

Modified WRR Scheduling Algorithm for WiMAX Networks

Wail Mardini

Dept. of Computer Science, Jordan University of Science and Technology

Irbid, 22110 Jordan

Tel: +962-777-306646 E-mail: mardini@just.edu.jo

Mai M. Abu Alfoul

Dept. of Computer Science, Jordan University of Science and Technology

Irbid, 22110 Jordan

Tel: +962-776-072942 E-mail: mmabualfoul@just.edu.jo

Received: June 27, 2011 Accepted: July 24, 2011 DOI: 10.5296/npa.v3i2.751

Abstract

Worldwide Interoperability for Microwave Access (WiMAX), which is also known as IEEE 802.16 standard, supports last-mile broadband access wireless networks. WiMAX has many advantages including wide coverage area and high bandwidth. These advantages enable WiMAX to support long transmission range and high data rate compared to cellular and WiFi network. WiMAX technology uses a number of scheduling techniques in the Medium Access Control layer, which is responsible for the utilization of available resources in the networks and distribute them among users in order to ensure the desired quality of service. In this study, we propose a Modified Weighted Round Robin (MWRR) scheduler in order to decrease the average end-to-end delay and improve the average throughput. The proposed scheduling technique has been designed and simulated using the QualNet 5.0.2 network simulator. In order to evaluate the performance of our proposed approach, we compared our results to the results of well-known scheduling techniques (Weighted Round Robin, Strict Priority, and Weighted Fair Queuing). The average percentage of improvement was around 4%.

Keywords: WRR, WiMAX, Scheduling, Call Admission Control.

1. Introduction

1.1 Overview of Wireless Network

Recently, the world has witnessed great improvement in the means of communication such as laptops and mobile phones. Wireless network is a type of network where links between the participating nodes is implemented without using wires.

According to their dependence on the infrastructure, the wireless networks can be divided into two distinct types. The first type is known as infrastructure based where the wireless network have an access point that enables the communication between host devices within the scope of the transmission. Worldwide Interoperability for Microwave Access (WiMAX) is an example of this type of networks, where mobile hosts can communicate through some access points. When the host changes his position and comes out of the scope of an access point, he connects with a new access point and resumes the connection. This process is called handover [1]. One problem associated with operation of the transformation is the strict requirement for a smooth handover of the connection through an access point to another [2].

The second type of wireless networks is infrastructureless ad hoc wireless networks. As the name indicates, these networks are established in an ad hoc manner, and there are no central access points. Therefore, the host cell is self-organizing without the help of any infrastructure [3] [4].

1.2 WiMAX Networks and Characteristics

Worldwide Interoperability for Microwave Access (WiMAX), which is also known as 802.16 standard, has many advantages that enable it to use in many applications, such as; support long transmission range and high data rate compared to cellular and WiFi network. WiMAX in some applications covers areas using 2.3 - 2.7GHz frequency which covers up to approximately 35 miles [5] [6] [7].

There are two models in WiMAX: Point to Multipoint (PMP) networks and Multipoint to Multipoint [1] [6] [8] [9]:

PMP is an access network which includes a small number of Subscriber Stations (SSs) which are connected to a full functional Base Station (BS). When users install user equipment, they can get immediate access to the network. In this model, the SSs can direct the antennas to the base station where the base station is the clustering point between all SSs [3].

Multipoint to Multipoint, which also called mesh network, is a network without centralized base station, and each subscriber station has the ability to connect directly to another subscriber station or via intermediate subscriber stations [3].

The communications between the SSs and the BS is managed by Access Service Network (ASN). For connectivity between SSs, WiMAX also uses functions provided by Connectivity Service Network (CSN), which allows any two SSs to communicate directly when each one of them is within the other transmission range. Otherwise, they need another

intermediate node such as BS or SS to perform the connection [2]. Fig. 1 shows an example of WiMAX network that involves a collection of subscriber stations connected to one base station.

Many applications are currently using this type of network to get fast and wide range connection, such as broadband home networking [10].

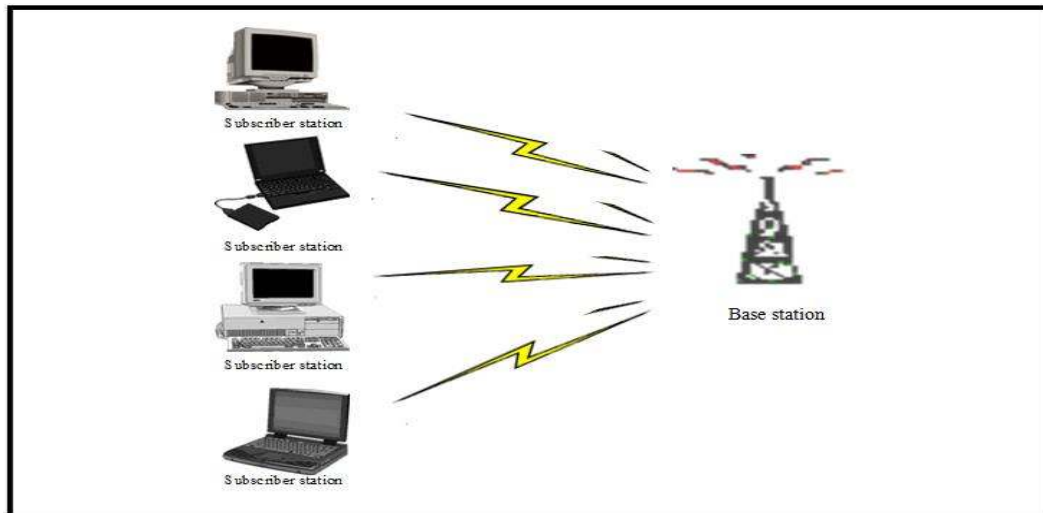


Figure. 1: A collection of subscriber stations connected to one base station forming a Point to Multipoint WiMAX network

In the urban and suburban areas, there is a problem in using wired based technologies such as digital subscriber line (DSL) and cable because of the need to satisfy rural areas requirements at lower cost and higher speed of deployment. WiMAX can be the best solution for such cases. Moreover, even in large cities, WiMAX can be used to extend services to any area efficiently [11] [12].

Recently, the world has witnessed the evolution of many multimedia applications in different fields and the demand of such applications is increasing. Internet Protocol TeleVision (IPTV) is the most important application that uses broadband multimedia, and it is expected to contribute to the next generation of wireless networks. WiMAX is becoming more wide spreading than DSL and cable in providing high performance of Quality of Service (QoS) in multimedia applications [13].

The data rate can reach up to 70 Mbps enabling WiMAX to serve all SSs with the required QoS and to support different classes of services in many applications such as web browsing, VoIP, and multimedia applications [14] [15].

When SSs ask for specific class of service, the BS locates bandwidth suitable for this class with attention to required delay. There are five levels of QoS: Unsolicited Grant Scheme (UGS), real time Polling Service (rtPS), extended real time Polling Service (ertPS), non-real time Polling Service (nrtPS), and Best Effort (BE). We will explain them later in section two [3].

1.3 Statement of the Problem

In WiMAX, there are many challenges and issues that need to be tackled such as the level of Quality of Service (QoS), bandwidth allocation and transmission rate limitations. It is a challenging task to choose a suitable scheduling technique that supports different QoS requirements for different SSs. We studied this problem and proposed a new scheduling technique.

1.4 Motivation

Scheduling is concerned with the distribution of the available resources among the users in a fair manner. It aims to achieve maximum throughput with minimum delay to ensure fairness among all users. To provide QoS, we must consider the queue status and the priority for each packet to guarantee fairness between all users.

In this study, we propose a Modified Weighted Round Robin (MWRR) scheduler in order to decrease the average jitter and average end-to-end delay while maintaining or increasing the average throughput.

The proposed scheduling technique has been designed and simulated using the QualNet 5.0.2 network simulator [16]. In order to evaluate the performance of our proposed approach, we compared our results to the results of well-known scheduling techniques (weighted Round Robin (WRR), Strict Priority (SP), and Weighted Fair Queuing (WFQ)). The decision in choosing these schedulers was based on the ability of each of them to serve all of service classes.

The reason behind choosing the WRR scheduling techniques is that it has the ability to serve all classes of service, though it does not treat all the classes of services with the same manner. Furthermore, it has a lot of significant features such as provides fairness among queues. This is because the WRR algorithm assigns weight to the SSs according to their Minimum Reserved Traffic Rate (MRTR) and serves all the classes in rounds. In other hand, WRR will not provide good performance in the presence of variable size packets, so we turned in our scheduler to use fixed size packets for better results. In addition, WRR is easy to implement which makes it a good selection.

1.5 Methodology

This paper evaluates the performance of three scheduling techniques; WRR, WFQ, and SP. The evaluation results will be based on how each scheduler can serve different classes of services in a fair manner. In each scheduler, when a new connection arrives, the Call Admission Control (CAC) ensures whether the network is able to provide the required QoS or not, then it decides to reject or accept this connection. The required QoS for each class is usually mapped into a certain bandwidth allocation depending on the type of desired class of service. There are five classes of service at MAC level: UGS, rtPS, ertPS, nrtPS, and BE.

To assess the performance of the four scheduling techniques, we used the QualNet simulator. Different metrics were used to compare them: average throughput, average delay jitter and end-to-end delay metrics.

1.6 Paper Organization

The remainder of this paper is organized as follows. Next section discusses scheduling and call admission control techniques in wireless networks. In Section 3, we discuss our proposed modified approach after discussing more details about Weighted Round Robin (WRR) technique. Simulation environment and results are discussed in Section 4. Finally, Section 5 concludes this work and provides directions for future work.

Table 1: Type of Service Classes and Theirs QoS Parameters

Class of Service	QoS Parameters	Type of Applications
UGS	Delay Jitter	VoIP Without Silence Suppression
	End-to-End Delay	
	Minimum Reserved Traffic Rate	
	Maximum Sustained Traffic Rate	
	Maximum Latency	
	Packet Loss	
ertPS	Delay Jitter	VoIP With Activity Detection
	Minimum Reserved Traffic Rate	
	Maximum Sustained Traffic Rate	
	Maximum Latency	
	Traffic Priority	
rtPS	Delay Jitter	Internet Shopping; Video on Demand (VoD); Audio on Demand (AoD)
	End-to-End Delay	
	Minimum Reserved Traffic Rate	
	Maximum Sustained Traffic Rate	
	Maximum Reserved Traffic Rate	
	Packet Loss	
nrtPS	Delay Jitter	Multimedia Messaging; E-Commerce; FTP
	End-to-End Delay	
	Minimum Reserved Traffic Rate	
	Maximum Sustained Traffic Rate	
	Packet Loss	
	Traffic Priority	
BE	Delay Jitter	Web Browsing; E-mail
	End-to-End Delay	
	Maximum Sustained Traffic Rate	
	Packet Loss	
	Traffic Priority	

2. Scheduling and Call Admission Control in WiMAX Networks

Resources management is responsible for the distribution of resources in the networking

system. This management aims to achieve the greatest possible utilization from these resources by distributing them among users in fair manner based on the desired QoS [1] [5].

