www.ietdl.org

Published in IET Communications
Received on 23rd August 2010
Revised on 17th November 2010
doi: 10.1049/iet-com.2010.0770

ISSN 1751-8628

Low-overhead uplink scheduling through load prediction for WiMAX real-time services

W. Nie[1] *H. Wang*[1] *N. Xiong*[2]

[1]*School of Automation Engineering, University of Electronic Science and Technology of China, Chengdu 610054, People's Republic of China*
[2]*Department of Computer Science, Georgia State University, Atlanta, GA 30303, USA*
E-mail: weiuestc@gmail.com

Abstract: As WiMAX achieves increasing deployment, the large overhead in its uplink scheduling when providing real-time services has become a major challenge. In this study, the authors present effective, low-overhead scheduling algorithms for WiMAX uplink scheduling. The authors adaptively predict users' load and select a small set of active users to be served. This addresses the major source of overhead in WiMAX uplink scheduling: the Markovian arrival process information elements (IEs) and media access control layer (MAC) service data unit (SDUs) subheader overheads grow with the number of active users. The authors introduce additional novel techniques, including piggybacking, to reduce MAC overhead. The authors implement their algorithms and conduct extensive evaluations. The results show that their algorithms not only provide quality-of-service guarantees, but also substantially reduce the scheduling overhead compared with existing schemes.

1 Introduction

The industry claims well over 300 trials worldwide. These include WiMAX networks by Sprint and Clearwire in the USA and WiBro (wireless broadband access service) in South Korea. Broadband wireless access networks compete with existing wire networks [1]. Over the past few years, there has been a rapid growth of new services such as streaming audio and video, and multimedia for residential and business customers. Multimedia communication entails diverse quality-of-service (QoS) requirements for different applications. Multimedia services may cause a sharp increase in demand for bandwidth, which places greater strain on the infrastructure of internet service provider (ISPs). AT&T, a leader in telecommunication services, phases out unlimited iPhone data plans for those who stream video on their iPhones all day. Thus, it is very important for us to design a scheduling algorithm that maximises bandwidth utilisation in WiMAX real-time communication networks.

Real-time polling services (rtPS) are designed to support real-time services that generate variable size data packets on a periodic basis, such as moving pictures experts group (MPEG) video and video phone, but which are also sensitive to delay. An important principle of WiMAX is that it is connection oriented. This means that an subscriber station (SS) must register to the base station (BS) before it can send or receive data. In order to ensure constant connection, all the SSs should keep in touch with the BS, which requires frequent indication of scheduling information. A Markovian arrival process (MAP) message is transmitted at the beginning of each frame to provide scheduling information to receivers. However, the system has to keep in constant contact with all mobile users, which requires a lot of control signalling between BS and SSs. In IEEE 802.16 standard [2], the main MAC overhead comes from the uplink map (UL-MAP) and downlink map (DL-MAP) overhead. UL-MAP information element (IE) describes the resource allocation of data bursts belonging to the same basic connection ID (CID), that is to say, from the same SS. Hence, the size of the UL-MAP is the function of the number of SSs concerned by the UL-MAP. Since MAP messages should be delivered to all SSs even in a bad channel condition, they have to be transmitted with the most robust MCS level and also with some repetitions in order to achieve the maximum robustness. Furthermore, the MAP messages are transmitted in every frame with the lowest data rate, and therefore they consume a large amount of radio resource. The analysis above shows that if the BS scheduler allocates slots to many SSs in each frame, the size of UL-MAP may become quite big. At the same time, the slot number for every connection or SS may be relatively small. In order to fit data in the assigned burst, it is often necessary to have an SDU fragmentation. If packing is turned on for a connection, multiple MAC SDUs can be packed into a single MAC protocol data unit (PDU) packing subheader (3 bytes) should be inserted before each SDU, Therefore the subheader overhead will also occupy a large percentage of the radio resource in a frame.

As a result, the WiMAX system suffers a problem of huge MAC overhead, which causes the reduction of the radio resources available for data transmission and leads to the degradation of the system throughput; for example, the simulation results form shows that if there is no MAP optimisation, the MAP messages alone occupy up to 20–60% of system resources [3].

In this paper we propose a scheduling scheme to reduce the overhead and maximise bandwidth utilisation. We schedule the deadline packets by the earliest deadline first (EDF) algorithm to avoid the delay violation. For the remaining bandwidth, because the number of MAP IEs increases with the number of users, we use an adaptive uplink Bandwidth Scheduling Scheme to reduce the number of scheduled users. That is to say, we try to allocate more bandwidth to a user in one frame and so we predict the rtPS packets that will be queued between the time it makes the request for bandwidth and the time the BS responds. This estimate is combined with the number of rtPS packets that are waiting to be transmitted to find the total bandwidth necessary, and to estimate time-slot requests for the SS.

Moreover, in order to save the system resources, control messages are concatenated to data packets (piggybacking MAP IEs) [4] instead of the BS providing periodic unicast request opportunities for contention-free bandwidth request. The remainder of this paper is organised as follows: In Section 2, we give some background information in the area of IEEE 802.16. In Section 3, we describe the EDF algorithm in detail; we also include the batch Markovian arrival process (BMAP) model and Newton's interpolation polynomial function, which predict the bandwidth requirement of rtPS scheduler, and a smoothing parameter to modify the requested bandwidth based on the difference between previous predictions and actual requirements. In Section 4, we present the preliminary simulation results and compare them with weighted round robin (WRR) and weighted fair queuing (WFQ) algorithm. Finally, Section 5 concludes the discussion and presents our future work.

2 Background

2.1 Overview on WiMAX system

IEEE 802.16 (also known as Worldwide Interoperability for Microware Access System) basic system consists of one BS and several SSs. BS acts as the central entity, transferring data to and receiving data from the SSs in a point to multipoint mode. Transmissions take place through two independent channels: Downlink Channel (from BS to SS) and Uplink Channel (from SS to BS). Uplink Channel is shared between all SSs while Downlink Channel is used only by BS. A time division duplex (TDD) frame structure is adopted, where the BS and SS each transmit on the same frequency separated in time.

