

Efficient Uplink Scheduler Architecture of Subscriber Station in IEEE 802.16 System*

Woo-Jae Kim¹, Joo-Young Baek¹, Sun-Don Lee¹, Young-Joo Suh¹,
Yun-Sung Kim², and Jin-A Kim²

¹ Department of Computer Science and Engineering,
Pohang University of Science and Technology (POSTECH),
San 31, Hyoja-Dong, Nam-Gu, Pohang 790-784, Korea
{hades15, nalsunia, sdonlee, yjsuh}@postech.ac.kr

² WiBro Technology Lab., Telecommunication R&D Center,
Samsung Electronics Co., LTD,
Maetan-Dong, Paldal-Gu, Suwon 442-742, Korea
{tseliot, jin0420}@samsung.com

Abstract. The IEEE 802.16, broadband fixed wireless access standard, defines four service classes, USG, rtPS, nrtPS and BE on high speed wireless networks. To guarantee the QoS requirement of these classes, the subscriber station and base station require scheduling architecture and algorithm. However, the IEEE 802.16 does not define any scheduling architecture or algorithm, and the most existing scheduling mechanisms only focus on working at the BS. In this paper, we propose two types of scheduling architecture working at the SS. In the one-level scheduler, we use a flow queue and class queue by differentiating flows with their class priority. The two-level scheduler can provide more organized QoS service with complementing the one-level scheduler. Adapting these architectures makes scheduler efficiently control all types of traffic defined in the IEEE 802.16. In the proposed architecture, any scheduling algorithms such as SCFQ and EDF can be applied. We evaluate the proposed scheduling architecture by simulation. The results of the simulation show that our proposed architecture can use the bandwidth efficiently.

1 Introduction

Broadband Wireless Access (BWA) is emerging as a last mile broadband access technology with several advantages: rapid deployment, high scalability, low maintenance and upgrade costs, and granular investment to match market growth [1]. BWA systems are designed to support quality of service (QoS) for real time applications such as video conference, video streaming, and voice over IP. The newly developed IEEE 802.16 standard is one of the BWA systems receiving wide attention from the industry and researchers.

* This work was supported in part by the Samsung Electronics and protected by the patent (10-2005-0036409) which is assigned by the Samsung Electronics.

The IEEE 802.16 standard specifies QoS signaling mechanisms (per connection or per station) such as bandwidth requests and bandwidth allocation. To support QoS, the IEEE 802.16 standard uses the concept of service flow. The upstream service flow types defined in IEEE 802.16 are Unsolicited Grant Service (UGS), Real-Time Polling Service (rtPS), Non-Real-Time Polling Service (nrtPS), and Best Effort (BE). It also specifies the transmission grant mechanisms, such as Grants per Connection (GPC) and Grants per Subscriber Station (GPSS). When a SS (Subscriber Station) requests a bandwidth to the BS (Base Station) in the GPC mode, the BS allocates the requested bandwidth to the SS per connection basis. This means the SS can send only the packets belong to the connection granted by the BS. Therefore, the SS does not need a scheduler to schedule its multiple uplink connections since the uplink transmissions are scheduled by the BS. On the other hand, the SS should have a scheduler when it operates in the GPSS mode. When the BS receives a bandwidth request from the SS in the GPSS mode, it allocates the requested bandwidth to the SS regardless of the number of connections of the SS. Thus, the SS should schedule its multiple uplink connections to efficiently utilize the allocated bandwidth. However, the scheduler architecture and algorithm that determine the uplink and downlink bandwidth allocation and the packet scheduling are not defined in the IEEE 802.16 standard [2].

