Priority-Based Scheduling Algorithm for Downlink Traffics in IEEE 802.16 Networks

Jia-Ming Liang*, Jen-Jee Chen*, You-Chiun Wang*, Yu-Chee Tseng*[†], and Bao-Shuh P. Lin*[‡]
*Department of Computer Science
National Chiao-Tung University, Hsin-Chu 30010, Taiwan
[†]Department of Information and Computer Engineering
Chung-Yuan Christian University, Chung-Li, Tao-Yuan 32023, Taiwan
[‡]Information and Communications Research Laboratories
Industrial Technology Research Institute, Chu-Tung, Hsin-Chu 31040, Taiwan
Email: {jmliang, chencz, wangyc, yctseng}@cs.nctu.edu.tw, bplin@itri.org.tw

***Abstract*—The IEEE 802.16 standard is proposed to provide a wide-range broadband wireless service, but it leaves the implementation of the wireless resource scheduler as an open issue. We thus propose a *priority-based scheduling algorithm* to arrange resources for downlink traffics in an IEEE 802.16 broadband wireless network. The idea is to differentiate the mobile subscriber stations with good channel conditions from those with bad channel conditions, and to differentiate the urgent real-time traffics from the non-real-time ones. Thus, the network throughput can be improved while the delays of real-time traffics can be alleviated. In addition, our design also keeps fairness in mind, so non-real-time traffics will not be starved. Simulation results show that our scheduling algorithm can increase the network throughput, maintain the long-term fairness, and keep packet dropping ratios of real-time traffics low, as compared with existing results.**

***Index Terms*—IEEE 802.16, fair scheduling, quality of service (QoS), resource management, WiMAX.**

I. Introduction

Recently, the IEEE 802.16 standard [1], [2] has been proposed to support wide-range broadband wireless access. The standard's objective is to use a more flexible and economical way to solve the last mile problem in a metropolitan area network, as compared with traditional wired access networks such as fiber optics or T1 links [3], [4]. IEEE 802.16 supports the *point-to-multipoint (PMP)* mode, where one *base station (BS)* can directly communicate with several *mobile subscriber stations (MSSs)*. The BS will manage network resources for these MSSs. Based on the standard, the resource unit is defined by physical layer specifications. In this paper, we use *slot* as the resource unit which is defined by the mandatory physical layer specification, called *orthogonal frequency division multiplexing (OFDM)* and *orthogonal frequency division multiplexing access (OFDMA)*.

The IEEE 802.16 standard also defines five types of scheduling services to support QoS (quality of service). They are *unsolicited grant service (UGS)*, *real-time polling service (rtPS)*, *extended rtPS (ertPS)*, *non-real-time polling service (nrtPS)*, and *best effort (BE)*. Briefly, these five types of scheduling services can be classified into real-time services (including UGS, rtPS, and ertPS) and non-real-time services (including nrtPS and BE). In the IEEE 802.16 MAC layer, a scheduler is defined to manage wireless resources for these services. However, how to implement the scheduler leaves an open issue in the standard. Therefore, in this paper, we propose a scheduling algorithm to arrange resources for downlink traffics in an IEEE 802.16 broadband wireless network.

In the literature, several studies also consider scheduling downlink traffics in an IEEE 802.16 broadband wireless network. The work in [5] proposes a *modified proportional fair (MPF)* method to increase the network throughput while maintaining fairness. In [6], a utility function is proposed to evaluate the tradeoff between network throughput and fairness. In [7], a proportional fairness scheme based on signal-to-noise ratio is proposed to achieve rate maximization. However, the above studies consider only non-real-time traffics. The work in [8] assigns priorities to different traffics to satisfy their QoS requirements, but it does not consider the fairness issue. The work in [9] models the scheduling problem as an M/M/1/K queuing system, whose objective is to minimize the blocking probability. However, it may not guarantee the delays of real-time traffics.