The following sections will discuss Call Admission Control (CAC) and scheduling algorithms in WiMAX networks.

2.1 Call Admission Control

Call admission control algorithm is responsible for the acceptance of new connections to the network. Before accepting any new connections, the CAC make sure that the network is able to provide the required QoS [17].

WiMAX supports different classes of QoS. The QoS requirements for each class are usually mapped into a certain bandwidth allocation depending on the class specifications [18]. There are five levels of QoS at MAC level: [19] [20].

- 1) Unsolicited Grant Services (UGS)
- 2) Real-time Polling Services (rtPS)
- 3) Extended real-time Polling Services (ertPS)
- 4) Non real-time Polling Service (nrtPS)
- 5) Best Effort Services (BE)

Table 1 above illustrates the service classes, theirs QoS parameters and the applications types in which they are used.

2.2 Overview of Scheduling Techniques

The scheduling techniques are solely responsible for the utilization of available resources in the networks and distributing them among users in order to ensure the desired quality of service. Therefore, the scheduling is a real challenge to serve all the users and meet their demands in fair manner. There are three categories of scheduling algorithms for the traffic in WiMAX: homogeneous, hybrid and opportunistic [6] [17].

1) Homogeneous Scheduling Algorithms

Homogeneous scheduling algorithms include a set of traditional scheduling techniques. These techniques aim at solving some problems such as providing high QoS and distributing the resources among all users fairly. Many of these algorithms were proposed mainly for wired networks. However, they are used in WiMAX networks to satisfy the requirements of the five class services (UGS, ertPS, rtPS, nrtPS, and BE) [18] [3] [14] [15]. The following subsections will discuss eight different homogenous scheduling algorithms.

- First In First Out (FIFO) queuing system can be considered as a simple scheduling algorithm in which the received packets is served based on the order of arrival. This system does not take the priority or QoS into consideration when packets are queued or dequeued [21].
- Round Robin (RR) is a simple scheduling algorithm. It works in a circular manner between the existing queues in a time sharing system to avoid starvation on queues with

lower priority. There will be a number of queues; each queue will have a time slot without any priority assigned by the scheduler. In RR scheduler each queue in a single service round is served once and not visited again until all queues are served within the same service round. RR cannot guarantee all QoS requirements for all users and it is not relevant to WiMAX conditions where different QoS requirements may exist [22] [23].

- Weighted Round Robin (WRR) scheduler is designed to serve different processing capacities. It takes into account the queue status. Each queue in WRR is assigned a weight, and the queue with higher weight takes the priority to get a connection first. Moreover, higher priority queues get more connections than the lower weight queues. At the beginning of any request slot, the scheduler checks to see which customer is next to serve. The SSs of the rtPS class are given the higher weights compared to the weight assigned to SSs of the nrtPS and BE classes [17] [22] [18]. The weight of each queue is determined by the average size of the packets in each queue, the minimum reserved traffic rate for those packets and the maximum sustained traffic rate [18] [24].

- Earliest Deadline First (EDF) scheduler is proposed to serve the users of high priority classes (UGS, ertPS, and rtPS) in wide area networks. The scheduler allocates bandwidth to all SSs and assigns deadlines to each packet based on maximum delay requirements. The drawback of this scheduler is that starvation can potentially occur to SSs with lower class of service since they do not have delay requirements [18] [25] [26] [27].

- Strict priority (SP) is a simple scheduling algorithm that serves all the higher priority traffic of the SSs first. In the lower classes of service a starvation occurs between the SSs especially when more loads of higher classes is present. But sometimes, it is possible that the behaviors of the lower-class may affect or delay the behaviors of the higher-class in SP scheduling algorithm under some improper mixing of the traffics and over a high speed links [17] [14] [15].

- Weighted Fair Queuing (WFQ) is a scheduling technique commonly used in uplink traffic in WiMAX networks. WFQ sorts the packets in an increasing order according of its finish time which is computed depending on the size and weight assigned to the packets of the SSs. The main disadvantage of WFQ is that it does not take the start time of the packets [18] [24].

- Deficit Round Robin (DRR) is similar to RR scheduling technique in serving the packets; the difference is in the order of the packet size. DRR gives different amounts of quantum to each queue depending on the required QoS that the SSs desire. Each SS in DRR receives a fixed quantum of service depending on the packet size. But in case the SS can't send a packet or when the packet size is less than the determined quantum, the remainder quantum is stored in a deficit counter and added to the quantum in the next round for this SS [18] [11].

- Adaptive rtPS Scheduler mainly designed for the rtPS class. When the packets arrive, the classifier takes just the packets of rtPS class and allocates the requested bandwidth to them. The purpose of the adaptive rtPS scheduler is to grant the requested bandwidth for

both data packets in the rtPS queue and for the data packets that may arrive. This adaptive scheduler uses a special prediction technique to estimate the time at which the data packet will arrive [18] [24].

2) Hybrid Scheduling Algorithms

Each hybrid scheduling algorithm is a combination of a number of Homogeneous Scheduling Algorithms. In the following we discuss number of popular hybrid scheduling algorithms.

- Hybrid Scheduling Algorithm of (EDF, WFQ and FIFO): In [18] and [24], EDF, WFQ and FIFO scheduling algorithms are combined in one scheduling algorithm. Upon the arrival of every packet, the EDF, WFQ, and FIFO are executed, but the bandwidth allocated to the SS is not executed until the beginning of every frame. SP is the responsible to allocate the overall requested bandwidth. EDF scheduling algorithm is used to serve SSs of ertPS and rtPS classes, while the WFQ scheduling algorithm is used to serve the SSs of nrtPS class and FIFO is used to serve SSs of BE class since the SSs of BE do not require any QoS. This hybrid algorithm has one drawback which is the starvation of the lower priority SSs in case of a large number of SSs has the higher priority, because of the overall available bandwidth are allocated by SP to the SSs of the high priorities.

- Hybrid Scheduling Algorithm of (EDF and WFQ): This technique works is similar manner to the above approach where WFQ scheduler will help in serving BE and nrtPS, while EDF will help in serving the traffic of rtPS class. It uses a fairer manner than strict priority in order to allocate the bandwidth among the classes. The EDF and WFQ are executed when each packet arrives [18] [28].

- WRR and RR Algorithms: In this hybrid algorithm, there are two phases to distribute the bandwidth among all users. In the first phase, WRR is used to allocate part of the bandwidth to all SSs of rtPS and nrtPS classes. Then in the second phase, the remaining bandwidth is allocated to the SSs of the BE class by using the RR algorithm. Therefore, in this algorithm, the lower priority SSs will suffer from starvation in the existence of a large number of higher priority SSs.

3) Opportunistic Scheduling Algorithms

This class of algorithms primarily focuses on utilizing the variability in channel conditions in WiMAX network [18].

As an example belongs to this class of algorithms we discuss maximum Signal-to-Interference Ratio (mSIR). This type of scheduling algorithms take into consideration the quality of the SSs channels and arranges the SSs in a decreasing order based on their received Signal-to-Interference Ratio (SIR). Then, mSIR allocates the resources to the SSs that have the highest SIR while the SSs having the least SIR are scheduled after much delay [24].

2.3 Scheduling Characteristics

The scheduling techniques are distinguished from each other in terms of their characteristics because these characteristics determine the performance of each scheduler. The following subsections will explain some of them [3] [11] [13] [14] [28]

- 1) Flexibility: Scheduling techniques must be flexible enough to fulfill the minimum QoS and user requirements. Therefore, the scheduler requires a minimum number of changes to go smoothly with the requirements of a different network deployment.
- 2) Simplicity: Simplicity is perceived mechanically and conceptually. Mechanically, a scheduling algorithm must be simple to learn and used in a large scale, but conceptually, it must make the analyses of the QoS statistics such as delay and throughput parameters easier.
- 3) Protection: Users who enter service level agreement (SLA) deserve a scheduling algorithm to protect them from variability causes and insurgent users. It must ensure that such fluctuations in the network do not affect them.
- 4) Fairness: Scheduling algorithm should satisfy the QoS requirements, serve all users from different service classes fairly, prevent dropping low priority packets and keep them from corruption before arriving at the destination. The scheduler is fair when it gives a level of fairness to each user suitable for allocated bandwidth.
- 5) Link Utilization: Link utilization is the allocated bandwidth received by SSs and the distribution of the resources between them. Scheduling algorithm should distribute the resources among all users without wasting them, and should not allocate large resources to users if they do not have enough data to transmit.

2.4 Comparison between Different Scheduling Algorithms

The scheduling algorithms differ from each other in terms of their supported class of services and their complexity. This section contains discussion and comparisons between different scheduling algorithms related to our work in this paper.

RR is one of the simplest scheduling algorithms designed especially for a time sharing system, it is well known as the most naive algorithm. RR can provide a fair resource access to each SS, and every queue is allocated with the same portion of system resources regardless of the channel condition. However, the RR scheduler has the same bandwidth efficiency as a random scheduler. Also, it cannot guarantee different QoS requirements for each queue [29].

The WRR scheduler is the same as the RR scheduler except that lower priority tasks are executed for a shorter time-slice. WRR has the ability to serve all classes of service, though it does not treat all the classes of services with the same manner. Furthermore, it provides fairness among all queues. WRR will not provide good performance in the presence of variable packets size. In addition, WRR is easy to implement [18] [24].

DRR is a variation of RR. The difference between them is that when a SS is not able to send a packet, the remainder quantum is stored in a deficit counter. The value of the deficit counter is added to the quantum in the following round. DRR is flexible enough as it allows

provision of quantum of different sizes depending on the QoS requirements of the SSs [24].

WFQ achieves nearly perfect fairness, but it is usually expensive to implement. WFQ's behavior is ideal in a wired network; it distributes the available bandwidth fairly to all active flows by giving each flow the proper priority as indicated by its weight.

