INTERNATIONAL JOURNAL OF COMMUNICATION SYSTEMS
Int. J. Commun. Syst. 2003; **16**:81–96 (DOI: 10.1002/dac.581)

Packet scheduling for QoS support in IEEE 802.16 broadband wireless access systems

Kitti Wongthavarawat*,† and Aura Ganz‡

Multimedia Networks Laboratory, Electrical and Computer Engineering Department, University of Massachusetts, Amherst, MA 01003, U.S.A.

SUMMARY

In this paper we introduce a scheduling algorithm and admission control policy for IEEE 802.16 broadband wireless access standard. The proposed solution which is practical and compatible to the IEEE 802.16 standard, provides QoS support to different traffic classes. To the best of our knowledge this is the first such algorithm. The simulation studies show that the proposed solution includes QoS support for all types of traffic classes as defined by the standard. We have shown the relationship between traffic characteristics and its QoS requirements and the network performance. Copyright © 2003 John Wiley & Sons, Ltd.

KEY WORDS: packet scheduling; QoS; IEEE 802.16; wireless access

1. INTRODUCTION

Broadband wireless access (BWA) systems, e.g. IEEE 802.16 standard [1], provide fixed-wireless access between the subscriber station (residential or business customers) and the internet service provider (ISP) through the base station. BWA systems complement existing last mile wired networks such as cable modem and xDSL. Due to the upcoming air interface technologies which promise to deliver high transmission data rates, BWA systems become an attractive alternative. Their main advantage is their fast deployment which can result in cost savings. For example, such installations can be beneficial in (1) very crowded geographical areas such as cities or in (2) rural areas where there is no wired infrastructure. Without new cable wiring for the whole city, the antenna of the base station and subscriber customers are easily set up at the rooftop of their buildings to form the wireless network. BWA systems are expected to support quality of service (QoS) for real time applications such as video conferencing, video streaming, and voice over IP.

*Correspondence to: Kitti Wongthavarawat, Multimedia Networks Laboratory, Electrical and Computer Engineering Department, University of Massachusetts Amherst, MA 01003, U.S.A
†E-mail: kwongtha@ecs.umass.edu
‡E-mail: ganz@ecs.umass.edu

Contract/grant sponsor: NSF; contract/grant number: NSF-CISE 0087945; NSF-CISE 0080119; NSF-CISE 9812589; NSF-ANI 0230812
Contract/grant sponsor: DARPA; contract/grant number: F33615-02-C-4031

Received 31 March 2002
Revised 30 June 2002
Accepted 31 August 2002

Such applications are delay and delay variation sensitive, i.e. in case packets incur large delays and delay variation, the quality of the application is severely degraded. QoS support in BWA systems will provide additional sources of revenue and differentiation for wireless ISPs. In other words, if customers require a better access (guaranteed QoS) to the Internet, they will have to pay a premium for such services.

The IEEE 802.16 broadband wireless access standard developed by the IEEE 802.16 working group on broadband wireless access [2] was recently approved. IEEE 802.16 media access control, which is based on the concepts of connections and service flows, specifies QoS signaling mechanisms (per connection or per station) such as bandwidth requests and bandwidth allocation. However, IEEE 802.16 standard left the QoS based packet scheduling algorithms, that determine the uplink and downlink bandwidth allocation, undefined.

In recent years, several packet scheduling algorithms for broadband wireless networks were published [3–9]. To the best of our knowledge, there is no proposed packet scheduling solution specifically designed for IEEE 802.16. In this paper, we propose a packet scheduling algorithm that provides QoS support for a wide range of real time applications as defined in IEEE 802.16. The proposed solution is practical and compatible with the IEEE 802.16 QoS signaling mechanisms. The simulation results we obtained show that the proposed solution can support diverse traffic classes of traffic with different QoS requirements in terms of bandwidth and maximum delay.

The paper is organized as follows. In Section 2 we introduce the IEEE 802.16 broadband wireless access systems. In Section 3, we describe the existing IEEE 802.16 QoS architecture as well as our proposed QoS architecture. The terminology used in this paper is provided in Section 4. In Section 5, we describe in details the proposed uplink packet scheduling (UPS) algorithm. The admission control policy is introduced in Section 6. Section 7 provides simulation results of our proposed UPS algorithm and Section 8 concludes the paper.

2. IEEE 802.16 BROADBAND WIRELESS ACCESS SYSTEMS

The Physical layer operates at 10–66 GHz (IEEE 802.16) and 2–11 GHz (IEEE 802.16a) with data rates of 32–130 Mbps depending on the channel frequency width and modulation technique. IEEE 802.16 architecture consists of two kinds of fixed (non-mobile) stations: subscriber stations (SS) and a base station (BS). The BS regulates all the communication in the network, i.e. there is no peer-to-peer communication directly between the SSs. The communication path between SS and BS has two directions: uplink (from SS to BS) and downlink (from BS to SS). When the system uses time-division multiplexing (TDM), for uplink and downlink transmissions, the frame is subdivided into an uplink subframe and a downlink subframe (see Figure 1). The duration of these subframes is dynamically determined by the BS. Each subframe consists of a number of time slots. SSs and BS have to be synchronized and transmit data into predetermined time slots.

IEEE 802.16 can support multiple communication services (data, voice, video) with different QoS requirements. The media access control (MAC) layer defines QoS signaling mechanisms and functions that can control BS and SS data transmissions. On the downlink (from BS to SS), the transmission is relatively simple because the BS is the only one that transmits during the downlink subframe. The data packets are broadcasted to all SSs and an SS only picks up the packets destined to it. One of the modes of uplink arbitration (from SS to BS) uses a TDMA MAC. The BS determines the number of time slots that each SS will be allowed to transmit in an

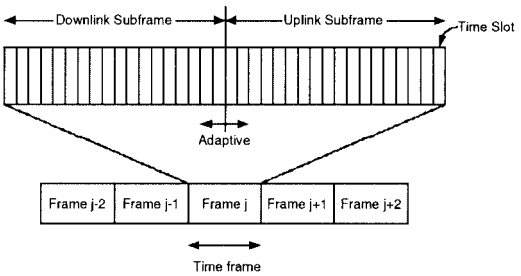

Figure 1. IEEE 802.16 TDM frame structure.