The main purpose of uplink scheduler is bandwidth allocation. IEEE 802.16 standard defines the frame Structure of WiMAX as shown in Fig. 1 [2]. The vertical axis in this figure is frequency or subcarriers and the horizontal axis is time. The time is divided into frames (typically 5 ms duration) [5].Each frame consists of DL and UL subframes. A preamble is used for time synchronisation. The DL-MAP and UL-MAP define the burst-start time, burst-end time, modulation types and forward error control for each SS. The SS allocation is in terms of bursts.

The IEEE 802.16 standard specifies the use of information elements in the UL-MAP message.

A TDD frame structure is adopted, where the BS and SS each transmit on the same frequency separated in time. The IEEE 802.16 standard specifies the use of information elements in the UL-MAP message [6]. For our purpose, the most important components of an IE are the CID and the number of symbols allocated to the SS. The list of IEs constitutes the UL-MAP message that is broadcast to all the SSs. In the pseudo-code that follows, the function that creates the UL-MAP message will be referred to as CreateIE(). For our purpose, the most important components of an IE are the CID and the number of symbols allocated to the SS. The list of IEs constitutes the UL-MAP message that is broadcast to all the SSs.

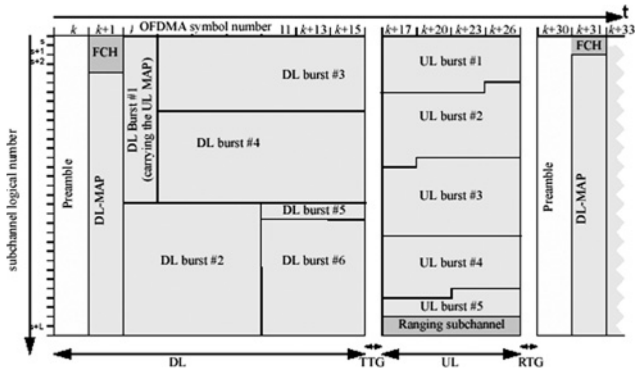

Fig. 1 *WiMAX frame structure*

The rtPS is designed to support real-time uplink service flows that transport variable size data packets on a periodic basis, such as MPEG video and video phone.

Maximum Latency specifies the maximum latency between the reception of a packet by the BS or SS on its network interface and the forwarding of the packet to its radio frequency interface.

The BS shall be able to satisfy bandwidth requests for a service flow up to its minimum reserved traffic rate. The value of this parameter is calculated from the byte following the MAC header check sequence (HCS) to the end of the MAC PDU payload. The rate is usually expressed in bits per second and specifies the minimum amount of data to be transported on behalf of the SS when averaged over time.

Maximum sustained traffic rate specifies the peak information rate of the SS. The value, expressed in bits per second, does not limit the instantaneous rate of the SS but it is used to police the SS to ensure that it conforms to the value specified, on average, over time.

The mandatory QoS parameters are minimum reserved traffic rate, maximum sustained traffic rate and maximum latency. The QoS requirements may be either per connection based grant per connection (GPC) or per SS based grant per subscriber (GPSS) [6]. In this paper, we limit the discussion to GPC – the responsibility of a connection to collect its service requirements.

Requests refer to the mechanism that SSs use to indicate to the BS that they need uplink bandwidth allocation. A Request may come as a PiggyBack Request. Because the uplink burst profile can change dynamically, all requests for bandwidth shall be made in terms of the number of bytes needed to carry the MAC header and payload. An SS shall not request bandwidth for a connection if it has no PDU to transmit on that connection. When the BS receives an incremental Bandwidth Request, it shall add the quantity of bandwidth requested to its current perception of the bandwidth needs of the connection.

2.2 Related work

The packet schedulers operating at the MAC layer are very important for QoS delivery. The IEEE 802.16 standard does not specify the scheduling algorithm to be used. It is up to vendors to implement an algorithm based on their network traffic. Vendors and operators have the choice among many existing scheduling techniques or they can develop their own scheduling algorithms. However, several papers have been published analysing the best bandwidth request strategy and proposing scheduling for WiMAX network such as RR scheduling [7, 8], fair scheduling [9–11], MaxMin fair scheduling [12], channel state dependent round robin [13], distributed fair scheduling [14], energy-efficient scheduling [15], packet-by-packet generalised processor sharing scheme [16] and multi-rate power-controlled collision-free scheduling [17]. However, none of these algorithms can be directly used for WiMAX owing to the specific features of the technology. Let us take, for example, two of the better-known algorithms: PGPS and WF2Q. The queuing disciplines used by PGPS and WF2Q are based on a timestamp mechanism to determine the packet service sequence. The timestamp mechanism for all packets, however, entails implementation complexity. RR scheduling algorithms are the simplest scheduling algorithms designed especially for a time-sharing system. RR algorithm can be considered the very first simple RR fairly assigns the allocation one by one to all connections. With packet-based allocation, stations with larger packets have an unfair advantage. Moreover, RR may be non-work conserving in the sense that the allocation is still made for connections that may have nothing to transmit. Since RR cannot assure QoS for different service classes.

We will only discuss those that have been proposed and evaluated in WiMAX: WRR and WFQ algorithms.

WRR [18, 19]: WRR has been applied for WiMAX scheduling [20]. The weights can be used to adjust for the throughput and delay requirements. WRR is an extension of the RR algorithm. It is a work-conserving algorithm in that it will continue allocating bandwidth to the SSs as long as they have backlogged packets. The WRR scheduling algorithm originally proposed for asynchronous transfer mode (ATM) traffic in [21] has been implemented in [22] to evaluate the IEEE 802.16 MAC layer on how effectively it supports QoS requirements of the multi-class traffic.