There are some research results on the scheduling architectures and algorithms of the IEEE 802.16 standard [3-5]. However, most of them focus on the scheduler architectures and algorithms at the BS side while those at the SS side are left unaddressed or just suggest some conceptual scheduling algorithms with no validations. In this paper, we propose uplink scheduling architectures for SSs to efficiently utilize the allocated uplink bandwidth in the GPSS mode: one-level scheduling and two-level scheduling schemes. In the one-level scheduling scheme, only one scheduler exists that manipulates multiple output queues. The two-level scheduling schemes consists of five schedulers and multiple queues - four class schedulers for each service class and one aggregate scheduler for coordinating the class schedulers and sharing the bandwidth among them. We evaluate the proposed scheduling architectures by simulation. The simulation results show that the proposed scheduler utilizes the allocated bandwidth more efficiently than other schedulers.

2 IEEE 802.16 QoS Mechanism

In the IEEE 802.16 standard, there are two channels between a SS and BS: uplink channel (from the SS to the BS) and downlink channel (from the BS to the SS). The downlink channel is the broadcast channel, while the uplink channel is shared by multiple SSs. The frame size is fixed and a frame is consist of uplink and downlink subframes. The BS dynamically determines the duration of uplink and downlink subframes based on its scheduling algorithm. The downlink data transmission is relatively simple because the BS is the only transmitting station during the downlink period. The data packets are broadcast to all SSs and a

SS picks up only the packets destined to it. In the uplink subframe, the BS determines the number of time slots that each SS will be allowed to transmit. This information is broadcast by the BS through the uplink map message (UL-MAP) at the beginning of each frame. After receiving the UL-MAP, each SS knows how long and when it can send data.

In the IEEE 802.16 scheduling architecture, two modules (The Admission Control (AC) module and Uplink Packet Scheduling (UPS) module) reside in the BS for supporting QoS. The AC module handles connection establishments including handshaking connection requests and responses on starting a communication between a SS and BS. On the connection establishment time, bandwidth allocation is performed between the BS and the SS. When the SS requests bandwidth based on its backlogged traffic, the UPS module in the BS sees the request on per station basis and then grants the requested bandwidth to the SS. This type of bandwidth association is called GPSS mode. The standard specifies another bandwidth association scheme called GPC mode. If GPC mode is used, the BS grants bandwidth per connection so that it guarantees the QoS. So the SS scheduler does not need to maintain QoS among its connections and control for sharing the bandwidth among the connections for fairness. Thus, we assume GPSS scheme is used in this system.

The UPS lets the flows that destined to the same destination have the same connection ID in the SS. The connection classifier in the SS classifies data packets with each connection ID and let all packets generated from application layer get into the proper queue. Then the scheduler of the SS picks up the data packet from the queues and transmits it in the appropriate time slots as indicated in the UL-MAP sent by the BS [6]. Based on this architecture, the IEEE 802.16 standard [2] defines the following four categories of service flows to fulfill each flow's various QoS requirements.

Unsolicited Grant Service (UGS): This service is designed for supporting constant bit-rate real-time flows such as voice over IP. To service this kind of traffic, the BS provides fixed size unsolicited data grants on a periodic basis.

Real Time Polling Service (rtPS): The rtPS is for real-time VBR-like flows that generate variable bit-rate data in a period such as MPEG video. The applications belong to this category receive specific bandwidth for not missing deadline.

Non-Real Time Polling Service (nrtPS): This service supports for non-real time flows which are variable size data and requires delay-tolerant data stream service, such as servicing high bandwidth FTP. The nrtPS offers periodic timely unicast request opportunities, so the SS should contend to request bandwidth to the BS.

Best Effort (BE) Service: The service BE is for best effort traffic such as HTTP. There is no QoS guarantee. The applications of this category receive the remained bandwidth after the bandwidth is allocated to the previously mentioned three service flows.