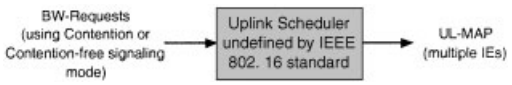

Figure 2. Undefined uplink scheduler in IEEE 802.16.

uplink subframe. This information is broadcasted by the BS through the uplink map message (UL-MAP) at the beginning of each frame. UL-MAP contains information element (IE) which include the transmission opportunities, i.e. the time slots in which the SS can transmit during the uplink subframe. After receiving the UL-MAP, each SS will transmit data in the predefined time slots as indicated in IE. The BS uplink scheduling module determines the IEs using bandwidth request PDU (BW-request) sent from SSs to BS. In IEEE 802.16 standard, there are two modes of transmitting the BW-Request: contention mode and contention-free mode (polling). In contention mode, SSs send BW-Request during the contention period. Contention is resolved using back-off resolution. In contention-free mode, BS polls each SS and SSs reply by sending BW-request. Due to the predictable signaling delay of the polling scheme, contention-free mode is suitable for real time applications. IEEE 802.16 defines the required QoS signaling mechanisms described above such as BW-Request and UL-MAP, but it does not define the Uplink Scheduler (see Figure 2), i.e. the mechanism that determines the IEs in the UL-MAP.

3. QOS ARCHITECTURE

IEEE 802.16 defines four types of service flows, each with different QoS requirements and corresponding uplink scheduler policy:

1. *Unsolicited grant service* (*UGS*)—this service supports constant bit-rate (CBR) or CBR-like flows such as Voice over IP. These applications require constant bandwidth allocation.
   - BW-Request: Not required.
   - Uplink Scheduler: BS determines the IEs for the UL-MAP—it allocates a fixed numbers of time slots in each time frame.
2. *Real-time polling service* (*rtPS*)—this service is for real-time VBR-like flows such as MPEG video. These applications have specific bandwidth requirements as well as a deadline (maximum delay). Late packets that miss the deadline will be useless.
   - BW-Request: used only in the contention-free mode. The current queue size that represents the current bandwidth demand is included in the BW-Request.
   - Uplink Scheduler: Not defined in the current IEEE 802.16.

3. *Non-real-time polling service* (*nrtPS*)—this service is for non-real-time flows which require better than best effort service, e.g. bandwidth intensive file transfer. These applications are time-insensitive and require minimum bandwidth allocation.
   - BW-request: uses either contention-free mode or contention mode. Current queue size is included in BW-request.
   - Uplink scheduler: Not defined in current IEEE 802.16.
4. *Best effort service* (*BE*)—this service is for best effort traffic such as HTTP. There is no QoS guarantee. The applications in this service flow receive the available bandwidth after the bandwidth is allocated to the previous three service flows.
   - BW-request: uses only contention mode. Current queue size is included in BW-request.
   - Uplink scheduler: Not defined in current IEEE 802.16.

Figure 3(A) shows the existing QoS architecture of IEEE 802.16. Uplink packet scheduling (UPS) resides in the BS to control all the uplink packet transmissions. Since IEEE 802.16 MAC protocol is connection oriented, the application first establishes the connection with the BS as well as the associated service flow (UGS, rtPS, nrtPS or BE). BS will assign the connection with a unique connection ID (CID). The connection can represent either an individual application or a group of applications such as multiple tenants in an apartment building (all in one SS) sending data with the same CID. IEEE 802.16 defines the connection signaling (connection request, response) between SS and BS but it does not define the admission control process. All packets from the application layer in the SS are classified by the connection classifier based on CID and are forwarded to the appropriate queue. At the SS, the Scheduler will retrieve the packets from the queues and transmit them to the network in the appropriate time slots as defined by the UL-

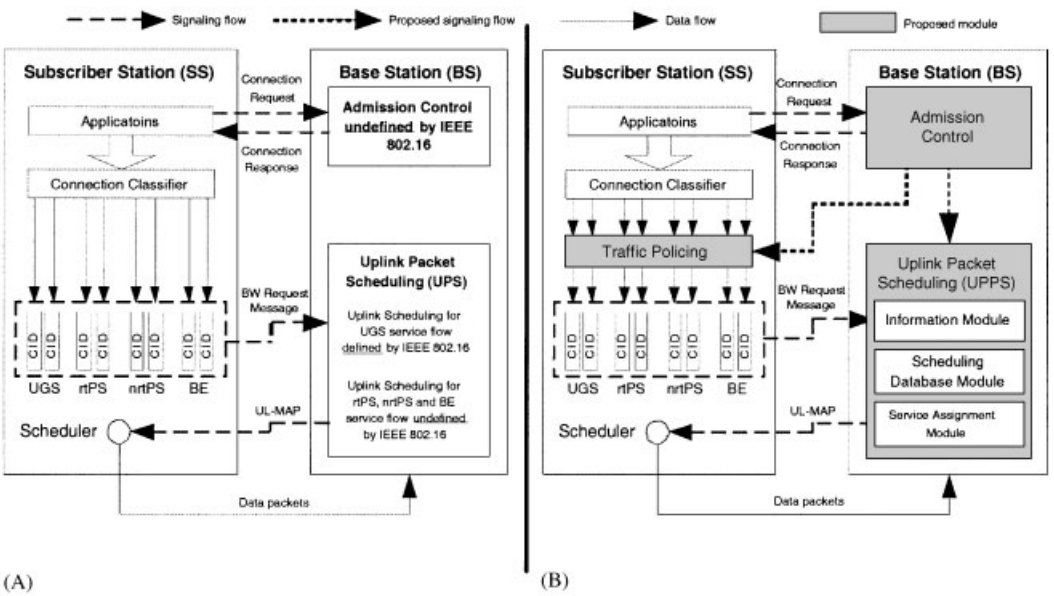

Figure 3. QoS architecture. (A) IEEE 802.16 QoS architecture and (B) proposed QoS architecture.