WFQ [23]: it is a packet-based approximation of the generalised processor sharing (GPS) algorithm. GPS is an idealised algorithm which assumes that a packet can be divided into bits and each bit can be scheduled separately. The WFQ algorithm results in superior performance compared to the WRR algorithm in the presence of variable size packets. The finish time of a packet is essentially the time the packet would have finished service under the GPS algorithm. The disadvantage of the WFQ algorithm is that it will service packets even if they would not have started service under the GPS algorithm. This is because the WFQ algorithm does not consider the start time of a packet.

3 Low-overhead uplink scheduling scheme

We adopt the EDF [24] scheduling algorithm for those uplink rtPS packets that have to receive service to avoid the delay violation. In addition, we design an adaptive bandwidth scheduling scheme for the remanding bandwidth. Bandwidth utilisation is maximised by piggybacking MAP IEs to reduce MAC overhead.

We assume that a MAP message is transmitted with fixed modulation and coding scheme level. It is composed of multiple MAP IEs and each MPA IE describes the resource region of a burst, which is a collection of data packets whose modulation and coding scheme levels are the same. Consequently, the number of MAP IEs in a MAP message is equal to the number of the number of bursts transmitted in the frame. We assume that each burst is composed of data packets which will be delivered to a user so that the number of MAP IEs is equal to the number of scheduled users in the frame.

3.1 EDF scheduling scheme

EDF is one of the most widely used scheduling algorithms for real-time applications as it selects SSs based on their delay requirements. Ruangchaijatupon evaluated the performance of the EDF algorithm. EDF is a work-conserving algorithm originally proposed for real-time applications in wide-area networks. The algorithm assigns deadline to each packet and allocates bandwidth to the SS that has the packet with the earliest deadline. Deadlines can be assigned to packets of an SS based on the SS's maximum delay requirement. The EDF algorithm is suitable for SSs belonging to rtPS scheduling services, since SSs in this class have stringent delay requirements.

The UL scheduler comprises three modules: information module, database module and service assignment, which are shown in Fig. 2.

Following are variables/functions:

f: frame size (ms), uplink and downlink subframe contains;
d_i: the maximum delay of connection i (ms);
$q_i(t)$: the queue length of connection i at time t(bit);
$s_i\,[t, t+f]$: the number of bits required to be transmitted for connection i in the time interval $[t, t+f]$;
$a_i[t, t+f]$: the number of bits arriving for connection i in the time interval $[t, t+f]$;
$Nd_i[t, t+f]$: the number of bits waiting in the queue for connection i, which will expire in the time interval $[t, t+f]$.

3.1.1 Information module: This module extracts the queue size information, for example, the number of waiting packets and the size of each packet of each connection from the BW-request messages. The process decides time bound that is given by the sum of the packet's arrival time and the packet's maximum delay requirement.

rtPS connection input information module is

$$q_i(t),\ s_i[t-f, t],\ d_i$$

Firstly, it shows in the Fig. 3, using the services curve and arrival curve, we determine the number of bits which arrived in the previous frame $a_i[t-f, t]$. According the figure, input is $t = nf$, $(n = 1, 2, 3\ldots)$, queue size $= qi(nf)$, service $= [(n-1)f, nf]$;

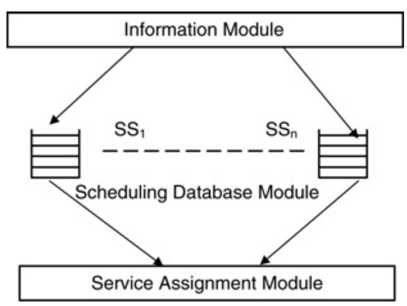

Fig. 2 *Uplink scheduler architecture*

doi: 10.1049/iet-com.2010.0770

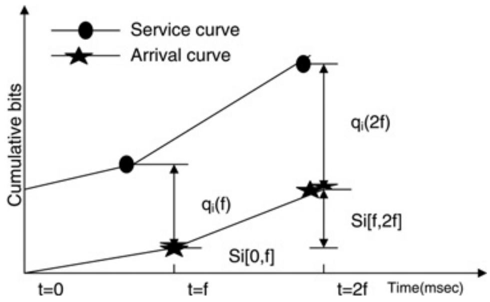

Fig. 3 *Input information module*

Output is

$$a_i[(n-1)f, nf] = q_i(nf) + s_i[(n-1)f, nf] - q_i((n-1)f) \quad (1)$$

Secondly, according to $a_i[t-f, t]$, we determine the expiration time which equals arrival time plus the maximum packet delay requirement. As we show in Fig. 4.

Suppose $[t-f, t]$ is the upper bound of arriving packets in the time $[t-f, t]$, packets in the queue waiting time for f.

In order to avoid violating the delay requirements, these packets must be serviced in the frame.

$$[(t-f)+(d_i-f), (t-f)+d_i] \quad (2)$$

So, we obtain that

$$Nd_i[(t-f)+(d_i-f), (t-f)+d_i] = a_i[t-f, t] \quad (3)$$

Thus, the second step could be summed up as when $t = nf$, $(n = 1, 2, 3\ldots)$, our input is $a_i[(n-1)f, nf]$; the output is

$$Nd_i[(t-f)+(d_i-f), (t-f)+d_i] = a_i[(n-1)f, nf] \quad (4)$$

In Fig. 4, f_{last} indicates the frame in which the packets have to receive service to avoid the delay violation.

3.1.2 Scheduling database module: For all the SS in the system, the scheduling database module serves as a database of information for all connections; we show the database structure of scheduling database module as shows in Table 1 rtPS database. This is a two-dimensional (2D) database, including the connection and the expiration time (frame) component. The item $(i, [t, t+f])$ contains the number of bits Nd_i $[t, t+f]$ in the frame $[t, t+f]$ to be sent, (received from the information module).

