

## An Efficient Qos Scheduling Algorithm for Ieee 802.16 Networks

K.S.S. Prasad

Department of Electrical and Electronics,  
Bharath University, Chennai, India

**Abstract:** Worldwide Interoperability for Microwave Access (WiMAX) is one of the most familiar broadband wireless access technologies that support multimedia transmission. IEEE802.16 Medium Access Control (MAC) covers a large area for bandwidth allocation and Quality of Service (QoS) mechanisms for various types of applications. Nevertheless, the standard lacks a MAC scheduling algorithm that has a multi-dimensional objective of satisfying QoS requirements of the users, maximizing channel utilization while ensuring fairness among users. So a novel Priority based Scheduling Algorithm using Fuzzy logic that addresses these aspects are proposed. The initial results show that maximum channel utilization is achieved with a negligible increment in processing time while keeping the priority intact. and fairness.

**Key words:** WiMAX • Fuzzy • Priority • Neural Networks • Scheduling Algorithm

### INTRODUCTION

There are basically three different ways to access Internet they are Broadband Access, Wi-Fi Access and Dial up access. WiMAX(IEEE802.16)-Worldwide Interoperability for Microwave Access by the WiMAX Forum is defined as “a standards-based technology enabling the delivery of last mile wireless broadband access as an alternative to cable and DSL”. This provides access up to 30 miles (50km) for fixed stations and 3-10miles (5-15km) for mobile stations adopting OFDMA technology and covering 10-66GHz spectrum range. There are two types of scheduling followed in IEEE 802.16 networks [1]. They are Uplink Scheduling (UL) which is transmission from Subscriber Stations (SS) to Base Station (BS) and Downlink Scheduling (DL) which is the transmission from BS to SS. WiMAX architecture consists of one BS and one or more SS. It supports adaptive modulation and coding, such that the distance between the subscriber station and the base station determines the type of modulation to be used.

A high level of QoS and scheduling support is one of the interesting features of the WiMAX standard. Quality of service is an architecture that treats packets differently by providing guaranteed services to the end user. QoS is enabled by the bandwidth request and grant mechanism

between various subscriber stations and base stations. BS can communicate in either point to multipoint (PMP) or mesh mode [2]. In PMP, the BS controls all the communication within its' own coverage. In mesh mode, the SS can form a mesh connection without the need of BS. The five scheduling services supported in WiMAX are; Unsolicited Grant Service (UGS), real time polling service (rtPs), extended real time polling service (ertPs), non-real time polling service (nrtPs) and best effort (BE) to provide the service-class classification for video, audio and data services. The service scheduler provides scheduling for different classes of services for a single user. This would mean meeting SLA requirements at the user level.

### The Five Service Flows Are Explained Below:

- Unsolicited grant service (UGS): This comes under real time service with CBR and can provide guaranteed data throughput and latency.
- Real-time polling service (rtPS): The minimum reserved rate and the latency are guaranteed in this application and comes under real time with VBR.
- Enhanced Real-time polling service (ertPS): It especially concentrates on real time Voice over IP. It is similar to UGS but have a silent suppression in it.

- Non-real-time polling service (nrtPS): This service tolerates delay while streaming variable-sized data packets.
- Best effort (BE): The channel access mechanism of this service is based on the contention and provides no QoS guarantees.

The IEEE802.16 standard defines five scheduling classes, but it does not provide any information about the standard algorithms that should be used to provide QoS to the service flows (SFs). Even though there are lots of conventional scheduling algorithms they are not meeting all the required QoS parameters [3]. The performance affecting parameters like fairness, bandwidth allocation, throughput and latency are studied and found out that none of the algorithms perform effectively for both fairness and maximum bandwidth utilization simultaneously [2]. So a decision has been made to optimize those two parameters by using an algorithm based on Artificial Intelligence (AI) [4]. This paper is organized as follows: Section 2 describes the related work. Section 3 and 4 explain the proposed scheme. Section 5 shows the performance of WiMAX using the newly proposed scheduling algorithm and the conclusions in section 6 [5].

