Simulating IEEE 802.16 Uplink Scheduler Using NS-2 [*]

Seungwoon Kim
School of Engineering
Information and
Communication University
Daejeon, South Korea
swkim@icu.ac.kr

Minwook Lee
Dept. of Computer Science
Korea Advanced Institute of
Science and Technology
Daejeon, South Korea
leeminwook@kaist.ac.kr

Ikjun Yeom
Dept. of Computer Science
Korea Advanced Institute of
Science and Technology
Daejeon, South Korea
yeom@cs.kaist.ac.kr

ABSTRACT

Since the IEEE 802.16 standard has been defined for wireless technology, there have been many efforts to analyze and evaluate the performance of 802.16 networks. In this paper, we present an ns-2 simulation module for IEEE 802.16 uplink scheduling. We describe in detail the algorithm used, and validate our module through simulations of a typical network architecture.

Keywords

IEEE 802.16, ns-2 simulator, Uplink scheduling

1. INTRODUCTION

IEEE 802.16 MAN [1] is a promising technology for wireless access links. It overcomes several of the drawbacks held by legacy wireless technologies, such as limited coverage and high communication costs. Many recent studies have examined the performance of IEEE 802.16 networks [2, 16, 18]. It has also been receiving wide attention as a commercial service; in 2006, for example, a mobile service named *WiBro* based on IEEE 802.16e [4] was launched in South Korea.

Novel network technologies can be evaluated and analyzed through a number of methods, including mathematical analysis, simulations, and experiments. Mathematical analysis can be used to provide the theoretical background of a given technology, but most tractable models are over-simplified. Experiments with real equipment are useful for monitoring the status of a system, but they are hard to perform in early stages of development. Thus, simulations play a very important role in network research. They are especially helpful in evaluating the performance of emerging technologies, since they do not need any real equipment and produce repeatable results. Recent performance evaluations of IEEE 802.16 have mostly relied on simulations. The importance of network simulations is thoroughly discussed in Paxson and Floyd's landmark essay "Why we don't know how to simulate the Internet" [8].

There are several well-known tools for network simulation, such as ns-2 [5], OPNET [9], and Qualnet [10]. The first of these, ns-2, is free, open-source software widely used in academia. It can be used to set up accurate packet-level simulations for a variety of network types, including wireless and optical. ISI (Information Sciences Institute), NIST (National Institute of Standards and Technology), and other organizations are currently developing a new version of ns-2. CMU's (Carnegie Mellon University) Monarch project has used ns-2 to develop simulation modules for IEEE 802.11-based WLAN and ad-hoc networks, and these tools are now widely used.

In this paper, we present a new simulation module[1] for IEEE 802.16. In our earlier works [11, 12], we have evaluated best-effort traffic performance in IEEE 802.16 networks using ns-2. This paper extends that work to all classes of traffic: UGS (Unsolicited Grant Service), rtPS (Real-time Polling Service), nrtPS (non-real-time Polling Service), and BE (Best-Effort). There have been several recent simulation studies of IEEE 802.16 [2, 16] networks, most of which have developed their own tools for obtaining specific performance measures. Such works are difficult to adapt to other purposes. Two works, however, have developed IEEE 802.16 simulation modules for ns-2 [13, 14]. These tools focus on the physical characteristics of networks and mobility. The module presented in this paper is designed to simulate the uplink scheduling of various traffic models, which is one of the most interesting features of IEEE 802.16.

The simulation module in this paper conforms to the IEEE 802.16 standard. We design simple and straightforward schemes for uplink scheduling and admission control, so that these building blocks can be easily adapted to future research. Since the module is based on a layered architecture, it can also be easily merged with modules simulating the physical characteristics and mobility of the network [13, 14]. We believe that the simulation module presented in this paper will be useful for future research on IEEE 802.16.

The rest of this paper is organized as follows: In Section 2, we summarize the MAC protocol of the IEEE 802.16 standard and describe how the simulation module conforms to the standard. In Section 3, we present the uplink schedul-

---

[*] This work was supported by SK Telecom Co.

