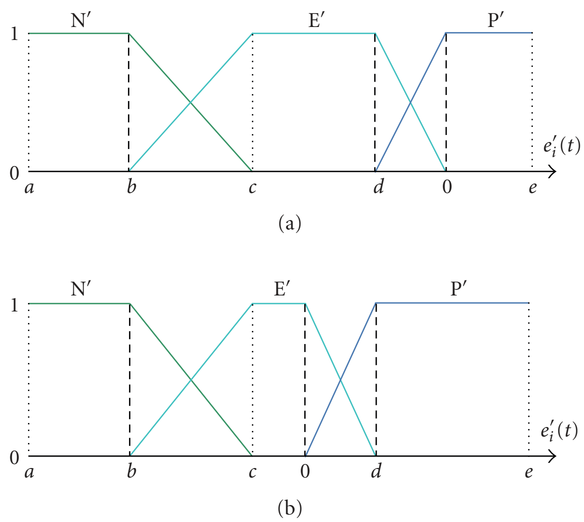

Figure 7: Membership functions of linguistic values for $e_i'(t)$.

The other input of the controller is the difference between two errors, which is defined as $e_i'(t) = e_i(t) - e_i(t-1)$. Substituting $e_i(t) = d_i(t) - g_i(t)$ to $e_i'(t)$, we obtain $e_i'(t) = d_i(t) - d_i(t-1)$. The universe of $e_i'(t)$ is $[-d_i(t-1), D_i(t) - d_i(t-1)]$. The linguistic values of $e_i'(t)$, N′, E′, and P′ also representing fuzzy concepts "Negative," "Equal," and "Positive," respectively, are characterized by the membership functions as shown in Figure 7, where $a = -d_i(t-1)$, $b = g_i^{\text{low}}(t) - d_i(t-1)$, $c = g_i(t) - d_i(t-1)$, $d = g_i^{\text{up}}(t) - d_i(t-1)$, and $e = D_i(t) - d_i(t-1)$. Sign of these values constitutes four cases as shown in Figure 7. The membership functions are time-variant and change along with the variable $d_i(t-1)$.

We consider four cases to design the fuzzy rule base as follows.

Case 1. If HOL delay is too large, that is, $d_i(t-1) > g_i^{\text{up}}(t)$, the priority should be increased with the large (L) step.

Case 2. If $g_i^{\text{up}}(t) > d_i(t-1) > g_i(t)$, maintaining priority at the median (M) level is fine.

Case 3. If $g_i(t) > d_i(t-1) > g_i^{\text{low}}(t)$, maintaining priority at the median (M) level is fine.

Case 4. If HOL delay is too small, that is, $d_i(t-1) < g_i^{\text{low}}(t)$, the priority should be decreased with negative decrement (S).

Therefore, expanding the above cases with changing rate $e_i'(t)$, we have the linguistic inference rules

(R1) If $e_i(t)$ is P and $e_i'(t)$ is P′, then $P_i(t)$ is L,
(R2) If $e_i(t)$ is P and $e_i'(t)$ is E′, then $P_i(t)$ is M,
(R3) If $e_i(t)$ is P and $e_i'(t)$ is N′, then $P_i(t)$ is M,
(R4) If $e_i(t)$ is E and $e_i'(t)$ is P′, then $P_i(t)$ is M,
(R5) If $e_i(t)$ is E and $e_i'(t)$ is E′, then $P_i(t)$ is M,
(R6) If $e_i(t)$ is E and $e_i'(t)$ is N′, then $P_i(t)$ is M,
(R7) If $e_i(t)$ is N and $e_i'(t)$ is P′, then $P_i(t)$ is M,
(R8) If $e_i(t)$ is N and $e_i'(t)$ is E′, then $P_i(t)$ is M,
(R9) If $e_i(t)$ is N and $e_i'(t)$ is N′, then $P_i(t)$ is S.

Using Mandamni implication and the centroid defuzzifier, we obtain the control action responding each HOL delay $d_i(t)$.

The priority controller makes the delay fall in the tolerable range which is below the deadline. Hence, each connection in the real-time class achieves the QoS specification, while intraclass fairness is guaranteed. When the delay is below the tolerable range, the controller decreases the priority for releasing the resources. This scheme guarantees the jitter and interclass fairness at the same time.

(c) *Priority Adaptation for Fairness.* For making the connections within the same class achieve equal degree of QoS, the priority controller adapts $P_i(t)$ by further considering the packet loss rate. All connections should receive the same packet loss rate. To compensate the packet losses in the tth frame, we define the loss rate as $\text{loss}_i(t)$ and the scaling by

$$P_i(t) = \begin{cases} P_i(t), & \text{if } \text{loss}_i(t) = 0, \\ P_i(t) \times \max\left(\dfrac{\lambda}{-\log(\text{loss}_i(t))}, 1\right), & \text{if } \text{loss}_i(t) > 0, \end{cases} \tag{5}$$

where λ is a constant to normalize the loss rate according to the predefined precision. According to (5), if the connection drops packets due to out of the deadline, the priority controller allocates more resources by increasing the priority for achieving intraclass fairness. Even if all connections are in an extremely bad environment, they will suffer the same loss rate.

3.1.3. TXOP Controller. The TXOP controller initiates the TXOP based on frame duration t_{frame} and packet interval of the ith connection t_{pi} as (6)

$$\text{TXOP}_i(0) = \left\lceil \frac{t_{\text{frame}}}{t_{\text{pi}}} \right\rceil. \tag{6}$$

According to deficit round robin [20], the TXOP increases as the number of packets in a queue increases. Let $Q_i(t)$

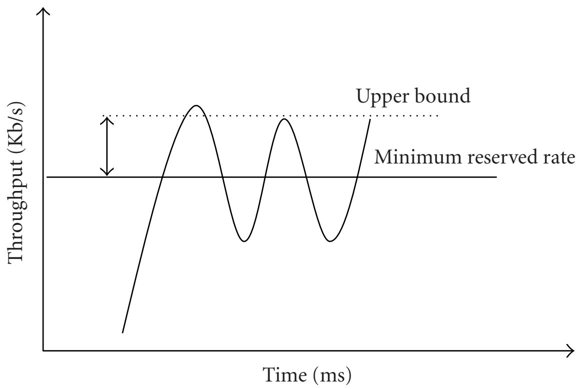

Figure 8: Control mechanism for nrtPS connection.