3.1.3 Service assignment module: The uplink subframe allocation in terms of the number of bits per SS is determined by the service assignment module. The physical layer of the wireless network determines the number of bits per time slot. According to physical modulation, the system will finally calculate the number of time slots and the units used in the IE of the UL-MAP generation.

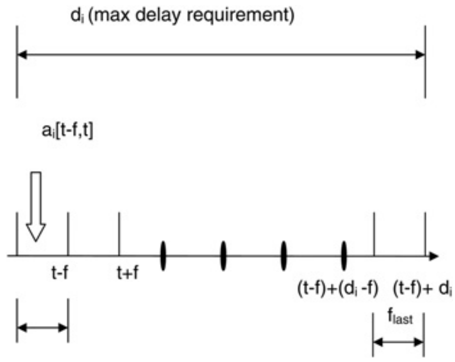

Fig. 4 *rtPS expire time and arrival time of connection i*

3.1.4 Specific implementation steps: At the beginning of each frame, the system checks all rtPS available bandwidth BW_{rtPS} and $\text{Buffer}_{i\_\text{deadline}}$ that is the bandwidth required by the deadline frame in a current time. SS piggybacking requires bandwidth BW_{i_require} $(i \in \text{rtPS})$ if $BW_{\text{rtPs}} < \sum_{i\in\text{rtPS}} \text{Buffer}_{i\_\text{deadline}}$, we allocate the bandwidth by the proportional method as shown below then the connection i is assigned bandwidth. The remaining packets that are not scheduled will be discarded.

$$BW_{i\_\text{allocate}} = BW_{\text{rtPS}} \bullet \frac{\text{Buffer}_{i\_\text{deadline}}}{\Sigma_{i\in\text{rtPS}}\text{Buffer}_{i\_\text{deadline}}}$$

if

$$\sum_{i\in\text{rtPS}} \text{Buffer}_{i\_\text{deadline}} \le BW_{\text{rtPS}} < BW_{i\_\text{require}}$$

we guarantee the bandwidth of deadline packets, then we try to allocate the more bandwidth to active SS requirement. So the connection i is assigned bandwidth

$$BW_{i\_\text{allocate}} = Br_{i\_\text{deadline}} + BW_{i\_\text{require}}$$

if

$$BW_{i\_\text{require}} \le BW_{\text{rtPS}}$$

The bandwidth requirement for each rtPS connection will be scheduled

$$BW_{i\_\text{allocate}} = BW_{i\_\text{require}}$$

In the pseudo-code that follows in Fig. 5, the function that

Table 1 rtPS database structure of scheduling database module

rtPS database	Number of bits waiting in queue with deadline at time frame interval [*a*, *b*]			
	$[t, t+f]$	$[t+f, t+2f]$	$[t+2f, t+3f]$	$[t+3f, t+4f]$
connection *i*	$N_{di}[t, t+f]$	$N_{di}[t+f, t+2f]$	$N_{di}[t+2f, t+3f]$	$N_{di}[t+3f, t+4f]$
connection *j*	$N_{dj}[t, t+f]$	$N_{dj}[t+f, t+2f]$	$N_{dj}[t+2f, t+3f]$	$N_{dj}[t+3f, t+4f]$
connection *k*	$N_{dk}[t, t+f]$	$N_{dk}[t+f, t+2f]$	$N_{dk}[t+2f, t+3f]$	$N_{dk}[t+3f, t+4f]$
connection *l*	$N_{dl}[t, t+f]$	$N_{dl}[t+f, t+2f]$	$N_{dl}[t+2f, t+3f]$	$N_{dl}[t+3f, t+4f]$

```
1. //Drop packets of rtPS SSs that have missed their deadline
2. Drop(rtPS)
3. Assign deadline upon arrival of a packet.
4. //Assign bandwidth to the SS with the packet with earliest deadline, execute
 the EDF algorithm for rtPS SSs
5. While(C>0)
6. Bi= Bi+ amount(min_deadline(P), ξi)
7. CreateIE()
8. Set SS flag to 1
9. C = C - amount(min_deadline(P), ξi)
10. end while

11.//Carry-over any bandwidth remaining from execution of EDF, we try to
allocate more bandwidth to a SS in one frame
12. while(C>0 and Request Bandwidth>0 )
13. if get SS flag equals 1 then
14. Bi= Bi+ dequeue(P)
15. C = C - Request Bandwidth (P, ξ)
16. end if
17. end while

18.//if there still bandwidth remaining , we execute adaptive algorithm allocate
bandwidth for other SSs
19. While(C>0 and Request Bandwidth>0
20. Bi= Bi+ deque(P)
21. CreateIE()
22. C = C - Request Bandwidth (P, ξ)
23. end while
```

Fig. 5 *Pseudo-code of low-overhead algorithm*

creates the UL-MAP message will be referred to as CreateIE(). $\mathrm{Min}_{\mathrm{deadline}}(\mathrm{P})$ refers to the packet with the earliest deadline. The algorithm below is executed upon arrival of every packet.

Below are the variables/functions M

C: the uplink channel capacity;
Φ: set of all SSs belonging to the rtPS class;
Bi: bandwidth allocated to connection i;
Dequeue i: remove packet P from the queue of connection i;
amount(P, ξi): retrieve the packets P from the connection i. Convert the packets to number of symbols according to the signal-to-interference noise ratio [SINR(ξi)] of connection i.
CreateIE(amount(P, ξi)): create an IE for connection i with amount(P, ξi) number of symbols. Then, IE is added to the UL-MAP message.
Drop(rtPS): drop packets from the queues for all connections.