In this paper, we propose a *priority-based scheduling algorithm* to manage downlink traffics in IEEE 802.16 broadband wireless networks. Our objectives are to improve the network throughput, to satisfy the delay constraints of real-time traffics, and to achieve fair resource distribution among MSSs. The basic idea is to assign priorities to MSSs according to their channel conditions and buffered traffics. In particular, the MSSs with good channel conditions will have a higher priority compared with those with bad channel conditions, so the network throughput can be increased since MSSs can use a higher rate to transmit their data. In addition, the MSSs with urgent real-time traffics will be assigned with a high priority to alleviate their traffic delays. On the other hand, the priorities of those MSSs that have queued a large amount of non-real-time traffics will be raised to prevent them from starving. In this way, both the delays of real-time traffics can be alleviated while the long-term fairness can be maintained.

The rest of this paper is organized as follows. Section II formally defines our resource allocation problem. Section III presents our scheduling algorithm. Simulation results are given in Section IV. Section V concludes this paper.

TABLE I
SUMMARY OF NOTATIONS.

notation	definition
n	the number of admitted MSSs in the system
M_i	the ith MSS in this system
r_i^R, r_i^N	the request average real-time data rate and the request minimal non-real-time data rate of M_i, respectively
b_i^R, b_i^N	the amounts of real-time and non-real-time buffered data of M_i, respectively
a_i^R, a_i^N	the amounts of resources allocated to M_i for real-time and non-real-time data, respectively
c_i	the current channel rate of M_i
c_i^{avg}	the average channel rate of M_i in the recent f_T frames
s_i^N	the non-real-time rate satisfaction ratio of M_i in the most recent f_T frames
δ	the ratio of MSSs for prior real-time data allocation
$\mathcal{F}$	the number of free slots in the current downlink subframe
f_c	the current frame index
f_T	the window size for fairness measurement
w^R, w^N	the weights of real-time and non-real-time data, repsectively

II. Problem Statement

We consider the downlink communication in an IEEE 802.16 OFDMA system with one BS supporting multiple MSSs under the PMP mode. When an MSS needs to initiate a traffic flow, it has to ask for the BS's permission. The BS can admit the connection if it has enough resource to support the QoS requirement of that traffic flow; otherwise, the traffic flow will be dropped.

We are given n MSSs, where each of them requests an average real-time data rate of r_i^R (in bits/frame) and a minimum non-real-time data rate of r_i^N (in bits/frame), and uses a channel rate of $c_{i,k}$ (in bits/slot) at frame k. The *scheduling problem* asks how to determine the resource $a_{i,k}$ (in bits) allocated to each MSS M_i, $i = 1..n$, in every frame k, such that the network throughput is maximized, the long-term fairness among MSSs is satisfied, and the delays of real-time traffics are guaranteed. Here, referring to [10], [11], we define a *fairness index (FI)* to evaluate the long-term fairness of a scheduling algorithm as:

$$FI = \frac{(\sum_{i=1}^{n} SD_i)^2}{n\sum_{i=1}^{n}(SD_i)^2}, \tag{1}$$

$$SD_i = \frac{w^R \sum_{j=0}^{f_T-1} a_{i,f_c-j}^R}{f_T \times r_i^R} + \frac{w^N \sum_{j=0}^{f_T-1} a_{i,f_c-j}^N}{f_T \times r_i^N},$$

where f_c is the current frame index, f_T is the window size (in frames) that we measure fairness, a_{i,f_c-j}^R and a_{i,f_c-j}^N (both in bits) are the resources allocated to M_i for real-time and non-real-time traffics at frame $f_c - j$, respectively, and w^R and w^N are weights that we put on real-time and non-real-time traffics, respectively, such that $w^R + w^N = 1$. In particular, $0 < FI \leq 1$ and a scheduling algorithm is considered to be more fair if its FI is larger.

Table I summarizes the notations used in this paper.