denote the number of packets in queue i in the tth frame. The controller stores the bounded difference $DC_i(t) = Q_i(t) \ominus Q_i(t-1) = \max(Q_i(t) - Q_i(t-1), -1)$ in the deficit counter of connection i in the tth frame. Then, we add $DC_i(t)$ to $\text{TXOP}_i(t)$as follows:

$$\text{TXOP}_i(t) = \max\left(\text{TXOP}_i(t-1) + DC_i(t), 0\right). \tag{7}$$

To avoid burst transmission, TXOP has an upper bound in

$$\text{TXOP}_i^{\text{up}}(t) = \left\lceil \frac{d_i(t) - g_i^{\text{low}}(t)}{t_{\text{pi}}} \right\rceil. \tag{8}$$

3.2. Controller Design for nrtPS. The nrtPS connection supports delay-tolerant variable-rate data streams and guarantees minimum reserved rate. The control mechanism for nrtPS connections can be divided into three steps: (1) setting up minimum reserved rate and an upper bound, (2) adjusting the priority according to average throughput of nrtPS connections and the required jitter of real-time connections, and (3) adjusting TXOP of nrtPS connections according to the required jitter. Figure 8 shows the control mechanism for nrtPS connection.

If the average throughput is lower than minimum reserved rate, the priority controller raises the priority to increase the throughput. Moreover, the controller needs to prevent large jitter from over-high priority. Besides, if the average throughput exceeds the upper bound, the controller decreases the priority to release the resource. We depict the controller design as follows.

3.2.1. Priority Controller. The priority controller in nrtPS class is easier than in the real-time class. The QoS requirement is only to guarantee the minimum reserved rate. Hence, we do not use fuzzy control and simply use the priority-based scheduler for nrtPS connections. Let $T_i(t)$ denote the average throughput of connection i in the tth frame, and let $R_i(t)$ denote the instantaneous data rate of connection i in the tth frame. The average throughput in the tth frame is usually estimated over a time constant t_c using moving average as

$$T_i(t+1) = \begin{cases} \left(1 - \dfrac{1}{t_c}\right) \times T_i(t) + \dfrac{1}{t_c} \times R_i(t), & \text{if } i = i^*, \\ \left(1 - \dfrac{1}{t_c}\right) \times T_i(t), & \text{if } i \neq i^*, \end{cases} \tag{9}$$

where i^* means connection i is scheduled in the tth frame.

For an nrtPS connection i, the priority $P_i(t)$ in the tth frame is defined as

$$P_i(t) = \begin{cases} P_{\text{nrtPS}} - \delta_i(t), & \text{if } F_i(t) \geq 1,\ r_i(t) \neq 0, \\ P_{\text{nrtPS}}, & \text{if } F_i(t) < 1,\ r_i(t) \neq 0, \\ 0, & \text{if } r_i(t) = 0, \end{cases} \tag{10}$$

where $\delta_i(t)$ is the priority decrement, P_{nrtPS} is the maximum priority of nrtPS connection, and $F_i(t)$ is the throughput requirement indicator which is the ratio of average throughput with respect to the minimum reserved rate $T_i^{\min}$

$$F_i(t) = \frac{T_i(t)}{T_i^{\min}}. \tag{11}$$

If $F_i(t) \geqq 1$, the throughput requirement is satisfied, and the controller decreases the priority to release resource. When $F_i(t) < 1$ implying that the average throughput is less than the minimum reserved rate, the connection should get more resources immediately to achieve the requirement. Hence, at this time, the priority is set to the maximum P_{nrtPS}. The priority decrement $\delta_i(t)$ is further defined as

$$\delta_i(t) = k \times \frac{\text{TXOP}_i(t) \times L_{\text{packet}}}{T_i^{\text{up}} - T_i^{\min}}, \tag{12}$$

where L_{packet} is the packet length, T_i^{up} is the upper bound of $T_i(t)$ which is the maximum sustained rate in the traffic specification, and k is a constant representing system load.

3.2.2. TXOP Setting. For nrtPS, the FQFC sets $\text{TXOP}_i(t)$ according to the throughput upper bound T_i^{up} as

$$\text{TXOP}_i(t) = \left\lceil T_i^{\text{up}} \times \frac{t_{\text{frame}}}{L_{\text{packet}}} \right\rceil. \tag{13}$$

3.2.3. TXOP Adaptation for Fairness. For intraclass fairness, all nrtPS connections should have the same throughput ratio of average throughput with respect to minimum reserved rate. Via setting the upper bound T_i^{up} in (13), we control the average throughput within the range between the minimum reserved rate and the upper bound, and we make the throughput ratio of all nrtPS connections the same. For interclass fairness, the average throughput will not exceed the upper bound. Hence, we can release more resources to the connections without QoS requirements.

3.3. Controller Design for BE

3.3.1. Priority Setting. For a BE connection i, the priority $P_i(t)$ in the tth frame is defined as

$$P_i(t) = \begin{cases} P_{BE}, & \text{if } r_i(t) \neq 0, \\ 0, & \text{if } r_i(t) = 0, \end{cases} \quad (14)$$

where P_{BE} is the maximum priority of BE connection. All BE connections have the same priority. For intraclass fairness, we adopt the round robin scheduling for BE connections.

3.3.2. TXOP Setting. For fair resource allocation, the FQFC sets the $\text{TXOP}_i(t)$ according to the frame duration t_{frame} and the packet interval of the ith connection t_{ip} as (14):

$$\text{TXOP}_i(t) = \left\lceil \frac{t_{\text{frame}}}{t_{\text{ip}}} \right\rceil. \quad (15)$$

In this paper, we also perform priority adaptation. Therefore, the overhead, especially the complexity, will be slightly higher than that of the priority-only method. Since in centralized PMP mode, all traffic flows are managed by base stations which have much more powerful computing ability than SSs, the additional computation overhead will not give any sensibly negative effect. Moreover, the proposed controllers do not use any control/management packets for fairness and QoS purposes. There is no additional network overhead caused by the proposed FQFC.

4. Evaluations and Simulation Results

We first introduce intraclass and interclass fairness criteria and then according to these criteria, we evaluate the performances of the fairness.