3.2 Adaptive scheduling scheme

We study the scheduling algorithm and propose an adaptive bandwidth scheduling scheme for rtPS at each SS to maximise bandwidth utilisation. The BMAP model is used as the arrival process model since it considers different sizes of packets and batch arrivals and it is analytically tractable. We use a BMAP and Newton's interpolation polynomial function to predict the bandwidth requirement of rtPS packets that will be queued between the time the SS makes the request for bandwidth and the time the BS responds. This estimate is combined with the number of rtPS packets that are waiting to be transmitted to find a total bandwidth necessary and estimate time-slot requests for the SS.

We have developed the analytical models in the following subsections for proposed adaptive algorithm to estimate, cumulative data flow at SS, data-flow equations, time-slot request function, time allocation for transmission time for an SS, evaluation of data arrival rate at SS and estimation of adaptive time-slots. The analytical model that we have developed is generic in nature (applicable for all data arrival patterns).

3.2.1 Modelling the arrival process: We evaluate our algorithm, with the aim of showing that it outperforms the classical one, by simulation and by considering BMAP [25, 26] traffic for each considered class. In this paper, we identify the BMAP as the arrival process. In this model, the key idea is to customise the BMAP such that the different lengths of IP packet. Therefore the BMAP model is used as the arrival process model since it considers different lengths of packets and batch arrivals. As an example, if, in the WiMAX network, there are three different PDU types, the BMAP will correspondingly consider the different packet lengths. In this way, BMAP proves to be a very accurate model for characterising the aggregated traffic, especially with regard to the burstness and the self-similarity properties of the IP traffic.

The key idea of this aggregated traffic model is to customise the BMAP such that different lengths of IP packets are represented by reward values, that is, batch sizes of arrivals, of the BMAP.

In our model, we take the arrival process for each class of traffic to be a BMAP. Lucantoni was one of the first to mention this type of process. It belongs to the same class as many well-known input process such as Markovian arrival process, Markovian-modulated Poisson process, PHtype renewal process, interrupted Poisson process etc.

Given a continuous time Markov chain (CTMC) with $(0, 1, \ldots, N)$ states where 0 is an absorbing state and the others $(1, 2, \ldots, N)$ are transient, the BMAP can be constructed in the following way: allow the CTMC to evolve until an absorption in state 0 occurs. At this point, the chain is restarted in one of the transient state.

The phase-type (PH) distribution is a special case of the BMAP. PH distribution is usually used to analyse absorbing Markov chains that contain one or more absorbing states (absorbing Markov chains). We can transit from each non-absorbing state to the absorbing state in one or more time steps. The distribution of time from transient state i to absorbing state $(N+1)$ is called PH distribution. Below, we give the infinitesimal generator of the PH distribution

$$D = \begin{Bmatrix} \boldsymbol{B} & \boldsymbol{B}_0 \\ 0 & 0 \end{Bmatrix} \tag{5}$$

The $N^{*}N$ matrix $\boldsymbol{B}$ gives the transitions among the transient states. The $N^{*}1$ matrix $\boldsymbol{B}_0$ gives the transitions from the transient states to the absorbing state $(N+1)$.

In BMAP, we encounter only finite and absorbing Markov chains. Given a 2D Markov process $\{V(t), U(t)\}$ with state space $\{(i, j);\ \ i \geq 0,\ \ 1 \leq j \leq N\}$, $V(t)$ gives the total arrivals in the interval $(0, t)$, and $U(t)$ gives the underlying Markov chain. Below is the structure of an infinitesimal generator D

$$D = \begin{Bmatrix} D_0 & D_1 & D_2 & D_3 & \cdots \\ 0 & D_0 & D_1 & D_2 & \cdots \\ 0 & 0 & D_0 & D_1 & \cdots \\ 0 & 0 & 0 & D_0 & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \end{Bmatrix} \tag{6}$$

doi: 10.1049/iet-com.2010.0770

In the above, $\boldsymbol{D}_0$ is the rate matrix of transitions without arrivals and $\boldsymbol{D}_m$ is the rate matrix of transitions with arrivals of batch size m with $1 \le m \le M$. The infinitesimal generator D of the CTMC associated with the BMAP is

$$D = \boldsymbol{D}_0 + \sum_{m=1}^{M} \boldsymbol{D}_m \tag{7}$$

The BMAP is taken to be in a transient state i, with an exponentially distributed time of rate λ_i. The sojourn time passes with probability $P_{mi,j}$. In this event, the BMAP goes into the absorbing state 0 and an arrival of batch size m takes place. The process then restarts in state j. If N is the state number of the CTMC and m is the batch size, $1 \le m \le M$, then the selection of state j with $1 \le j \le N$ is determined solely by $P_{mi,j}$. Therefore with probability $P_{0i,j}$, the BMAP transfers to another state j with $j \ne i$, without arrivals. The submatrix $D_{0i,j} = \lambda_i P_{0i,j}$ for $i \ne j$ and $D_{0i,j} = -\lambda_i$ and $D_{mi,j} = \lambda_i P_{mi,j}$. The probability density functions that tell us the probabilities for the state changes from state i to state j with batch size m at time t are given in the matrix $f_m(t)$

$$f_m(t) = eD(0)tD(m) \tag{8}$$

It can be shown that the cumulative distribution function of the inter-arrival time for the batch size m is

$$F(t) = \pi(1 - eD(0)t)(-D(0))^{-1}D(m) \tag{9}$$

Since D is the infinitesimal generator of the BMAP, we know that $\sum_{k-0}^{M} D_K e = 0$, where e is the $m \times 1$ column vector of size m with all entries being 1. Furthermore, as $D = \sum_{k-0}^{M} D_K\, e$ is also the infinitesimal generator for the associated Markov chain, we know that $\pi D = 0$, $\pi e = 1$ for some stationary probability vector $\boldsymbol{\pi}$.

The average arrival rate λ and the average batch arrival rate λ^{b} of the stationary BMAP are given by and $\lambda = \pi\sum_{k=1}^{M} D_K\, e$ and $\lambda^{\mathrm{b}} = \pi\sum_{k=1}^{M} D_K\, e$.

