World Applied Sciences Journal 15 (4): 525-531, 2011
ISSN 1818-4952
© IDOSI Publications, 2011

# Bandwidth Allocation in Wimax Networks Using Reinforcement Learning

[1]*Saeid M. Jafari*, [2]*Majid Taghipour and* [3]*M.R. Meybodi*

[1]Department of Computer and IT Engineering, Qazvin, Iran
[2]University of Applied Science and Technology, Urmia, Iran
[3]Department of Electrical and Computer Engineering, Amirkabir, Tehran, Iran

**Abstract:** An important problem for the WiMAX networks is how to provide a guaranteed quality of service for applications. A key aspect of this problem is how base stations should share bandwidth capacity between different classes of traffic. The decision needs to be made for each incoming packet and is known as the packet scheduling problem. A major challenge in packet scheduling is that the behavior of each traffic class may not be known in advance and can vary dynamically. this paper has described how the packet scheduling problem has been modeled as an application for reinforcement learning. We have demonstrated how our reinforcement learning approach could learn scheduling policies that satisfy the quality of service requirements of multiple traffic classes under a variety of conditions. The proposed solution has been designed to have an ability to accommodate integrated traffic in the networks with effective scheduling schemes. A series of simulation experiments have been carried out to evaluate the performance of the proposed scheduling algorithm. Results revealed that the proposed solution performs effectively to the integrated traffic composed of messages with or without time constraints and achieves proportional fairness among different types of traffic.

**Key words:** WiMAX % Scheduling Algorithms % Channel assignment % Learning Automata % Quality of Service

## INTRODUCTION

WiMAX technology based on the IEEE 802.16 standard [1] has a very rich set of features [2]. Indeed, it is a very promising Broadband Wireless Access (BWA) technology. The major attractions of WiMAX systems come from their ability to provide broadband wireless access and potential ability to compete with existing wired systems such as fiber optic links, coaxial systems using cable modems and digital subscriber line (DSL) links with much scalability [3]. The WiMAX networks have the capacity to provide flexibility and efficiency to allow coexistence of different types of traffic, such as real-time and multimedia traffic.

The IEEE 802.16 standard provides specification for the medium access control (MAC) and physical (PHY) layers for the air interface. The standard includes details about the various flavors of PHY layers supported and characteristics of the MAC layer such as bandwidth request mechanisms and the scheduling services supported [6].

One important issue in the WiMAX networks design is to support QoS services to different types of traffic. IEEE802.16d standard [1] ratified in June 2004, has specified all the techniques of the WiMAX systems to deliver broadband service in the fixed point-to-point (PTP) or point-to-multipoint (PMP) topologies and it has proposed a framework for the QoS services for four types of traffic. Unsolicited Grant Service (UGS), real-time Polling Service (rtPS), non real-time Polling Service (nrtPS) and Best Effort (BE) QoS classes [5]. UGS supports real-time service flows that have fixed-size data packets on a periodic basis. RtPS supports real-time service flows that generate variable data packets size on a periodic basis. The BS provides unicast grants in an unsolicited manner like UGS where as the UGS allocations are fixed in size. NrtPS is designed to support non real-time service flows that require variable size bursts on a regular basis. BE is used for best effort traffic where no throughput or delay guarantees are provided. Those service classes are defined in order to satisfy different types of Quality of Service (QoS) requirements. However, the IEEE 802.16 standard does not specify the scheduling algorithm to be used [7]. Vendors and operators have to choose the scheduling algorithm(s) to be used. Three types of schedulers must be defined; an uplink and a downlink scheduler both in the Base Station (BS) and just an uplink scheduler for the Subscriber Station (SS) between the different simultaneous connections of the SS.

**Corresponding Author:** Saeid M. Jafari, Department of Computer and IT Engineering, Qazvin, Iran.

This paper has presented a system for packet scheduling that is based on Learning Automaton. In our approach, Learning Automaton is used to learn a scheduling policy in response to feedback from the network about the delay experienced by each traffic class. Key advantages of our approach are that our system does not require prior knowledge of the statistics of each traffic flow and can adapt to changing traffic requirements and loads. In practice, this helps network providers to deliver a guaranteed QoS to customers, while maximizing network utilization and minimizing the need for manual intervention.

We make three key contributions in this paper: (1) we present a model for using RL to address the problem of packet scheduling in Base Station with QoS requirements; (2) we demonstrate the advantages of RL in terms of convergence time in comparison to other scheduling schemes; and (3) we provide an insight into the relative merits of two alternative RL algorithms in the context of this application. We begin by describing the application of packet scheduling. We then describe our solution based on LA and demonstrate its effectiveness in a range of simulated traffic conditions.

