Performance Comparison of Scheduling Algorithms for Real-Time Applications over WiMAX Networks

Nada M. El-Shennawy¹, Mahmoud M. Fahmy², Mohamed N. El-Derini³, and Moustafa A. Yousef⁴

Computers and Automatic Control Dept. Faculty of Engineering, Tanta University, Tanta, Egypt¹,². Vice Rector for Education and Students Affairs, Pharos University³, Computer Science and Engineering, E-JUST, Alexandria, Egypt⁴

Nada_elshennawy@f-eng.tanta.edu.eg¹, mfn_288@hotmail.com², nazih.elderini@pua.edu.eg³, and moustafa@cs.umd.edu⁴

ABSTRACT

WiMAX (Worldwide Interoperability for Microwave Access) is an emerging broadband wireless network for providing Last-mile problem solutions for supporting higher bandwidth and many service classes with dissimilar quality of service requirement. Real-time applications are widely implemented over the Internet. So the Internet needs a network access with strong support for these applications. There is an emerging broadband wireless access network, namely, WiMAX networks. WiMAX has efficient and reliable quality of service (QoS) architecture which can achieve the real-time applications requirements. A powerful scheduling algorithm is essential in WiMAX to fulfill the growth of using dissimilar applications. Video conferencing and high quality video are the most popular real-time applications. In this paper, we introduce performance comparison of some uplink scheduling algorithms to measure the enhancement of real-time applications performance. One of the algorithms considered is our proposal. Delay and jitter of applications are used as performance metrics. The results show that the proposed algorithm outperforms the other algorithms considered with respect to delay and jitter of real-time applications.

KEYWORDS

WiMAX, Scheduling Algorithms, QoS, Real-Time Applications

1. INTRODUCTION

The rapid growth of new services such as online video games, video conferences, and multimedia services is demanding a reliable and an efficient Internet access. Wireless broadband access networks [1, 2] represent a viable solution to provide last-mile access to the Internet. WiMAX is one of the emerging broadband wireless access networks [1, 3]. The term WiMAX is commonly used to refer to collection of standards, products, and service offerings derived from the IEEE 802.16 family of standards [3]. WiMAX is a good choice for multimedia applications because of many features. These features include high data rate, large spanning area, adaptive modulation and coding rate, security management, and diverse quality of service (QoS) for all types of real-time applications [1, 3]. To support these applications efficiently, strong and reliable QoS architecture is crucial to work with the different requirements of end users [4].

A powerful scheduling algorithm is essential in WiMAX networks to satisfy the growth of end user requirements for different applications. There is no specific scheduling algorithm stated in IEEE802.16 standard to use. The selection of the algorithm is left for service providers to pick a suitable one, which is able to satisfy dissimilar application requirements [1, 5].

Scheduling algorithms can be classified into two categories [5]: channel-aware and channel-unaware algorithms. Channel-aware algorithms take the channel information into account in the bandwidth allocation decision. But, channel-
unaware algorithms do not use any channel information in the bandwidth allocation decision.

There are two types of Internet applications: real-time and non-real-time. A real-time application is an application that functions within a time frame. The latency must be less than a defined value for the application, usually measured in seconds. Video conferencing and high quality video are the most famous real-time applications in the Internet. Video conferencing is a communications technology that integrates video and audio to connect users anywhere in the world as if they were in the same room.

A good survey about scheduling algorithms in WiMAX networks is presented by So-In et al. [6]. L. Jin-Cherng and et al. in [7], provided a performance simulation study of some scheduling schemes such as weighted fair queuing (WFQ), random early detection (RED), fair queuing (FQ), deficit round robin (DRR) and drop-tail. The authors reported that the weighted queuing scheme with dynamic bandwidth allocation functions give the best performance in WiMAX networks. H. Guesmi and et al. in [8] introduced a performance study of some scheduling algorithms: first-in first-out (FIFO), fair queuing (FQ), deficit round robin (DRR) and weighted fair queuing (WFQ). The authors conclude that WFQ guarantee the QoS for each class in WiMAX. Patrik Dhronoe and et al. in [9] presented performance study of some uplink scheduling algorithms using simulation analysis. The authors divided the scheduling algorithms into three groups and studied examples from each group. A complete analysis of these scheduling algorithms is given. The authors recommended that none of the algorithms considered is capable of effectively supporting all WiMAX classes of service.

Research workers, in the area of WiMAX networks, often recommend weighted scheduling algorithms with dynamic weight functions and the performance study of WiMAX networks with respect to scheduling algorithms are presented only. In this paper, we introduce performance study of real-time applications based on some scheduling algorithms one of these algorithms is our proposal and it is published in [10].