**Previous Work:** Several research works have been conducted in order to provide QoS in IEEE 802.16 networks. In [6], the authors propose a hybrid of Earliest Due Date (EDD) and Weighted Fair Queue (WFQ). In EDD, all the arriving packets get a deadline stamp and are scheduled according to the increasing order of deadlines. The algorithm intends to serve the real time traffic first and only if real time buffer is empty will they consider BE traffic. This will certainly lead to starvation. In [7], the authors consider two types of queues. The first type is used to schedule data grants for UGS and allocate request opportunities for rtPS and nrtPS. These grants are scheduled in a first in first out (FIFO) manner. Once the first queue type has been served, the scheduler will consider the second type leading to scarcity. The authors in [8] propose an architecture consisting of three schedulers. The first scheduler is concerned with UGS, rtPS and ertPS flows. The second scheduler is concerned with flows requiring a minimum bandwidth mainly nrtPS. The third scheduler is used for BE traffic the third scheduler comes into picture only when the first two schedulers have become free. In [9], the suggested uplink scheduling algorithm is Weighted Round Robin (WRR) with GPSS grant mode. The duration of contention slots and uplink data slots are dynamically distributed

according to bandwidth requirements. The authors did not comment on what weights to use for WRR scheduling or BS downlink scheduling. In [10], the authors suggest downlink bandwidth allocation algorithms based on flow type and strict priority from highest to lowest-UGS, rtPS, ertPS, nrtPS and BE. Here an Opportunistic fair scheduling was used. Here BE traffic is served whenever an opportunity is available, but for most of the time BE starves for bandwidth.

### Setting Priority Using Fuzzy

**Fuzzy Logic Controller:** Fuzzy logic was introduced by Zadeh in 1965 as a mathematical way to represent uncertainty in everyday life. It can provide the means to represent vague and fuzzy information, manipulate it and to draw inferences from it. Fuzzy logic has been introduced to deal with vague, imprecise and uncertain problems. A fuzzy logic controller can be regarded as an expert system that is able to process qualitative variables and to infer crisp values out of uncertainty. In ordinary mathematics, information is of a crisp kind. It belongs to a set or it does not. The choice of a yes-or-no answer is possible and usually applied, but information could be lost in such a choice, as the degree of belonging is not taken into consideration. A fuzzy model is the idea of a fuzzy set. A fuzzy set differs from conventional (crisp) sets in its semi-permeable boundary membrane. Instead of a characteristic function that has 2 states, inclusion (1) or exclusion the fuzzy set has a function that admits a degree of membership in the set from complete exclusion (0) to absolute inclusion.

The value zero is used to symbolize complete non-membership, the value 1 is used to symbolize complete membership and values in between are used to symbolize intermediate degrees of membership.

A fuzzy concept is a linguistic variable used to define a fuzzy subset, as CLOSE or FAR for a range of obstacle. A Fuzzy set comes as a generalization of conventional set theory. It is a superset of conventional (crisp) logic that has been extended to handle the concept of partial truth-truth values between “completely true” and “completely false”; allowing intermediate values between crisp values.

In Fig 1, The fuzzifier receives the input variables and converts them to suitable linguistic values needed by the inference engine. The inference engine simulate the decision making, based on fuzzy control rule comprises of IF-THEN statements at the fuzzy rule base. The defuzzifier receives the aggregated linguistic values and generates a nonfuzzy control output.

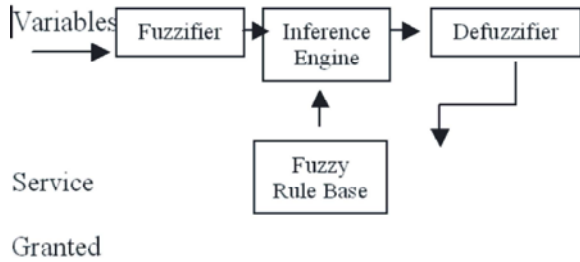


Fig 1: The structure of fuzzy logic controller

Table 1: Fuzzy Rule Base,

(a) Expiry Time Vs Waiting Time

	Waiting Time		
	L	M	H
L	H	L	L
M	M	H	L
H	L	M	H

(b) Packet size Vs Queue length

	Queue Length		
	L	M	H
L	H	M	M
M	L	H	M
H	L	L	H

(c) (a) Vs (b)

	(b)		
	L	M	H
L	VL	L	M
M	L	M	H
H	M	H	VH

**The Fuzzy Scheduler:** The incoming requests in the WiMAX have different variables that play a key role in setting the priority of that particular request. The variables are Expiry Time, Waiting Time, Queue Length, Packet Size and Type of Service. In the proposed fuzzy scheduler two different stages namely the Primary Scheduler, FS1 and the Dynamic Scheduler, FS2 are used. This proposed scheduler is named as Dynamic Fuzzy based Priority Scheduler (DFPS) which uses five inputs namely, Expiry time (E), Waiting time (W), Queue length (Q), Packet size (P) and Type of service and one output namely Final Priority index as shown in Figure 3. Multiple Input and single Output System is used here.