Both WFQ and WRR scheduling algorithms assign weights to SSs. Unlike the WRR algorithm, the WFQ algorithm also considers the packet size and the channel capacity when allocating bandwidth to the SSs. The complexity of WFQ is high due to two main reasons: selection of the next queue to serve and the computation of the virtual time. The complexity of the former is $O(\log N)$ whereas the complexity of the latter is $O(N)$, where N is the number of SSs [18].

EDF is one of the most widely used scheduling algorithms for real-time applications as it selects SSs based on their delay requirements, so it is suitable for SSs having traffic contains UGS and rtPS classes of service. The value of maximum latency for SSs with nrtPS and BE traffic is set to infinity [18] [24].

WRR, EDF, and WFQ schemes satisfy QoS requirements of their users in different ways. The EDF algorithm allocates bandwidth according to the delay requirements of the SSs whereas the WRR and WFQ algorithms allocate bandwidth according to the weight assigned to the SSs. In other hand, WRR will be similar to the WFQ if packets are of fixed size. Such a scheduler is simpler to implement and also analytically tractable [18].

The complexity of the WRR algorithm is known to be constant with respect to the number of SSs $O(1)$. The complexity of the WFQ algorithm is $O(N)$. The complexity of the EDF algorithm is also $O(N)$ [18].

In Table 2, the comparison between the different scheduling algorithms is illustrated.

Table 2: Comparison between different Scheduling Algorithms

Scheduling Algorithm	Fairness	Complexity	Simplicity
SP	low fairness	$O(1)$	Easy to implement
RR	fairness when the data packets are equally sized	$O(1)$	Easy to implement
DRR	high fairness	$O(1)$	Easy to implement
WRR	high fairness	$O(1)$	Easy to implement
WFQ	high fairness	$O(n)$	Expensive to implement
EDF	Medium fairness	$O(n)$	Expensive to implement

3. Proposed Scheme

3.1 Overview

A scheduler is the entity that is responsible for distributing the network resources among all active users. Each user could ask for different class of service. There are five different service classes. Each class of service guarantees a minimum average end-to-end delay and bandwidth requirements. Thus, the design of each scheduling algorithm must take in consideration the desired parameters of QoS. It must also differentiate between service classes.

Round Robin technique gives equal priority to each queue regardless of the QoS requirements. WRR is designed to take QoS requirements into account, so traffic is serviced in a different way according to the requirements.

WRR assigns a specific weight to each queue. This weight, which is obtained from allocated bandwidth to the queues, allows the users to send a quantum of packets in the same service round. It can then move to the next service round after all queues have been served in the current round based on their weights.

Weight given to the queue is determined based on the QoS requirements at each SS. There are different QoS parameters that determine the weight of each queue: the average size of the packets, the minimum reserved traffic rate for those packets and the maximum sustained traffic rate [18] [24]. In the following, we provide some details about number of different parameters:

Minimum Reserved Traffic Rate (MRTR): This parameter represents the minimum value of rate, expressed in bit per second (bps), which is reserved for service flow. The BS has to fulfill the bandwidth requests of a connection up to its MRTR. So, when the SS requests bandwidth is less than the value of MRTR, the BS retains the rest of the official value to other use. If the value of MRTR equals zero, then no minimum traffic rate is reserved.

Maximum Sustained Traffic Rate (MSTR): This parameter expresses the upper bound of information rate allowed for the SS expressed in bps. MSTR is used to check whether or not the user uses the limit of the allocated bandwidth.

Maximum Latency (ML): This parameter specifies the maximum value of time between the reception of a packet in a SS and the time of forwarding the packet to another SS.

Packet Loss (PL): This parameter specifies the allowed percentage of dropped packets from the queue due to reaching the maximum delay requirements without service.

Fig. 2 illustrates how WRR scheduling technique gives weight to the queues and serve them based on it. When the available bandwidth is 10, and the assigned weights are 0.1, 0.8 and 0.4 to queues number one, two and three respectively, if the value of MRTR equals 5; The BS allocates 10% of all bandwidth to queue number one because the requested bandwidth is less than the MRTR value. Also queue number three will assigned 40% of all bandwidth because it requests bandwidth less than MRTR value. But for queue number two,

the requested bandwidth is larger than MRTR value, so the BS allocates the remaining bandwidth to it which is 50%.

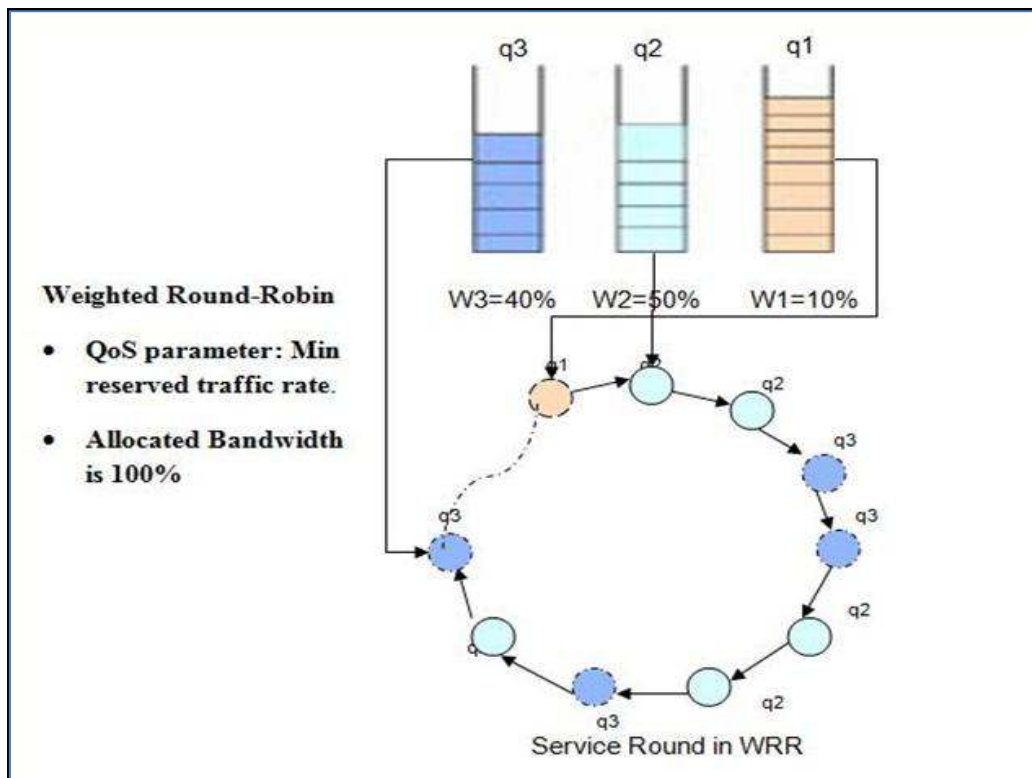


Figure. 2: WRR Scheduling Algorithm

According to WRR scheduler technique, the number of packets served from different queues is based on their priorities. However, the packets which have low priority suffer from delay and delay jitter compared to the packets of high priority. Moreover, as the number of users increases, the delay and jitter increases as well. This causes a problem of starvation which negatively affect the performance of the network overall.

In this paper, we propose a Modified Weighted Round Robin (MWRR) scheduler in order to decrease the average jitter and average end-to-end delay while maintaining or increasing the average throughput.

3.2 The Proposed Scheme

We propose a Modified Weighted Round Robin (MWRR) scheduling algorithm which is supposed to provide an enhancement to the original WRR scheduler explained before. In WRR, a weight for each queue is usually specified according to the priority of the data in each queue using the following equation 1 where sum Queue Priority is sum of all active queues [16]:

$$\text{QueueData}[i].\text{Weight} = (\text{QueueData}[i].\text{priority}) / \text{sum_Queue_Priority} \quad (1)$$

This weight allows the users to send a quantum of packets in the same service round. It can then move to the next service round after all queues have been served in the current

round. It is worth noting that at the same service round, the maximum number of packets of each queue a scheduler can serve represents queue weight counter multiplied by a multiplier factor.

A typical WRR algorithm uses Greatest Common Divisor (GCD) function to evaluate weight counter for WRR packet scheduler. This function returns the greatest common divisor of two numbers. These numbers could be weights of the queues, so the returned value is used to calculate weight counter for WRR packet scheduler. Service round value represents the overall packets that WRR should serve. The scheduler continues to serve packets till all queues become empty and the weight counter for every queue becomes zero. Weight counter for each queue is computed as follows Where WRR Weight Multiplier for service count calculation is usually set to 100 [16]:

$$\text{QueueInfo [i].Weight Counter} = \frac{(\text{QueueData[i].Weight} \times \text{WRR Weight Multiplier})}{\text{GCD Info}} \quad (2)$$

To facilitate the understanding of WRR algorithm and the proposed modification later, we present the following example on how to calculate the weight counter and the service round.

In this example, we will divide all the packets into five classes of service set to 0, 1, 3, 4, and 5. These classes of service will be assigned weights (0.1, 0.3, 0.5, 0.7, and 0.9 respectively) to guarantee specific QoS requirements.

The GCD value for the weights after multiplied by 100 is 10. Then, we will use this value in equation 2 to calculate weight counter for each queue. The result will be 1, 3, 5, 7, and 9 for each queue, respectively.

So, the service round, which is the summation of all queues weight counter, will be 25 packets. This is explained in Fig. 3.

While service round is small and since we need to make these calculations at the start of each service round, the average delay and jitter will be negatively affected. So we suggest increasing the service round to improve the performance to guarantee that the number of packets the scheduler will serve be higher than that in WRR, and the scheduler will invest the time consumed in calculation steps to serve more packets. We will explain this idea by using the same values of the above example.

So, for example, in equation 2, the returned results will be for queue number 1, queue weight counter equals 10, for queue number 2, queue weight counter equals 30, for queue number 3, queue weight counter equals 50, for queue number 4, queue weight counter equals 70, finally for last queue, queue weight counter equals 90. So the new service round will be 250 packets instead of 25 packets.

Here, we will multiply each queue weight counter by constant integer value (r) in order to maximizing the service round. After set of experiments, we noticed that using r=10 achieve the best results for one scenario and r=20 give the best results for another scenario.

In the second scenario, where the network size is relatively smaller, the value of r is higher. This is due to the fact that in small area with short distances between the SSs and BSs, a large exert pressure on BS arises. So, we need to restrict the number of calculations by maximizing the service round to acceptable limit to serve a large number of packets. On average, the results show that for nrtPS Traffic the large values of r give better throughput, but for rtPS traffic the small number of r give better throughput.