MAP sent by the BS. The UL-MAP is determined by the UPS module based on the BW-request messages that report the current queue size of each connection in SS.

In summary, IEEE 802.16 defines: (1) the signaling mechanism for information exchange between BS and SS such as the connection set-up, BW-request, and UL-MAP and (2) the uplink scheduling for UGS service flow. IEEE 802.16 does not define: (1) the uplink scheduling for rtPS, nrtPS, BE service flow and (2) the admission control and Traffic Policing process.

Figure 3(B) shows the proposed QoS architecture that completes the missing parts in the IEEE 802.16 QoS architecture. At the BS we add: a detailed description of the UPS module (scheduling algorithm that which supports all types of service flows), and admission control module. At the SS we add the Traffic Policing module. Here is a brief description of the connection establishment using the QoS architecture in Figure 3(B): (1) An application that originates at an SS establishes the connection with BS using connection signaling. The application includes in the connection request the traffic contract (bandwidth and delay requirement), (2) The admission control module at the BS accepts or rejects the new connection and (3) If the admission control module accepts the new connection, it will notify the UPS module at the BS and provide the token bucket parameters to the traffic policing module at the SS. After the connection is established, the following steps are taken: (1) Traffic policing enforces traffic based on the connection's traffic contract, (2) At the beginning of each time frame, the UPSs information module collects the queue size information from the BW-requests received during the previous time frame. The information module will process the queue size information and update the scheduling database module, (3) the service assignment module retrieves the information from the scheduling database module and generates the UL-MAP, (4) BS broadcasts the UL-MAP to all SSs in the downlink subframe and (5) SS's scheduler transmits packets according to the UL-MAP received from the BS.

4. TERMINOLOGY

In this section we define the terminology used in the remainder of the paper for a description of the arrival, service and frame durations:

1. f = duration of a time frame (ms) which includes uplink and downlink subframes.
2. d_i = maximum delay requirement of connection i (ms).
3. $q_i(t)$ = queue size (bits) of connection i at time t.
4. $s_i[t, t+f]$ = number of bits of connection i that transmit during time frame interval $[t, t+f]$.
5. $a_i[t, t+f]$ = number of bits of connection i that arrive during time frame interval $[t, t+f]$.
6. $Nd_i[t, t+f]$ = number of bits waiting in queue of connection i with deadline at time interval $[t, t+f]$, i.e. in order to avoid delay violation, these bits must be transmitted before the end of time interval $[t, t+f]$.
7. C_{uplink} = total capacity (bps) allocated for uplink transmission.
8. C_{downlink} = total capacity (bps) allocated for downlink transmission.
9. C_{total} = total capacity (bps) of the wireless network, $C_{\text{total}} = C_{\text{uplink}} + C_{\text{downlink}}$.
10. C_{UGS} = total capacity (bps) allocated for current UGS connections.
11. C_{rtPS} = average capacity (bps) allocated for current rtPS connections.
12. C_{nrtPS} = average capacity (bps) allocated for current nrtPS connections.

13. C_{BE} = average capacity (bps) available for current BE connections = $C_{uplink} - C_{UGS} - C_{rtPS} - C_{nrtPS}$.
14. $C_{NRT} = C_{nrtPS} + C_{BE} = C_{uplink} - C_{UGS} - C_{rtPS}$.
15. N_{uplink} = total number of bits that SSs are allowed to transmit in an uplink subframe, $N_{uplink} = fC_{uplink}$.
16. $N_{UGS,i}$ = number of bits of UGS connection i that are required to transmit in one time frame.
17. r_i = token bucket rate (average data rate) of connection i.
18. b_i = token bucket size of connection i, (*Note*: the definition and analysis of token bucket can be found in Reference [10]).

The frame duration f, needs to be chosen such that:

1. $f < \min(d_i/2)$,
2. frame size (f) must be the common divisor of the delay requirement of all connections.

For example, the network administrator might define four connections classes (A,B,C,D): (1) Class A for connection with 20 ms delay bound, (2) Class B for connections with 50 ms delay bound, (3) Class C for connections with delay bound 100 ms, (4) Class D for connections with no delay bound. In this example, the possible frame sizes are 1, 2, 5 and 10 ms.

5. PROPOSED UPLINK PACKET SCHEDULING

To support all types of service flows (UGS, rtPS, nrtPS and BE), the proposed uplink packet scheduling uses a combination of strict priority service discipline, earliest deadline first (EDF) [11] and weight fair queue (WFQ) [12]. The hierarchical structure of the bandwidth allocation is shown in Figure 4. The proposed UPS consists of three modules: information module, scheduling database module and service assignment module. The proposed UPS principles:

1. *Overall bandwidth allocation*: bandwidth allocation per flow follows strict priority, from highest to lowest: UGS, rtPS, nrtPS and BE. One disadvantage of the strict priority service

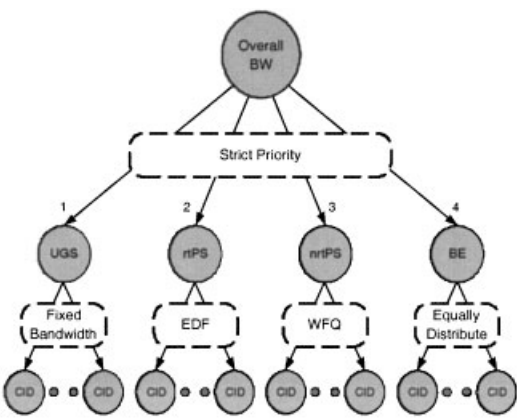

Figure 4. Hierarchical structure of bandwidth allocation.

discipline is that higher priority connections can starve the bandwidth of lower priority connections. To overcome this problem, we include the traffic policing module in each SS which forces the connection's bandwidth demand to stay within its traffic contract. This will prevent the higher priority connections from using bandwidth more than their allocation.