Membership in a fuzzy subset should not be on a 0 or 1 basis, but rather on a 0 to 1 scale; that is the membership should be an element of the interval [0, 1]. The fuzzy rule table is created based on the membership

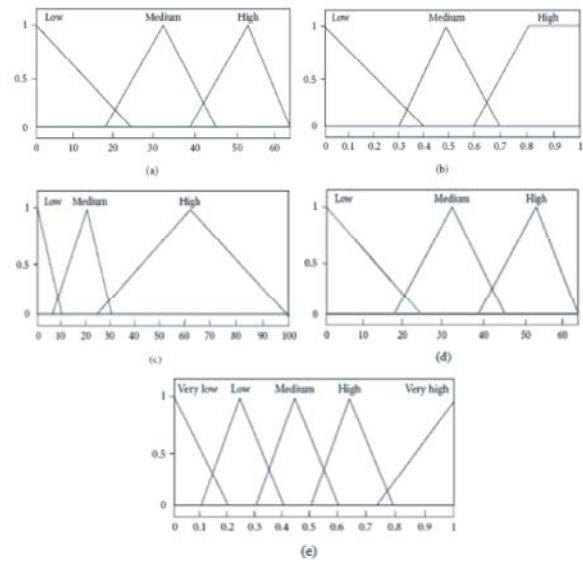


Fig. 2: Membership functions

- (a) Expiry time (in sec) (b) Packet size (in Kbytes)  
(c) Queue length (in bytes) (d) Waiting time (in sec) (e) Priority Index

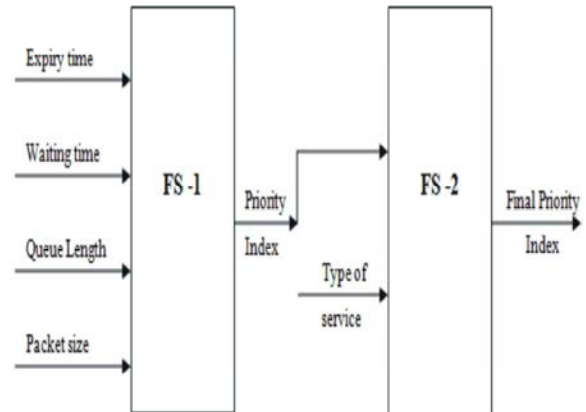


Fig. 3: Dynamic Fuzzy scheduler

functions (Figure 2) that are carefully designed as explained in table 1. The linguistic parameters associated with the input variables are  $FS1=f(L, M, H)$  and  $FS2=f(VH, H, M, L, VL)$ . The bases of functions are chosen so that they result in optimal value of performance measures.

**Definition of the Membership Functions Are:** VL: Very Low, L: Low, M: Medium, H: High, VH: Very High.

For a Primary Scheduler (FS1), the output variable (priority index) has five linguistic parameters. Only triangular functions are used for the output. The Fuzzy Rule Base is shown in table 1. For illustration the ninth rule is interpreted as “If packet size is high and queue

Table 2: Dynamic Fuzzy Rule Base

Priority	UGS	rtPS	ertPS	nrtPS	BE
VL	VH	L	L	VL	VL
L	VH	M	L	L	VL
M	VH	H	M	L	L
H	VH	H	M	M	L
VH	VH	VH	H	M	L

length is high, then priority index is high". Similarly, the other rules are framed. The priority index, if high, indicates that the packets are associated with the highest priority and will be scheduled immediately. If the index is low, then packets are with the lowest priority and will be scheduled only after higher priority packets are scheduled.