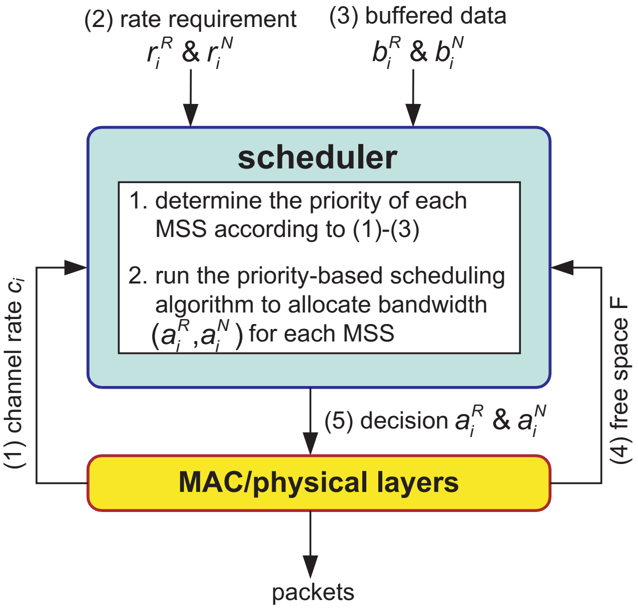
(2) rate requirement
r_i^R & r_i^N
(3) buffered data
b_i^R & b_i^N
scheduler
1. determine the priority of each MSS according to (1)-(3)
2. run the priority-based scheduling algorithm to allocate bandwidth (a_i^R,a_i^N) for each MSS
(1) channel rate c_i
(4) free space F
(5) decision a_i^R & a_i^N
MAC/physical layers
packets

Fig. 1. The system architecture of our proposed scheduler.

III. The Priority-Based Scheduling Algorithm

Fig. 1 illustrates the system architecture of our proposed scheduler. Since the scheduler will handle each downlink subframe, we omit the frame index k in the following. When scheduling each frame, the scheduler will first query the MAC/physcial layers for the current channel rate c_i of each MSS and the total free space $\mathcal{F}$ (in slots) of the current downlink subframe. Then, given the rate requirement (r_i^R, r_i^N) and the buffered data (b_i^R, b_i^N) of each MSS, the scheduler will calculate its priority. Based on these priorities, the scheduler can execute our proposed priority-based scheduling algorithm to determine the bandwidth (a_i^R, a_i^N) allocated to each MSS. Such a decision will be sent to the MAC/physical layers to arrange free slots for transmission.

Given the current channel rate c_i, buffered real-time data b_i^R (in bits), real-time data rate r_i^R, and non-real-time traffic satisfaction ratio s_i^N of an MSS M_i, we can calculate its priority p_i as follows:

$$p_i = c_i \times \frac{c_i}{c_i^{\text{avg}}} \times \frac{b_i^R}{r_i^R} \times \frac{1}{s_i^N}, \tag{2}$$

where c_i^{avg} is the average channel rate of M_i and s_i^N is defined as

$$s_i^N = \min\left\{1, \frac{\sum_{j=0}^{f_T-1} a_{i,f_c-j}^N}{f_T \times r_i^N}\right\}. \quad (3)$$

Note that M_i has a higher priority if its p_i is larger.

In the above Eq. (2), the first term means that we will assign a higher priority to those MSSs that can use higher channel rates. The second term means that we will assign a higher priority to those MSSs that have a better channel condition (as compared with their historical channel conditions). These two terms benefit the MSSs with good channel qualities to improve the network throughput. The third term means that we will assign a higher priority to those MSSs that require more time to transmit their buffered real-time data. This term is to alleviate the delays of real-time traffics. The last term means that we will give a higher priority to those MSSs that have queued a large amount of non-real-time data. This term is to prevent non-real-time traffics from starvation.

