An Efficient Scheduling Algorithm for Multiple MSSs in IEEE 802.16e Network

Wen-Hwa Liao and Chen Liu
Department of Information Management
Tatung University
Taipei, Taiwan
whliao@ttu.edu.tw

Sital Prasad Kedia
Samsung India Software Operations Pvt. Ltd.
Bangalore, India
sital.kedia@samsung.com

***Abstract*—The IEEE 802.16e standard introduced the concept of mobile subscriber stations (MSS) to provide mobility support. Five quality of service (QoS) classes have been defined to support QoS requirement different connections between the base station and the subscriber station. Out of these QoS classes, UGS has been designed to support real-time service flows that periodically generate fixed-size data packets. Most of the existing scheduling schemes for UGS class consider scheduling of a single MSS or even if they consider scheduling of multiple MSSs, the QoS requirement of the MSSs is not satisfied properly after scheduling. In this paper we have proposed a scheduling scheme, which schedules multiple MSSs with UGS connections so that QoS requirement of each MSS can be satisfied after scheduling.**

Keywords- IEEE 802.16e; scheduling; WiMAX; wireless.

I. INTRODUCTION

Wireless network is often considered as a cheaper and time saving alternative to its wired counterpart. In addition, some developing countries and uncivilized regions lack in infrastructures for deployment of wired network. To overcome the above-mentioned limitations of the wired network and to satisfy the huge demand for wireless services, worldwide interoperability for microwave access (WiMAX) was advocated as IEEE 802.16 wireless technology with high throughput over long distance (up to 30 miles). Later, the IEEE 802.16e standard, also known as Mobile WiMAX was proposed, which introduced the concpet of Mobile Subscriber Station (MSS) [1]. Basic WiMAX network includes a Base Station (BS) and several Subscriber Stations (SS) that are served by the BS. There are five QoS classes defined in 802.16e standard: UGS (Unsolicited Grant Service), rtPS (real-time Polling Service), ertPS (Extended Real-time Polling Service), nrtPS (non-real-time Polling Service), and BE (Best Effort). UGS is designed for services that periodically generate fixed-size data, such as T1/E1 and Voice over IP (VoIP). The BS assigns fixed grant to UGS connections. Hence, the MSSs need not send bandwidth request to BS every time they need to transmit data and thus saving the bandwidth used to send the bandwidth request to the BS.

There are several research works which have focused on optimizing the power consumption in IEEE 802.16e networks. The works in [3][4] apply the Chinese remainder theorem to decide the start time of each connections of an MSS. Due to different start time combination, the wake up time of the MSS is reduced. However, they didn’t take into account the bandwidth used by each connection. So an MSS may need to handle more number of connections than it can in a certain time, which debase the feasibility of their approach. The goal of works in [5][9] is to minimize the wake up time of an MSS having multiple connections of different service classes. They gather the bursts of all the connections of different service classes and transmit these bursts together so that the wakeup time of the MSS can be reduced and thus saving significant amount of energy. Their approach is efficient but they didn’t consider the multiple MSSs environment. Three energy efficiency scheduling algorithms for multiple MSSs were proposed in [6][7][8]. The work in [6] classifies MSSs into two catagories i.e., primary and secondary, based on their QoS requirement. A primary MSS is allowed to use the bandwidth in burst mode, whereas a secondary MSS is given the necessary bandwidth only to meet the requirement of its delay constraint. This approach can save significant amount of energy and also can avoid interference between the MSSs. However, in the real world environment, when the traffic load of all the MSSs is high, it is hard to classify the MSSs as primary and secondary. The work in [7] proposes a scheduling algorithm for mulitple MSSs environment. The algorithm gathers the bursts of all the connections in an MSS and transmit them together in order to minimize the wake up time of the MSSs. Minimum wake up time and multiple MSSs environment were considered for the first time in their work. In [8], authors have applied the Ford-Fulkerson algorithm to decide the time slots used by the MSSs. However, after applying the algorithm, the QoS requirement of the MSSs is not guranteed to be satisfied.