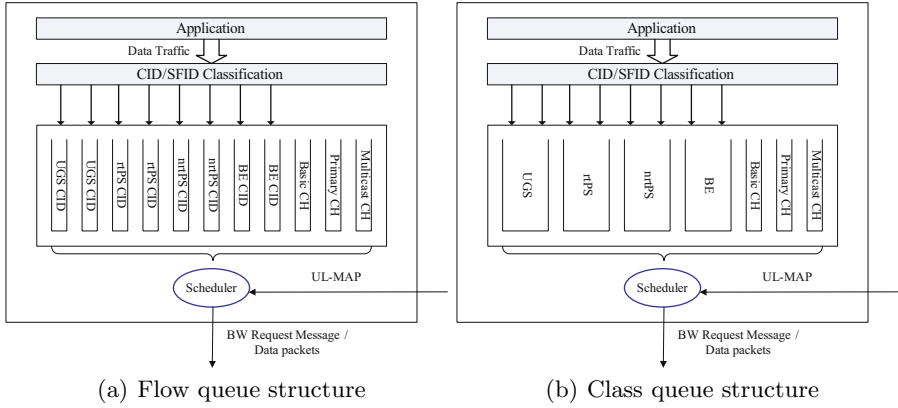


Fig. 1. The architecture of one-level scheduling scheme

3 Proposed Scheduler Architecture

3.1 Normal Scheduler: One-Level Scheduling Scheme

To fulfill various QoS requirements of flows, a scheduling algorithm is required to schedule packets among multiple flows. The general scheduling algorithm, such as SCFQ [7], EDF [8, 9], etc., has one criteria to set a priority of each packet. Therefore one-level scheduling scheme, which is shown in Figure 1, treats all flows equally no matter which class they belong to. The scheduling algorithm used in one-level scheduler has multiple queues and, in general, one queue corresponds to one service flow. But if the scheduler makes one queue whenever new connection is established (one queue per one service flow), the queue management operation may lead to much overhead at the SS. To reduce this overhead, we can use a static queue allocation method. Because the IEEE 802.16 defines only four classes of service flows, it is enough to the scheduler with four queues (one queue per one service class). In this paper, we consider the two cases and refer them to as *flow queue* structure and *class queue* structure, respectively.

There are tradeoffs between the flow queue and class queue. If a scheduler uses a *flow queue* structure, the queue management operation may cause much overhead. But the priority of packets of each flow is not affected by other flows because all packets in the same queue belong to only one flow. Therefore a scheduler can guarantee fairness among all flows. As you can see in Figure 1(a), all flow queues are treated equally without discriminating a flow's priority when the scheduler selects a packet to transmit. If the flows have similar characteristics, it has no problem to use traditional scheduling algorithm. However, if the type of flows is different, the traditional scheduling algorithm may not schedule properly because of its lack of scheduling criteria. The *class queue* divides incoming packets into four categories based on their service classes. When a packet arrives at MAC layer, the packet is inserted into an appropriate queue among four class

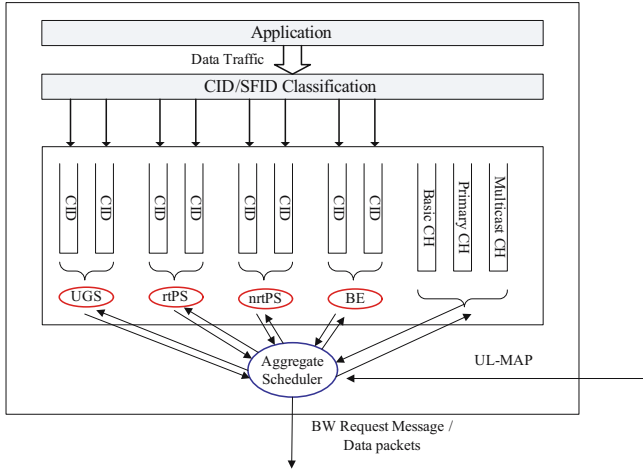


Fig. 2. The architecture of two-level scheduling scheme

queues (see Figure 1(b)). The static allocation of queues can reduce the queue management overhead, but the packets of flows that belong to the same class are inserted into one queue. Therefore an early arrived packet is served earlier than other packets within the same class queue, and the scheduler cannot guarantee fairness among flows belong to the same class.