3.2.2 Estimation of time: We define Newton's interpolation polynomial function to predict the response time when BS allocates the bandwidth to SS.

$$P_n(x) = f[x_0] + \sum_{K=1}^{n} f[x_0, \ldots, x_k]e_k(x) \tag{10}$$

Hence

$$f[x_0, \ldots, x_k] = \frac{f[x_0, \ldots, x_k] - f[x_0, \ldots, x_{k-1}]}{x_k - x_0} \tag{11}$$

The polynomials of Newton's basis e_k, are defined by

$$\begin{aligned} e_k(x) &= \textstyle\coprod_{i=0}^{k-1} (x - x_i) \\ &= (x - x_0)(x - x_1)\ldots(x - x_{k-1}),\ k = 1, \ldots n \end{aligned} \tag{12}$$

With the following convention

$$e_0 = 1$$

Newton's interpolation polynomial function has the Runger phenomenon, given $n = 2$. We assume that the bandwidth request time is $T_{\mathrm{r}0}$, $T_{\mathrm{r}1}$, $T_{\mathrm{r}2}$ and the bandwidth response time is T_0, T_1, T_2. In the next round, when to SS request bandwidth time is T_{r}, we can use the previous the history record and Newton's interpolation polynomial to predict the response time T_{pn}.

We define the E_0 and E_1 as follows

$$E_0 = \frac{T_1 - T_0}{T_{\mathrm{r}1} - T_{\mathrm{r}o}} \tag{13}$$

and

$$E_1 = \frac{T_2 - T_1}{T_{\mathrm{r}2} - T_{\mathrm{r}1}} \tag{14}$$

So

$$T_{pn} = T_0 + E_0(T_{\mathrm{r}} - T_{\mathrm{r}0}) + \frac{E_1 - E_0}{T_{\mathrm{r}2} - T_{\mathrm{r}0}}(T_{\mathrm{r}} - T_{\mathrm{r}0})(T_{\mathrm{r}} - T_{\mathrm{r}1}) \tag{15}$$

3.2.3 Adaptive time slots calculating: We try to allocate more time slots at a time, and so we predict the rtPS bandwidth requirement size between the time it makes the request for bandwidth and the time the BS responds. We use a smoothing parameter to modify the requested bandwidth based on the difference between previous predictions and actual requirements. The SS will iterate the process in the next required bandwidth cycle. Eventually, we will converge on the optimal bandwidth allocation solution.

Let, $X\,(t)$ = queue length in rtPS scheduler of SS

M = maximum transmission rate at output link for rtPS

We use the following the formula to calculated the expected bandwidth time slots

$$N_p = S_i \frac{X(T_{rn}) + (T_{pn} - T_{rn})\lambda}{M} + (1 - S_i)N_{\mathrm{r}} \tag{16}$$

Given the buffer size, we can calculate the required the time slots. $S_i(0, 1)$ is the smooth parameter, and it will give the ratio of the actual allocation bandwidth to previous predictions and requirements. We set the initial value $S_i = 0.5$. N_{r} is the time sots of required bandwidth that SS calculates and sends to BS, according to the buffer size.

We define the function to modify it.

$$\varepsilon = \left\{ \begin{array}{ll} \left|\dfrac{N - N_{\mathrm{r}}}{N - N_p}\right| & , N_p \ne N \\ 1 & , N_p = N \end{array} \right\} \tag{17}$$

N is real allocation bandwidth. If $\varepsilon > 1$, this shows that the calculated bandwidth is closer to predicted bandwidth . If $\varepsilon < 1$, it shows calculated bandwidth is closer to requested bandwidth. So, we use ω to adjust S_i. In the following formula we define $\omega = 0.05$.

$$S_i = \left\{ \begin{array}{cc} \min\,[S_{i-1} + \omega, 1] & \varepsilon > 1 \\ S_{i-1} & \varepsilon = 1 \\ \max\,[S_{i-1} - \omega, \omega] & \varepsilon < 1 \end{array} \right\} \tag{18}$$

The SS will iterate the process in the next required bandwidth cycle. After a long time, we could obtain the optimal bandwidth allocation solution.

4 Simulation results

In this section we evaluate the performance of the new proposed scheduler through simulation using the OPNET Technologies Inc. (OPNET) simulator. The arrival of data packets is described by BMAP-3 with various traffic rates λ [27].

We simulated a simple WiMAX network in OPNET modeller version 16 [28]. The network consists of one BS and eighteen SSs in each cell as illustrated in Fig. 7.

In real-time communication over the WiMAX network, there are several important parameters, such as: minimum reserved traffic rate, maximum sustained traffic rate, average throughput, average queuing delay and packet loss. QoS parameters values are specified by the SS on admission into the network. Both average delay and packet loss will allow us to determine how effectively a scheduling algorithm satisfies the QoS requirements of real-time SSs.

We compare these QoS parameters with the WFQ and WRR algorithm. We use the following parameters shown in Table 2 to do the simulation.

In real-time communication over the WiMAX network, there are four important parameters: frame utilisation, average throughput, average queuing delay and packet loss, which we compare with the WRR and WFQ algorithm.

Fig. 6 is frame utilisation which is the number of symbols utilised for data out all the symbols in the uplink subframe. The metric is used to determine how effectively the scheduling algorithm utilises the frame. We know low-overhead (LOH) scheduling algorithm guarantees the higher frame utilisation and gets the best performance.