For a dynamic scheduler (FS2), the output of the primary scheduler (FS1) and the type of service variable is also added. The Dynamic Fuzzy Rule Base is shown in table 2. To illustrate any rule, consider the first column contents. The Priority Index of the Primary Scheduler may be from VH to VL. If the type of service is UGS then that request must be given higher level priority than the other type of services even if the Primary Scheduler FS1 allots them higher priority indices. The final priority index is? referred as  $\varphi$  which is the standard notation used in the literature. The illustrations were designed using the fuzzy tool available in the MATLAB.

#### Algorithm 1: Setting up the priority

**Input:** Expiry Time, Waiting Time, Packet Size, Queue Length, Type of Service

**Assumptions:** The above said first four inputs are in 3 scales (H, M, L) and last input is in 5 scales (VH, H, M, L, VL).

**Output:** Highest Priority Request For  $i=1$  to  $n$  do

    Compare Expiry Time and Waiting Time which arrives at Priority index 'a'.

    Compare Packet Size and Queue Length which arrives at priority index 'b'.

    Compare 'a' and 'b' (3 scale output) which arrives at Intermediate Priority (5 scale output).

    Compare Intermediate Priority with Type of Service which arrives at Final Priority Index.

Arrange the request in descending order.

#### Proposed ANN

**Scheduling using ANN:** The next step is scheduling of the prioritized input received from the DPFS. Since neural

networks have high computational speeds we decided to use ANN. A neural network is a parallel system, capable of resolving paradigms that linear computing cannot. The first artificial neuron was produced in 1943 by the neurophysiologist Warren McCulloch and the logician Walter Pitts. Neural networks, with their remarkable ability to derive meaning from complicated or imprecise data, can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques. A trained neural network can be thought of as an "expert" in the category of information it has been given to analyze. Other advantages include:

- Adaptive learning
- Self-Organization
- Real Time Operation
- Fault Tolerance via Redundant Information Coding

The artificial neuron is a device with many inputs and one output, designed to mimic the first-order characteristics applied, each representing the output of another neuron. Depending on the specific model used, it can receive different names, such as semi-linear unit, Nv neuron, binary neuron, linear threshold function or McCulloch–Pitts neuron. The artificial neuron receives one or more inputs (representing the one or more dendrites) and sums them to produce an output (synapse). Usually the sums of each node are weighted and the sum is passed through a non-linear function known as an activation function or transfer function. The transfer functions usually have a sigmoid shape, but they may also take the form of other non-linear functions, piecewise linear functions, or step functions. They are also often monotonically increasing, continuous, differentiable and bounded.

The neuron has two modes of operation; the training mode and the using mode. In the training mode, the neuron can be trained to fire (or not), for particular input patterns. In the using mode, when a taught input pattern is detected at the input, its associated output becomes the current output. If the input pattern does not belong in the taught list of input patterns, the firing rule is used to determine whether to fire or not. For a given artificial neuron as shown in Fig 4, let there be  $m + 1$  inputs with signals  $x_0$  through  $x_m$  and weights  $w_0$  through  $w_m$ . Usually, the  $x_0$  input is assigned the value +1, which makes it a *bias* input with  $w_{k0} = b_k$ . This leaves only  $m$  actual inputs to the neuron: from  $x_1$  to  $x_m$ .

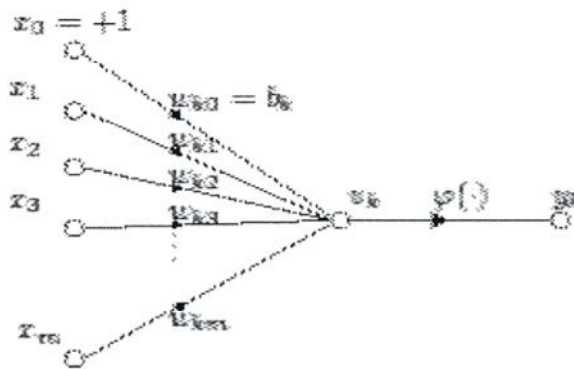


Fig. 4: Basic Structure of a neuron

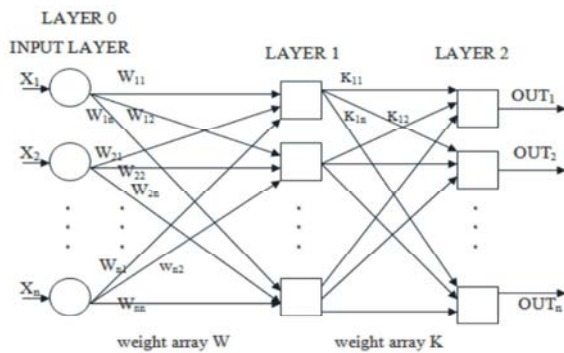


Fig. 5: Proposed ANN

The output of  $k$ th neuron is:

$$y_k = \phi \left( \sum_{j=0}^m w_{kj} x_j \right)$$

Where  $\phi$  is the transfer function.