2. *Bandwidth allocation within UGS connections*: The UPS allocates fixed bandwidth (fixed time duration) to UGS connections based on their fixed bandwidth requirement. This policy is determined by the IEEE 802.16 standard.
3. *Bandwidth allocation within rtPS connections*: We apply earliest deadline first (EDF) service discipline to this service flow. Packets with earliest deadline will be scheduled first. The information module determines the packets' deadline.
4. *Bandwidth allocation within nrtPS connections*: We apply weight fair queue (WFQ) service discipline to this service flow. We schedule nrtPS packets based on the weight of the connection (ratio between the connection's nrtPS average data rate and total nrtPS average data rates).
5. *Bandwidth allocation within BE connections*: The remaining bandwidth is equally allocated to each BE connection.

5.1. Information module

The information module performs the following tasks: (1) retrieves the queue size information of each connection from the BW-Request messages, (2) determines the number of packets (in bits) that arrived from rtPS connection in the previous time frame using the arrival-service curve concept [13], (3) determines rtPS packets' arrival time and deadline and updates this information in the scheduling database module, (4) queuing information from nrtPS and BE BW-Requests is passed directly to scheduling database module and (5) *Note*: Since UGS requires only fixed bandwidth allocation and does not need BW-Requests, there is no need for processing UGS connections.

Information module for rtPS connections: The information module needs to find the rtPS deadline information (see Figure 5). Based on this deadline information, the UPS will know exactly when to schedule packets such that packets' delay requirements are met. We apply the arrival-service curve concept to determine the packets' arrival and deadline. The packets' deadline is their arrival time plus the connection's maximum delay requirement. Figure 6 shows

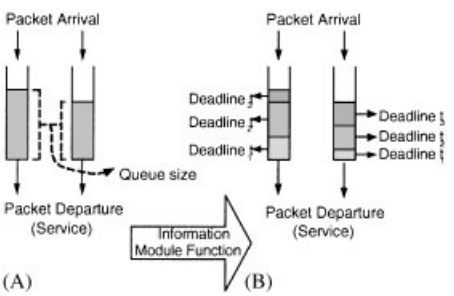

Figure 5. Concepts of information module operation for rtPS messages.

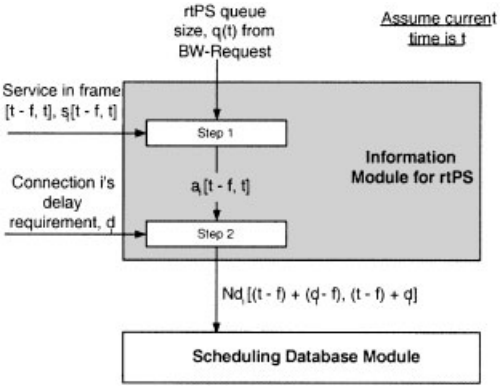

Figure 6. Information module for rtPS connections.

the block diagram of the information module for rtPS connection. It includes the following two processes (assume current time is t and the current frame is $[t, t+f]$):

1. *Determine arrival time*—the process determines the arrival time of packets that arrived during the previous frame $[t-f, t]$ using the arrival-service curve concept.
2. *Determine deadline*—the process determines the deadline of packets that arrived during the previous frame $[t-f, t]$. The deadline is given by the sum of the packet's arrival time and the packet's maximum delay requirement (as determined by the connection QoS parameters).

Information Module input for rtPS connections (at time t):

1. $q_i(t)$—current queue size of rtPS connection i as obtained from BW-Request
2. $s_i[t-f, t]$—the service that connection i actually receives in frame $[t-f, t]$. The BS can compare the service assigned by the Service Assignment Module to the actual service allocated to the SS. This will provide the channel condition information for UPS which will be used to adapt the bandwidth allocation.
3. d_i-delay requirement as obtained from connection i's traffic contract.

Step 1: Determine the number of arrivals during the previous time frame, $a_i[t-f, t]$. This is accomplished by using the arrival-service curve concept (Figure 7). At time $t = nf$, $(n = 1, 2, 3,)$

- *Input*: queue size $= q_i(nf)$, Service $= s_i[(n-1)f, nf]$
- *Ouput*: $a_i[(n-1)f, nf] = q_i(nf) + s_i[(n-1)f, nf] - q_i((n-1)f)$

Step 2: Determine the packets' deadline given the packets' arrival information during the previous frame, $a_i[t-f, t]$ as determined in Step 1. The deadline is determined from the packet's arrival time plus the packet's maximum delay requirement.

Let us first describe this step using the example shown in Figure 8. We assume as the upper bound that the packets that arrived in $[t-f, t]$, denoted by $a_i[t-f, t]$, have already waited in the queue for time f. Therefore, to avoid delay violation, these packets have to receive service

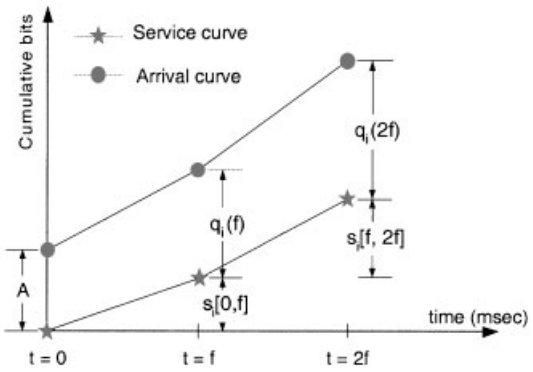

Figure 7. Information module—Step 1.