The rest of this paper is organized as follows. In Section 2, an overview of WiMAX networks and scheduling algorithms are reviewed. Section 3 presents the details of the proposed approach. Simulation results are introduced in Section 4. Finally, conclusions and trends for future work are reported in Section 5.

2. WiMAX NETWORKS AND SCHEDULING ALGORITHMS: An OVERVIEW

WiMAX stands for worldwide interoperability for microwave access. It is designed based on the IEEE 802.16 standard [1, 3, 11]. WiMAX networks bases on IEEE 802.16 standard are divided into two main layers: physical layer (PHY) and medium access control layer (MAC). The PHY layer can use many physical layer types such as: wireless MAN-OFDM (orthogonal frequency division multiplexing), wireless MAN-SC (single carrier), wireless MAN-SCa, and wireless MAN-OFDMA (orthogonal frequency division multiple access) [3].

MAC layer is the intermediate layer between the WiMAX PHY layer and the higher layers. It is responsible of many important jobs outlined as follows: header suppression, packet scheduling, bandwidth allocation, QoS management, and security and authentication issues.

To facilitate the MAC layer work, the MAC layer is divided into three sublayers. Each sublayer is responsible for some of MAC functions. The three sublayers are [4, 9, 12]: convergence sublayer, common-part sublayer, and security sublayer.

Convergence sublayer, it is designed for making the convergence between the higher layers and the WiMAX MAC layer. Its main function is mapping data from the upper layer into appropriate MAC service data unit (SDUs). Also it makes a classification of data into suitable service class and the header suppression operation. Common-part sublayer, it is responsible for connection establishment, QoS management, service flow management, bandwidth allocation, and scheduling services. Security sublayer, it is developed for authentication, security key exchange, and encryption. Security is maintained by encryption of data and secure key distribution.

WiMAX has two types of communication modes: Point-to-multipoint (PMP) mode and mesh mode. In PMP mode, the communications between all subscriber stations (SSs) are organized and passed through the base station (BS). But in mesh mode, the communications can be achieved directly between SSs.

The main feature of WiMAX is the QoS support [11, 13, 14]. WiMAX is designed to manage dissimilar applications, including voice, video, and data by defining five different service classes for constant and variable bit rate applications.

The service classes are [3, 5, 12]: unsolicited grant service (UGS), used to support constant data rate real-time applications such as VoIP without silence suppression; real-time polling service (rtPS), defined to support real-time applications with variable data rate such as a MPEG compressed video; extended real-time polling service (ertPS), used to support real-time applications with variable data rate such as VoIP with silence suppression; non-real-time polling service (nrtPS), defined for variable bit rate non-real-time applications; finally, the best effort (BE) service class defines non-real-time applications with no need of any special requirements.

QoS plays major role in determining the network performance. It has three main parameters, namely, throughput, delay, and jitter. QoS has two main control key architectures which are used to enhance the QoS and the overall network performance. These architectures are: scheduling algorithm and admission control algorithm.

A scheduling algorithm is a part of QoS architecture. Its function is the allocation of bandwidth among SSs in such a way to maximize throughput and minimize both delay and jitter. The scheduler should be simple, fair, and efficient. To ensure good performance of QoS in WiMAX networks, suitable bandwidth allocation algorithm is needed [4, 12, 14]. The scheduling algorithm is a significant part of QoS architecture. There are two types of scheduling algorithms are defined in BS [5]: downlink algorithm (from BS to SSs), and uplink algorithm (from SSs to BS). Also, SS has an internal scheduling algorithm to use when SS has many application types.

Scheduling algorithms classified into two main classes [5]: channel-aware scheduling algorithms and channel-unaware algorithms. In channel-unaware algorithms, the bandwidth allocation is worked without any use of information about the channel. Weighted round robin (WRR) [5, 9, 16], Deficit round robin (DRR) [8, 14, 15], and modified deficit round robin (MDRR) [17] are classified under this type of algorithms.

In channel-aware algorithms, the decision of the bandwidth allocation makes based on the channel information such as signal strength, signal-to-noise ratio, and received signal power. There are many channel-aware schedulers [5], for example, MLWDF [18] and link Adaptive largest weighted throughput (LWT) [19].