[1] Available at http://cnlab.kaist.ac.kr.

SimulationWorks 2008, March 04, Marseille, France

DOI 10.4108/ICST.SIMUTOOLS2008.3100

ing and admission control algorithms for the various QoS classes defined in the IEEE 802.16 standard. In Section 4 we validate our module through various simulation scenarios, and show that it provides a useful and accurate performance evaluation. Section 5 summarizes our results.

2. IEEE 802.16 COMMUNICATION ARCHITECTURE

An IEEE 802.16 network consists of one base station (BS) and a set of subscribe stations (SS) sharing a channel. By default, the channel is shared between uplinks and downlinks using TDD (Time Division Duplexing). Both TDM (Time Division Multiplexing) and FDM (Frequency Division Multiplexing) are also supported by the standard. Our simulator supports TDD and TDM, but excludes FDM since we wish to focus on the MAC protocol associated with scheduling rather than on the physical characteristics of the network.

In Fig. 1, we present a typical IEEE 802.16 network architecture. The IEEE 802.16 standard supports various QoS requirements for the individual connections. If an SS wants to open a connection to the BS, it first sends a request. Upon receiving the message, the BS performs admission control based on the requested traffic, the QoS specification, and currently available resources. Once the connection is established, the SS may obtain a particular bandwidth by sending a class-specific request (e.g. polling, contention-based, etc.). The BS then aggregates all requests and allocates bandwidth to each connection or SS through an appropriate scheduling scheme.

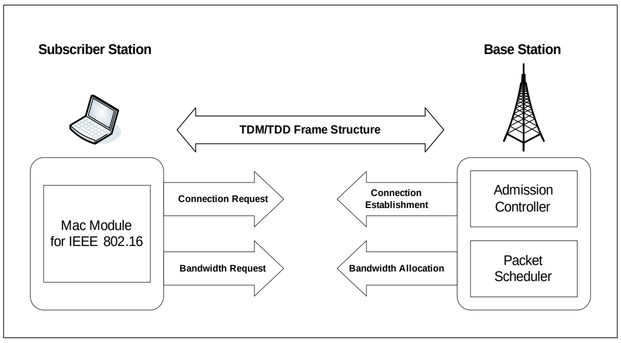


Figure 1: IEEE 802.16 network architecture

In Fig. 2, we illustrate the frame structure of an IEEE 802.16 network. Data is transmitted in a sequence of frames, each one divided into two subframes for the downlink and uplink. A subframe consists of multiple data bursts, and the actual downlink or uplink transmission is performed via reading and writing to a data burst. The BS performs uplink and downlink scheduling, and allocates data bursts to each SS. Information on the burst allocation is transmitted in DL (Downlink) and UL (Uplink) MAPs, which are located at the beginning of each downlink subframe.

3. UPLINK SCHEDULER

One of the advantages of an IEEE 802.16 network is that its uplink scheduling supports multiple QoS classes. To accommodate heterogeneous traffic with various QoS requirements, the MAC protocol of IEEE 802.16 is connection-oriented and employs strict admission control. When establishing a new connection, the SS sends a request message containing its class and traffic specifications. The connection is accepted when enough resources are available to accommodate it. The SS can then send or receive data through the allocated bandwidth. The bandwidth allocation is performed by the BS according to an appropriate scheduling scheme. Although the IEEE 802.16 standard does not specify a scheduling scheme, it is obvious that downlink scheduling is much simpler than uplink scheduling since the BS has complete information on downlink traffic. It has been shown that a simple Deficit Round Robin (DRR) scheduler [15] works well for downlink scheduling [16]. The simulation module presented in this paper thus focuses on the more difficult problem of uplink scheduling.