In these two queue architectures, the scheduler cannot satisfy QoS requirements of each flow, because the scheduling algorithm has only one criteria but flows have different characteristics and QoS requirements. For example, there are two flows: a rtPS class flow requiring guaranteeing service delay bound, and a nrtPS class flow. Suppose that the one-level scheduler uses the SCFQ scheduling algorithm and a class queue. When a packet arrives at queues, the scheduler classifies a class of the packet, calculates its weight, and inserts it into an appropriate class queue. But the SCFQ algorithm does not consider a delay and delay variation of flows, and thus it cannot guarantee a delay bound requirements of flows belong to the rtPS class. When the EDF algorithm is applied, it cannot guarantee a required bandwidth of flows belong to the nrtPS class because it has no methods to guarantee bandwidth. In the class queue structure, the EDF algorithm also cannot guarantee the delay bound of flows belong to same class because all packets on same class are inserted into one queue. If the flow queue is applied, the problems stated above except guaranteeing delay bound among flows belong same class in EDF algorithm, cannot be solved.

3.2 Efficient Scheduler: Two-Level Scheduling Scheme

The two-level scheduling scheme is designed for serving packets as well as satisfying specific QoS requirements of each flow. The two-level scheduling scheme can support various QoS requirements of each flow using a hierarchical scheduler ar-

chitecture. As you can see in Figure 2, two level scheduling scheme consists of aggregate scheduler and four class schedulers. The aggregate scheduler distributes bandwidth to each class scheduler when the BS allocates bandwidth to the SS. When the class scheduler receives bandwidth from the aggregate scheduler, it can serve packets of its flow queues in order of class priority. The class scheduler uses flow queue for classifying packets and various scheduling algorithms can be applied to each class scheduler. In each class scheduler, the two-level scheduling scheme chooses an efficient scheduling algorithm which can guarantee QoS requirements of each service class. Therefore, the two-level scheduling scheme has multiple scheduling criteria and schedules well packets based on appropriate QoS requirements per class. In each class scheduler, the backlogged packets have similar QoS properties and constraints, and the class scheduler only schedules flows having similar QoS constraints. Therefore the flows of each service class can receive more fair scheduling service than the one-level scheduling scheme. The aggregate scheduler distributes bandwidth to each class with proportional ratio based on class priority and the amount of backlogged packets in each class.

3.3 Considering the Aggregate Scheduler

Each class scheduler only transmits packets to the BS based on the allocated bandwidth from the aggregate scheduler. If a class scheduler does not receive sufficient bandwidth from the aggregate scheduler, it may not guarantee the QoS requirement. Therefore the distributing methods of the aggregate scheduler should be carefully designed. For efficient distributing of allocated bandwidth from the BS, the aggregate scheduler should know the amount of backlogged packets of each class scheduler. Because the class scheduler knows the amount of backlogged packets in its queues, the aggregate scheduler retrieves that value from the class scheduler before it distributes the bandwidth.

One possible distribution method is that the aggregate scheduler provides an opportunity for using bandwidth to the UGS class scheduler first. After serving all backlogged packets of the UGS class, the remaining bandwidth is distributed to the rtPS class scheduler. After that, the remaining bandwidth is distributed to the nrtPS and BE class scheduler. Using this distribution method, aggregate scheduler differentiates each class based on the priority of service classes. This method is simple but rtPS or nrtPS flows may not receive sufficient bandwidth. Another possible method is that the aggregate scheduler divides bandwidth into four pieces to satisfy proportional fairness of service classes, and distributes each bandwidth piece to each class scheduler. This method can prevent the starvation of a relative low priority class such as the nrtPS. We compare the performance of these two distribution methods by simulation in the following section.