4.1. Fairness Criteria. The descriptions of fairness indices are as follows.

4.1.1. Intraclass Fairness Index. Intraclass fairness means that the connections within the same class achieve equal QoS guarantees. Because the connections in different service classes have different QoS requirements, we define respective intraclass fairness indices for real-time, nrtPS, and BE classes.

(a) *Real-Time Connection.* A connection belonging to the real-time class requires strict maximum allowable latency (deadline) and the tolerated jitter. Packet loss occurs when packet delay is out of the deadline. Hence, we use loss rate and jitter to evaluate the intraclass fairness of real-time connections. We define a real-time indicator $I_{\text{RT},i}$ as

$$I_{\text{RT},i} = \begin{cases} 1, & \text{if jitter}_i > \text{jitter}_{\text{tolerated}}, \\ \text{loss}_i, & \text{if jitter}_i \leq \text{jitter}_{\text{tolerated}}, \end{cases} \quad (16)$$

where loss_i, jitter_i, and $\text{jitter}_{\text{tolerated}}$ are the loss rate, jitter, and the tolerated jitter of connection i, respectively. If the jitter is larger than the tolerated jitter, the connection does not achieve QoS guarantees and we set the real-time indicator to one. Otherwise, we set the real-time indicator as the loss rate. Then, we utilize the real-time indicator to compute the real time fairness index. If the real-time indicators of all connections are closer to each other, the better intraclass fairness is achieved. We define the real-time fairness index FI_{RT} as the standard deviation of the real-time indicators of all rtPS connections as follows:

$$\text{FI}_{\text{RT}} = \sqrt{\frac{1}{N_{\text{RT}} - 1} \sum_{j=1}^{N_{\text{RT}}} \left(I_{\text{RT},j} - I_{\text{RT,avg}}\right)^2}, \quad (17)$$

where N_{RT} is the number of connections in the real time class, and $I_{\text{RT,avg}}$ is the average real-time indicator. Thus, a smaller value of FI_{RT} represents better intraclass fairness of the real-time class.

(b) *nrtPS Connection.* A connection belonging to the nrtPS class requires minimum reserved rate. Hence, we use the average throughput to evaluate the intraclass class fairness of nrtPS connections. We define a nrtPS indicator $I_{\text{nRT},i}$ as

$$I_{\text{nRT},i} = \frac{T_i}{T_i^{\min}}, \quad (18)$$

where T_i and $T_i^{\min}$ are the average throughput and minimum reserved rate of connection i, respectively. Then, we introduce the throughput indicator to compute the nrtPS fairness index. The nrtPS fairness index FI_{nRT} is defined as the standard deviation of the throughput indicator of connections in the same nrtPS class as follows:

$$\text{FI}_{\text{nRT}} = \sqrt{\frac{1}{N_{\text{nRT}} - 1} \sum_{j=1}^{N_{\text{nRT}}} \left(I_{\text{nRT},j} - I_{\text{nRT,avg}}\right)^2}, \quad (19)$$

where N_{nRT} is the number of connections in nrtPS class, and $I_{\text{nRT,avg}}$ is the average nrtPS indicator. Similar to the FI_{RT}, a smaller FI_{nRT} value represents better intraclass fairness of the nrtPS class.

(c) *BE Connection.* A connection belonging to BE requires no QoS metrics. We introduce the average throughput to compute the BE fairness index. The BE fairness index is defined as the standard deviation of the average throughput of connections in the same BE class i as follows:

$$\text{FI}_{\text{BE}} = \sqrt{\frac{1}{N_{\text{BE}} - 1} \sum_{j=1}^{N_{\text{BE}}} \left(T_j - T_{\text{avg}}\right)^2}, \quad (20)$$

where N_{BE} is the number of connections in BE class, and T_{avg} is the average throughput in the BE class. Smaller FI_{BE} represents better intraclass fairness of the BE class.

4.1.2. Interclass Fairness Index. According to the definition of interclass fairness, the interclass fairness has two folds: (1) the connections with QoS requirements achieve the demands;

(2) the connections without QoS requirements equally share the remaining resources.

For the first fold, we introduce a requirement indicator $I_{R,i}$ to show the degree of the connection close to the demands as

$$I_{R,i} = e^{-k|x_i - G_i|}, \quad (21)$$

where k is a tunable parameter which determines the tolerable range. x_i and G_i are the average state and the QoS goal of class i, respectively. In the real-time class, the average state is the mean loss rate, and its goal loss rate is zero. In the nrtPS class, the QoS parameter is the average throughput, and the goal is the minimum reserved rate. The smaller the difference between the average state and the QoS goal is, the larger requirement indicator is. When the mean allocated resources for a class are away from the requirement, no matter above or below the goal, the requirement indicator decreases. When the allocated resources reach the requirements exactly, not only the QoS is guaranteed but also the remaining resources are most preserved at the same time.

The BE class has no QoS requirement. For the second part, we introduce Jain's fairness index [21] as the BE fairness index:

$$I_{\text{BE}} = \frac{\left(\sum_{i=1}^{n} T_i\right)^2}{\left(n \cdot \sum_{i=1}^{n} T_i^2\right)}, \quad (22)$$

where n is the number of connections without QoS requirements. The index equals to one indicates perfect fairness in the class without QoS requirements. Then, we utilize the requirement indicator $I_{R,i}$ and the BE fairness index I_{BE} to define the interclass fairness index as follows:

$$\begin{aligned} \text{FI} &= \alpha \times \sum_{i=1}^{m} I_{R,i} w_i + \beta \times I_{\text{BE}}, \\ &\sum_{i=1}^{m} w_i = 1, \\ &\alpha + \beta = 1. \end{aligned} \quad (23)$$

In (23), m is the number of classes with QoS requirements, and w_i is the weighting factor of class i, which determines the importance of the class. α and β are the weighting factors of the classes with and without QoS requirements, respectively. In contrast to the indices of intraclass fairness, a larger FI value indicates better interclass fairness.