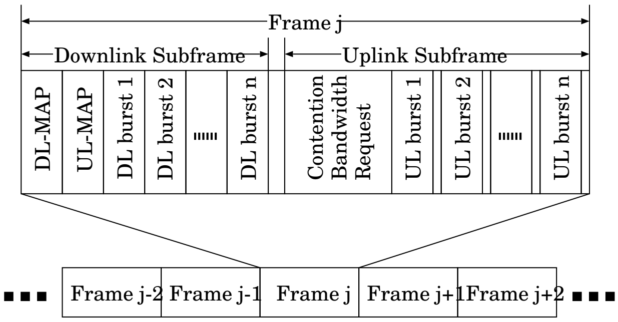


Figure 2: Frame structure

In uplink scheduling, the BS allocates bandwidth among the SS based on their requests. The mechanism is different for each service class, however, as follows:

- UGS (Unsolicited Grant Service) has the highest priority. This class is used for legacy voice and VoIP traffic services, which generate fixed length packets at uniform intervals. They are thus not required to request bandwidth for individual packets. Rather, the BS regularly assigns slots according to the traffic specification once the connection has been accepted.
- rtPS (real-time Polling Service) has the next highest priority, and is used for multimedia streams. Most multimedia streams also send packets at uniform intervals, but the packets are of variable size. Thus, the BS polls the connection periodically to find out how much bandwidth is needed.
- nrtPS (non real-time Polling Service) is for any other service which requires more resources than a Best-Effort connection. It is essentially the same as the BE class, except that it may be given additional bandwidth through occasional (irregular) polling.
- BE (Best-Effort) services are only allowed to make contention-based requests; that is, they must compete for several shared bandwidth slots along with other services of this class.

In the remainder of this section, we will describe the implementation of our simulation module in detail.

3.1 Connection Establishment

The connection establishment process consists of a connection request and a grant, along with an appropriate admission control scheme. When an SS wants to be connected to the BS, it sends a connection request message containing {traffic class, start time, end time, sending rate, packet size}. Connection establishment for each class is then performed as follows:

- UGS: The BS examines the time interval defined in the request, and checks where there are enough resources to grant it. If so, bandwidth is allocated to the connection during the requested interval. Note that the packet size of a UGS connection is expected to be constant.
- rtPS: As the UGS connection, except that packet size is taken as an average value.
- nrtPS and BE: For an nrtPS or BE connection, there is no admission control because they do not require a specific QoS. Upon receiving a request for these two classes of connection, the BS simply accepts it and sends a connection identifier back to the SS. Note that traffic class information is only required for the nrtPS and BE connections.

The UGS and rtPS connections should be strictly controlled, since they require a uniform QoS. In our simulation module, admission control is based on bandwidth. We initially set bandwidth limits for the UGS and rtPS classes, only accepting new connections when the total traffic of current activity and accepted requests does not exceed this limit. It may be that more elaborate methods provide better admission control, but our modules focuses on providing a general framework for uplink scheduling in IEEE 802.16 network simulations. A discussion of specific admission control schemes lies beyond the scope of this paper. We believe that this simple scheme can easily be extended in future studies.

3.2 Bandwidth Scheduling

For each frame, the BS performs uplink scheduling on all established UGS and rtPS connections, as well as any nrtPS or BE connections that successfully transmitted bandwidth requests. The scheduling results are stored by UL-MAP in the downlink subframe. In the rest of this section, we describe the scheduling procedure for each service class.

3.2.1 UGS

UGS traffic is scheduled with the highest priority without the need for explicit bandwidth requests. Since a UGS connection needs a fixed amount of bandwidth periodically, we simply record the packet size and period of each connection. In our simulation module, we also maintain a variable which indicates the time of the next bandwidth allocation. This procedure is described in lines 13–14 of Algorithm 1.

In our simulation module, scheduling is performed for each frame but no particular order is forced on the data within a frame. Thus, higher priority traffic is always located near the beginning of the frame. When a frame is not large enough to contain all the scheduled UGS connections, some of them are moved to the next frame. To prevent any one connection from being scheduled in the next frame over and over again, we randomize the scheduling order of UGS connections.