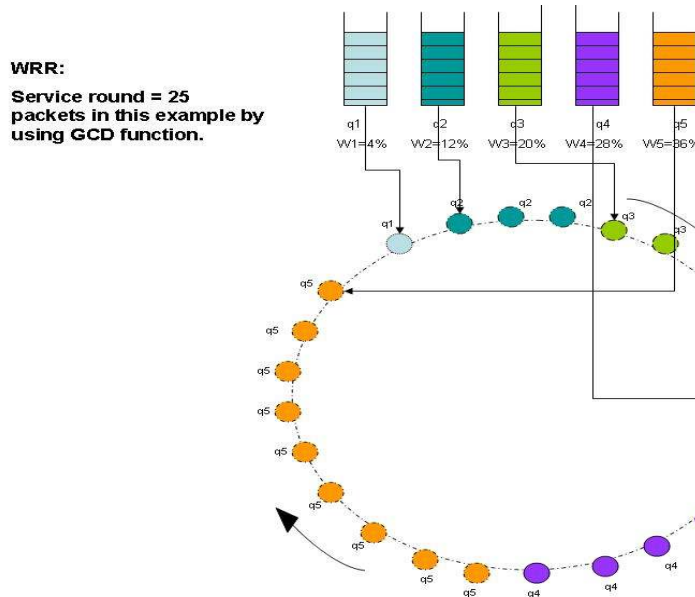


Figure. 3: An Example of WRR Scheduling Algorithm

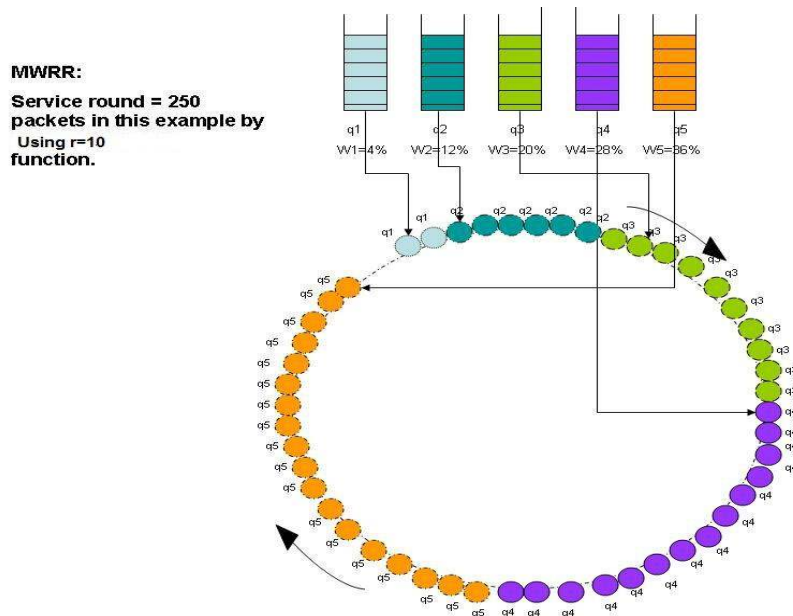


Figure. 4: An Example of MWRR Scheduling Algorithm

In MWRR, retrieve function checks weight assignments before running MWRR scheduling technique. If no weight is assigned to any queue, retrieve function call Auto weight assignment function. Auto weight assignment function gives weight to each active queue based on its priority. This weight allows the users to send a quantum of packets in the

same service round. It can then move to the next service round after all queues have been served in the current round. It is worth noting that at the same service round, the maximum number of packets of each queue a scheduler can serve represents queue weight counter multiplied by a multiplier factor.

WRR algorithm uses GCD function to evaluate weight counter for WRR packet scheduler. This function returns the greatest common divisor of two numbers. These numbers could be weights of the queues, so the returned value is used to calculate weight counter for WRR packet scheduler. Service round value represents the overall packets that WRR should serve. The scheduler continues to serve packets till all queues become empty and the weight counter for every queue becomes zero.

Notice that while service round is small and since we need to make these calculations at the start of each service round; we suggest maximizing the service round to improve the performance to guarantee that the number of packets the scheduler will serve be higher than that in WRR, and the scheduler will invest the time consumed in calculation steps to serve more packets. So, we multiply each queue weight counter by constant integer value (r) in order to maximizing the service round. As mentioned before, r will depend on the network size; the larger the network size the smaller value of r is better and vice versa. The pseudo code shown in Fig. 5 is a modified version of the pseudo code in [16].

```
WRR scheduler retrieve packet.
// this function checks weight assignments before running WRR scheduling technique.
If (user not assigned weight):
// retrieve function call Auto weight assignment function.
// auto weight assignment assigns to each active queue weight.
Auto weight assignment
// calculate sum of all active queues (priority).
For (i=0; i < num queues; i++):
Sum Queue Priority += (QueueData [i]. priority).
1.      End if.
// gives weight to each active queue based on its priority.
2.      Assign weight to each active queue.
3.      For (i=1; i <= num queues; i++):
4.      QueueData [i]. Weight = (QueueData [i]. priority) / sum Queue Priority.
// this function calculates the weight counter value for WRR packet scheduler
5.      Assign Queue Weight Counter ().
6.      For (i=0; i < num queues; i++):
7.      QueueInfo [i]. Weight Counter = (QueueData [i]. weight × WRR Weight Multiplier) / gcd Info * r.
8.      Calculate WRR service Round.
9.      For (i=0; i < num queues; i++):
10.     Service Round += QueueInfo [i]. Weight Counter.
```

Figure. 5: MWRR Algorithm

4. Simulation Methodology and Results

This section discusses the simulation environment and the results obtained in order to

assess the performance of the MWRR scheduling algorithm.

4.1 Simulation Environment

This section discuss the details of the simulation environment we have followed to evaluate the performance of our proposed scheduling algorithm. This involves overview of QualNet simulator, simulation test cases, simulation parameters as well as the performance metrics.

1) Overview of QualNet Simulator

QualNet is a collection of inclusive tools for design large wired and wireless networks [16]. It comprises simulation and emulation to evaluate the performance of networks and improve their design, operation and management. QualNet could be used to evaluate and analyze the networks by:

- Modeling new protocol.
- Improve new and existing models.
- Design large wired and wireless networks using pre-configured or the designed models by users.
- Analyze network performance and conduct their optimum.

The Important features of QualNet can be summarized as follows:

- **Speed:** QualNet can support real-time speed beside the simulation time to enable network emulation and hardware-in-the-loop modeling. Faster speed enables the developers to run multiple analyses in a short time.
- **Scalability:** QualNet enables high level of fidelity for design large networks that contain a thousand of nodes. It takes the advantages of latest hardware and parallel computing technology to run on cluster, multi core, and multi-processor systems.
- **Model Fidelity:** QualNet tend to use advanced models to design the wireless environment and enable more accurate modeling of real-world networks.
- **Portability:** QualNet runs on different platforms such as Windows XP, Mac OS X, and Linux operating systems, distributed and cluster parallel architectures, and both 32- and 64-bit computing platforms.
- **Extensibility:** QualNet can interconnect to other hardware and software applications, such as real networks to enhance the Importance of the network model.

2) Simulation runs

Many simulation runs has been conducted. Each simulation run simulates 20 minutes of real operation of the network which took around 1 minute as simulation time. This due to high traffic rate used to enable us measure the performance of the algorithm under high traffic volume.

Each experiment is repeated ten times with different random seeds, and the average value of their results is taken to ensure integrity. We calculated 95% confidence error margins in all of the simulation experiments. The tables of error margins at 95% confidence for all experiments are illustrated in details in Appendix A.

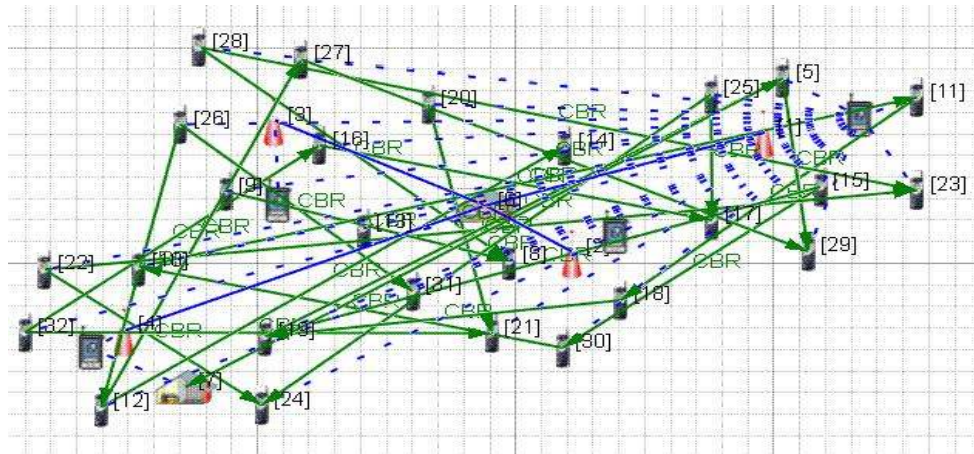
We considered two scenarios in our simulation; one scenario simulates large network area and one for relatively smaller network area. For the first scenario we have 30 nodes randomly located in a space of 30000 m × 40000 m and for the second scenario we have 3000 m × 8000 m. The IEEE 802.16 is used as a MAC layer communication protocol.

In the application layer, the nodes communicate using Constant Bit Rate (CBR) traffic generators over UDP with random source/destination pairs. The CBR is a model for sending data packets from sources to destinations. If the value of the items to be sent equals 0, CBR will run until the end of the time value or the end of the simulation, depending on which one occurs first. If the end time equals 0, CBR will be running until all the items are sent or until we reach the end of the simulation depending on which occurs first.