4.2. Simulation Configuration. The parameters used in this simulation are listed in Table 1, where OFDM FFT size represents the number of subcarriers an OFDMA symbol composes. The packet length is 1024 bits, and the maximum priority of each service class is $P_{\text{rtPS}} = 1.0$ and $P_{\text{nrtPS}} = 0.8$, and for best effort, $P_{\text{BE}} = 0.6$. The weighting factors w_i, α, β in (23) are all 0.5. Each connection uses a fixed modulation. The FQFC allocates fixed number of time slots to UGS connections. The FQFC adopts persistent resource allocation [1, 22, 23] for UGS service because it has the highest priority. We focus on the performance of real-time, nrtPS, and BE connections. Besides, in our survey, the priority-based scheduler was proposed only for WiMAX OFDM PHY [4]. FQFC outperforms many state-of-the art schedulers for WiMAX OFDM PHY. To present the improvement by FQFC, we modify the priority-based scheduler in [4] to work with WiMAX OFDMA PHY by using the Raster algorithm and regard it as the priority-only scheduler. Then, FQFC fuzzy controllers further improve the fairness and QoS performance of the priority-only scheduler. There are four simulation scenarios as follows.

(i) *Scenario* 1. We set 20 real-time connections. The QoS requirements of real-time connection are the loss rate, deadline, and required jitter. The traffic rates of connections are 8 connections in 1 Mbps, 10 connections in 500 kbps, and 2 connections in 250 kbps. This scenario is to verify the guarantees of maximum latency, the tolerated jitter, and the intraclass fairness in real-time class. It is difficult to find out the mapping between priority and QoS requirements. We prove that the FQFC can efficiently control the delay.

(ii) *Scenario* 2. We set 10 real-time connections and 10 nrtPS connections. The QoS parameter of nrtPS connection is the minimum reserved rate. The traffic rates are 2 real-time connections in 1 Mbps, 8 real-time connections in 500 kbps, 5 nrtPS connections in 1 Mbps, and 5 nrtPS connections in 500 kbps. This scenario is to verify the guarantees of minimum reserved rate and fair resource allocation of the FQFC scheme.

(iii) *Scenario* 3. We set 10 real-time connections, 10 nrtPS connections, and 10 BE connections. BE connection has no QoS requirement. The traffic rates are 1 real-time connection in 1 Mbps, 9 real-time connections in 500 kbps, 3 nrtPS connections in 750 kbps, 2 nrtPS connections in 500 kbps, 5 nrtPS connections in 1 Mbps, and 10 BE connections in 100 kbps. This scenario is to verify the fair resource allocation of FQFC.

(iv) *Scenario* 4. In this scenario, we simulate the wireless link degrades. This will cause the modulation to change. The experiment is designed to test the robustness of the FQFC whether it can efficiently track the goal delay when the channel quality degrades. The simulated network consists of 1 BS and 10 SS (numbered from 1 to 10). In the downlink, each SS with number i ($i = 1 \sim 10$) has 1 real-time, 1 nrtPS, and 1 BE connection with CID i, $10 + i$, $20 + i$, respectively. The connections from SS1 to SS5 apply with QPSK modulation, and connections from SS6 to SS7 apply with 16-QAM modulation. All the other connections initially adopt 64-QAM modulation. This is for simulating the different channel conditions.

Table 1: Simulation parameters.

Parameter	Value
System bandwidth	10 MHz
Frame duration	5 ms
OFDMA FFT size	1024
Number of subchannels	30
Number of OFDMA symbols for DL	28

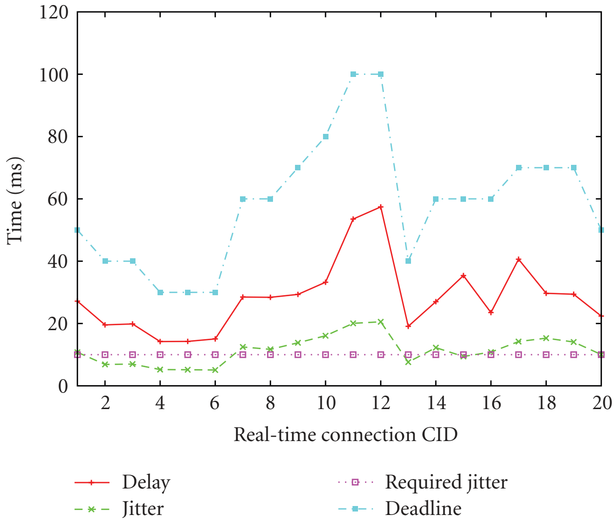

Figure 9: Delay and jitter performances of real-time connections in priority-only scheduler.

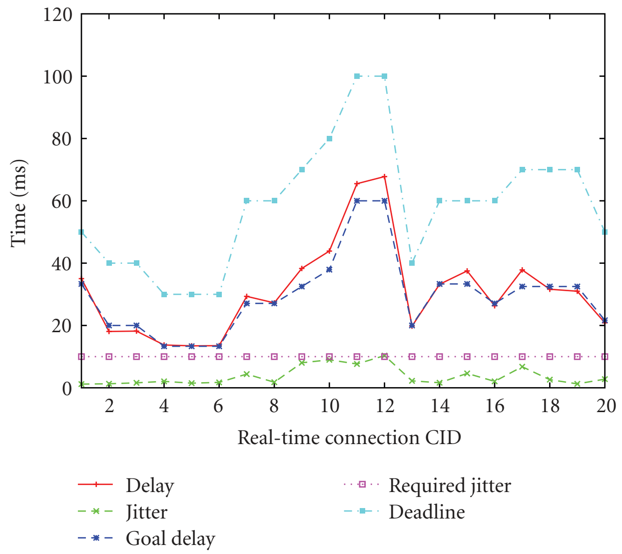

Figure 10: Delay and jitter performances of real-time connections in FQFC.

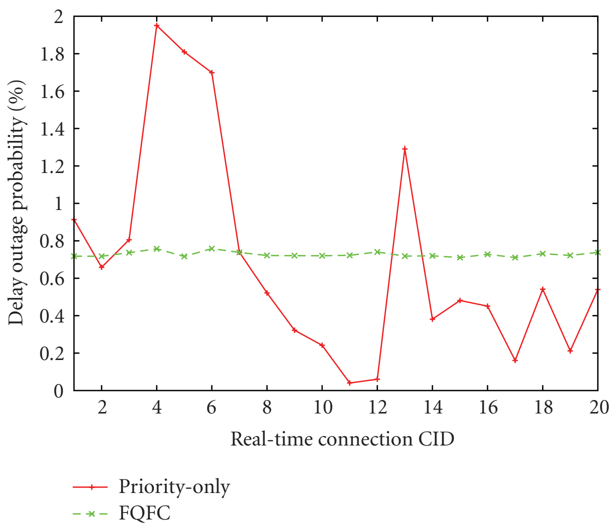

Figure 11: Average outage probability of rtPS connections.