The scheduler then allocates slot resources to MSSs according to their priorities. However, to alleviate the delays of real-time traffics, we should first allocate resources to those MSSs that have *urgent data*, which are real-time data that will be dropped if they are not sent in the current frame. Then, we should select a δ ratio of high-priority MSSs to serve their real-time data, where $0 \leq \delta < 1$. If we still have free resource, we can distribute it to MSSs according to their priorities. In particular, our proposed priority-based scheduling algorithm involves in the following steps:

1. We sort n MSSs by their priorities in a descending order. Below, we examine each MSS using this order.
2. Let d_i be the size of M_i's urgent data, $i = 1..n$. For each M_i with $d_i > 0$, we allocate it with a resource of

$$a_i = \min\{c_i \times (\mathcal{F} - \sigma_i), d_i\}, \quad (4)$$

   where $\mathcal{F}$ is the number of free slots in the current downlink subframe and σ_i is the summation of allocated slots for all MSSs except M_i, that is,

$$\sigma_i = \sum_{\forall j, j \neq i} \left\lceil \frac{a_j}{c_j} \right\rceil.$$

   Here, $c_i \times (\mathcal{F} - \sigma_i)$ means the total remaining bits that the BS can give M_i in the current downlink subframe, using M_i's channel rate c_i.
3. We then select the first $\lceil \delta n \rceil$ MSSs to serve their real-time data. For each such MSS M_i, we allocate it with a resource of

$$a_i = \min\left\{c_i \times (\mathcal{F} - \sigma_i), b_i^R\right\}. \quad (5)$$

4. For each M_i, $i = 1..n$, we allocate it with a resource of

$$a_i = \min\left\{c_i \times (\mathcal{F} - \sigma_i), b_i^R + b_i^N\right\}. \quad (6)$$

5. For each M_i, $i = 1..n$, we set $a_i^R = \min\{b_i^R, a_i\}$ and $a_i^N = a_i - a_i^R$.

Note that in step 3, we serve the real-time traffics of the first $\lceil \delta n \rceil$ MSSs if there still remains free space. This is to avoid cumulating too much urgent real-time data in the following frames. If there are free slots after step 3, we can distribute them among MSSs, as shown in step 4. Finally, in step 5, we transform the result of assignment a_i into (a_i^R, a_i^N) to serve the real-time and non-real-time traffics of each MSS M_i.

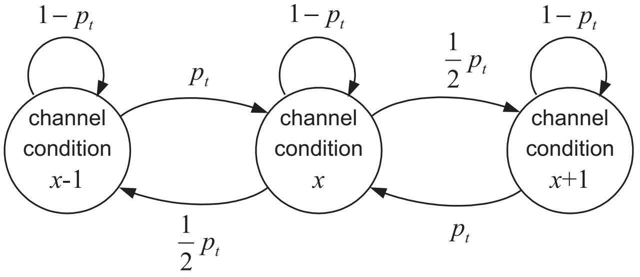

Fig. 2. The three-state Markov chain to model the change of channel conditions of MSSs.

Since $d_i \leq b_i^R \leq b_i^R + b_i^N$, we have Eq. (4) $\leq$ Eq. (5) $\leq$ Eq. (6). So, M_i can be allocated with enough slots by Eq. (5) to support its urgent data. Similarly, M_i can be allocated with enough slots by Eq. (6) to support all its real-time data.

IV. Simulation Results

In this section, we present some simulation results to verify the effectiveness of our algorithm. We develop a simulator by C++ language. Table II lists the system parameters used in our simulator, which follows those defined in the IEEE 802.16 standard.

For each MSS, its admitted real-time data rate r_i^R and non-real-time data rate r_i^N are randomly selected from $[0, 400]$ bits per frame. The channel condition of an MSS will change during the simulations. We use a three-state Markove chain [12] to model the change of channel conditions, as shown in Fig. 2. In particular, let MCS = {QPSK1/2, QPSK3/4, 16QAM1/2, 16QAM3/4, 64QAM1/2, 64QAM3/4} be the list of modulation and coding schemes and $MCS[x]$ denote the scheme with index x. Suppose that an MSS uses the scheme of $MCS[x]$ under its channel condition at the current frame. There is a probability of $\frac{1}{2}p_t$ that it has to use the scheme of $MCS[x-1]$ when the channel condition becomes worse at the next frame. Also, there is a probability of $\frac{1}{2}p_t$ that it can use the scheme of $MCS[x+1]$ when the channel condition becomes better at the next frame. In addition, there is a probability of $1 - p_t$ that the channel condition of the MSS remains the same at the next frame. We set the transition probability $p_t = 0.5$ and the x value of each MSS is randomly selected from 2 to 5.