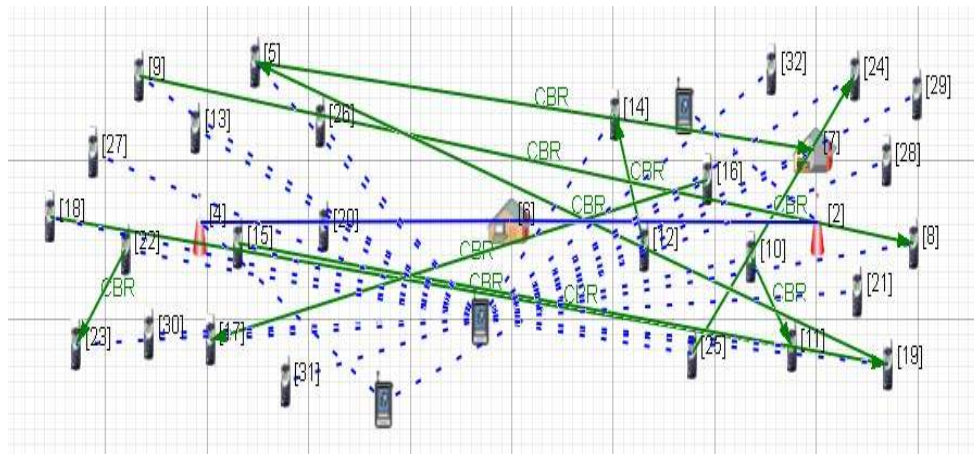
Table 3. Simulation Parameters [30] [25] [26] [27] [29]

Parameter	Value
Antenna Highest	Scenario 1: 10 m (BSs and SSs) Scenario 2: 10 m for BSs and 1.5 m for SSs
Antenna Model	Omnidirectional
Channel Frequency	2.4 GHz
Interval between Packets	1 milli-seconds
Item Size	512 bytes
Link Bandwidth (between base stations)	50 Mbps
Link Bandwidth (between subscriber stations)	10 Mbps
Node Placement	Random
Number of CBR	Scenario 1: 10, 25, 50 and 75 CBR Scenario 2: 25, 50 and 75 CBR
Precedence Values	0, 1, 2, 3, 4, 5, 6, and 7
Radio Type	802.16 Radio
Scheduling Algorithms Evaluated	SP, WFQ, WRR, MWRR
Simulation Grid Size	Scenario 1: 30000 m × 40000 m Scenario 2: 3000 m × 8000 m
Simulation Time	60 seconds
Simulator	QualNet 5.0.2
Traffic Type	CBR
Transmission Power	20 GHz

To comprehensively measure the performance of our algorithms, we studied the effect of the number of traffic generators by assuming 10, 25, 50, and 75 CBR sources. The overall system traffic load increases with the number of sources and the CBR value. Table 3 illustrates the simulation parameters depending on the related work.



(a)



(b)

Figure 6: QualNet Simulation Screenshots

3) Performance Metrics

For the purpose of evaluating our proposed scheduling technique and comparing its performance with the performance of the WRR, WFQ, and SP scheduling techniques, we used some common performance metrics: Average Jitter, Average end to end Delay, Packet Loss, and Average Throughput.

- **Average Jitter:** This performance metric represents the inter-packet arrival time to the receiver and is required to be reasonably stable by the real-time applications [30] [31].
- **Average End-to-End Delay:** This performance metric represents the average delay between the time when the data packet was originated at the source node and the time it reaches the destination node. The end to end delay metric includes delays due to route discovery, queuing and transmissions at the MAC level [17] [32] [33].
- **Average Throughput:** Average throughput represents the amount of data transmitted by user per unit time. The value is expressed in Kbps [31] [20].

- Packet Loss: This parameter specifies the allowed percentage of dropped packets from the queue due to reaching the maximum delay requirements without service.

In Fig. 6 above, we illustrate the topology of the scenarios used in the simulation.

4.2 Simulation Results

In this section we present the results of different simulation experiments performed to evaluate the proposed scheduler MWRR. The results are presented for scenario 1 and 2 discussed above. The performance metrics discussed are the end to end delay, average Jitter, and average throughput.

The experiments compare four scheduling algorithms: SP, WFQ, WRR and MWRR. The performance metrics are presented for different traffic loads.

1) The First Scenario

Fig. 7 represents the average end to end delay for different number of connections for the four scheduling algorithms. The figure shows that our MWRR scheduler performs the best at different number of CBR connections except for the first point of 10 connections. At the first point the number of connections is low and the traffic is considered to be light traffic and no difference between the different scheduling algorithms is noticed. After that, at points 25, 50 and 75, MWRR performed the best because in MWRR we have increased the service round which in turn reduced the number of calculations, and thus the time saved is invested to serve lined up packets reducing the delay. Notice that the delay at 75 connections is less than the delay at 50 connections because the drop packet rate at 75 is much higher and thus there is higher chance for packets to be served faster than before and thus less delay is recorded.

Fig. 8 and Fig. 9 represents the details of the results presented above divided into non real time traffic and real time traffic, respectively. Same behavior can be noticed in these figures. In Fig. 8, where around third of the connections are non-real time, same behavior are noticed for the four algorithms.

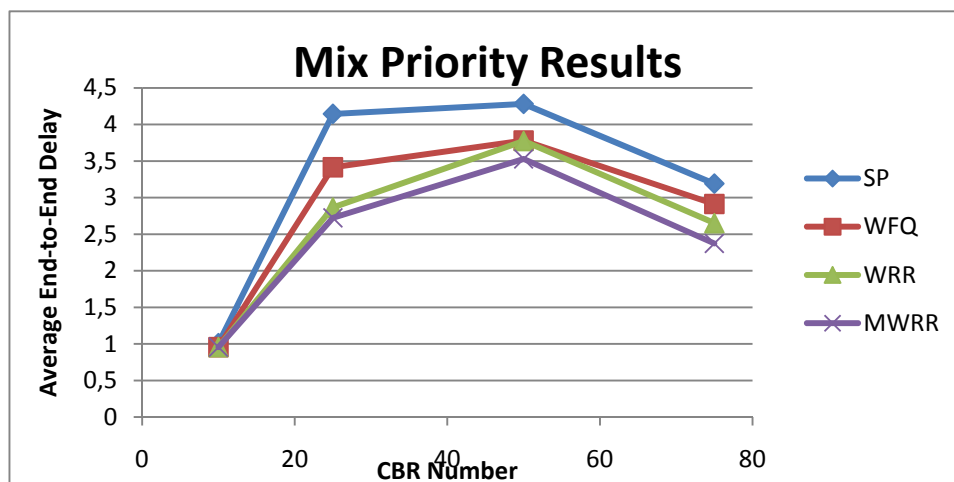


Figure. 7: End-to-End Delay of Mix Priority Classes and Different Numbers of Traffic Generators

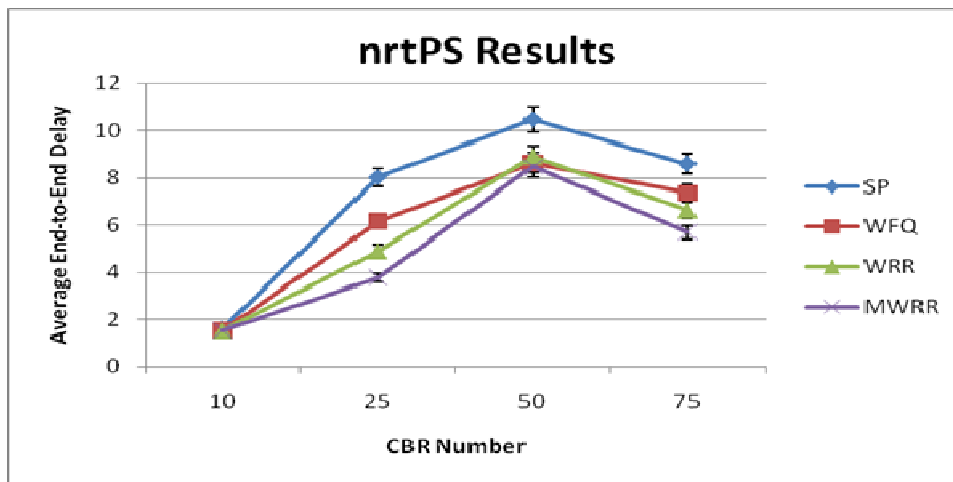


Figure. 8: End-to-End Delay of nrtPS Class and Different Numbers of Traffic Generators

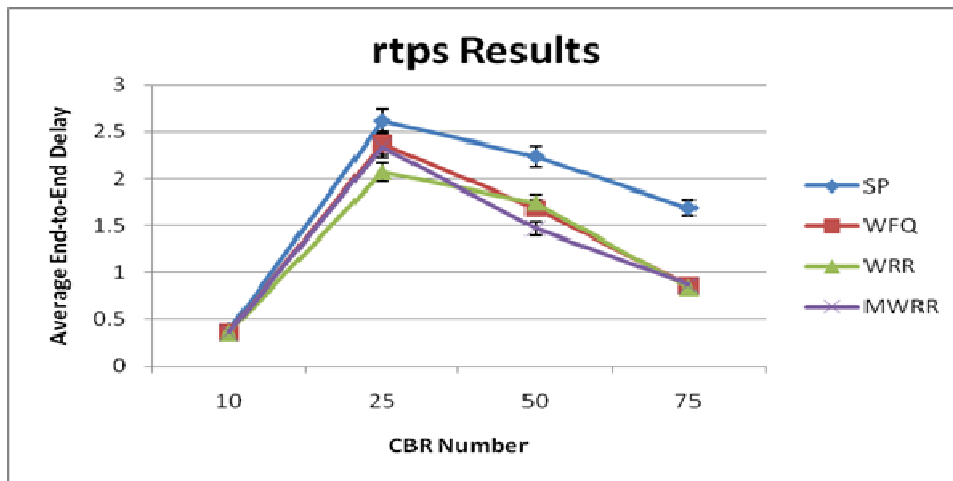


Figure. 9: End-to-End Delay of rtPS Class and Different Numbers of Traffic Generators

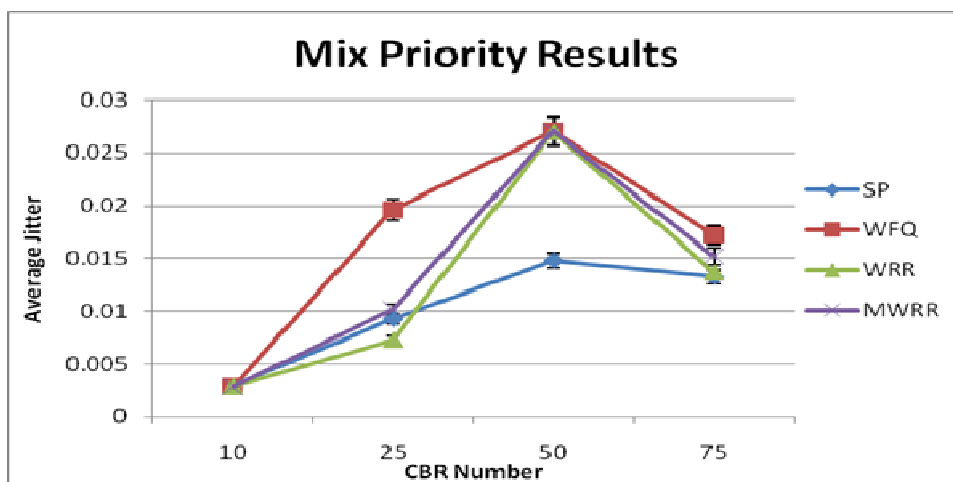


Figure. 10: Average Jitter of Mix Priority Classes and Different Numbers of Traffic Generators

In Fig. 9, where around two thirds of the connections are real time, the delay is

increasing from 10 to 25 connections; however it is decreasing from 25 to 50 and from 50 to 75. This can be explained because the real time traffic are dropped earlier than other traffic classes which allows more time and less delay for the later traffic.