4 Performance Evaluations

4.1 Simulation Environments

We use a simulation to investigate the effect of scheduling architectures (one-level *vs.* two-level) and types of queue (flow queue *vs.* class queue). We implement a

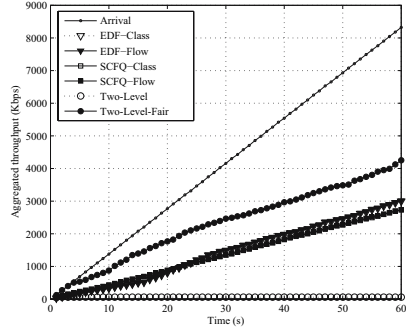
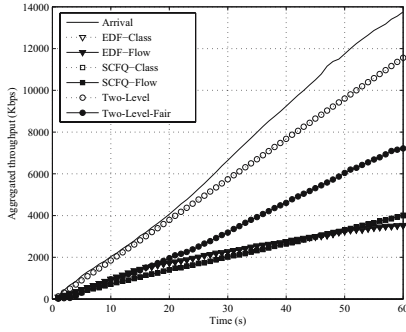
simulator using the C programming language including a BS module, a SS module, a network module, etc. In the BS module, we use a simple round-robin scheduling algorithm to allocate a requested uplink bandwidth because we focus on the efficiency of the scheduling architecture in the SS. The BS module also manages all SSs in its cell and we evaluate the performance of scheduling architectures in one of SSs. When the BS module allocates a requested bandwidth, the SS sends packets according to its scheduling algorithms. In this paper, we use the SCFQ and EDF algorithms in one-level and two-level architectures because of their simplicity and efficiency. Therefore, we also implement the SCFQ and EDF algorithms. We assumed that the size of a queue is unlimited, so there is no packet drop caused by the buffer overflows. In the channel model, we assume that the channel is static and the QPSK modulation scheme which is the default modulation method in the IEEE 802.16 uplink packet transmission is used.

We compare the performance of each scheduling architecture when there are 10 SSs in the cell. A SS has multiple uplink flows belong to each service class. In this simulation, we set the number of flows at each SS is 4 (one flow per one service class) or 8 (two flows per one service class). The simulation runs during 60 simulation seconds, and five simulation results are averaged. In one-level scheduler, we use SCFQ and EDF algorithms with flow and class queues (SCFQ-Flow, SCFQ-Class, EDF-Flow, EDF-Class). In two-level scheduler, the FIFO algorithm is used in the UGS and BE class scheduler, and the EDF and SCFQ algorithms are used in the rtPS and nrtPS class scheduler. In the aggregate scheduler, a priority based distribution method (Two-Level) and a proportional fair distribution method (Two-Level-Fair, which distributes the bandwidth to rtPS and nrtPS class scheduler with 2:1) are used. Therefore the performances of 6 scheduler architectures are compared in this simulation.

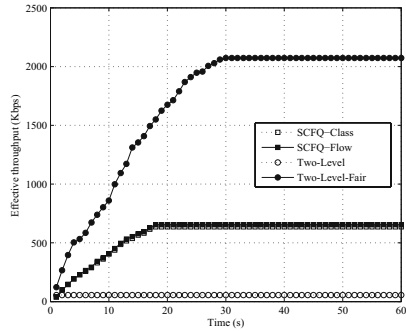
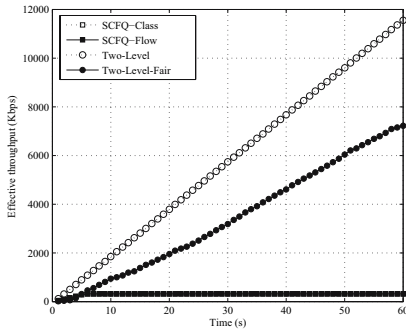
The performance metrics of our simulation results are the aggregate and effective throughput. The aggregate throughput is a cumulative throughput. The effective throughput is an amount of packets which is meaningful at the receiver. When the SCFQ algorithm is used to schedule packets belong to a rtPS class, packets which arrive at queues lately, may remain in the queue because of limited bandwidth. In this case, remaining packets are transmitted at next allocated frame from the BS. However, if these packets do not arrive at the receiver in the required delay bound, they are just dropped and thus waste wireless bandwidth. We refer this wasted bandwidth to useless throughput, and the effective throughput is derived from subtracting the useless throughput from the aggregate throughput.