4.3. Performance Evaluation for Fairness and QoS Guarantees

4.3.1. Scenario 1: Intraclass Fairness and QoS Guarantees of Real Time Connections. We compare the FQFC with the priority-only scheduler [4]. The QoS is in terms of average delay, delay jitter, and packet loss rate. As illustrated in Figure 9, although the priority-only scheduler controls delay of connection to be below the deadline, it cannot guarantee tolerant jitter. Under the same simulation conditions, FQFC guarantees both delay and jitter requirements as shown in Figure 10. The average delay is close to the goal delay. The result also shows that it is useful by controlling the HOL delay in the tolerable range to guarantee the required jitter.

For intraclass fairness evaluation, from Figure 10, we can see that the jitter of the connections using FQFC is still smaller than the tolerated jitter in Figure 9. Figure 11 shows the delay outage probabilities of the FQFC and the priority-only scheduler. The FQFC disperses the outage probability for intraclass fairness. Moreover, as summarized in Table 2, the intraclass fairness index of the FQFC is much lower than the one of the priority-only scheduler and is almost near to zero. Hence, the FQFC guarantees the intraclass fairness for real-time connections.

Table 2: Intraclass fairness index.

		FQFC	Priority-only
Scenario 1	Real-time	0.000131	0.504639
Scenario 2	Real-time	0	0.489360
	nrtPS	0.012748	0.081995
Scenario 3	Real-time	0	0.502625
	nrtPS	0.003088	0.107002
	BE	6.686637	84.312975

4.3.2. Scenario 2: Intraclass Fairness and QoS Guarantees of Real-Time and nrtPS Connections. For QoS evaluation, we introduce the throughput indicator defined as the ratio of the average throughput with respect to the minimum reserved rate. In Figures 12 and 13, even adding nrtPS connections, the FQFC still guarantees the delay and jitter specifications of real-time connections. Then, we evaluate nrtPS connections

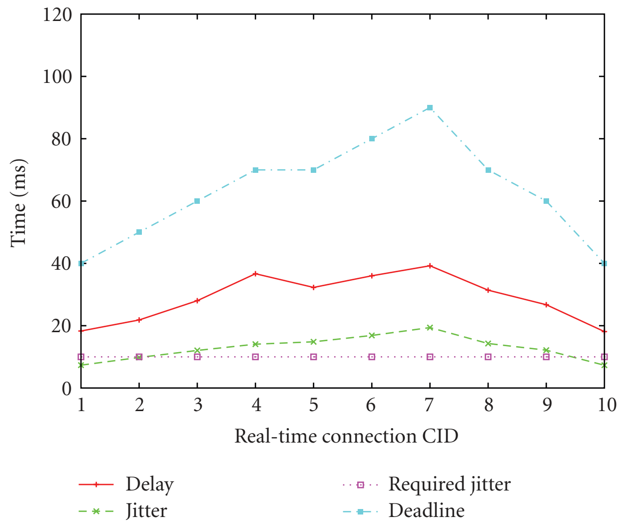

Figure 12: Delay and jitter performances of real-time connections in priority-only scheduler.

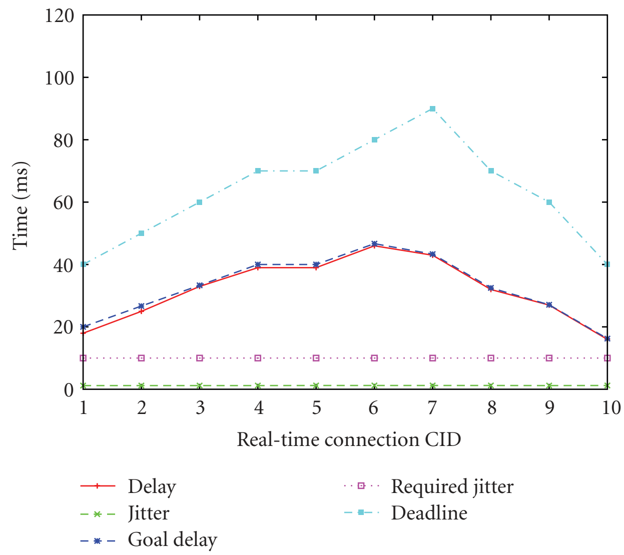

Figure 13: Delay and jitter performances of real-time connections in FQFC.

by throughput indicators. Figure 14 shows that all nrtPS connections with FQFC control keep their throughput indicators almost the same about 1.15. The result means that the FQFC guarantees minimum reserved rate. For the connections with priority-only-based scheduler [4], the throughput indicators of the last four nrtPS connections are higher than the others since priority-only-based scheduler provides more resources to the connections using high-bitrate modulation. The FQFC focuses on making the connections of the same class achieve the equal degree of QoS. As illustrated in Table 2, for nrtPS, the intraclass fairness index of the FQFC is close to zero which is much lower than the one of the priority-only scheduler. Hence, the FQFC also guarantees the intraclass fairness for the connections of the nrtPS class.

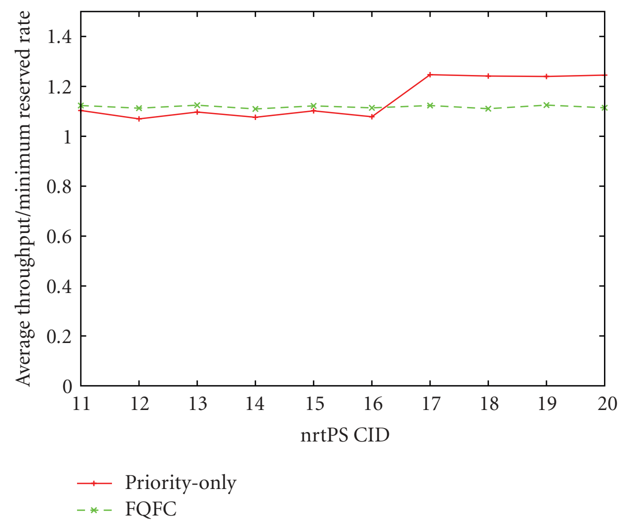

Figure 14: Average throughput/minimum reserved rate of nrtPS connections.