The number of MSSs (i.e., n) is ranged from 20 to 70. Due to the limited resource, the system can totally serve at most 70 MSSs. We compare our proposed algorithm against the Max-Throughput (MT) scheme and the MPF scheme [5]. The MT scheme always selects the MSS with the best channel condition c_i to serve. The MPF scheme assigns priorities to MSSs according to their c_i values and data rates. In our simulations, the values of weights w_R and w_N are set to 0.9 and 0.1, respectively.

TABLE II
SYSTEM PARAMETERS USED IN OUR SIMULATOR.

parameter	value
FFT (fast Fourier transform) size	1024
zone category	PUSC (partial usage of subchannel) with reuse 1
modulation and coding scheme	QPSK1/2, QPSK3/4, 16QAM1/2, 16QAM3/4, 64QAM1/2, and 64QAM3/4
frame duration	5 ms
types of real-time traffics	UGS and rtPS
types of non-real-time traffics	nrtPS and BE

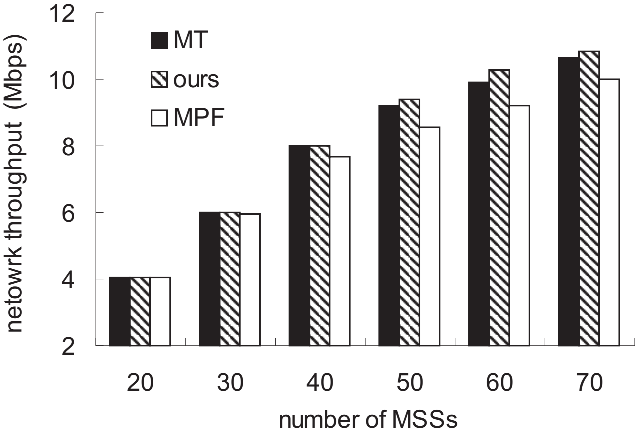

Fig. 3. Comparison on network throughput of different scheduling schemes.

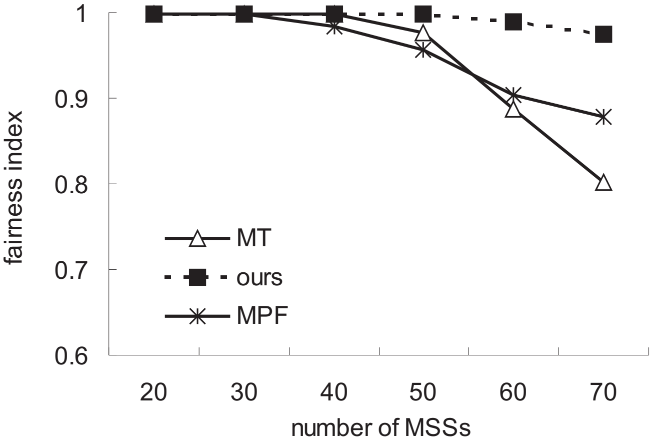

Fig. 4. Comparison on fairness indices of different scheduling schemes.

A. Network Throughput

We first compare the network throughput of these three scheduling schemes. Fig. 3 illustrates the effect of different numbers of MSSs on the network throughput. We can observe that when the number of MSSs is smaller than 40, the network throughput of all scheduling schemes are similar. This is because quite few MSSs ask for transmission, so the effect of different scheduling schemes is not significant. However, when the number of MSSs grows, our scheduling algorithm can improve the network throughput as compared with the MT and MPF schemes. This is because our scheduler will assign a higher priority to two kinds of MSSs: 1) the MSSs that have a better channel condition (i.e., a larger c_i) and 2) the MSSs whose current channel condition is better than their historical channel conditions (i.e., $c_i > c_i^{\mathrm{avg}}$). In this way, the MSSs can use a higher data rate to transmit their packets, so the network throughput can be increased.