**Previous Work:** In this section, we present some schedulers. The simplest scheduling algorithm is the Round Robin (RR) scheduler. It distributes equal channel resources to all the SSs without any priority. The RR scheduler is simple and easy to implement. However, this technique is not suitable for systems with different levels of priority and systems with strongly varying sizes of traffic. There is an extension of the RR scheduler, the Weighted Round Robin (WRR) scheduler, based on static weights. In the same context, we present the Deficit Round Robin (DRR) scheduler. The DRR scheduler associates a fixed quantum (*Q i*) and a deficit counter (DC i) with each flow i. At the start of each round and for each flow i, *DC I* is incremented by *Q i*. The head of the queue i is eligible to be queued if *DC i* is greater than the length of the packet waiting to be sent (*L i*). In this case, *DC i* is decremented by L i. At each round, one packet at most can be sent (and then queued) for each flow. Maximum signal to interference ratio (mSIR) Scheduler is based on the allocation of radio resources to subscriber stations which have the highest Signal to-Interference Ratio (SIR). This scheduler allows a highly efficient utilization of radio resources. However, with the mSIR scheduler, the users with a SIR that is always small may never be served.

The Temporary Removal Scheduler (TRS) scheduler [8] involves identifying the packet call power, depending on radio conditions and then temporarily removing them from a scheduling list for a certain adjustable time period *TR*. The scheduling list contains all the SSs that can be served at the next frame. When *TR* expires, the temporarily removed packet is checked again. If an improvement is observed in the radio channel, the packet could be topped up in the scheduling list again, otherwise the process is repeated for another *TR* duration. In poor radio conditions, the whole process could be repeated up to L times at the end of which, the removed packed is added to the scheduling list, independently of the current radio channel condition.

The Opportunistic Deficit Round Robin (O-DRR) scheduler [9] is used in the uplink direction. The BS polls subscribers periodically. After each period, the BS determines the set of subscribers that are eligible to transmit and their bandwidth requirements. This set is defined as the eligible set. A number of conditions must be verified by an SS to be in this set: (1) the queue is not empty. (2) The received SIR is above a minimum threshold denoted *SIRth*. Once these conditions are satisfied, the subscriber is eligible to transmit during a given frame of the current scheduling epoch. The scheduled set changes dynamically depending on the wireless link state of subscribers. At the beginning of each scheduling epoch, the BS resets the eligible and scheduled sets and repeats the above mentioned process.

The temporary TRS can be combined with the RR scheduler [7]. The combined scheduler is called TRS+RR. For example, if there are k packet calls and only one of them is temporary removed, each packet call has a portion, equal to 1/(k - 1), of the whole channel resources.

The TRS can be combined with the mSIR scheduler. The combined scheduler is called TRS+mSIR [10]. This scheduler assigns the whole channel resources to the packet call that has the maximum value of the Signal to Noise Ration (SNR). The station to be served has to belong to the scheduling list.

**Problem Definition:** This study is based on the model of packet scheduling in cellular network described by Hall and Mars [11]. The RL algorithm has been presented by Taghipoor firstly. But it had high overload because of use of probability matrix, then was not a real automata. Nevertheless in new proposed algorithm use same efficacy.

The aim of this study is to schedule N classes of traffic, where each traffic class has its own queue qi; i = 1: N. Let qN denote the queue for best-effort traffic, which has no predefined delay requirements. For each remaining queue qi; i = 1:::N - 1, there is a mean delay requirement Ri, which is the maximum acceptable mean queuing delay per packet for the traffic class assigned to qi. Let Mi denote the measured mean queuing delay of packets in qi over the last P packets. The aim is to learn a scheduling policy that minimizes MN while ensuring that Mi _ Ri for i = 1: N - 1. In other words, we want to satisfy the QoS constraints for queues qi; i = 1: N-1 while maximizing the available bandwidth to the best-effort queue qN.

In keeping with the model of Hall and Mars [11], all packets in our system have a constant fixed length. This is typical of the internal queues in routers that use a cell switching fabric. We can model this traffic using a discrete-time arrival process, where a fixed length timeslot is required to transmit a packet and at most one packet can be serviced at each timeslot. The arrival of packets is described by a Bernoulli process, where the mean arrival rate li for qi is represented by the probability of a packet arriving for qi in any timeslot. The role of the scheduler is to decide which queue should be serviced at each timeslot (Figure 1). At each timeslot, the scheduler must select an action a g {a1: aN}, where ai is the action of choosing to service the packet at the head of queue qi. The scheduler makes this selection by using a scheduling policy A, which is a function that maps the current state of the system s onto an action a. If the set of possible actions is denoted by A and the set of possible system states is denoted by S, then A: S6A.