**Proposed ANN:** The proposed ANN is shown in Figure 5. It consists of three layers. The first layer is the input layer and the second layer is the modified form of Kohonen layer. The final layer is the modified form of Grossberg layer. The proposed ANN deals with the efficient allocation of the available bandwidth based on the Priority Index set by the DFPS with a measure of fairness to all the service classes. The input layer receives the prioritized outputs from the DFPS. These inputs are organized in the order of their priority. Now the output of this layer is given as the input to the modified Kohonen Layer. The scheduling is performed as explained by the algorithm.

#### Algorithm 2: Scheduling using ANN

**Input:** Prioritized Request, Threshold Value

**Output:** Scheduling the request

For  $i=1$  to  $n$  do

In kohonen layer

- If input < threshold, send to Grossberg layer else the request is rejected.

**In Grossberg layer:**

- Compare Sum of bandwidth of requests with threshold
- If possible, set Sum as bandwidth of the request
- Else go for the next request.
- Sum = Sum + Bandwidth
- If threshold > Sum, Set the tag of request to not possible and store the request number as limit
- Else select low priority request starting from bottom
- Repeat steps b and c
- If threshold > Sum, tag the lower priority request as possible and select the next low priority request
- Else Tag the low priority request as not possible and
- select the next low priority request. Then, go to step g.
- If Low priority request number = Limit, stop
- Else go to step e.

**Simulation Results:** The simulation for evaluating the proposed scheduler is implemented using the tools available in MATLAB. The output graph explain the following.

**Effective Channel Utilization:** The scheduler should not assign a transmission slot to a session with a currently bad link since the transmission will simply be wasted. The figure clearly shows the amount of channel utilized by our proposed NFPS Algorithm. It begins from 10% for 10% of load to almost 90% for full load. So as the number of requests increases the channel utilization also increases. It is inferred that as the requested bandwidth nears the total load, the percentage of channel utilization increases.

**Fairness:** The scheduling algorithm must provide fairness to all the requests with different quality of service classes. The channel starving lower priority BE requests and nrtPS requests must be satisfied too leading to fairness. In the Figure 6 all the requests of UGS i.e. 100% are granted. 75% of the requests of the rtPS are granted.

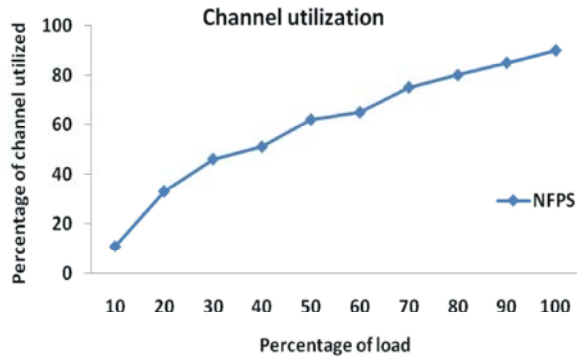


Fig. 6: Graph showing percentage of channel utilized using NFPS Algorithm

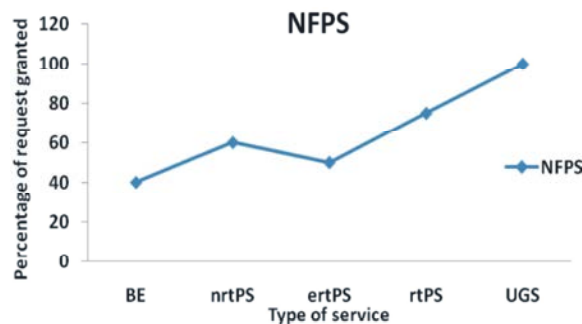


Fig. 7: Graph showing percentage of request granted for different types of services using NFPS Algorithm