Most of the related works consider single MSS for scheduling, or even if they consider multiple MSSs, the QoS requirement of the MSSs is not satisfied after scheduling. Therefore, we have proposed a scheduling scheme to assign time slots used by multiple MSSs with UGS connection so that QoS requirement of each connection is satisfied after scheduling.

This paper is organized as follows. Section II discusses the scheduling scheme for multiple MSSs. Simulation results are presented in section III and Section IV concludes this paper.

ISBN: 978-1-61208-133-5

II. SCHEDULING ALGORITHM FOR MULTIPLE MSSS

In the IEEE 802.16e networks, one BS may serve several MSSs simultaneously. However, at a particular timeslot, the BS can serve only one MSS [2]. If two or more MSSs are scheduled for data transmission in the same time slot, it will cause interference at the BS side. Therefore, an efficient scheduling algorithm is required, which can avoid the potential interference between the MSSs and also can satisfy the QoS requirements of all the MSSs. Next, we list some notations used in the rest of the paper in Table I and then we present our scheduling scheme.

TABLE I. NOTATIONS

U	Set of connections waiting for transmission.
T	The repeat cycle length.
G_i	Grant transmit interval of connection *i*.
I_i	Idle interval of connection *i*.
ST_i	Start time of connection *i*.
C_i	Cycle of connection *i*, equals to G_i+I_i .
W_i	Weight of connection *i*.
ω_i	Waiting time of connection *i*.
ω_{max}	Maximum waiting time of all the connections.
d_i	Delay constraint of connection *i*.
t	Target connection.

The scheduling algorithm for multiple MSSs has been given in Fig. 1. Initially, set U contains all the connections that are waiting to be scheduled. The first step of our scheduling algorithm is to select a connection from the set U having maximum weight as the target connection (t) for scheduling. Then we will check if there are sufficient empty slots to satisfy the bandwidth requirement of the target connection. If not, we will remove the target connection from U because its bandwidth requirement cannot be fulfilled in this round of scheduling, but it will be given higher priority for selection in the next round. After that, we will check if the target connection needs to be modified. If the target connection is modified then we need to verify whether the requirement for delay constraint is satisfied after modification. If not, then the target connection cannot be scheduled in this round. Connection modification may produce some surplus connections, which are scheduled only after all the connections in set U have been scheduled. However, we need to reserve the timeslots used to schedule these additional connections in advance after connection modification. We then determine the start time of the target connection and perform the connection separation if required. At last, we will schedule the target connection based on its start time, grant transmit interval, and the target connection is removed from the set U.

Algorithm: Scheduling algorithm for multiple MSSs

```
1:  for all the connections in set U
2:        Select a connection with maximum weight
          from U as target connection t.
3:        if the remaining slots are not enough
4:            goto step 16.
5:        endif
6:        if  the target connection needs to be modified
7:             Perform connection modification.
8:              if the delay constraint is not satisfied
9:                  goto step 16.
10:             endif
11:             Reserve timeslots for scheduling
                additional connections.
12:       endif
13:       Determine the start time of the target connection.
14:       Perform connection separation if needed.
15:       Schedule the target connection.
16:       Remove target connection from set U.
17: endfor
18: Schedule the additional connections resulted from
    connection modification.
```

Figure 1. Scheduling algorithm for multiple MSSs.

A. Weight of a Connection

$$W_i = \frac{\frac{\omega_i}{\omega_{max}} + \frac{I_i}{d_i}}{C_i} \tag{1}$$

Weight formula is given in (1). The weight of connection i (W_i) is associated with waiting ratio (ω_i/ω_{max}), delay constraint ratio (I_i/d_i), and length of the cycle (C_i). The weight of the connection increases with increase in the waiting ratio because a connection with longer waiting time should be served earlier. Similarly, when the delay constraint ratio is large, it means that the connection has stringent delay constraint, hence it should be given higher priority for scheduling. The length of cycle (C_i) is inversely proportional to the weight because we found out that smaller is the length of the cycle, more is the chance that the selected connection can be scheduled without any modification. The connection modification will be described later in this section.