Algorithm 1 Pseudo code for uplink scheduling

```
 1: for each frame do
 2:   for each rtPS connection $r[i]$ do
 3:     if $r[i].next < now + 2\tau$ then
 4:       schedule unicast polling slot for $r[i]$
 5:     end if
 6:   end for
 7:   for each nrtPS connection $n[i]$ do
 8:     if $n[i].polling$ is set then
 9:       schedule unicast polling slot for $n[i]$
10:       reset $n[i].polling$
11:     end if
12:   end for
13:   for each UGS connection $u[i]$ do
14:     if $u[i].next < now + \tau$ then
15:       schedule slots for $u[i]$
16:       $u[i].next = u[i].next + u[i].interval$
17:     end if
18:   end for
19:   for each rtPS connection $r[i]$ do
20:     if $u[i].next < now + \tau$ then
21:       schedule slots for $r[i]$
22:       $r[i].next = r[i].next + r[i].interval$
23:     end if
24:   end for
25:   for each nrtPS and BE connection with non-empty
      request do
26:     schedule the remaining slots equally among them
27:     for each nrtPS connection whose request has just
        been fulfilled do
28:       set $n[i].polling$
29:     end for
30:   end for
31: end for
```

3.2.2 rtPS

After all the UGS traffic has been dealt with, rtPS traffic is scheduled. The amount of bandwidth available for rtPS traffic is determined periodically. The BS maintains two variables: the polling period and the time of the next bandwidth allocation. The rtPS scheduling algorithm operates on two consecutive frames. In the first frame, the BS schedules slots to the rtPS connections for their bandwidth requests. In the next frame, the BS schedules the actual data transmission. The rtPS scheduling procedure is described in lines 2–6 and lines 19–24 of the algorithm. Note that although polling for rtPS connections occurs prior to UGS scheduling, the *scheduling* of data transmission only occurs afterwards. Thus, rtPS scheduling does not have much impact on UGS scheduling.

3.2.3 nrtPS and BE

In the IEEE 802.16 standard, an nrtPS service is allowed to request both sporadic (non-periodic) polling and contention-

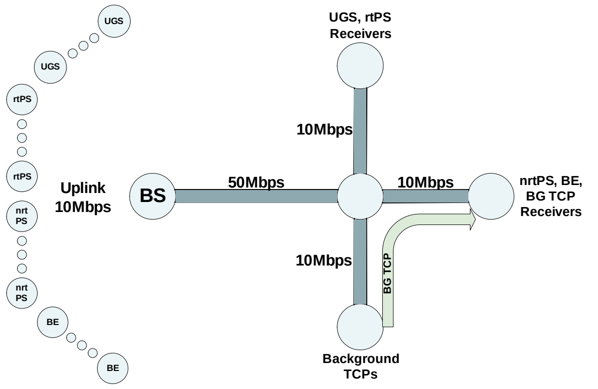


Figure 3: Simulation topology

based scheduling. The mechanism for sporadic polling, however, is not defined. In our simulation module, we allow both nrtPS and BE services to send contention-based requests. In each frame, a certain number of slots are allocated to contention-based requests (this number is set at the beginning of the simulation). For an nrtPS connection, we will poll the connection when it has been allocated the total amount of its requested bandwidth.

When nrtPS and BE connections have data to send, they send a contention-based bandwidth request to the BS. When multiple connections access the same slot by request, a collision occurs and they need to retransmit. Otherwise, the BS receives the request and allocates slots to the connection in the next frame. After allocating slots for the UGS and rtPS connections (lines 2–24 of the algorithm), the BS divides its remaining slots equally between nrtPS and BE connections. Thus, there is no difference between the nrtPS and BE connections once they have obtained their bandwidth. The benefit of an nrtPS service is that it has more chances to request bandwidth via non-periodic polling. In our simulation module, there is a chance for another request when a connection has been granted all of its requested bandwidth. If the connection still has data to send, then it may be able to request bandwidth without contention. Otherwise, like the BE service it must wait to send a contention-based request later.

4. SIMULATION RESULTS

In this section, we validate our simulation module under various scenarios. Fig. 3 presents the simulation topology. A set of mobile nodes are connected to a BS through an IEEE 802.16 wireless link. The BS is connected to a wired network. The uplink bandwidth is set to 10 Mbps, and the frame length is 5 ms. Each mobile node has one connection, belonging to one of the four classes. The total number of nrtPS and BE connections is set to 10, but we vary the numbers of UGS and rtPS connections. We use TCP for the nrtPS and BE connections, and UDP for the UGS and rtPS connections. To observe the throughput achieved by the nrtPS and BE connections, we send them through a separate link with background TCP traffic. Using this topology, we devise three simulation scenarios.