Fig. 10 represents the average jitter for different number of connections for the four scheduling algorithms mentioned earlier. The figure shows that our MWRR scheduler does not achieve any enhancement in terms of reducing the average jitter for all CBR connections. This is due to the fact that our schedule algorithm tends to serve more packets from nrtPS class, which affects serving the remainder packets in mix priority class.

Fig. 11 represents the average throughput for different number of connections. The figure shows that MWRR scheduler outperforms each of WRR and WFQ schedulers at different number of CBR connections. This is due to the fact that in MWRR we have reduced the calculations and invested the wasted time to serve more packets than those in WRR. The results for SP is better than the three other algorithms because that around two third of the traffic is real time and this is favored by the SP which made the overall average is better.

Fig. 12 and Fig. 13 represent the details of the results presented above divided into nrtPS traffic and rtPS traffic, respectively. In Fig. 12, where around third of the connections are nrtPS, the results show that MWRR, WRR and WFQ have larger throughput value than SP for all number of CBR applications, because the SP scheduler tends to serve rtPS traffic which affects throughput nrtPS traffic negatively. MWRR scheduler becomes evidently the most superior scheduler than WRR, WFQ and SP. The figure shows that the MWRR is more efficient than WRR in throughput packets of nrtPS traffic. The percentage of MWRR scheduler enhancement reaches 16% on average. In Fig. 13, where around two thirds of the connections are rtPS, the throughput is best for SP as expected.

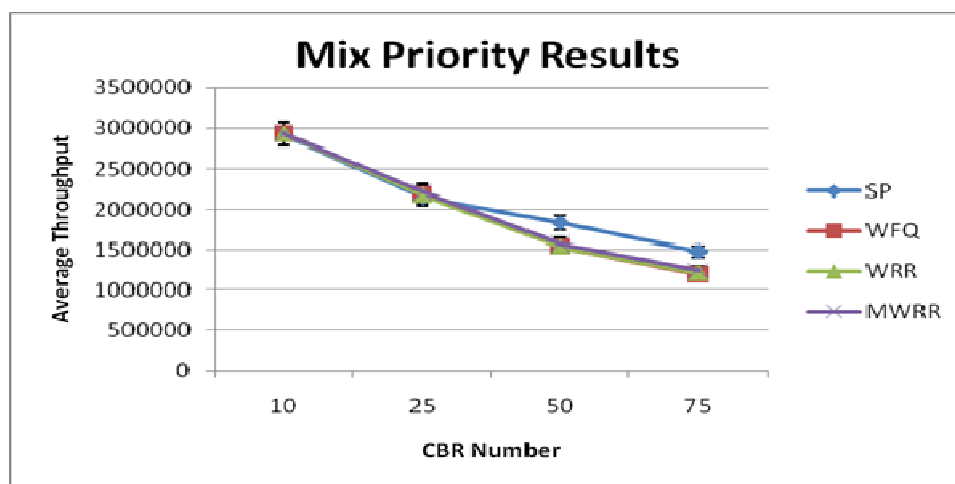


Figure. 11: Average Throughput of Mix Priority Classes and Different Numbers of Traffic Generators

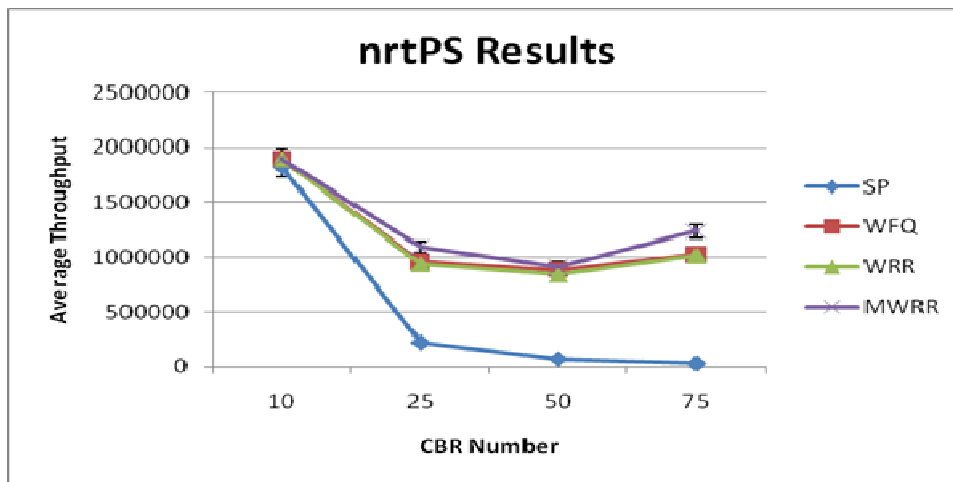


Figure. 12: Average Throughput of nrtPS Class and Different Numbers of Traffic Generators

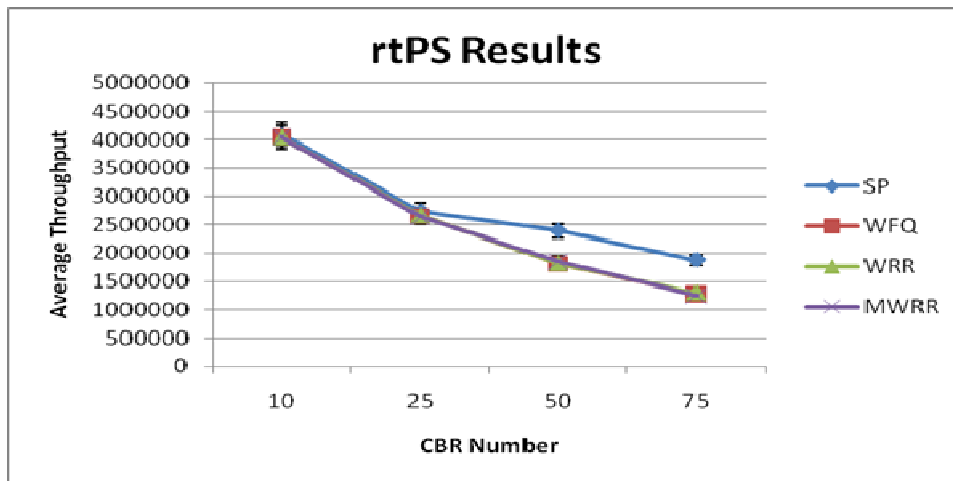


Figure. 13: Average Throughput of rtPS Class and Different Numbers of Traffic Generators

Fig. 14 represents the packet loss for different CBR generations. In general packet loss is high in this scenario because the traffic rate is high and the network size is large. In the next subsection we show smaller network with lower packet rates at which the packet loss is much lower. The results in Fig. 14 show that MWRR outperforms both of WRR and WFQ schedulers at different number of CBR connections. This is due to the fact that in MWRR we have reduced the calculations and invested the wasted time to serve a large number of packets than those in WRR and reduced the dropped packets. With regards to SP, The result is better than the three other algorithms because it tends to serve real time traffic which made the overall packet loss is better.

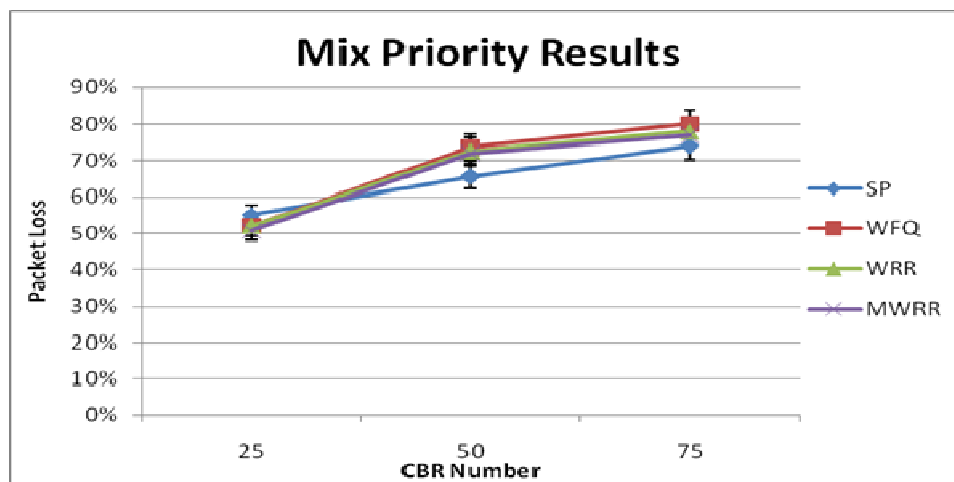


Figure. 14: Packet Loss of Mix Priority Classes and Different Numbers of Traffic Generators

2) The Second Scenario

For the second scenario, we discuss the four performance metrics for the mixed priority traffic only. The detailed performance for high and low priority traffic follow same trend of the first scenario and thus is not discussed here.

Fig. 15 represents the average end to end delay for different number of connections for the four scheduling algorithms. The figure shows that our MWRR scheduler performs the best for most number of CBR connections. At the first point the WRR gives lower end to end delay and performs all schedulers. After that, at points 50 and 75, MWRR performed the best because in MWRR we have increased the service round enough which in turn reduced the number of calculations, and thus the time saved is invested to serve lined up packets reducing the delay.

Fig. 16 represents the average Jitter for different number of connections for the four scheduling algorithms. The figure shows the same results of end to end delay, and the same reasons justify both of average jitter and end to end delay results.

Fig. 17 represents the average throughput for different number of connections. The figure shows that MWRR scheduler outperforms WRR scheduler at different number of CBR connections. This is due to the fact that in MWRR we have reduced the calculations and invested the wasted time to serve more packets than those in WRR.

Fig. 18 represents the packet loss for different CBR connections. The results show that MWRR outperforms both of SP and WFQ schedulers at 25 and 50 traffic generators, also it outperforms WRR at 50 traffic generators. But at 75 traffic generators MWRR does not give any enhancement, and all schedulers outperform it. This is due to the fact that MWRR tends to serve a large number of packets than those in WRR for dense traffic generators which affects the packet loss negatively, but for acceptable traffic the packet loss is the best in MWRR.