B. Connection Modification

In this step of scheduling, we will check whether the target connection is having interference with any of the already scheduled connection. If it is, then the target connection needs to be modified. We transform the target connection into another UGS connection having different grant transmit interval and idle interval to avoid interference with the already scheduled connections.

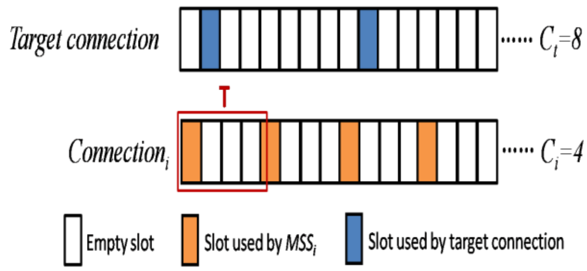

Figure 2. Example of the relationship between connections.

In Fig. 2, we assume that a connection i has already been scheduled, so the repeat cycle length T will be equal to the cycle (C_i) of connection i which is four in this case. Now we want to schedule the target connection t. We observe that if the cycle of target connection (C_t) is a multiple of T then t will have no interference with the already scheduled connections and hence it can be scheduled without any modification. Moreover, the new repeated cycle will be equal to C_t, and the grant transmit interval and idle interval of target connection will remain unchanged.

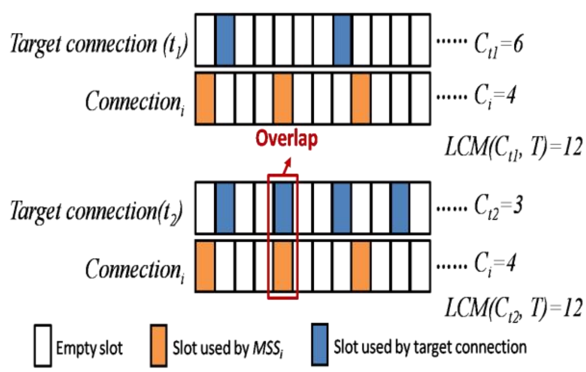

Figure 3. Interference between multiple connections.

However, if C_t is not a multiple of C_i, then the target connection may or may not overlap with the already scheduled connections. As shown in Fig. 3, C_t is not a multiple of C_i. In the first case, t_1 has no overlap with the already scheduled connection i, which means that t_1 can be scheduled without modification and the new repeated cycle T will be equal $LCM(T, C_{t1})$. However, in the second case, t_2 has an overlap with the already scheduled connection, so we need to modify the connection t_2. In general, let ST_t be the start time of the target connection t. If the $(ST_t + n*C_t)$th slot is not free, then the connection t has overlap with the already scheduled connections and it needs to be modified. Here n varies from 0 to $(LCM(T, C_{t1})/C_t) - 1$.

When a target connection needs to be modified, it is converted into another UGS connection having cycle (C_m) of length T. We use (2) to decide the new grant transmit interval (G_m) and idle interval (I_m) of the target connection.

$$G_m = G_t \times \left\lfloor \frac{T}{C_t} \right\rfloor \qquad (2)$$
$$I_m = T - G_t$$

Fig. 4 shows the modified connection for the target connection t_2 of Fig. 3. From (2), we can get G_m=1 and I_m=3 for the modified connection. We can see that t_2 has been converted to a new UGS connection having cycle of length 4 and it can be scheduled without any overlap. However, a surplus slot is left as shown in Fig. 4, which has to be put back as an additional connection to ensure the QoS requirement of the target connection. We see that the surplus slots are needed in each $LCM(T, C_t)$ slots. Thus we can calculate the length of the cycle(C_a), grant transmit interval (G_a) and idle interval (I_a) of the additional connection as given in (3). When the grant transmit interval of the additional connection is more than 1, we will further split it into G_a connections with grant transmit interval 1.