In the first scenario, we examine the performance of UGS traffic. The sending rate of each UGS connection is 0.1 Mbps, and the packet size is set to 0.5 kB. We increase the number of UGS connections by one every second, to a maximum of 50. There is one rtPS connection with a sending rate of 1 Mbps and a packet size of 1 kB. Fig. 4 presents the results of this simulation.

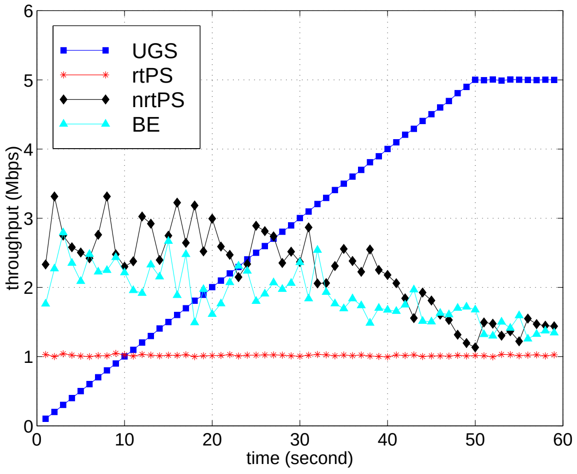
Fig. 4(a) presents the aggregated throughput achieved by each class of connections. Since they have the highest priority, the throughput achieved by the UGS connections increases to 5 Mbps over the course of the simulation. The throughput of the single rtPS connection is maintained, while the throughput of the nrtPS and BE connections decreases as the number of UGS connections increases. Note that the throughput of the nrtPS connections is also slightly higher than that of the BE connections. This simulation thus conforms to the standard of IEEE 802.16 uplink scheduling, in the sense that UGS and rtPS packets are scheduled with a higher priority than nrtPS and BE packets, and that nrtPS connections may acquire additional bandwidth through non-periodic polling.

In Fig. 4(b), we compare the worst-case delays of UGS and rtPS packets during each second of the simulation. (Delay is defined as the length of time a packet stays in the interface queue.) Note that the initial 5 ms delay of the UGS packets corresponds to the frame length. The rtPS packets observe additional 5 ms delay for the polling period. The worst case delay of both classes increases with the number of UGS connections. At 39 seconds, the delay of rtPS packets suddenly increases. This is due to the fact that some packets cannot be scheduled immediately and are delayed to the next frame. Five seconds later, the delay of UGS packets also jumps for the same reason. This result also confirms that our simulation module is properly scheduling the UGS and rtPS packets properly. UGS packets should observe very low delay when the amount of UGS traffic is managed properly (40 % in this scenario).

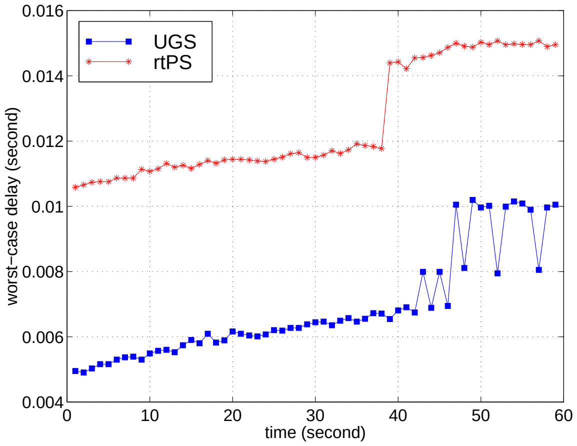
In the second scenario, we fix the number of UGS connections at 10 and vary the number of rtPS connections from 1 to 50. The sending rate of each rtPS connection is 0.1 Mbps. The throughput of each class in this simulation, shown in Fig. 5(a), is quite similar to that seen in Fig. 4(a). The rtPS packets are still scheduled with a higher priority than the nrtPS and BE packets. In Fig. 5(b), we present the worst-case delay of the UGS and rtPS packets. Unlike Fig. 4(b), we see here that the UGS delay is not affected by the increasing rtPS traffic. This confirms that UGS packets are still scheduled with the highest priority.