The second component of the scheduler is a reward function r: S ×A6R. When an action a ,A is executed in state s g S, the scheduler receives a reward r(s; a) from the system. This reward provides feedback about the immediate value of executing the action a.

The goal is to learn an optimal scheduling by iteratively refining an initial probability vector. Each time we use our current probability vector to select a scheduling action a, we observe the immediate reward R (a) and use this reward as feedback to update our current probability vector (p). This approach is known as reinforcement learning, which has been applied to a variety of scheduling and control tasks. In the next section, we describe our method for using learning automata to optimize the scheduling policy of our queue management system.

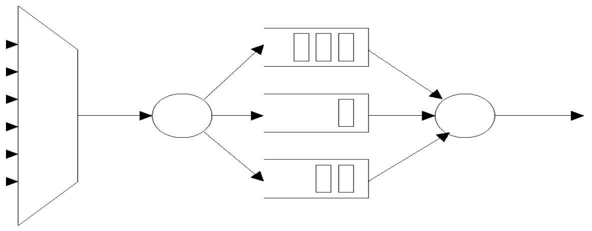

Fig. 1: Packet scheduling for multiple traffic classes

**Our Learning Automaton Approach:** There are three key components to our application of using Learning Automaton to learn scheduling policies for queue management. First, we require a representation of the state s of our system, which reflects the state of the traffic in our queues. Second, we require a suitable reward function: S×A6R, which reflects the immediate value of our scheduling actions. Finally, we require a learning algorithm to refine our policy function A(s) based on the feedback provided by our reward function. Let us now describe our solution for each of these components of our system.

**State Representation:** The reason for introducing the system state into the policy function is so that the scheduler can learn how to act in different situations. This is in contrast to the approach of using a SLA, which uses a single state in its policy function, i.e., the scheduling policy does not depend on the state of the queues. By introducing a more sophisticated state representation we can potentially gain greater control, albeit at the risk of greater complexity. However, we need to ensure that the state representation is not too complex; otherwise there may be too many parameters to be tuned, which may slow the convergence rate of the algorithm.

The aim of scheduling is to use different scheduling policies depending on which queues are not meeting their delay requirements. We represent the state of the system by a set of N -1 binary variables {s1: s-1}, where each variable si indicates whether traffic in the corresponding queue qi is meeting its mean delay requirement Ri,

$$S_i = \begin{cases} 0 & M_i \le R_i \\ 1 & M_i > R_i \end{cases}$$

Note that there is no variable corresponding to the best-effort queue $q_N$, since there is no mean delay requirement for that queue. For example, the state {0; 0;::: ; 0} represents that all queues have satisfied their mean delay constraint, while (1; 0;::: ; 0} represents that the mean delay requirements are being satisfied for all queues except $q_1$. Thus, if there are N queues in the system including one best-effort queue, then there are $2^{N-1}$ possible states. In practice, the number of traffic classes is normally small, e.g., four classes in Cisco routers with priority queuing, in which case the number of states is acceptable.

**Reward Function:** The role of the reward function is to provide feedback to the Learning Automaton algorithm about the effect of a scheduling action. Based on this feedback, the learning algorithm can decide how to update the current scheduling policy. The aim of reward function is to provide a positive reward when packets are serviced within their delay requirement and a negative reward when they are late. We also want to provide a positive reward when the system moves to a better state, i.e., when the measured mean delay for a queue falls below the required mean delay. Thus our reward function r comprises a time reward component rtime and a state reward component rstate, where r = rtime + rstate.

Every time a scheduling action is executed, the time reward rtime;i for each queue qi is calculated in terms of the mean delay requirement Ri and the measured mean delay Mi.

$$r_{time,i} = \begin{cases} C_1 M_1 \big/ Ri & if \quad M_i < R_i \\ C_1 & if \quad M_i = R_i \\ -C_2 & if \quad M_i > R_i \end{cases}$$

The time reward is positive when Mi = Ri and negative when Mi 6 Ri. It is maximized when the mean delay requirement is just satisfied. (Figure 2) There is a diminishi0ng reward as Mi approaches zero, since any reduction in Mi below Ri is wasting bandwidth that could be allocated to other queues. In general, it is possible to change the form of the reward function depending on the type of QoS requirements that need to be satisfied. The total time reward is a weighted sum of the rewards for each queue.