Figure 4. Modified connection for target connection t_2.

$$C_a = LCM(C_t, T)$$
$$G_a = G_t \times (C_a/C_t) - G_t \times \left\lfloor \frac{T}{C_t} \right\rfloor \times C_a/T \qquad (3)$$
$$I_a = C_a - G_a$$

C. Scheduling Additional Connections

Additional connections, which resulted from connection modification, are scheduled only after all the connections in set U are scheduled. However, we need to reserve the timeslots for these additional connections in advance. Let's consider the example shown in Fig. 5. The scheduling pattern has a cycle (T) of length 8 and the additional connection has a cycle (C_a) of length 12. Now suppose we select the third time slot as the start time of the additional connection, then the next timeslot occupied by the additional connection will be the 15th one. Here we can observe that the timeslots used by the additional connection is the third slot of each four timeslots which is equal to $GCD(8, 12)$. For example, the first timeslot used by the additional connection is the third slot of the first four slots and the second timeslot used is the third timeslot of fourth four slots. Thus we need to reserve one timeslot in each $GCD(T, C_a)$ cycle for scheduling of additional connections and the total number of timeslots that are reserved for scheduling of additional connection will be equal to $T/GCD(T, C_a)$.

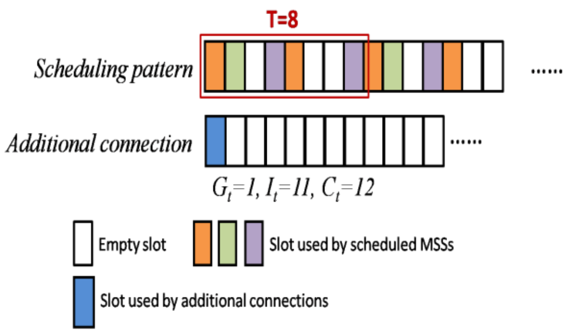

Figure 5. Scheduling additional connections.

Moreover, when more than one additional connections have the cycle of same length, they can use the same $GCD(T, C_a)$ cycle. For example in the previous case, the additional connection used the first and fourth $GCD(T, C_a)$ cycle, which means the second and third $GCD(T, C_a)$ cycle is empty and can be used to schedule another additional connection having cycle of length C_a. The number of connections that can be scheduled together is given by $C_a/GCD(T, C_a)$. For this reason, we split an additional connections into G_a connections with grant transmit interval 1 so that they can use the same $GCD(T, C_a)$ cycle. It should be noted that in case the number of additional connection having cycle of length C_a is less then $C_a/GCD(T, C_a)$, then some of the timeslots reserved for additional connections will remain empty, which results in wastage of bandwidth. We have considered this bandwidth waste as a parameter for performance evaluation in our simulation results.

D. Delay Constraint

For UGS connections, the idle interval between packet generation and packet transmission must be less than the predefined delay constraint. In our work, this idle period is the idle interval between two grant transmit intervals. Due to connection modification, the idle interval of the target connection may change. Hence, before scheduling the target connection, we have to check that the idle interval of the target connection is less than its predefined delay constraint. If not, it is not possible to satisfy the QoS requirements of the target connection, so it will not be scheduled in this round.

E. Assigning Start Time for Connections

For connections that are not modified, we will select the first empty slot of the cycle as the start time (ST_i) of those connections. The first empty slot should be larger than the summation of the grant transmit interval of all the connections scheduled so far. On the other hand, when assigning start time for the modified connections, we will find the last empty slot in the repeat cycle to meet requirements of the delay constraint. For example in Fig. 4, the start time of target connection will be the fourth timeslot. When we assign the last empty slot of repeat cycle as the start time, we can simply use the idle interval of target connection to check that the delay constraint is satisfied and ensure that the data needed to be sent is generated before the slot assigned to transmit it.