In the last scenario, we compare the nrtPS and BE classes. The aggregated sending rate of UGS and rtPS connections is fixed at 1 Mbps. We vary the number of request slots for nrtPS and BE connections from 2 to 10. In Fig. 6, we present the aggregated throughput of nrtPS and BE connections and the success rate of bandwidth requests. The parameter k is the maximum amount of data per request. We may notice the following features:

- nrtPS connections realize higher throughput than BE connections;

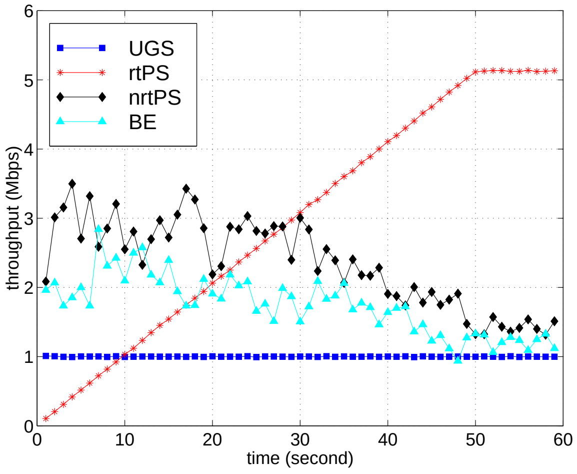


(a) Throughput

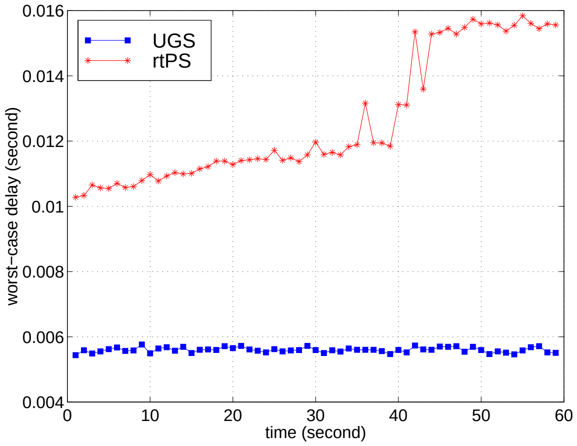


(b) Delay

Figure 4: Impact of UGS traffic



(a) Throughput



(b) Delay

Figure 5: Impact of rtPS traffic

- as the number of request slots increases, the success rate of requests increases;
- the difference between nrtPS and BE throughput decreases as the number of connections increases, since the principal advantage of nrtPS is less important;
- a larger value of k reduces the number of requests, so the success rate increases; and
- BE connections with a larger k achieve higher throughput, assuming the same number of available request slots.

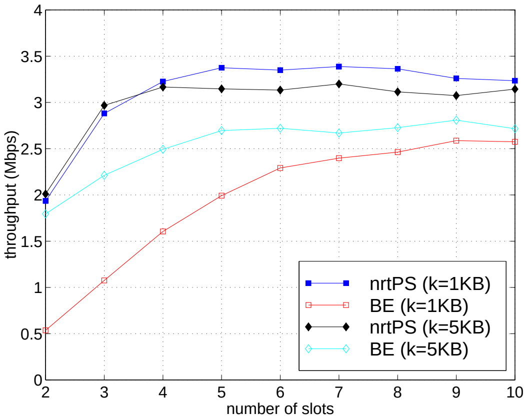
A larger value of k may induce unfair bandwidth sharing over the short term. There is thus a trade-off between throughput and fairness to be considered when choosing k. However, a detailed analysis of this parameter is beyond the scope of this paper. For now, the result simply confirms that our module effectively simulates the behaviors of nrtPS and BE traffic.