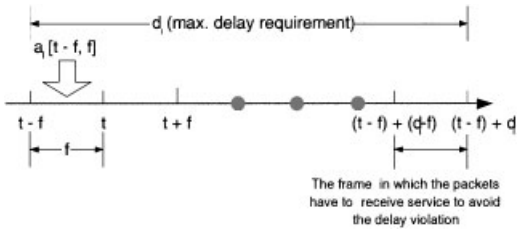

Figure 8. Arrival and deadline timing diagram for rtPS connection i.

in frame $[(t-f)+(d_i-f),(t-f)+d_i]$. Therefore, $Nd_i[(t-f)+(d_i-f),(t-f)+d_i] = a_i[t-f,t]$. In general form, Step 2 will be: At time $t = nf$, $(n = 1,2,3,\ldots,)$

- *Input*: $a_i[(n-1)f, nf]$
- *Output*: $Nd_i[(nf-f)+(d_i-f),(nf-f)+d_i] = a_i[(n-1)f, nf]$

In summary, the output of the Information Module which updates the Scheduling Database Module is given by:

1. *rtPS connections*—$Nd_i[a,b]$ the number of bits waiting in the queue of rtPS connection i with deadline in interval $[a,b]$ (computation given in Steps 1 and 2 above).
2. *nrtPS connections*—$q_j(t)$, the current queue size of nrtPS connection j.
3. *BE connections*—$q_k(t)$, the current queue size of BE connection k.

5.2. Scheduling database module

The scheduling database module serves as the information database of all connections in the network. Figures 9 and 10 show the database structure of the Scheduling Database Module. The database module (at time t) includes four types of databases based on each service flow as follows:

1. *UGS database* (*Figure 9*(*A*))—this is a per connection database. Each item i in the database contains the number of bits ($N_{\mathrm{UGS},i}$) of connection i that need to be serviced. This number is fixed and determined by the UGS connections' bandwidth requirement.

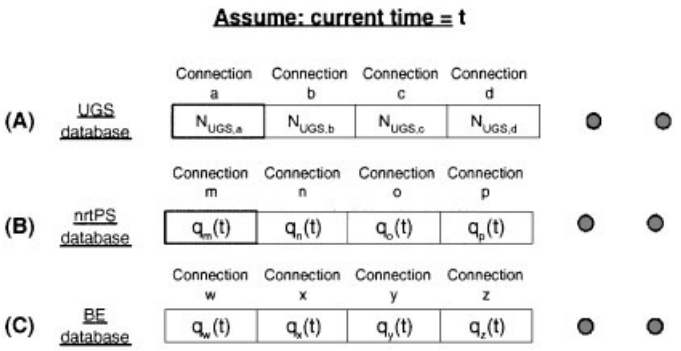

Figure 9. Database structure of UGS, nrtPS and BE in scheduling database module.

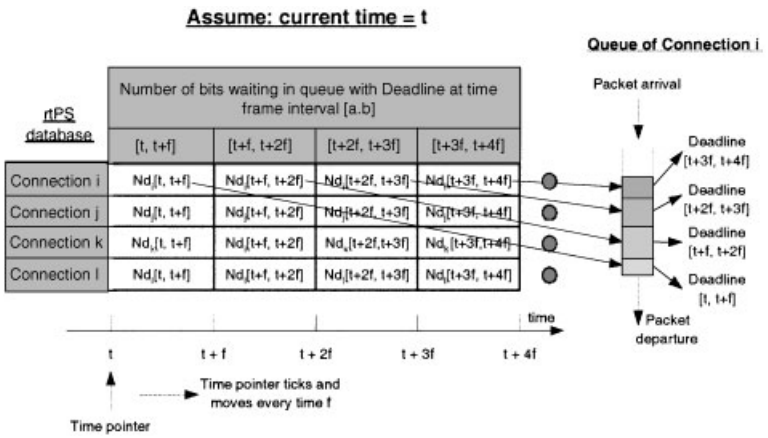

Figure 10. rtPS database structure in scheduling database module.

2. *nrtPS database* (*Figure 9*(*B*))—this is a per connection database. Each item m in the database contains $q_m(t)$ (as found by the information module), i.e. the number of bits (current queue size) of connection m.
3. *BE database* (*Figure 9*(*C*))—this is a per connection database. Each item w in the database contains $q_w(t)$ (as found by the information module), i.e. the number of bits (current queue size) of connection w.
4. *rtPS database* (*Figure 10*)—this is a two dimensional database, per connection and deadline (frame). Item $(i, [t, t+f])$ includes $Nd_i[t, t+f]$ (received from the information module) which is the number of bits to be transmitted in frame $[t, t+f]$. Figure 10 also shows the number of bits in the database table that correspond to the actual packets waiting in queue.

5.3. *Service assignment module*

The service assignment module determines the UL-MAP, using the database tables created in the scheduling database module. As depicted in Figure 4, we employ strict priority service

discipline for allocating bandwidth among service flows (i.e. UGS, rtPS, nrtPS and BE). Within each service flow we employ the following service disciplines: fixed bandwidth allocation for UGS, EDF for rtPS, WFQ for nrtPS and equal bandwidth allocation for BE. The service assignment module determines the uplink subframe allocation in terms of the number of bits per connection. The number of bits will eventually be converted to the number of time slots which are the units used in the information elements (IE) of the UL-MAP. The number of bits per time slot is determined by the physical layer of the wireless network.