4.3.3. Scenario 3: Intra- and Interclass Fairness and QoS Guarantees of All Classes. For QoS evaluation, Figures 15, 16, and 17 show that the FQFC guarantees the delay and jitter of real-time connections as well as guarantees the throughput of nrtPS connections. Even for users with diverse QoS requirements, the FQFC still provides QoS guarantees. For BE connections, although they have no QoS requirement, the remnant resources should be fairly allocated to all BE connections. In Figure 18, BE connections under FQFC control obtain throughputs and are not starved.

For intraclass fairness evaluation of the BE class, we compare the FQFC with the priority-only scheduler regarding average throughput. Figure 18 shows that the average throughputs of all BE connections under the FQFC control are nearly the same. The priority-only scheduler provides more resource to the last four BE connections since they employ higher rate modulation. Table 2 shows that the intraclass fairness index of the FQFC is close to zero, which is much lower than the one of the priority-only scheduler. For every real-time connection, FQFC sets the goal delay below the deadline for a certain distance in terms of the tolerable jitter. Since the goal is for priority and TXOP controllers to follow, intraclass fairness is achieved when real-time connections have almost the same loss rate and jitter performances based on the intraclass fairness criteria. For nrtPS connections, the FQFC control algorithm maintains their ratios of throughput achievement over minimum reserved rate as close to 1 as possible. Again, as long as BE connections can evenly share the remained resources from real-time and nrtPS connections, intraclass fairness of BE connections is achieved. Hence, the FQFC guarantees the intraclass fairness for the BE classes.

For interclass fairness evaluation, in Figure 17, the throughput indicator of the FQFC is lower than the one of the priority-only scheduler since the FQFC always preserves

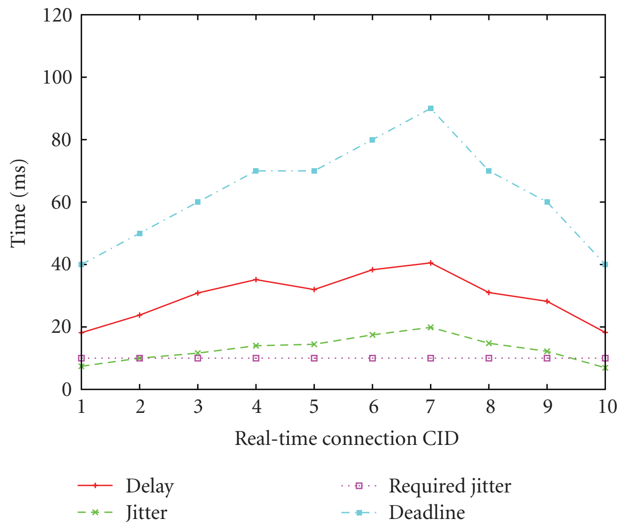

Figure 15: Delay and jitter performances of real-time connections in the priority-only scheduler.

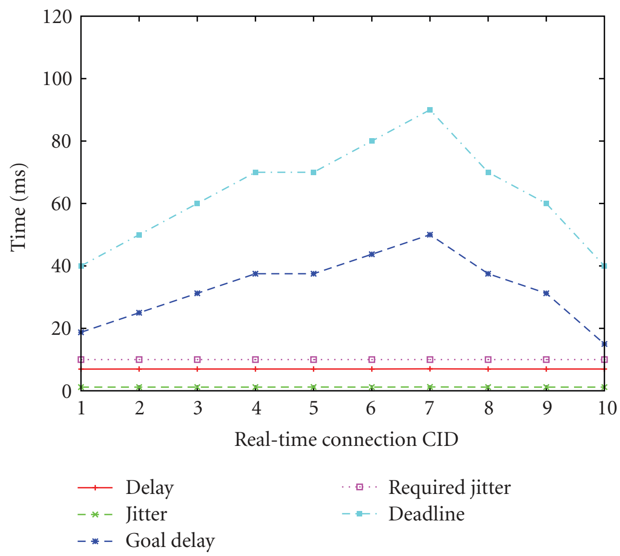

Figure 16: Delay and jitter performances of real-time connections in FQFC.

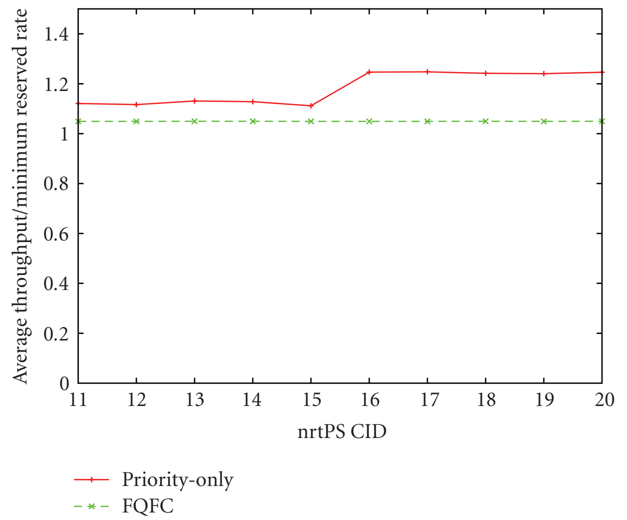

Figure 17: Average throughput/minimum reserved rate of nrtPS connections.

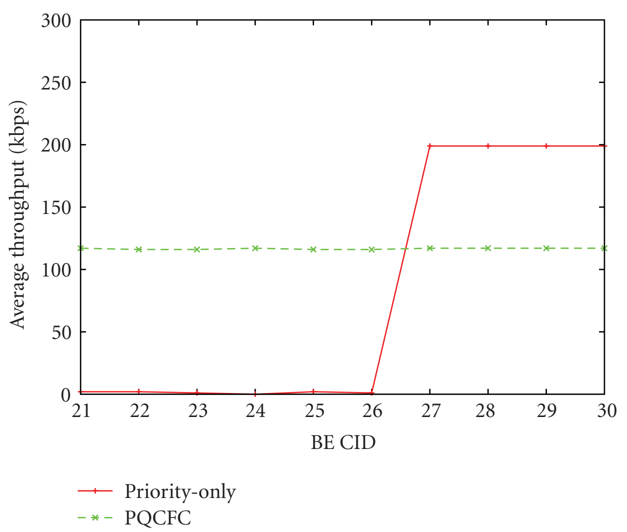

Figure 18: Average throughput of BE connections.

resources for lower priority classes. This causes the BE connections get more resources. For interclass fairness comparison, the FQFC outperforms priority-only scheduler as shown in Figure 18. Table 3 shows that the interclass fairness index of the FQFC is close to one. Hence, in addition to intraclass fairness, the FQFC also guarantees the interclass fairness. For priority-only scheduler, every real-time connection grabs as many channel resources as possible. Though delays can be lower than the deadlines, delay and jitter differences among connections are not maintained. For nrtPS connections, the differences of throughput ratios are not controlled in priority-only scheduler. The differences of channel resources grasped by the BE connections are, therefore, obvious.