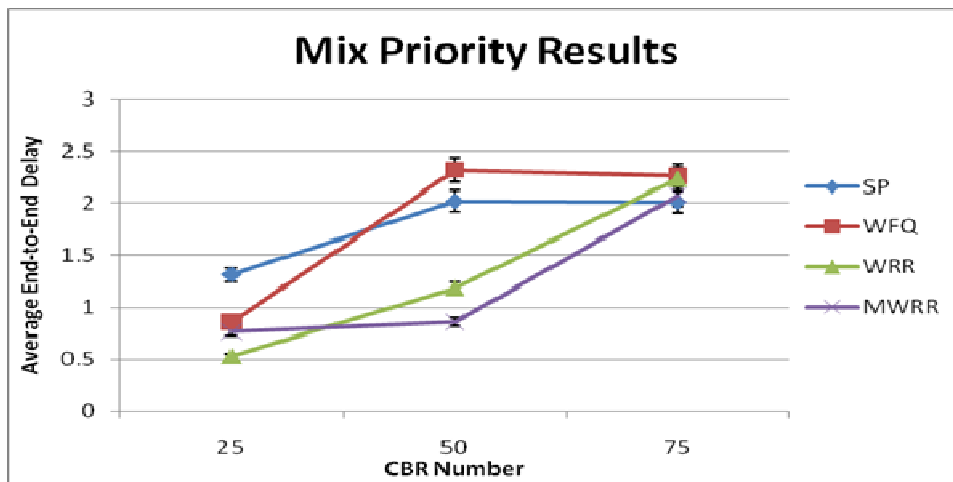


Figure. 15: End-to-End Delay of Mix Priority Classes and Different Numbers of Traffic Generators

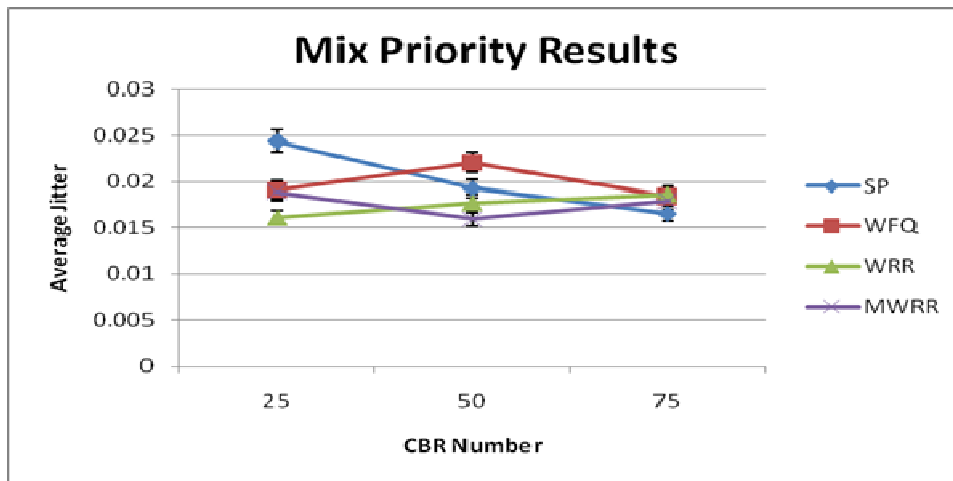


Figure. 16: Average Jitter of Mix Priority Classes and Different Numbers of Traffic Generators

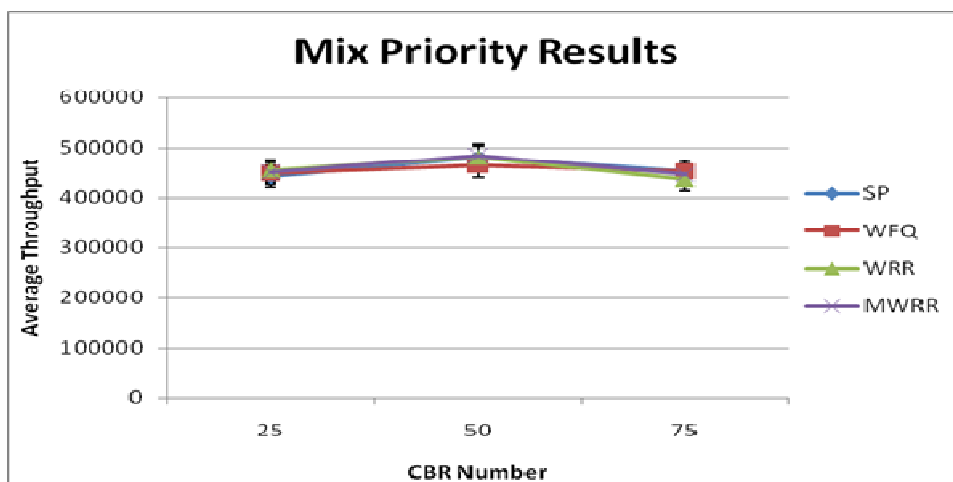


Figure. 17: Average Throughput of Mix Priority Classes and Different Numbers of Traffic Generators

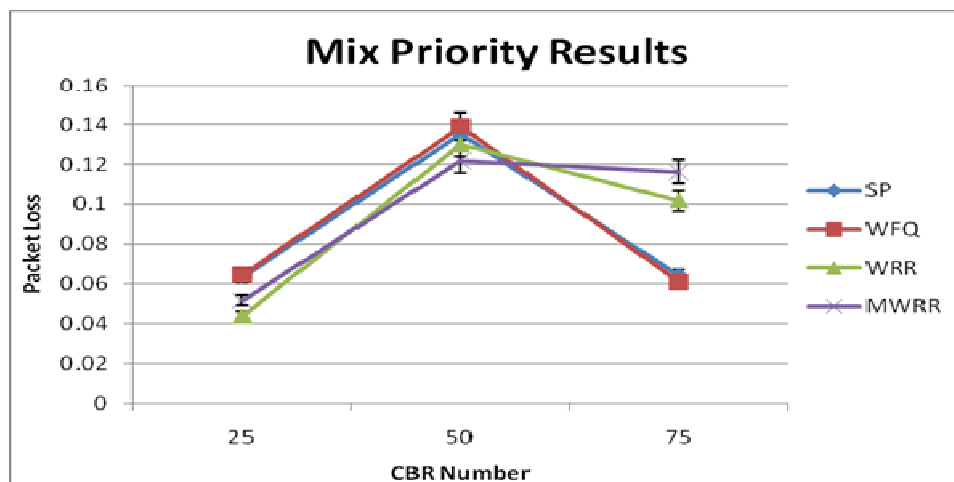


Figure. 18: Packet Loss of Mix Priority Classes and Different Numbers of Traffic Generators

4.3 Discussion and Analysis

In this section we discuss the results above according to the percentage of improvement achieved by MWRR over WRR for the first scenario.

In Table 4, Table 5 and Table 6, we show the percentage of improvement achieved in each case of mixed traffic, non-real time and real time traffic, respectively. In Table 4, we can see the average end to end delay and throughput improvements for MWRR and WRR schedulers under effectiveness of mix Priority traffics. With regards to end to end delay, it is clear from the table that MWRR outperforms WRR by 5%, 7% and 11% at 25, 50 and 75 connections, respectively. Also MWRR outperforms WRR with regards to average throughput by 2% on average.

However, in Table 5, the results showed that the average end to end delay and throughput are improved when adding more connections to each SS, With regards to end to end delay, the results show that MWRR outperforms WRR by 23%, 5% and 14% at 25, 50 and 75 connections respectively. For throughput, MWRR outperforms WRR by 16%, 9% and 22% at 25, 50 and 75 connections respectively. These results are better than in results showed in Table 4, this due to that when we maximizing the service round, the nrtPS have the higher chance to be served faster particularly when the number of traffic is increasing in contrast to rtPS.

In Table 6, for rtPS traffic, the results showed that the average end to end delay and throughput are increasing on average. Notice that for the above mentioned metrics at 50 connections, MWRR outperforms WRR which is affected by the number of drop packets.

At lower traffic rates, we see in Table 7 and Table 8, the average end to end delay for 100 and 10 packets per second traffic rates respectively. The average improvements when the rate equals 100 packets per second are similar to average improvements in discussed results at 1000 packets per second traffic rate. But by decreasing the rate of the packets by 10 packets

per second, the improvements become equals zero, in which the MWRR works as WRR in serving packets because the size of service round becomes similar.

Table 4. Improvement Percentages of MWRR over WRR for Mix Priority Classes

Modified Weighted Round Robin (MWRR)		
	End-to-End Delay	Average Throughput
10	0%	0%
25	5%	2%
50	7%	2%
75	11%	3%

Table 5. Improvement Percentages of MWRR over WRR for nrtPS Classes

Modified Weighted Round Robin (MWRR)		
	End-to-End Delay	Average Throughput
10	0%	0%
25	23%	16%
50	5%	9%
75	14%	22%

Table 6. Improvement Percentages of MWRR over WRR for rtPS

Modified Weighted Round Robin (MWRR)		
	End-to-End Delay	Average Throughput
10	0%	0%
25	-13%	0%
50	16%	2%
75	-3%	-4%

Table 7. Results and Improvement Percentages of MWRR over WRR for End-to-End Delay Metric with 100 Packet per second Traffic Rate

End-to-End Delay for 100 Packet/s Traffic Rate			
	WRR	MWRR	Improvement
nrtPS	3.580108176	3.216308347	10%
rtPS	0.439168298	0.447868299	-2%
Mix	1.237359663	1.151557382	7%

Table 8. Results and Improvement Percentages of MWRR over WRR for End-to-End Delay Metric with 10 Packet per second Traffic Rate

End-to-End Delay for 10 Packet/s Traffic Rate			
	WRR	MWRR	Improvement
nrtPS	0.151177104	0.151177104	0%
rtPS	0.101779355	0.101779355	0%
Mix	0.115430844	0.115430844	0%

5. Conclusions and Future Work

In this paper, we present the design and simulation of MWRR scheduling algorithm for WiMAX networks and also provide performance measurements using the QualNet 5.0.2 simulator. The major contributions of our protocol are:

- The MWRR is reliable scheduling algorithm that avoids the problems in WRR which causes a starvation and unnecessary delay for lower class of services.
- The main goal of our protocols is to reduce the average delay and increasing the average throughput especially to the lower classes by increasing the size of service round that in WRR. This is achieved by multiply each queue weight counter by 10. By applying this changes, our scheduler introduce a very low end to end delay compared to WRR, WFQ and SP schedulers, yet, it proves the efficiency of throughput large numbers of packets of the non-real time and mix priority traffics.
- The simulation experiments show that our proposed algorithms, MWRR significantly outperforms the WRR algorithm in terms of reducing end to end delay by 14% for non-real time traffic, and by 8% for mix priority traffic.
- In addition, MWRR substantially outperforms WRR in terms of average throughput, where MWRR outperforms WRR by 16% for non-real time traffic and by 2% for mix priority traffic. Regarding real time traffic, the experiments show that our scheduler does not achieve real enhancements in term of average delay and throughput.
- In terms of average jitter, we note that when the service round becomes large; the scheduler should serves a large number of packets from each queue. This causes the need for extra jitter time especially for real time traffic.