B. Fairness Index

Although our scheduling algorithm benefits those MSSs that can transmit data using high data rates, it still can maintain long-term fairness among all MSSs. This is verified in Fig. 4, which demonstrates the effect of different numbers of MSSs on the fairness index (i.e., Eq. (1)). We can observe that the fairness index of our scheduling algorithm is still close to 1, even though there are 70 MSSs in the system. This is because our scheduler will assign a higher priority to those MSSs that have queued a large amount of data. On the other hand, the fairness indices of both the MT and MPF schemes drop significantly when the number of MSSs grows, because they only allow the MSSs with better channel conditions to transmit their data first.

C. Packet Dropping Ratios of Real-Time Traffics

We then compare the packet dropping ratios of real-time traffics under different scheduling schemes. Fig. 5 illustrates the effect of different numbers of MSSs on the packet dropping ratios of real-time traffics. When the number of MSSs is more than or equal to 50, the network starts saturated. We can observe that the packet dropping ratios of both the MT and MPF schemes increase when the number of MSSs increases. This is because they do not differentiate real-time traffics from non-real-time ones, causing a large amount of non-real-time traffics to contend with urgent real-time traffics. On the other hand, by making MSSs transmit their urgent real-time traffics first, our scheduling algorithm can result in zero packet dropping ratio, even though there are 70 MSSs in the system. This verifies the effectiveness of our algorithm.

D. Satisfaction Ratios of Non-Real-Time Traffics

Although our scheduling algorithm benefits real-time traffics, it does not starve non-real-time traffics. This is verified in Fig. 6, which demonstrates the effect of different numbers of MSSs on the satisfaction ratio of non-real-time traffics (i.e., Eq. (3)). We can observe that the satisfaction ratio of our scheduling algorithm is still close to 1, even though there are 70 MSSs in the system. This is because our scheduler will assign a higher priority to those MSSs that have queued a large amount of non-real-time data. On the other hand, the

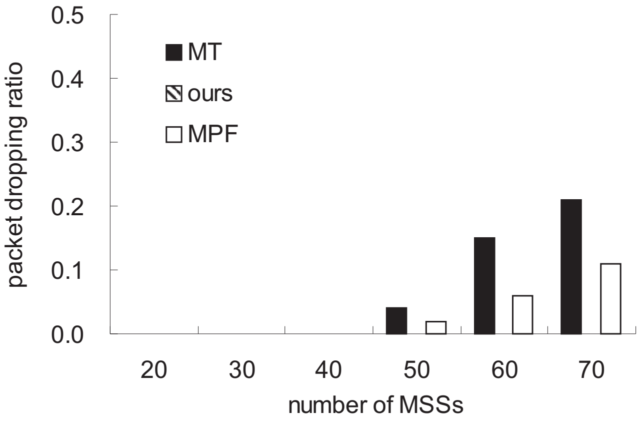

Fig. 5. Comparison on packet dropping ratios of real-time traffics under different scheduling schemes.

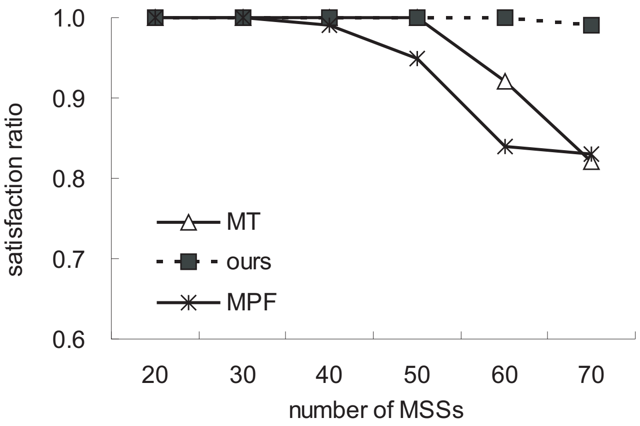

Fig. 6. Comparison on satisfaction ratio of non-real-time traffics under different scheduling schemes.

satisfaction ratio of both the MT and MPF schemes drop significantly when the number of MSSs grows, especially when the number of MSSs is more than 50. This is because they do not consider the queue lengths of MSSs, making non-real-time traffics starvation.