Fig. 7 shows the results for the average throughput, which are the amount of data transmitted by a user over the simulation time. The LOH scheduling algorithm indicates the highest average throughput for the rtPS class. The average throughput decreases with an increasing number of

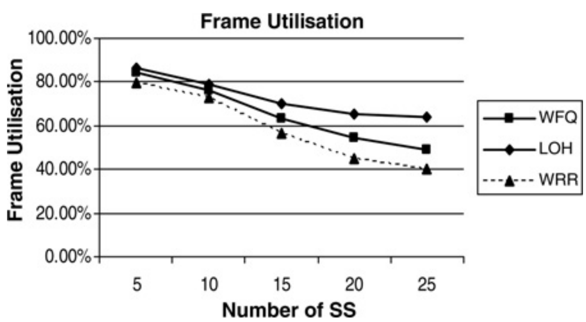

Fig. 6 *Frame utilisation*

Table 2 Simulation parameters

Parameter type	Parameter value
base frequency	2.5 GHz
duplexing	TDD
bits per OFDM symbol	192
overall bandwidth	5 Mbps
DL/UL ratio	2/1
initial ranging transmission opportunity	1
frame duration	5 ms
cyclic prefix duration	11.42 μs
basic symbol	91.43 μs
fast Fourier transform size	1024
number of subchannels	12
inter-arrival time between video frames	120 ms
minimum reserved traffic rate	32 kbps
maximum sustained traffic rate	384 kbps
maximum latency	160 ms

SSs because of decreasing load per SS and increase in bandwidth wasted by uplink burst preambles. When the number of SSs is high, the lower priority SSs are allocated very little bandwidth, indicated by a very low-average throughput. WRR has the lowest throughput and LOH scheduling algorithm presents best performance and enhances the system throughput. Even though the available bandwidth is insufficient, it still provides the highest throughput compared to the other algorithm.

Queuing delay is shown in Fig. 8, which is the time between the arrival and departure of a packet from the queue. To measure queue delay, an expression must be developed for queue size, which varies as packets arrive and depart. As the number of SSs increase, so does the average delay owing to competition for a fixed amount of bandwidth. As Fig. 8 illustrates, LOH has better performance than WFQ and WRR algorithms with an increase in the number of SSs, and its delay is small enough to not affect real-time performance. A videophone is one kind of real-time service. Videophones typically use two-way communications and have data rates between 32 and 384 kbps. With videophones, end-to-end delays <400 ms are acceptable, which means scheduled delays should be <160 ms. When watching videos online, the video traffic is more tolerant to delay since it is unidirectional only in the downlink direction.

Fig. 9 displays packet loss, which is the percentage of packets dropped from the queue out of all the packets that arrived into the queue. The metric indicates the percentage of packets that missed their delay bounds. Both average delay and packet loss will allow us to determine how effectively a scheduling algorithm satisfies the QoS

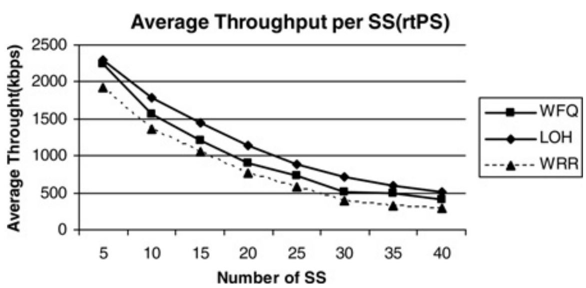

Fig. 7 *Average throughput*

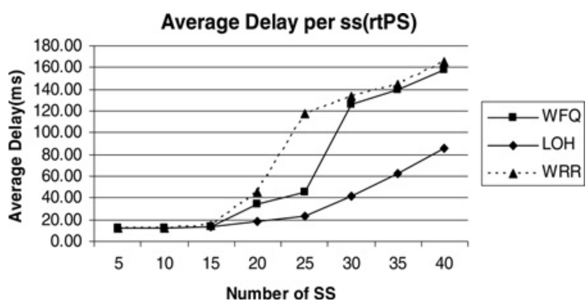

Fig. 8 *Queuing delay*

doi: 10.1049/iet-com.2010.0770

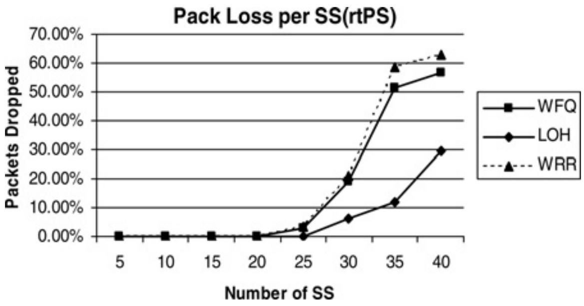

Fig. 9 *Packet loss*

requirements of real-time SSs. The packet loss increases with increasing number of SSs owing to increasing overhead of uplink burst preamble. The WFQ and WRR algorithms indicate higher average delay than LOH scheduling algorithm. They result in a lower packet loss using the LOH scheduling algorithm. The packet loss for the rtPS classes increases under the WFQ and WRR algorithms greater than LOH scheduling algorithm does under the same conditions.

5 Conclusion and future work

In conclusion, compared to the WRR and WFQ algorithm, the low-overhead uplink scheduling scheme for uplink real-time service sharply reduces the MAP and MAC SDUs subheader overhead and improves the system throughput. From the simulation results, we can see that not only is the QoS requirement satisfied, but also that the performance is optimised. It shows that the performance increase is most obvious in scenarios with a large number of users and short packet interval arrival time. For future work, we will combine the low-overhead uplink scheduling scheme with adaptive modulation and coding and channel state information at the physical layer, to try to maximise the throughput and spectral efficiency under the delay constraints. Moreover, we will research the bin packing problem or the rectangular packing problem and try to improve the bandwidth utilisation of the WiMAX system. The bin packing problem is known to be NP complete. Finding a mapping from the allocations into the downlink subframe for each burst is a variation of the bin packing problem. The mapping decision needs to be made within a few milliseconds for each WiMAX frame.