1. *UGS packet scheduling* (*fixed bandwidth allocation*)—schedule $N_{\mathrm{UGS},a}, N_{\mathrm{UGS},b}, N_{\mathrm{UGS},c}$, for transmission as shown in UGS database (Figure 12(A)). This schedule is included in IEs of the UL-MAP. After scheduling the packets, update N_{uplink} ($N_{\mathrm{uplink}} \leftarrow N_{\mathrm{uplink}} - \sum N_{\mathrm{uplink,i}}$)
2. *rtPS packet scheduling* (*EDF*)—schedule the rtPS packets in the rtPS database until either there are no rtPS packets left or there is no more bandwidth left (i.e. all N_{uplink} bits have been exhausted). After scheduling the packets, update N_{uplink} ($N_{\mathrm{uplink}} \leftarrow N_{\mathrm{uplink}} - \sum$ [all bits allocated for rtPS]). In case the total number of bits in the column is greater than N_{uplink}, N_{uplink} will be distributed to each connection based on its weight ($W_i = r_i / \sum r_i,\ i =$ rtPS connections). For example, connection i will be scheduled with $W_i N_{\mathrm{uplink}}$ bits. If there are still packets left in the current time frame interval and N_{uplink} is equal to zero, these packets will miss the deadline. We can take the following two actions for the packets that missed their deadline: (1) drop the packets, or (2) reduce the priority of the packets by moving them to the BE database, i.e. these packets will be scheduled with the same priority as BE. After scheduling the packets, update the UL-MAP and the rtPS database.
3. *nrtPS packet scheduling* (*WFQ*)—schedule the packets based on connections' weights ($W_i = r_i / \sum r_i,\ i =$ nrtPS connections) and update the UL-MAP to reflect these assignments. Also update the nrtPS database and N_{uplink} ($N_{\mathrm{uplink}} \leftarrow N_{\mathrm{uplink}} - \sum$ [all bits *allocated for nrtPS*])
4. *BE packet scheduling*—schedule packets equally, i.e. provide the same amount of bandwidth to each BE connection and update the UL-MAP to reflect these assignments. Also update the BE database.

6. ADMISSION CONTROL

Admission control is the QoS mechanism that decides whether a new session (connection) can be established. This mechanism will ensure that existing sessions' QoS will not be degraded and the new session will be provided QoS support. A connection is admitted if: (1) there is enough bandwidth to accommodate the new connection, (2) the newly admitted connection will receive QoS guarantees in terms of both bandwidth and delay and (3) QoS of existing connections is maintained.

Theorem 1
Assumptions:

1. There are n rtPS connections with traffic parameters: token bucket rate (r_i) in bps, token bucket size (b_i) in bits, and delay requirement (d_i) in seconds.
2. $C_{\mathrm{rtPS}}(bps) = \sum r_i$; $C_{\mathrm{NRT}}(bps) = C_{\mathrm{uplink}} - C_{\mathrm{UGS}} - C_{\mathrm{rtPS}}$; $W_i = r_i / \sum r_i = r_i / C_{\mathrm{rtPS}}$.

3. $m_i = d_i/f$, m_i must be an integer number following the definition of the frame duration in Section 4.

The necessary condition that a rtPS connection i can be scheduled by our uplink packet scheduling with delay guarantees (the deadline will not be missed) is given by

$$b_i \leqslant [(m_i - 1)(1 + C_{\mathrm{NRT}}/C_{\mathrm{rtPS}}) - 1]r_i f \tag{1}$$

Proof

Case 1: $m_i = d_i/f = 3$.

Current time frame is $[t, t + f]$. Figure 11 shows the timing diagram. The maximum number of packets (in term of arriving bits) that arrive in the time frame $[t - 2f, t - f]$ is $b_i + r_i f$. These arriving bits must be scheduled in time frame $[t - f, t]$ and $[t, t + f]$ to avoid delay violation. In the worst case scenario, all other rtPS sessions are active (i.e. all other rtPS sessions have packets to send) and the total bandwidth requirement of all rtPS connections in each time frame $[t - f, t]$ and $[t, t + f]$ are more than $C_{\mathrm{uplink}} - C_{\mathrm{UGS}}$ which equals $C_{\mathrm{rtPS}} + C_{\mathrm{NRT}}$. So each rtPS session will be allocated the following bandwidth based on its weight: $W_i(C_{\mathrm{uplink}} - C_{\mathrm{UGS}})f$ bits in each time frame. Therefore, the necessary condition will be:

$$b_i + r_i f \leqslant 2W_i(C_{\mathrm{uplink}} - C_{\mathrm{UGS}})f$$

$$b_i \leqslant [(3 - 1)(1 + C_{\mathrm{NRT}}/C_{\mathrm{rtPS}}) - 1]r_i f$$

For the general case, m_i, the necessary condition will be

$$b_i \leqslant [(m_i - 1)(1 + C_{\mathrm{NRT}}/C_{\mathrm{rtPS}}) - 1]r_i f$$

6.1. Admission control for rtPS connections

Using Theorem 1, the admission control policy for rtPS connections is given by the following algorithm:

Input:

1. a new rtPS connection requests with parameters b_i, r_i, d_i,
2. current network parameters: $C_{\mathrm{uplink}}, C_{\mathrm{UGS}}, C_{\mathrm{rtPS}}, C_{\mathrm{nrtPS}}, f$,
3. $C_{\mathrm{NRT}} = C_{\mathrm{uplink}} - C_{\mathrm{UGS}} - C_{\mathrm{rtPS}}$, $C_{\mathrm{NRT,new}} = C_{\mathrm{NRT}} - r_i$, $C_{\mathrm{rtPS,new}} = C_{\mathrm{rtPS}} + r_i$

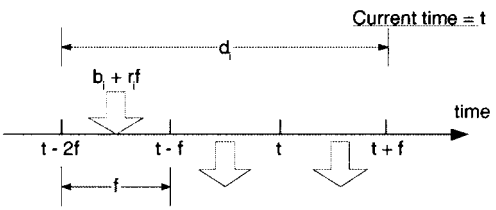

Figure 11. Case 1 timing diagram.