4.2 Simulation Results

Figure 3 shows the throughputs of rtPS and nrtPS traffics when there are 4 flows in the SS. During our simulation, the compared scheduling algorithms can serve UGS traffics well, so we do not include the throughput results of UGS traffics in this paper. Also, because the traffics of the BE class have no requirements of QoS, throughput results of the BE class are not included. Figures 3(a) and 3(b) show the aggregated throughput of rtPS and nrtPS traffics in one flow per one



(a) Aggregated throughput of rtPS traffics (b) Aggregated throughput of nrtPS traffics



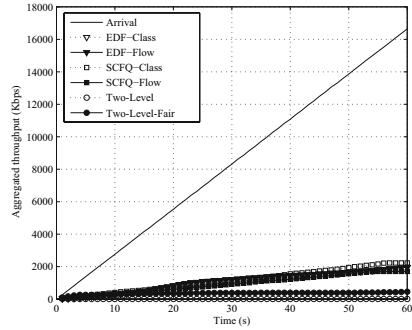
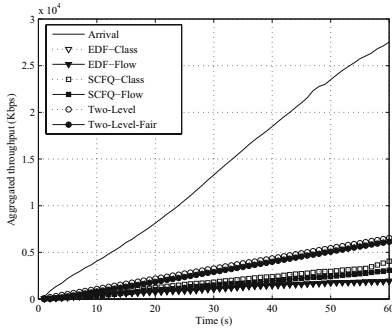
(c) Effective throughput of rtPS traffics (d) Effective throughput of nrtPS traffics

Fig. 3. Throughput of rtPS and nrtPS traffics (4 flows)

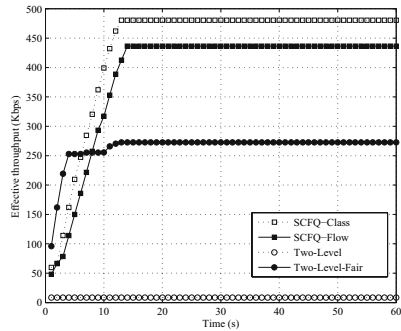
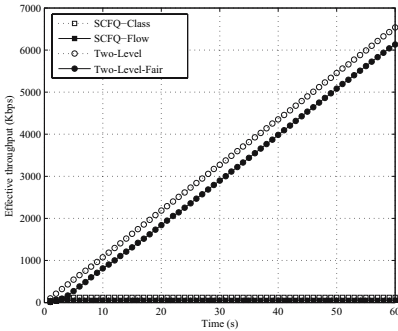
class case. When there is one flow per class, the class queue and flow queue is the same. Therefore, plots of the SCFQ-Class and SCFQ-Flow, EDF-Class and EDF-Flow are overlapped.

Comparing the throughput between two traffics, we can see that the two-level scheduler achieves the highest throughput for the rtPS traffics, but lowest throughput for the nrtPS traffics. This means that all the remaining bandwidth is used for serving rtPS traffics after serving UGS traffics. However, the two-level-fair can efficiently share the remaining bandwidth between rtPS traffics and nrtPS traffics. We also derive the useless throughput of each scheduler, but do not include in this paper. In the useless throughput plots, we can see that there are no useless throughput in two-level and two-level-fair schedulers. This means that the two-level-fair scheduler efficiently distributes the allocated bandwidth to rtPS and nrtPS traffics with no throughput degradation.