This paper finds a solution to the challenging problem of scheduling in WiMAX networks. However, there are still several research points that can be investigated further in order to extend the basic approach done in this paper. This section provides some suggestions for future work as follows:

- It would be interesting to work on a full mobility suite. And show how the system will work when put in a scenario when SSs are entering and leaving the network on a regular basis.
- Using Variable Bit Rate (VBR) application alters of CBR representing more realistic scenarios.
- Studying the effectiveness of network density and add more schedulers to compare with them.

References

- [1] Bo Han, Weijia Jia, Lidong Lin, “Performance evaluation of scheduling in IEEE 802.16 based wireless mesh networks”. *Computer Communications*, Vol. 30, Issue 4. Pp 782-792, <http://dx.doi.org/10.1016/j.comcom.2006.10.001>.
- [2] Koon Hoo Teo, Zhifeng Tao, and Jinyun Zhang, “The Mobile Broadband WiMAX Standard”. *IEEE Signal Processing Magazine*, Vol. 24, Issue 5. Pp 144- 148. Sept. 2007. <http://dx.doi.org/10.1109/MSP.2007.904740>
- [3] Michael Carlberg Lax and Annelie Dammander. “WiMAX - A Study of Mobility and a MAC-layer Implementation in GloMoSim”. Umea University Department of Computing Science; Apr. 2006; 1-99.
- [4] K.R.Santhi and G. Senthil Kumaran. “Migration to 4G: Mobile IP based Solutions”, *Advanced International Conference on Telecommunications and International Conference on Internet and Web Applications and Services*; Feb. 19-25, 2006. <http://dx.doi.org/10.1109/AICT-ICIW.2006.128>
- [5] Min Cao, Vivek Raghunathan, and P.R. Kumar. “A Tractable Algorithm for Fair and Efficient Uplink Scheduling of Multi-hop WiMax Mesh Networks. *Wireless Mesh Networks*”. 2nd IEEE Workshop; Sept. 25-28, 2006; 93–100. <http://dx.doi.org/10.1109/WIMESH.2006.288622>
- [6] Konark kelayia. “Routing & Scheduling Algorithm of IEEE 802.16 Mesh Backhaul Network for Radio Resource Management (RRM)”. *Mobile and Pervasive Computing*; Oct. 2008; 175-179.
- [7] Application Layer Threats to WiMAX Technology, [Online] [accessed March 2011]. Available from URL www.freewimaxinfo.com/application-layer-threat.html.
- [8] Fanchun Jin, Arora Amrinder, Jinho Hwang, and Hyeong-Ah Choi. “Routing and Packet Scheduling for Throughput Maximization in IEEE 802.16 Mesh Networks”. *Broadnets 2007, Fourth International Conference*; Sept. 2007; 574–582. <http://dx.doi.org/10.1109/BROADNETS.2007.4550485>
- [9] Hung-Yu Wei, Samrat Ganguly, Rauf Izmailov, and Zygmunt J. Haas. “Interference-Aware IEEE 802.16 WiMax Mesh Networks”. *Proceedings of 61st IEEE Vehicular Technology Conference*; spring 2005; 5: 3102 – 3106. <http://dx.doi.org/10.1109/VETECS.2005.1543918>
- [10] Shenghai Liu, Suili Feng, Wu Ye, and Hongcheng Zhuang. “Slot Allocation Algorithms in Centralized Scheduling Scheme for IEEE 802.16 Based Wireless Mesh Networks”. *Computer Communications*; 2009; 32: 943–953. <http://dx.doi.org/10.1016/j.comcom.2008.12.020>
- [11] Marie-Paule Gakuba, Mjumo Mzyece, and Anish Kurien. “An Enhanced Scheduler for the Guarantee of QoS Multimedia Services in WiMAX Networks”. *French South African Technical Institute in Electronics (F’SATIE) in Tshwane University of Technology (TUT)*;

July 2009; 1-6.

[12] Chen-feng wu. “Real-time Scheduling for Multimedia Services in IEEE 802.16 Wireless Metropolitan Area Network”. *Information Technology Journal*; 2010; 9: 1053-1067. <http://dx.doi.org/10.3923/itj.2010.1053.1067>

[13] Fen Hou, Lin X. Cai, James She, Pin-Han Ho, Xuemin (Sherman) Shen, and Junshan Zhang. “Cooperative Multicast Scheduling Scheme for IPTV Service over IEEE 802.16 Networks”. *IEEE International Conference*; May 2008; 2566 - 2570. <http://dx.doi.org/10.1109/ICC.2008.486>

[14] Y. Cao and V. Li. “Scheduling Algorithms in Broadband Wireless Networks”. *Proceedings of The IEEE*; Jan. 2001; 89: 76-87. <http://dx.doi.org/10.1109/5.904507>

[15] B.Skrikar. “Packet Scheduling Algorithms to Support QoS in Networks”. Master’s Paper, Indian Institute of Technology; Oct. 1999; 1-71.

[16] Qualnet 5.0.2 Advanced Wireless Model Library, Scalable Network Technologies, Los Angeles; 2010.

[17] Mikael Gidlund and Gang Wang. “Uplink Scheduling Algorithms for QoS Support in Broadband Wireless Access Networks. *Journal of Communications*”. *Journal of Communications*, Vol 4, No 2, Mar. 2009; Pp: 133-142. <http://dx.doi.org/10.4304/jcm.4.2.133-142>

[18] Pratik Dhrona, Najah Abu Ali, and Hossam Hassanein. “A Performance Study of Scheduling Algorithms in Point-to-Multipoint WiMAX Networks”. *LCN 2008. 33rd IEEE Conference*; Oct. 2008; pp: 843-850. <http://dx.doi.org/10.1109/LCN.2008.4664291>

[19] Sung-Min Oh, Sunghyun Cho, Jae-Hyun Kim, and Jonghyung Kwun. “VoIP Scheduling Algorithm for AMR Speech Codec in IEEE 802.16e/m System”. *IEEE Communications Letters*; May 2008; 12: 374 - 376. <http://dx.doi.org/10.1109/LCOMM.2008.072098>

[20] Hongfei Du, Jiangchuan Liu, and Jie Liang. “Downlink Scheduling for Multimedia Multicast/Broadcast over Mobile WsMAX: Connection-oriented Multi-state Adaptation”. *IEEE Wireless Communications*. Aug. 2009; Vol. 16, Issue: 4. Pp: 72-79. <http://dx.doi.org/10.1109/MWC.2009.5281258>

[21] Clifford E. Cummings and Peter Alfke. “Simulation and Synpaper Techniques for Asynchronous FIFO Design with Asynchronous Pointer Comparisons”. *SNUG San Jose*; 2002; 1-18.

[22] Kert Mezger and David W. Petr. “Bounded Delay for Weighted Round Robin”. *Technical Report TISL*; May 1995; 1-35.

[23] Mahasweta Sarkar and Harpreet Sachdeva. “A QoS aware Packet Scheduling Scheme for WiMAX”. *Proceedings of IAENG Conference on World Congress on Engineering and Computer Science*; Oct. 2009; 1: 1-6.

[24] M. Ahtsham Sarwar Khan, Ahsan Razza Sattar, Tasleem Mustafa, and Shehbaz Ahmad.

“Performance Evaluation and Enhancement of Uplink Scheduling Algorithms in Point to Multipoint Wimax Networks”. *European Journal of Scientific Research*; 2010; 42: 477-492.

[25] Nararat Ruangchaijatupont, Liping Wangt, and Yusheng Jit. “A Study on the Performance of Scheduling Schemes for Broadband Wireless Access Networks”. *International Symposium on Communications and Information Technologies, 2006. ISCIT '06. Sept. 2006; 1008-1012*. <http://dx.doi.org/10.1109/ISCIT.2006.339929>

[26] Matthew Andrews and Bell Laboratories. “Probabilistic End-to-End Delay Bounds for Earliest Deadline First Scheduling”. *Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies; Mar. 2000; 2: 603-612*. <http://dx.doi.org/10.1109/INFCOM.2000.832234>

[27] Ara Khil and Seungryoul Maeng. “Scheduling of Multimedia Traffic for Continuous Media in Packet-Switched Networks”. *Multimedia Computing and Networking; Jan. 1997; 3020: 29-40*. <http://dx.doi.org/10.1117/12.264305>

[28] Masood Khosroshahy and Vivien Nguyen. “A Study of WiMax QoS Mechanisms”. *RMOB Project, Telecom Paris; Apr. 2006; 1-23*.

[29] R. Ghazizadeh, P. Fan and Y. Pan. “A Two Layer Channel Aware A Scheduling Algorithm for IEEE 802.16 Broadband Wireless Access System”. *Journal of Applied Sciences; 2009; 9: 449-458*. <http://dx.doi.org/10.3923/jas.2009.449.458>

[30] Jianfeng Chen, Wenhua Jiao, and Qian Guo. “An Integrated QoS Control Architecture for IEEE 802.16 Broadband Wireless Access Systems”. *Global Telecommunications conference '05. IEEE; Dec. 2005; 6: 1-6*.

[31] Aymen Belghith and Loutfi Nuaymi. “Design and Implementation of a QoS-included WiMAX Module for NS-2 Simulator”. *Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops; Mar. 2008; 1-8*.

[32] Yi-Ting Mai, Chun-Chuan Yang, and Cheng-Jung Wen. “A New Cut-Through Mechanism in IEEE 802.16 Mesh Networks”. *Proceedings of World Academy of Science, Engineering and Technology; June 2009; 41: 98-104*.

Copyright Disclaimer

Copyright reserved by the author(s).

This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/3.0/>).