Admission control procedure:

1. *Check for available bandwidth*: If $r_i \leqslant C_{\text{uplink}} - C_{\text{UGS}} - C_{\text{rtPS}} - C_{\text{nrtPS}}$, there is available bandwidth for the new rtPS connection. Otherwise reject the new connection.
2. *Check for delay guarantees*: If $b_i \leqslant [(m_i - 1)(1 + C_{\text{NRT,new}}/C_{\text{rtPS,new}}) - 1]r_i f$, we can provide delay guarantees for the new rtPS connection. Otherwise reject the new connection.
3. *Check for delay violations of existing rtPS connections*: for each rtPS connection i, delay guarantee is maintained if $b_i \leqslant [(m_i - 1)(1 + C_{\text{NRT,new}}/C_{\text{rtPS,new}}) - 1]r_i f$. If for any connection this condition is not satisfied, reject the new connection.
4. *Update* C_{rtPS}: $C_{\text{rtPS}} \leftarrow C_{\text{rtPS}} + r_i$.
5. Pass token bucket parameters r_i, b_i to the traffic policing module.

6.2. Admission control for UGS connections

Input:

1. a new UGS connection request with parameter r_i,
2. current network parameters: $C_{\text{uplink}}, C_{\text{UGS}}, C_{\text{rtPS}}, C_{\text{nrtPS}}, f$,
3. $C_{\text{NRT}} = C_{\text{uplink}} - C_{\text{UGS}} - C_{\text{rtPS}}$, $C_{\text{NRT,new}} = C_{\text{NRT}} - r_i$.

Admission control procedure:

1. *Check for available bandwidth*: If $r_i \leqslant C_{\text{uplink}} - C_{\text{UGS}} - C_{\text{rtPS}} - C_{\text{nrtPS}}$, there is available bandwidth for the new UGS connection. Otherwise reject the new connection.
2. *Check for delay violations of existing rtPS connections*: for each rtPS connection i, delay guarantee is maintained if $b_i <= [(m_i - 1)(1 + C_{\text{NRT,new}}/C_{\text{rtPS}}) - 1]r_i f$. If for any connection this condition is not satisfied reject the new connection.
3. *Update* C_{UGS}: $C_{\text{UGS}} \leftarrow C_{\text{UGS}} + r_i$.
4. Pass parameter r_i to the traffic policing module.

6.3. Admission control for nrtPS connections

Input:

1. a new nrtPS connection request with parameter r_i, b_i,
2. current network parameters: $C_{\text{uplink}}, C_{\text{UGS}}, C_{\text{rtPS}}, C_{\text{nrtPS}}, f$.

Admission control procedure:

1. *Check for available bandwidth*: If $r_i <= C_{\text{uplink}} - C_{\text{UGS}} - C_{\text{rtPS}} - C_{\text{nrtPS}}$, there is available bandwidth for the new nrtPS connection. Otherwise reject the new connection.
2. *Update* C_{nrtPS}: $C_{\text{nrtPS}} \leftarrow C_{\text{nrtPS}} + r_i$
3. Pass token bucket parameters r_i, b_i to the traffic policing module.

Note: No admission control process is required for BE connections. BE connections are always admitted with no QoS support.

Traffic policing: Each SS will have a traffic policing module which monitors violations of QoS contracts by admitted connections, using the token bucket mechanism. The token bucket parameters for each connection are received from the admission control module.

7. SIMULATION RESULTS

We have developed a simulation model in C++ that demonstrates that our proposed uplink packet scheduling (UPS) provides QoS support to real time applications. We also investigate the factors that affect the performance of the proposed UPS. Packet arrivals occur at the beginning of each frame. The packet arrival process for each connection follows the token bucket envelope with parameters: token bucket rate (r_i), token bucket size (b_i) and maximum burst size. Each connection has specific QoS parameters in terms of (1) average bandwidth requirement which is equal to the token bucket rate, and (2) maximum delay requirement. Simulation output: (1) the arrival curve which depicts the arrival pattern of the input traffic, (2) the service curve which shows the service pattern provided by UPS, and (3) the percentage of packets that miss their deadline. The goal of this experiment is to show that the proposed UPS can provide QoS support in terms of bandwidth and delay for rtPS traffic. We perform the experiment that shows the QoS support provided by the uplink packet scheduling. We assume as the follows: (1) There are only two types of traffic (rtPS and BE). (2) All traffic is already admitted to the network. (3) BE traffic requires uplink bandwidth at all times and (4) $C_{\text{total}} = 10$ Mbps, $C_{\text{uplink}} = 5$ Mbps, $C_{\text{downlink}} = 5$ Mbps. (5) Frame size (f) = 10 ms and (6) Input traffic:

- Three rtPS sessions with average total bandwidth (C_{rtPS}) of 3 Mbps.
- rtPS traffic characteristics are shown in Table I. We specify token bucket rate (r_i), token bucket size (b_i), maximum burst size and maximum delay requirement for each session. The corresponding peak rate and burstiness (peak rate/average rate) are calculated from r_i, b_i, and maximum burst size.

Figure 12 shows the bandwidth allocation for rtPS and BE connections. Since our UPS is work conserving and there is always BE traffic available, rtPS and BE bandwidth allocation complement each other, i.e. in each frame the total rtPS and BE bandwidth equal to 5 Mbps. In this experiment there are no packets that miss their deadline.