F. Connection Separation

In case the grant transmit interval of the target connection is more than the number of consecutive empty slots starting from ST_t, then timeslots assigned to the target connection will overlap with the already scheduled connections. For example in Fig. 6, G_t of the target connection is 4, while there are only 2 consecutive empty slots after timeslot 2 in the scheduling pattern. To solve this problem, we will split the target connection into two separate connections, connection 1 and connection 2, as shown in Fig. 6. The grant transmit interval (G_1) of connection 1 is the number of consecutive empty slots starting from ST_t and the idle interval (I_1) will be $C_t – G_1$. Connection 1 will be immediately scheduled and its start time (ST_1) will be the original start time (ST_t) of the target connection. The connection 2 will be treated as the new target connection whose grant transmit interval (G_2) will be equal to $G_t – G_1$ and idle interval(I_2) will be $C_t – G_2$.

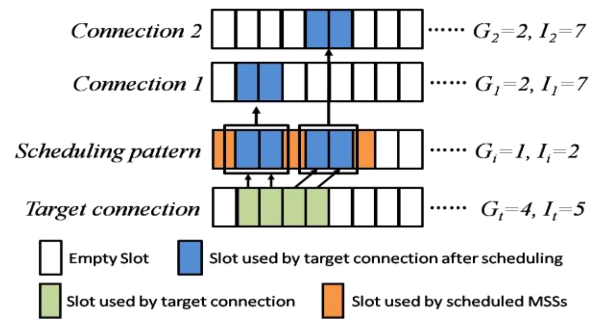

Figure 6. Connection separation.

III. SIMULATION RESULTS

In this section, we have presented the simulation results to evaluate the performance of our proposed scheduling algorithm. As already discussed in section I, most of the related works consider single MSS for scheduling. Hence, it is not possible to compare the performance of our proposed algorithm with any of the related works. We have used bandwidth utilization, connections selection rate, and bandwidth waste for arranging additional connections as the parameters for performance evaluation. Bandwidth utilization is the percentage of used slots to the total number of slots. Connection selection rate is the rate at which connections are being selected for scheduling. Bandwidth waste, as discussed previously, is the wastage of the bandwidth incurred by reserving time slots for scheduling of additional connections. The bandwidth waste is given by reserved slots*number of additional connections that can be scheduled/number of additional connections actually scheduled.

A C-coded custom simulator is used to evaluate the performance of our scheduling algorithm. All the simulation results were obtained by running the scheduling algorithm for 50 times and then taking the average. The simulation parameters are listed in Table II.

TABLE II. SIMULATION PARAMETERS

Parameters	Value
Grant transmit interval	1-5
Idle interval	1-25
Number of connections	1-15
Ratio of grant transmit	1:1 – 1:10
Delay constraint	2 * Idle interval

Fig. 7 and Fig. 8 present the bandwidth utilization and connection selection rate for varying number of connections and varying ratio of grant transmit interval to idle interval. The number of connection varies from 1 to 10 and the ratio of grant transmit interval to idle interval varies from 1:1 and 1:10.

In Fig. 7, the bandwidth utilization increases with increase in the number of connections and it reaches up to 90% when the number of connection is 10. This is because of the fact that, with increase in number of connections, more data is available for transmission and the bandwidth utilization increases. Similarly, when ratio of grant transmit interval to idle interval increases, the bandwidth utilization is increased due to the same reason.

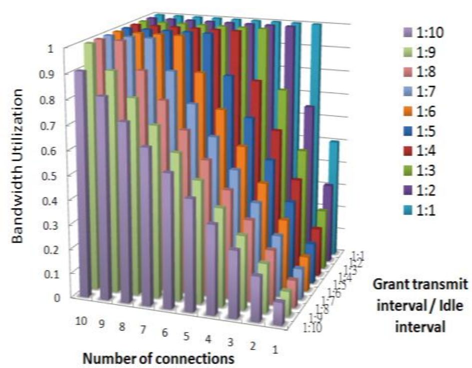

Figure 7. Bandwidth utilization.