5. SUMMARY

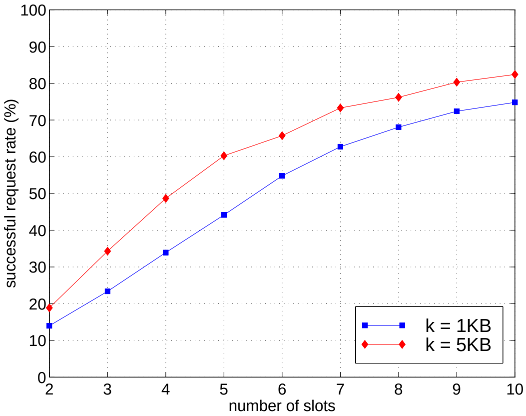
In this paper, we presented an ns-2 simulation module for the IEEE 802.16 uplink scheduler. The architecture of the module was described in detail. Simulations of several different scenarios have shown that our module conforms to the IEEE 802.16 standard and can be used to evaluate the performance of four service classes. We expect that our simulation module will be useful to anyone investigating IEEE 802.16 networks.

6. REFERENCES

[1] IEEE 802.16-2004, “IEEE Standard for Local and Metropolitan Area Networks-Part 16: Air Interface for



(a) Throughput



(b) Success rate of request

Figure 6: nrtPS and BE

Fixed Broadband Wireless Access Systems,” Oct. 2004.

[2] C. Cicconetti et al., “Quality of Service Support in IEEE 802.16 Networks,”, *IEEE Networks*, Mar./Apr. 2006, pp. 50-55.

[3] “WiBro: Wireless Broadband,” Available via http://www.wibro.or.kr.

[4] IEEE 802.16e Task Group, “Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands,” IEEE Std 802.16e-2005, Feb. 2006.

[5] L. Breslaau et al., “Advances in Network Simulation,” *IEEE Computer*, vol. 33, no. 5, May 2000, pp. 59-67.

[6] D. Bertsekas and R. Gallager, *Data Networks*, Prentice-Hall, Englewood Cliffs, New Jersey, 1992.

[7] D. Clark and W. Fang, “Explicit Allocation of Best Effort Packet Delivery Service,” *IEEE/ACM Trans. on Networking*, vol. 6, no. 4, Aug. 1998, pp. 362-373.

[8] V. Paxson and S. Floyd, “Why we don’t know how to simulate the Internet,” *Proc. of the 1997 Winter Simulation Conference*, Dec. 1997.

[9] OPNET. Available via http://www.opnet.com.

[10] QualNet. Available via http://www.scalable-networks.com.

[11] S. Kim and I. Yeom, “TCP-Aware Uplink Scheduling for IEEE 802.16,” *IEEE Communications Letters*, Feb. 2007.

[12] S. Kim and I. Yeom, “Performance Analysis of Best Effort Traffic in IEEE 802.16 Networks,” Tech. Report, KAIST, Nov. 2006.

[13] Seamless and Secure Mobility. Available via http://www.antd.nist.gov/seamlessandsecure.shtml.

[14] The design and implementation of WiMAX Module for ns-2 Simulator. Available via http://ndsl.csie.cgu.edu.tw/wimax_ns2.php.

[15] M. Shreedhar and G. Varghese, “Efficient Fair Queueing using Deficit Round Robin,” *IEEE/ACM Transactions on Networking*, vol.4 no. 3, June 1996.

[16] C. Cicconetti et al., “Performance Evaluation of the IEEE 802.16 MAC for QoS Support,” *IEEE Trans. on Mobile Computing*, vol. 6, no. 1, Jan. 2007, pp. 26-38.

[17] J. Chen et al., “The Design and Implementation of WiMAX Module for ns-2 simulator,” *Proc. ACM VALUETOOLS 2006(WNS2 ’06)*, vol. 202, Pisa, Italy, Oct., 2006.

[18] A. Sayenko et al., “Ensuring the QoS Requirements in 802.16 Scheduling,” *Proc. ACM MSWiM’06*, Malaga, Spain, Oct., 2006.