2-1 Channel-unaware algorithms
The channel-unaware algorithms are outlined as follows. The round robin (RR) algorithm [6] is a simple algorithm and fair in assigning one allocation for each connection in each serving cycle. Weighted round robin (WRR) [6, 12, 20] assign a weight to each connection then the connections served according to their weights. The main problem of WRR is that when the traffic has a variable packet size, WRR provides incorrect percentage of bandwidth allocation. Deficit Round Robin (DRR) [21] solves this problem of WRR. DRR defines two variables for each queue, deficit counter (DC) and quantum (Q). Q is set to constant value equal to the maximum traffic packet of the queue, and DC is initialized by a zero value when the queue created. When the queue is visited to serve, the value of Q is added to DC and the queue is still served until the head packet size is greater than DC. For each served packet, the value of DC decreases by the value of packet size. When the queue is empty, DC retunes to zero. Deficit weighted round robin (DWR) [22] is the same as DRR but adds a weight variable for each queue and the Q value depends on the weight value. Another modification on DRR named modified deficit round robin (MDRR) [22] works in the same way as DRR but a priority parameter is added for each queue to contribute to queue selection, it is a queue priority.

2-2 Channel-aware algorithms
Channel-aware algorithms such as: modified largest weighted delay first (MLWDF), proportional fairness schema (PFS), and maximum carrier to interference ratio (MAX C/I). MLWDF [7] is one of QoS guaranteed algorithms which support minimum throughput and delay. In this algorithm, for each queue j the scheduler computes a function $\rho_i * W_j(t) * r_j(t)$, where $\rho_i$ is a constant which should be take different value for each service classes, $W_j(t)$ can be either the delay of the head of line packet or the queue length, and $r_j(t)$ is the channel capacity for traffic i. The queue selection occurred based on the function value starting from the largest value. There are many modifications of MLWDF. PFS [23] belongs to fairness scheduler family which worked based on maximizing the long-term fairness. PFS uses a ratio of channel capacity $W_i(t)$ to the long-term throughput $R_i(t)$ to select the queue which will be served. The queue selection occurred based on the ratio value starting from the largest value. The main disadvantage of PFS is that there is no guarantee for delay. MAX C/I [23] used to maximizing the throughput. In MAX C/I, the queue is selected based on the best channel conditions. In WiMAX, the most used channel quality indicator is CINR. This algorithm checks the value of CINR for each queue and the queue with largest CINR is served first. The movement between the queues is occurred based on the CINR value in descending order.

3. THE PROPOSED ALGORITHM

In this paper, we introduce performance comparison of the proposed scheduling algorithm which is published in [10] with respect to the most famous real-time applications. These applications are video conference and high quality video. The proposed algorithm is a dynamic uplink channel-unaware scheduling algorithm for fixed WiMAX networks.

The proposed bandwidth allocation method for dividing bandwidth among n queues; that is n subscriber stations, depends on the formulation of a dynamic weight function in terms of the three QoS parameters: throughput, delay, and jitter. To this end, a weight $W_i(t)$ is assigned to queue i as a positive factor of the form in equation (1).

$$W_i(t) = \frac{N_i(t)}{\sum_{j=1}^{n} N_j(t)} \quad 1 \leq i \leq n \quad (1)$$

In equation (2), $N_i(t)$ is expressed as the sum of three terms corresponding to contributions of throughput, delay, and jitter, respectively. Specifically, we propose the following formula for a weight function $N_i(t)$:

$$N_i(t) = T_i + D_i(t) + J_i(t) \quad 1 \leq i \leq n \quad (2)$$

The first term $T_i$, in equation (2), is the fractional throughput contribution to $N_i(t)$, defined as:

$$T_i = \frac{X_i}{\sum_{j=1}^{n} X_j} \quad (3)$$

where $X_i$ is the minimum reserved traffic rate for queue i. The second term $D_i(t)$ is the fractional delay contribution

$$D_i(t) = \frac{\alpha_i \frac{Y_i(t)}{L_i}}{\sum_{j=1}^{n} \alpha_i \frac{Y_j(t)}{L_j}} \quad (4)$$

where $Y_i(t)$ is a time-varying average delay, $L_i$ is the given maximum latency, and $\alpha_i$ is a positive delay weighting factor. In equation (4), the ratio $Y_i(t)/L_i$ (less than unity) expresses the proportion of the delay of a particular queue relative to the maximum acceptable delay of the network. Further, the ratio $Y_i(t)/L_i$ is weighted by a factor $\alpha_i$, whose value varies according to the subscriber station (value of i). This is justifiable since each subscriber station is devoted to a particular application. The third term $J_i(t)$ is the fractional jitter contribution,

$$J_i(t) = \frac{\beta_i Z_i(t)}{\sum_{j=1}^{n} \beta_j Z_j(t)} \quad (5)$$

where $Z_i(t)$ is a time-varying average jitter, $K_i$ is the given maximum jitter and $\beta_i$ is a positive jitter weighting factor. The terms in equation (5) can be interpreted in the same way as in equation (4). Then, the uplink bandwidth divides among the n queues using the form in equation (6).

$$BW_i(t) = W_i(t) \times UL_{bw} \quad (6)$$
where $BW_i$ is the reserved bandwidth for queue $i$ and $UL_{BW}$ is the total bandwidth of the uplink subframe.