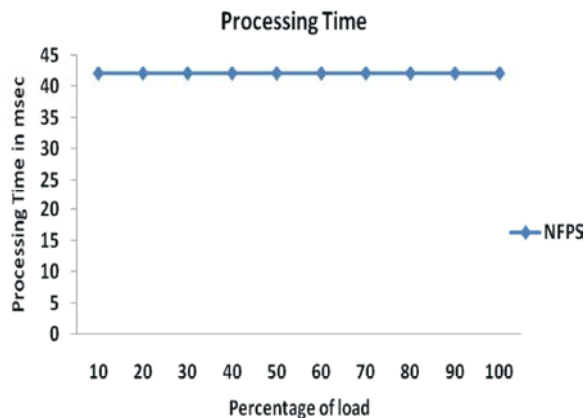


Fig. 8: Graph showing processing time using NFPS Algorithm

But in the case of ertPS 50% of the requests are granted. Even though nrtPS and BE have lower priority 60% and 40 % of their requests are granted respectively.

**Processing Time:** The algorithm must be able to provide delay bound guarantees for individual sessions in order to support delay-sensitive applications that largely depend on the processing time. Figure 7 shows that the processing time for the proposed algorithm to grant a full

load traffic and for lighter loads it was 42 milliseconds. But for multimedia applications using Internet permits delays upto 400 milliseconds as acceptable one. So as for as quality is concerned it is not on the wrong side but very much on the highly acceptable grounds.

## CONCLUSION

An Artificial Intelligence based QoS Scheduling Algorithm was designed. The fuzzy section dealt with the priority setting mechanism under uncertainty conditions by taking into consideration of variables such as expiry time, waiting time, queue length, packet size and the type of service for WiMAX requests. Artificial Intelligence section dealt with bandwidth allocation mechanism by considering fuzzy prioritized output as its input. The Simulation results show that a fair amount of fairness is attained while keeping the priority intact. The results also show that maximum channel utilization is achieved with a negligible increment in processing time.

## REFERENCES

1. Chen, Chao-lieh, Jeng-wei lee, Chi-yuan wu and yau-hwang kuo., 2009. Fairness and QoS Guarantees of WiMAX OFDMA Scheduling with Fuzzy controls, EURASIP Journal on Wireless Communications and Networking.
2. Chen, C.L., 2007. IEEE 802.11e EDCA QoS provisioning with dynamic Fuzzy control and cross-layer interface," in Proceedings of the 16<sup>th</sup> International conference on computer communications and networks (ICCCN'07) pp: 766-771 Honolulu, Hawaii, USA.
3. David Neels Pon Kumar, D., K. Murugesan and S. Raghavan, 2010. A Dynamic Fuzzy Based Priority Scheduler for Mobile WiMAX, Proceedings of ICVLSICOM10 pp 189-194, ISBN 978-93-80043070-8, India, pp: 8-10.
4. David Neels Pon Kumar, D., K. Murugesan and S. Raghavan, 2009. Quality of Service Scheduling for IEEE 802.16 Broadband Wireless Access System—An Overview, International journal Technology World Malaysia pp: 398-404, ISSN 2180-1614.
5. Dong, Guojun, Jufeng Dai School of Electronic Information Engineering and Tianjin University, 2007. An Improved Handover Algorithm for Scheduling Services in IEEE802.16e, China, IEEE.

6. Howon Lee, Taesoo Kwon and Dong-Ho Cho, 2004. An efficient uplink scheduling algorithm for VoIP services in IEEE 802.16 BWA systems, 60<sup>th</sup> IEEE Vehicular Technology Conference, 5: 3070-3074.
7. Yang, Kun, Jie zhang and Hsiao-hwa chen, 2007. A Flexible QoS aware service Gateway for heterogeneous wireless networks, IEEE NETWORK.
8. Sun, J., Y. Yao and H. Zhu, 2006. Quality of Service Scheduling for 802.16 Broadband Wireless Access Systems', IEEE Networks,
9. Wong, W.K., H. Tang, S. Guo and V.C. Leung, 0000 M., v. A fair scheduling for IEEE 802.16 Broadband Wireless Access Systems, ICC, 16-20, Seoul, Korea.
10. Mehrjoo, M., M. Dianati, X. Shen and K. Naik, 2006. Opportunistic Fair Scheduling for the Downlink of IEEE 802.16Wireless Metropolitan Area Networks, 3<sup>rd</sup> International Conference on QoS in heterogenous Wired /Wireless Networks, pp: 191.