V. Conclusions

In this paper, we have proposed a priority-based scheduling algorithm for the downlink communication in an IEEE 802.16 broadband wireless network. Our scheduling algorithm can allocate resources to MSSs based on their channel conditions and buffered data. The MSSs with good channel conditions and urgent real-time data will be served first. In this way, we can not only increase the network throughput but also alleviate the delays of real-time traffics. Our scheduling algorithm also addresses the fairness issue, so non-real-time traffics are not starved. Simulation results have shown that our scheduling algorithm can improve the network throughput, maintain the long-term fairness, alleviate the packet dropping ratios of real-time traffics, and increase the satisfaction ratios of non-real-time traffics.

Acknowledgement

Y.-C. Tseng's research is co-sponsored by MoE ATU Plan, by NSC grants 96-2218-E-009-004, 97-3114-E-009-001, 97-2221-E-009-142-MY3, and 98-2219-E-009-005, by MOEA under grant 98-EC-17-A-02-S2-0048, and by ITRI, Taiwan.

References

[1] IEEE Std 802.16-2004, "IEEE Standard for Local and Metropolitan Area Networks—Part 16: Air Interface for Fixed Broadband Wireless Access Systems," 2004.

[2] IEEE Std 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 1," 2006.

[3] A. Ghosh, D. Wolter, J. G. Andrews, and R. Chen, "Broadband wireless access with WiMax/802.16: current performance benchmarks and future potential," *IEEE Communications Magazine*, vol. 43, no. 2, pp. 129–136, 2005.

[4] C. Eklund, R. B. Marks, K. L. Stanwood, and S. Wang, "IEEE Standard 802.16: a technical overview of the WirelessMAN$^{\rm TM}$ air interface for broadband wireless access," *IEEE Communications Magazine*, vol. 40, no. 6, pp. 98–107, 2002.

[5] J. Kim, E. Kim, and K. S. Kim, "A new efficient BS scheduler and scheduling algorithm in WiBro systems," in *IEEE International Conference on Advanced Communication Technology*, vol. 3, 2006, pp. 1467–1470.

[6] J. Shi and A. Hu, "Maximum utility-based resource allocation algorithm in the IEEE 802.16 OFDMA System," in *IEEE International Conference on Communications*, 2008, pp. 311–316.

[7] Y. Ma, "Rate-maximization scheduling for downlink OFDMA with long term rate proportional fairness," in *IEEE International Conference on Communications*, 2008, pp. 3480–3484.

[8] X. Zhu, J. Huo, S. Zhao, Z. Zeng, and W. Ding, "An adaptive resource allocation scheme in OFDMA based multiservice WiMAX systems," in *IEEE International Conference on Advanced Communication Technology*, 2008, pp. 593–597.

[9] N. A. Ali, M. Hayajneh, and H. Hassanein, "Cross layer scheduling algorithm for IEEE 802.16 broadband wireless networks," in *IEEE International Conference on Communications*, 2008, pp. 3858–3862.

[10] D. M. Chiu and R. Jain, "Analysis of the increase and decrease algorithms for congestion avoidance in computer networks," *Journal of Computer Networks and ISDN*, vol. 17, no. 1, pp. 1–14, 1989.

[11] Y. C. Wang, Y. C. Tseng, and W. Chen, "MR-FQ: a fair scheduling algorithm for wireless networks with variable transmission rates," *SCS Simulation*, vol. 81, no. 8, pp. 587–608, 2005.

[12] P. Y. Wu, J. J. Chen, Y. C. Tseng, and H. W. Lee, "Design of QoS and admission control for VoIP Services over IEEE 802.11e WLANs," *Journal of Information Science and Engineering*, vol. 24, no. 4, pp. 1003–1022, 2008.