Equation (2) is valid for both real- and non-real time applications; this implies that the weighting factors $\alpha_i$ and $\beta_i$ for real-time applications should be greater than those for non-real-time applications. The reason is the fact that real-time applications are more highly sensitive to delay and jitter.

The values of the weighting factors $\alpha_i$ and $\beta_i$ are chosen in such a way that both delay and jitter are given greater attention in real-time applications than in non-real-time applications. In these specific applications, it is found that the best possible values of $\alpha_i$ and $\beta_i$ are in the ratio 1:10 in non-real-time and real-time applications [3].

The processes of the proposed algorithm are illustrated using the flowchart shown in Figure 1.

4. EXPERIMENTAL SCENARIOS AND SIMULATION RESULTS

Here, we present the performance study of the proposed algorithm. The simulation results are obtained using OPNET [24]. The used network consists of four WiMAX service classes: ertPS, rtPS, nrtPS and BE with applications: VoIP, video conference, FTP and HTTP, respectively. The traffic parameters for each service class are listed in Table (1) [9]. Aother scenario is used by substituting the video conference application by high quality video.

![Figure 1: The Proposed Algorithm Processes](image-url)
Delay and jitter for real-time applications are studied by using several scenarios by varying the number of SSs. Each scenario consists of one BS, serving a number of SSs, communicating in PMP mode of operation. The frame duration is 5 msec, with 50% for each uplink and downlink subframe. A random topology in 1000 x 1000 m square space is used. The number of SSs varies from 6 to 36 with ratio 1:2:2:1 SSs for service classes ERTS:RTS:NRTPS:BE, respectively. The proposed algorithm is compared with MDRR and WRR. Simulation time is 10 minutes [9].

From Figure 2 and Figure 3, we conclude that the proposed algorithm has better values for both video conferencing delay and jitter. This is due to use of weighting factors for real- and non-real-time applications which are gave high importance for real-time applications than in non-real-time applications.
The percentage of enhancement of the second real-time application, high quality video, delay and jitter is shown in Table (4) and Table (5), extracted from Figure 4 and Figure 5 respectively. The maximum enhancement occurred at SSs=36 for delay and SSs=30 for jitter. Where the minimum enhancement occurred at SSs=12 for delay and at SSs=12 for jitter.

Table (4) proposal enhancement in high quality video delay

<table>
<thead>
<tr>
<th>Feature Algorithms</th>
<th>Max. Enhancement (SSs=36, Proposal=0.0562)</th>
<th>Min. Enhancement (SSs=12, Proposal=0.016)</th>
</tr>
</thead>
<tbody>
<tr>
<td>WRR</td>
<td>Value 0.09125 % 38 Value 0.0167 % 4</td>
<td></td>
</tr>
<tr>
<td>MDRR</td>
<td>Value 0.0776 % 28 Value 0.0161 % 6</td>
<td></td>
</tr>
</tbody>
</table>

Table (5) proposal enhancement in high quality video jitter

<table>
<thead>
<tr>
<th>Feature Algorithms</th>
<th>Max. Enhancement (SSs=30, Proposal=0.000143)</th>
<th>Min. Enhancement (SSs=12, Proposal=0.00057)</th>
</tr>
</thead>
<tbody>
<tr>
<td>WRR</td>
<td>Value 0.000458 % 69 Value 0.00007 % 18</td>
<td></td>
</tr>
<tr>
<td>MDRR</td>
<td>Value 0.00035 % 59 Value 0.00085 % 33</td>
<td></td>
</tr>
</tbody>
</table>

5. CONCLUSIONS AND TRENDS FOR FUTURE WORK

To meet the QoS requirements of multimedia applications, a scheduling algorithm is needed to allocate the bandwidth to users to satisfy bounds on delay and jitter and to maximize throughput. In this paper, we introduce a performance comparison of our proposed scheduling algorithm which is published in [10] with respect to the most famous real-time applications. These applications are video conference and high quality video. The simulation results reveal that our algorithm outperforms the other two algorithms with respect
delay and jitter for the two real-time applications as functions of number of subscriber stations.

In a future research work, an interesting challenging task will be focused on the application of the proposed algorithms to the newly established Long-Term Evolution (LTE) networks [25].

REFERENCES