The SCFQ based algorithm (SCFQ-Class and SCFQ-Flow) uses the weighted sharing of the allocated bandwidth from the BS. In our simulation, the SCFQ uses the same weight to each service class. When the application generates more data than the scheduler can serve, data packets are stored in the queue of the



(a) Aggregated throughput of rtPS traffic (b) Aggregated throughput of nrtPS traffic



(c) Effective throughput of rtPS traffic (d) Effective throughput of nrtPS traffic

Fig. 4. Throughput of rtPS and nrtPS traffic (8 flows)

SS and served at the next interval. In this case, many data packets experience much delay resulting drops at the destination because the delay constraints of the rtPS traffic are violated. But the SCFQ-based algorithm just sends data packets in enqueueing order because there is no delay consideration in the SCFQ-based algorithm. So we derive the effective throughput results as shown in Figures 3(c) and 3(d). The effective throughput is derived by subtracting the useless throughput from the aggregated throughput. In these figures, we also know that the two-level-fair scheduler can efficiently use the bandwidth sharing between rtPS and nrtPS traffic although the throughput of rtPS traffic is lower than the two-level algorithm.

Figure 4 shows the aggregate and effective throughput of rtPS and nrtPS traffic when there are 8 flows. In Figures 4(a) and 4(b), the aggregate throughput of rtPS traffic in the two-level scheme is almost same as that of two-level-fair, but the aggregate throughput of nrtPS traffic in the two-level scheme is almost zero. In Figures 4(c) and 4(d), we can see that the effective throughput of rtPS traffic in the two-level-fair scheme is almost same as that of two-level, and the two-level-fair scheme also can support the nrtPS traffic much well than the two-level scheme.

5 Conclusions

The IEEE 802.16 defines four service classes, USG, rtPS, nrtPS and BE. To guarantee the required QoS of these classes, the SS and BS should have a scheduling architecture and algorithm which are not defined in the standard.

In this paper, we propose two types of scheduling architectures: One-level scheduler and Two-level scheduler. Between them, the two-level scheduler can provide more organized QoS service than the one-level scheduler. Adapting these architectures makes scheduler control efficiently all types of traffic defined in the IEEE 802.16. We evaluate the proposed scheduling architecture by simulation. The results of the simulation show that the proposed scheduler can use the bandwidth efficiently than other schedulers. In the two-level scheduler, the bandwidth distribution method of the aggregate scheduler is a critical factor on the performance and should be carefully designed.

References

1. GuoSong Chu, Deng Wang, and Shunliang Mei, *A QoS Architecture for the MAC Protocol of IEEE 802.16 BWA System*, IEEE International Conference on Communications, Circuits and Systems and West Sino Expositions, vol. 1, pp. 435-439, Jul. 2002.
2. IEEE 802.16-2004, *Air Interface for Fixed Broadband Wireless Access Systems*
3. Mohammed Hawa and David W. Petr, *Quality of Service Scheduling in Cable and Broadband Wireless Access Systems*, IEEE International Workshop on Quality of Service, pp. 247-255, May 2002.
4. G.Nair, J.Chou, T.Madejski, K.Perycz, D.Putzolu and J.Sydir, *IEEE 802.16 medium access control and service provisioning*, Intel Technology Journal, vol. 8, no. 3, pp. 213-228, Aug. 2004.
5. Dong-Hoon Cho, Jung-Hoon Song, Min-Su Kim, and Ki-Jun Han, *Performance Analysis of the IEEE 802.16 Wireless Metropolitan Area Network*, IEEE International Conference on Distributed Frameworks for Multimedia Applications, 2005.
6. Kitti Wongthavarawat and Aura Ganz, *Packet scheduling for QoS support in IEEE 802.16 broadband wireless access systems*, International Journal of Communication System, vol. 16, pp. 81-96, John Wiley & Sons, 2003.
7. S.Jamaloddin Golestani, *A Self-Clock Fair Queueing Scheme for Broadband Applications*, IEEE INFOCOM'94, Jun. 1994.
8. L.Georgiadis, R.Guerin, and A.Parekh, *Optimal multiplexing on a single link: delay and buffer requirements*, IEEE Transactions on Information Theory, vol. 43, no. 5, pp. 1518-1535, Sep. 1997.
9. J.Liebeherr, D.Wrege, and D.Ferrari, *Exact admission control for networks with a bounded delay service*, IEEE/ACM Transactions on Networking, vol. 4, no. 6, pp. 885-901, Dec. 1996.