6 References

1 Shankar, A.R., Hegde, R.: 'WIMAX on the road to future'. IET Int. Conf. on Wireless, Mobile and Multimedia Network, 11–12, 2008, pp. 275–278

2 IEEE P802.16Rev2/D2: 'DRAFT standard for local and metropolitan area networks – part 16: air interface for broadband wireless access systems', December 2007, 2094 pp

3 Kwon, T., Lee, H., Choi, S., *et al.*: 'Design and implementation of a simulator based on a cross-layer protocol between mac and phy layers in a WiBro compatible IEEE 802.16e OFDMA system', *IEEE Commun. Mag.*, 2005, **43**, (12), pp. 136–146

4 Kim, J.Y., Cho, D.-H.: 'Piggybacking scheme of MAP IE for minimizing MAC overhead in the IEEE 802.16e OFDMA system'. PIEEE VTC 2007-Fall, September 2007, pp. 284–288

5 WiMAX Forum: 'WiMAX system evaluation methodology V2.0', December 2007, 230 pp

6 Ma, M., Lu, J., Fu, C.P.: 'Hierarchical scheduling framework for QoS service in WiMAX point-to-multi-point networks', *IET Commun.*, 2010, **4**, (9), pp. 1073–1082

7 Sayenko, A., Alanen, O., Hamalainen, T.: 'Scheduling solution for the IEEE 802.16 base station', *Spec. Issue Comput. Netw.*, 2008, **52**, (1), pp. 96–115

8 Boedihardjo, A.P., Liang, Y.: 'Hierarchical smoothed round robin scheduling in high-speed networks', *IET Commun.*, 2009, **3**, (9), pp. 1557–1568

9 Lu, S., Bharghavan, V., Srikant, R.: 'Fair scheduling in wireless packet networks', *IEEE/ACM Trans. Netw.*, 1999, **7**, pp. 473–489

10 Kim, T., Lim, J.T.: 'Quality of service supporting downlink scheduling scheme in wordwide interoperability for microwave access systems', *IET Commun.*, 2010, **4**, (1), pp. 32–38

11 Kim, H., Han, Y.: 'A proportional fair scheduling for multicarrier transmission systems', *IEEE Commun. Lett.*, 2005, **9**, pp. 210–212

12 Tassiulas, L., Sarkar, S.: 'Maxmin fair scheduling in wireless networks'. Proc. IEEE Computer Communication Conf., New York, NY, 2002, vol. 2, pp. 763–772

13 Bhagwat, P., Bhattacharya, P., Krishna, A., Tripathi, S.K.: 'Enhancing throughput over wireless LANs using channel state dependent packet scheduling'. Proc. IEEE Computer Communication Conf., San Francisco, CA, 1996, vol. 3, pp. 1133–1140

14 Vaidya, N.H., Bahl, P., Gupta, S.: 'Distributed fair scheduling in a wireless LAN', *IEEE Trans. Mob. Comput.*, 2005, **4**, pp. 616–629

15 Jung, E., Vaidya, N.H.: 'An energy efficient MAC protocol for wireless LANs'. Proc. IEEE Computer Communication Conf., New York, NY, 2002, vol. 3, pp. 1756–1764

16 Parekh, A.K., Gallager, R.G.: 'A generalized processor sharing approach to flow control in integrated services networks: the single node case', *IEEE/ACM Trans. Netw.*, 1993, **1**, (3), pp. 344–357

17 Friderikos, V., Papadaki, K., Wisely, D., Aghvami, H.: 'Multi-rate power-controlled link scheduling for mesh broadband wireless access networks', *IET Commun.*, 2007, **1**, (5), pp. 909–914

18 Katevenis, M., Sidiropoulous, S., Courcoubetis, C., *et al.*: 'Weighted round-robin cell multiplexing in a general purpose ATM switch chip', *IEEE J. Sel. Areas Commun.*, 1991, **9**, pp. 1265–1279

19 So-In, C., Jain, R., Al-Tamimi, A.: 'Generalized weighted fairness and its support in deficit round robin with fragmentation in IEEE 802.16 WiMAX'. IEEE Sarnoff Symp. 2009, December 2008, Available at http://www.cse.wustl.edu/~jain/papers/gwf.htm, Submitted for publication

20 So-In, C., Jain, R., Al-Tamimi, A.: 'Scheduling in IEEE 802.16e mobile WiMAX networks: key issues and a survey', *IEEE J. Sel. Areas Commun. (JSAC)*, 2009, **27**, (2), pp. 156–171

21 Katevenis, M., Sidiropolous, S., Courcoubetis, C.: 'Weighted round-robin cell multiplexing in a general-purpose ATM switch chip', *IEEE J. Sel. Areas. Commun.*, 1991, **9**, pp. 1265–1279

22 Cicconetti, C., Erta, A., Lenzini, L., Mingozzi, E.: 'Performance evaluation of the IEEE 802.16 MAC for QoS support', *IEEE Trans. Mob. Comput.*, 2007, **6**, (1), pp. 26–38

23 Dianati, M., Shen, X., Naik, S.: 'A new fairness index for radio resource allocation in wireless networks'. Proc. IEEE Wireless Communications and Networking Conf., March 2005, pp. 712–717

24 Shakkottai, S., Srikant, R.: 'Scheduling real-time traffic with deadlines over a wireless channel', *ACM/Baltzer Wirel. Netw.*, 2002, **8**, pp. 13–26

25 Lucantoni, D.M.: 'The BMAP/G/1 queue'. A Tutorial – Models and Techniques for Performance Evaluation of Computer and Communications Systems, 1993, pp. 330–358

26 Klemm, A., Lindemann, C., Lohmann, M.: 'Modeling IP traffic using the batch markovian arrival process (extended version)', *Perform. Eval.*, 2003, **54**, pp. 149–173

27 Lindemann, C., Lohmann, M.: 'IP2BMAP software package', available at: www.ip2bmap.de

28 http://www.opnet.com/solutions/network_rd/modeler_wireless.html