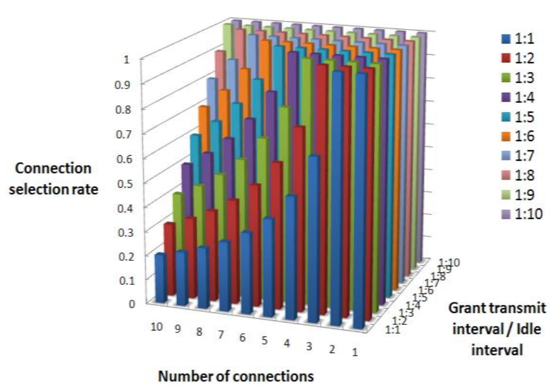

Figure 8. Connection selection rate.

In Fig. 8, the connection selection rate decreases as the number of connections increases because more is the number of connections, lesser is the chance for a connection to be selected. The connection selection rate also decreases with increase in the ratio of grant transmit interval to idle interval due to increase in amount of data for transmission.

We have also considered the parameters of UGS connection for simulation. The UGS parameters are listed in Table III. We classified UGS connections into three classes i.e., class A, class B, and class C having packet size 32 bytes, 64 bytes, and 128 bytes respectively.

TABLE III. UGS SIMULATION PARAMETERS

Parameters	Value
Packet Size	32, 64, 128 bytes
Frame duration	5 ms
Slot duration	0.1 ms
Data rate	30 kbps
Idle interval	10 – 50 ms
Delay constraint	150 ms – 400 ms
Number of connections	1-15

The simulation results of bandwidth utilization and connection selection rate for varying UGS parameters have been presented in Fig. 9 and Fig. 10 respectively. In Fig. 9, the bandwidth utilization for class A and class B is less than that for others because of smaller packet size of class A and class B. On the other hand, the bandwidth utilization for class B and class C is highest because of their large packet size. As shown in Fig. 10, the connection selection rate is highest for class A and class B. We can observe that the connection selection rate decreases with increase in the packet size and is lowest for class B and class C.

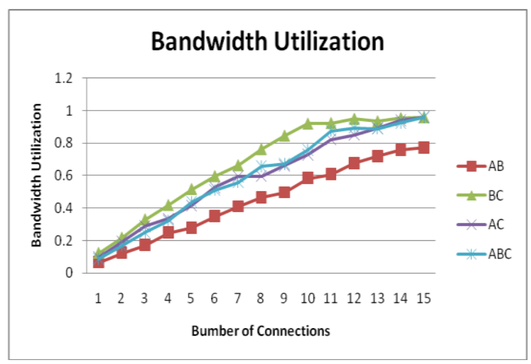

Figure 9. Bandwidth utilization with UGS parameters.

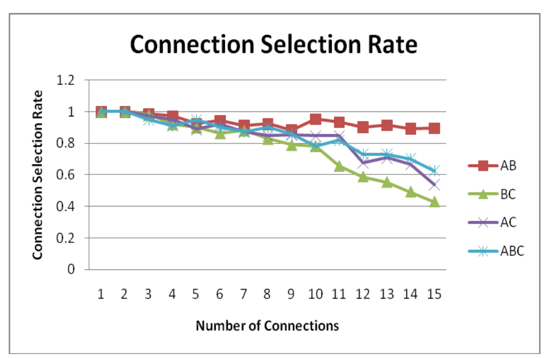

Figure 10. Connection selection rate with UGS parameters.