Table 3: Interclass fairness index.

	FQFC	Priority-only
Scenario 3	0.994669	0.677405

4.3.4. Scenario 4: Link Degradation. In this scenario, we evaluate the robustness of the FQFC against wireless link degradation. At the 4.0th second, the wireless link from BS to SS3 degrades, and the PHY layer adaptation mechanism changes the modulation over this link from 64-QAM to QPSK. At the 6.0th second, this link recovers to 64-QAM. Figures 19 and 20, respectively, show the PDU delay of real-time connection 3 and the average throughput of nrtPS connections 13 in SS3, where the link degradation occurs at the 4.0th second. Figure 20 also shows the average throughput of nrtPS connection 13 which is an external connection

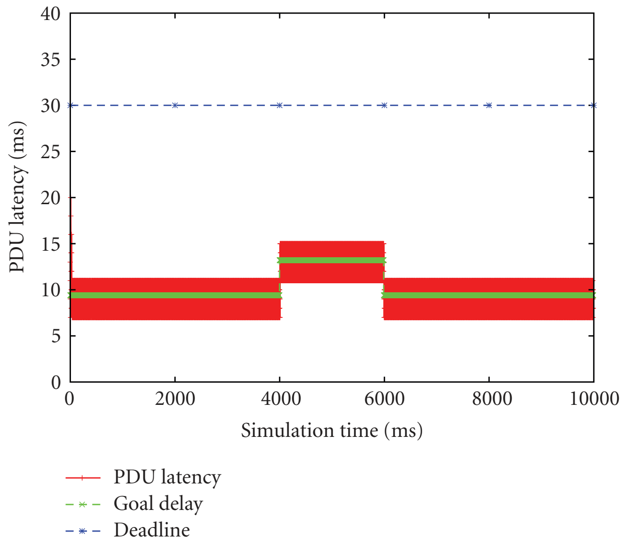

FIGURE 19: PDU delay of real-time connection 3.

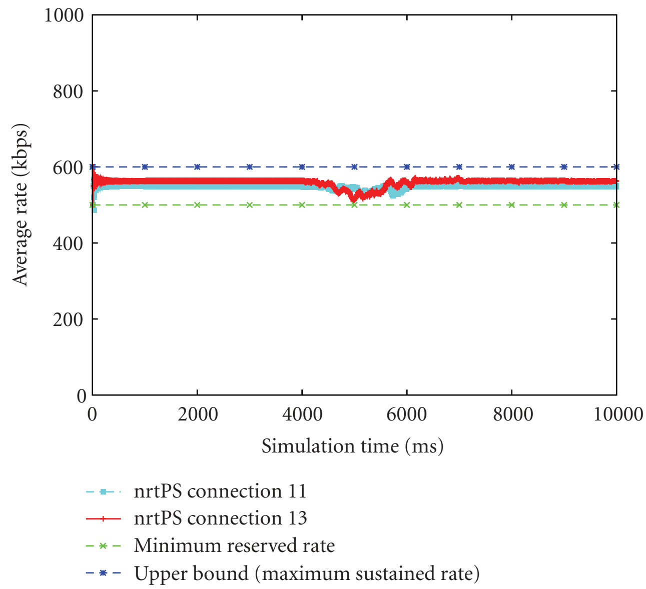

FIGURE 20: Average throughput of nrtPS connections 11 and 13.

out of SS3. Figure 21 shows the average throughput of BE connections 23 and 25. The simulation shows that

(i) when the link degradation occurs, the FQFC adjusts the goal delay and the tolerable range according to the updated modulation. The FQFC continues to make the delay of real-time connection 3 fall in the tolerable range as shown in Figure 19. Hence, the FQFC can efficiently control the delay according to the goal delay and the tolerable range;

(ii) the service curves of nrtPS connections 11 and 13 in Figure 20 distinguish a throughput drop from the 4.0th second to the 6.0th second, whereas FQFC still maintains the throughput to meet the QoS requirements. The service curves of BE connections 23 and 25 in Figure 21 also distinguish a throughput drop from the 4.0th second to the 6.0th second. The resources are released to guarantee the QoS of real-time connection 7 as shown in Figure 19. For intraclass fairness in nrtPS connections and BE connections, all nrtPS connections keep almost the same resource usage ratio. For interclass fairness, nrtPS and BE connections release resources to guarantee the QoS of real-time connections. Hence, the FQFC guarantees both QoS and fairness even in case that wireless link degrades.

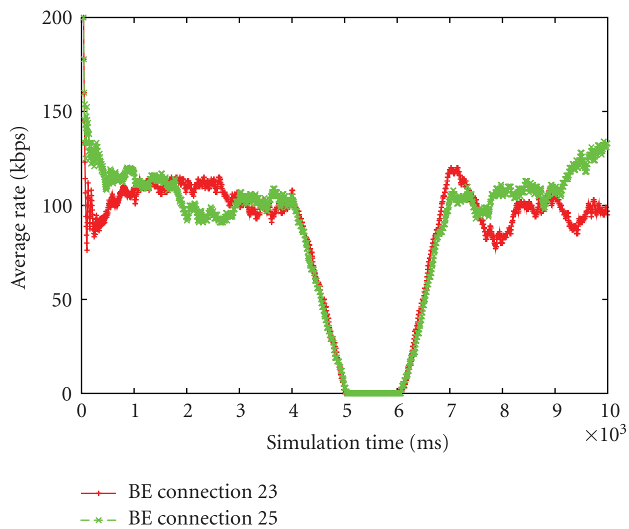

FIGURE 21: Average throughput of BE connections 23 and 25.