Figure 13 shows the arrival and service curves of all three rtPS connections. The graphs clearly show that the service curve adapts and follows the arrival curve. Our UPS dynamically allocates bandwidth based on the bandwidth demand of each session. The delay of each session is also guaranteed since there are no packets that miss their deadline. We observe that the horizontal distance between these two curves (arrival curve and service curve) of each session is bound by the maximum delay of each session.

Table I. Input traffic.

Session	Token bucket rate, r_i (kbps)	Token bucket size, b_i (bits)	Maximum burst size (ms)	Peak rate, p_i (kbps)	Burstiness (p_i/r_i)	Max. delay req. (ms)
1	500	10000	10	1500	3	20
2	1000	20000	10	3000	3	40
3	1500	30000	10	4500	3	60

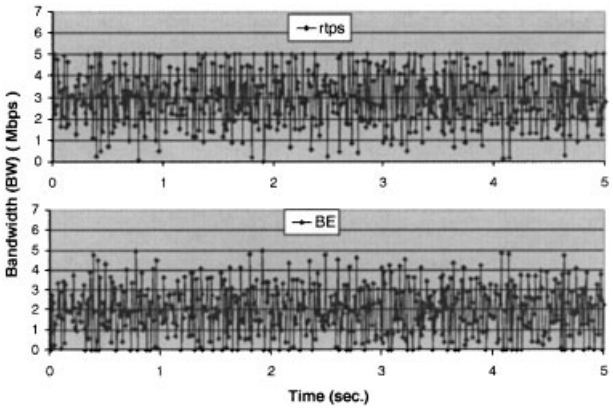

Figure 12. Bandwidth allocation for rtPS and BE connections ($C_{\text{uplink}} = 5$ Mbps, $C_{\text{rtPS}} = 3$ Mbps, $C_{\text{BE}} = 2$ Mbps).

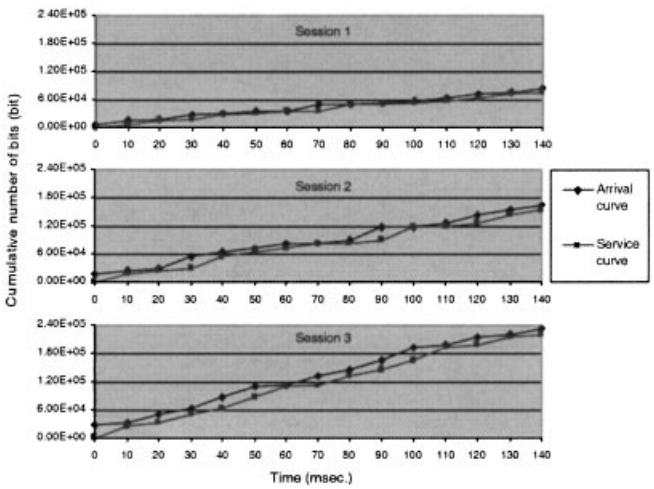

Figure 13. Arrival and service curves for Sessions 1–3.

8. CONCLUSION

In this paper we have presented a scheduling algorithm and admission control policy for IEEE 802.16 standard. The proposed solution is practical and compatible to the existing IEEE 802.16 standard. The simulation studies show that the proposed solution provides QoS support in terms of bandwidth and delay bounds for all types of traffic classes as defined by the standard.

ACKNOWLEDGEMENTS

This work was supported in by NSF-CISE 0087945, NSF-CISE 0080119, NSF-CISE 9812589, NSF-ANI 0230812 and DARPA F33615-02-C-4031.

REFERENCES

1. IEEE 802.16 Standard—Local and Metropolitan Area Networks—Part 16. *IEEE Draft P802.16/D3-2001*
2. IEEE 802.16 Working Group on Broadband Wireless Access. http://wirelessman.org.
3. Bhagwat P, Krisna A, Tripathi S. Enhancing throughput over wireless LAN's using channel state dependent packet scheduling. *IEEE INFOCOM 96*; March 1996; 1133–1140.
4. Fragouli C, Sivaraman V, Srivastava M. Controlled multimedia wireless link sharing via enhanced class-based queuing with channel-state dependent packet scheduling. *IEEE INFOCOM 98*; March 1998; 572–580.
5. Lu S, Bharghvan V. Fair scheduling in wireless packet networks. *IEEE/ACM Transactions on Networking* 1999; **7**(4):473–489.
6. Eugene TS, Stoica I, Zhang H. Packet fair queuing algorithms for wireless networks with location-dependent errors. *IEEE INFOCOM 98*; March 1998; 1103–1111.
7. Ramanathan P, Agrawal P. Adapting packet fair queuing algorithms to wireless networks. *ACM/IEEE MOBICOM 98*; Dallas, TX; 1998; 1–9.
8. Gome J, Campbell AT, Morikawa H. The Havana framework for supporting application and channel dependant QoS in wireless networks. *Proceedings of ICNP'99*; November 1999; 235–244.
9. Cao Y, Li VOK. Scheduling algorithms in broad-band wireless networks. *Proceeding of the IEEE* 2001; **89**(1):76–86.
10. Parekh AK, Galager RG. A generalized processor sharing approach to flow control in integrated services networks: the single-node case. *IEEE/ACM Transactions on Networking* 1993; **1**(3):344–357.
11. Georgiadis L, Guerin R, Parekh A. Optimal Multiplexing on a Single Link: Delay and Buffer Requirements. *Proceedings of IEEE INFOCOM 94*; vol. 2, 1994; 524–532.
12. Demers A, Keshav S, Shenker S. Analysis and Simulation of a Fair Queuing Algorithm. *SIGCOMM CCR 19* 1989; 4.
13. Cruz RL. A Calculus for network delay, Part I: Network elements in isolation. *IEEE Transaction of Information Theory* 1991; **37**(1):114–121.