 ISBN: 978-1-61208-133-5

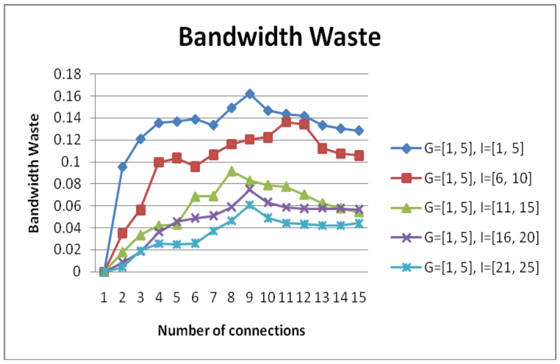

Figure 11. Bandwidth waste.

Fig. 11 shows the bandwidth waste for different values of ideal interval and number of connections. The grant transmit interval value is set to a random number between 1 to 5. Initially, the bandwidth waste increases with increase in the number of connections but it tends to decrease as the number of connections increases further. This is because of the fact that initially with increase in the number of connections, more number of additional connections are produced and we have to reserve more timeslots to schedule these additional connections. Hence, the bandwidth waste increases. However, when the number of connections increases further, the chances that additional connections can be scheduled together increases, which result in decrease in the bandwidth waste. Thus, our approach will not have large bandwidth waste when number of connections is large.

IV. Conclusion

In this paper, we proposed a scheduling algorithm for multiple MSSs with UGS connection in IEEE 802.16e networks. Our approach can avoid the potential interference between the MSSs and satisfy the QoS requirement of the MSSs after scheduling. The simulation results show that our approach can achieve more than 90% bandwidth utilization and it will not have large bandwidth waste when number of connections is large. Thus, our approach can achieve good performance and scalability in the real world environment.

Acknowledgments

This work was supported in part by the National Science Council, Taiwan, under grant NSC 99-2221-E-036-036-MY2, and Tatung University, under grant B99-N03-065.

References

[1] IEEE Standard 802.16e-2005, “IEEE standard for local and metropolitan area networks part 16: air interface for fixed and mobile broadband wireless access system,” Proceedings of IEEE Std., Feb. 2006.

[2] S.-I. Chakchai, R. Jain, and A.-K. Tamimi, “Scheduling in IEEE 802.16e mobile WiMAX networks: key issues and a survey,” IEEE Journal on Selected Areas in Communications, Vol. 27, No. 2, pp. 156-171, February 2009.

[3] T.-C. Chen, Y.-Y. Chen, and J.-C. Chen, “An efficient energy saving mechanism for IEEE 802.16e wireless MANs,” IEEE Transactions on Wireless Communications, Vol. 7, No. 10, pp. 3708-3712, October 2008.

[4] T.-C. Chen, J.-C. Chen, and Y.-Y. Chen, “Maximizing unavailability interval for energy saving in IEEE 802.16e wireless MANs,” IEEE Transactions on Mobile Computing, Vol. 8, No. 4, pp. 475-487, April 2009.

[5] C.-Y. Lin, H.-L. Chao, Y.-J. Liao, and T.-J. Tsai, “Least-awake-slot scheduling with delay guarantee for IEEE 802.16e broadband wireless access networks,” IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), 2008.

[6] G. Fang, E. Dutkiewicz, Y. Sun, J. Zhou, J. Shi, and Z. Li, “Improving mobile station energy efficiency in IEEE 802.16e WMAN by burst scheduling,” IEEE GLOBECOM, 2006.

[7] S. C. Huang, R. H. Jan, and C. Chen, “Energy efficient scheduling with QoS guarantee for IEEE 802.16e broadband wireless access networks,” International Wireless Communications and Mobile Computing Conference (IWCMC), 2007.

[8] S.-C. Huang, C. Chen, R.-H. Jan, and C.-C. Hsieh, “An energy-efficient scheduling for multiple MSSs in IEEE 802.16e broadband wireless,” IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), 2008.

[9] C.-Y. Lin and H.-L. Chao, “Energy-saving scheduling in IEEE 802.16e networks,” IEEE Conference on Local Computer Networks (LCN), 2008.