5. Conclusions

A fairness and QoS guaranteed scheduling approach with fuzzy controls FQFC algorithm is proposed for WiMAX OFDMA systems. Different from the utility-fairness, new fairness and QoS evaluation criteria in terms of loss rate, jitter, and throughput are proposed for different classes. The proposed FQFC scheme controls the delay, jitter, and throughput QoS parameters efficiently providing both fairness and QoS guarantees. Rather than using hard computation approaches such as utility-based optimizations, we use fuzzy controller to perform scheduling and resource allocations to resolve mapping among priority, transmission opportunity, and QoS requirements. The proposed FQFC scheme provides both intra- and interclass fairness guarantees in addition to QoS guarantees while implementation is with low complexity.

References

[1] IEEE Std. 802.16-2004 (Revision of IEEE Std. 802.16-2001), "IEEE standard for local and metropolitan area networks—part 16: air interface for fixed broadband wireless access systems," Revision of IEEE Std. 802.16-2001, October 2004.

[2] IEEE 802.16e-2005, "IEEE Standard for Local and metropolitan area networks—part 16: air interface for fixed and mobile broadband wireless access systems amendment 2: physical and medium access control layers for combined fixed and mobile operation in licensed bands and corrigendum," February 2006.

[3] H. Fattah and C. Leung, "An overview of scheduling algorithms in wireless multimedia networks," *IEEE Wireless Communications*, vol. 9, no. 5, pp. 76–83, 2002.

[4] Q. Liu, X. Wang, and G. B. Giannakis, "A cross-layer scheduling algorithm with QoS support in wireless networks," *IEEE Transactions on Vehicular Technology*, vol. 55, no. 3, pp. 839–847, 2006.

[5] G. Song and Y. Li, "Cross-layer optimization for OFDM wireless networks—part I: theoretical framework," *IEEE Transactions on Wireless Communications*, vol. 4, no. 2, pp. 614–624, 2005.

[6] M. Ergen, S. Coleri, and P. Varaiya, "QoS aware adaptive resource allocation techniques for fair scheduling in OFDMA based broadband wireless access systems," *IEEE Transactions on Broadcasting*, vol. 49, no. 4, pp. 362–370, 2003.

[7] C.-L. Chen, J.-W. Lee, S.-Y. Chen, and Y.-H. Kuo, "Hierarchical cross-layer fuzzy control for compromise of multiple objectives in wireless mobile networks," in *Proceedings of the International Conference on Mobile Technology, Applications, and Systems (Mobility '08)*, pp. 1–7, Yilan, Taiwan, September 2008.

[8] D. Zheng and J. Zhang, "A two-phase utility maximization framework for wireless medium access control," *IEEE Transactions on Wireless Communications*, vol. 6, no. 12, pp. 4299–4307, 2007.

[9] K.-H. Liu, L. Cai, and X. Shen, "Multiclass utility-based scheduling for UWB networks," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 2, pp. 1176–1187, 2008.

[10] N. Lu, J. Bigham, and N. Nasser, "An intra-class and interclass utility-fair bandwidth adaptation algorithm for multiclass traffic in wireless networks," in *Proceedings of the Asia-Pacific Conference on Communications (APCC '06)*, pp. 1–5, Busan, Korea, August-September 2006.

[11] N. Lu and J. Bigham, "On utility-fair bandwidth adaptation for multi-class traffic QoS provisioning in wireless networks," *Computer Networks*, vol. 51, no. 10, pp. 2554–2564, 2007.

[12] C.-L. Chen, "IEEE 802.11e EDCA QoS provisioning with dynamic fuzzy control and cross-layer interface," in *Proceedings of the 16th International Conference on Computer Communications and Networks (ICCCN '07)*, pp. 766–771, Honolulu, Hawaii, USA, August 2007.

[13] C.-L. Chen, "Morphisms from IEEE 802.11 DCF specifications to its EDCA QoS practice with cross-layer interface," in *Proceedings of the 13th International Conference on Parallel and Distributed Systems (ICPADS '07)*, vol. 2, pp. 1–8, Hsinchu, Taiwan, December 2007.

[14] C. Douligeris and G. Develekos, "Neuro-fuzzy control in ATM networks," *IEEE Communications Magazine*, vol. 35, no. 5, pp. 154–162, 1997.

[15] I. W. Habib, "Applications of neurocomputing in traffic management of ATM networks," *Proceedings of the IEEE*, vol. 84, no. 10, pp. 1430–1441, 1996.

[16] K. Yang, J. Zhang, and H.-H. Chen, "A flexible QoS-aware service gateway for heterogeneous wireless networks," *IEEE Network*, vol. 21, no. 2, pp. 6–12, 2007.

[17] H. B. Kazemian and L. Meng, "Neuro-fuzzy control for MPEG video transmission over bluetooth," *IEEE Transactions on Systems, Man and Cybernetics, Part C*, vol. 36, no. 6, pp. 761–771, 2006.

[18] Y. Ben-Shimol, I. Kitroser, and Y. Dinitz, "Two-dimensional mapping for wireless OFDMA systems," *IEEE Transactions on Broadcasting*, vol. 52, no. 3, pp. 388–396, 2006.

[19] C. Wengerter, J. Ohlhorst, and A. G. E. von Elbwart, "Fairness and throughput analysis for generalized proportional fair frequency scheduling in OFDMA," in *Proceedings of the 61st IEEE Vehicular Technology Conference (VTC '05)*, vol. 3, pp. 1903–1907, Stockholm, Sweden, May-June 2005.

[20] M. Shreedhar and G. Varghese, "Efficient fair queuing using deficit round-robin," *IEEE/ACM Transactions on Networking*, vol. 4, no. 3, pp. 375–385, 1996.

[21] R. Jain, D. Chiu, and W. Hawe, "A quantitative measure of fairness and discrimination for resource allocation in shared computer systems," DEC Research Report TR-301, Digital Equipment, Littleton, Mass, USA, September 1984.

[22] J. Freitag and N. L. S. da Fonseca, "Uplink scheduling with quality of service in IEEE 802.16 networks," in *Proceedings of the 50th Annual IEEE Global Telecommunications Conference (GLOBECOM '07)*, pp. 2503–2508, Washington, DC, USA, November 2007.

[23] M.-H. Fong, R. Novak, S. McBeath, and R. Srinivasan, "Improved VoIP capacity in mobile WiMAX systems using persistent resource allocation," *IEEE Communications Magazine*, vol. 46, no. 10, pp. 50–57, 2008.