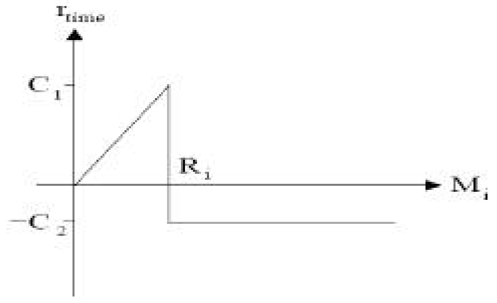

Fig. 2: Time reward function

$$r_{time,i} = \sum_{i=1}^{N-1} w_i r_{time,i}$$

Where the weights wi depend on which queue was serviced by the last scheduling action. In practice, we have found that suitable weights are wi = 0:3 if qi was the queue serviced by the last action, otherwise wi = 1:0. These weights discourage the scheduler from servicing queues with satisfactory performance if there are other queues experiencing unsatisfactory performance. Although the choice of weight values is not critical, we found that we can significantly improve the convergence rate of our system by using a non-zero weight for queues that were not serviced by the last action.

The state reward is positive if a scheduling action causes the system to move to a better state Ð compared to the previous state s. State Ð is considered to be better than s if it has more queues whose mean delay requirements are being met, e.g., Ð= {0; 0;::: ; 0} is better than s = {1; 0;::: ; 0}. Thus

$$t_{state} = \begin{cases} C_3 & \text{if S' Is better than S} \\ O & \text{Else} \end{cases}$$

**Learning Algorithm:** The updating algorithm (function) is used to enable the automaton to learn the state of the random environment based on the obtained feedback and choose the best possible action at any point of stage.

For a multi-action system, the updating algorithm writes as follows:

When a positive response is obtained for an action, its probability is increased and the probabilities of all other actions are decreased. If a negative feedback is received for an action, the probability of that action is decreased and that of others is increased.

$P^i(n + 1) = p^i(n) + a(r^i)$
$P^i(n + 1) = p^i(n) – ((1 – r^i)/3), j ... i$

When a negative feedback is obtained for action i, the automaton updates its action probability set based below Eq.

$P^i(n + 1) = p^i(n) – a(r^i)$
$P^i(n + 1) = p^i(n) + "((r^i)/3), j ... i$

In all our simulations, we assume a = 5 and we refer to this as learning parameter.

Table 1: Slot size for OFDM PH

| Modulation | Channel Coding | Slot Size(byte) |
|---|---|---|
| 64 –QAM | 3/4 | 108 |
| 64-QAM | 2/3 | 96 |
| QPSK | 3/4 | 36 |
| QPSK | 1/2 | 24 |

Table 2: WiMAX parameter

| Parameter | Value |
|---|---|
| PHY | OFDM |
| Bandwidth | 7MHz |
| Frame per Second | 400 |
| Duplexing mode | TDD |
| ARQ/CRC | OFF |

Table 3: The scenario used in our simulation

| Scenario | UGS | rtps | nrtps | BE |
|---|---|---|---|---|
| 1 | 3 | 3 | 3 | 1 |

**Evaluation:** This section presents the simulation results for the proposed scheduling solution. For testing performance of proposed mechanism, the introduced RL is implemented in the Network Simulator (NS-2) [13] and WiMAX module [7] that is based on the WiMAX NIST module [10]. The MAC implementation contains the main features of the 802.16 standard, such as downlink and uplink transmission. We have also implemented the most important MAC signaling messages, such as UL-MAP and DL-MAP, authentication (PKM), capabilities (SBC), registration (REG), dynamic service addition (DSA) and dynamic service change (DSC). The implemented PHY is OFDM.

The current implementation also supports different MCSs. Table 1 shows presents slot size for different modulations and channel coding types.

We present a simulation scenario to study thoroughly the proposed scheduling solution. The scenario will present a multi-service case, in which a provider has to support connections with different 802.16 classes and traffic characteristics. The purpose of this scenario is to ensure that the scheduler at the BS takes the service class into account and allocates slots based on the QoS requirements and the request sizes sent by SSs. Another purpose is to test that the scheduler at the BS takes the MAC overhead into account. Table 1 presents information about which applications are active at scenario.

Regardless of the simulation scenario, the general parameters of the 802.16 network are the same (Table 2). There is one BS that controls the traffic of the 802.16 network. The physical layer is OFDM. The BS uses the dynamic uplink/downlink slot assignment for the TDD mode. Both the BS and all Sss use packing and fragmentation in all simulation scenarios. The MAC level uses the largest possible PDU size. ARQ is turned off; neither the BS nor SSs use the CRC field while sending packets.

Table 3 presents information number applications are active at different times.

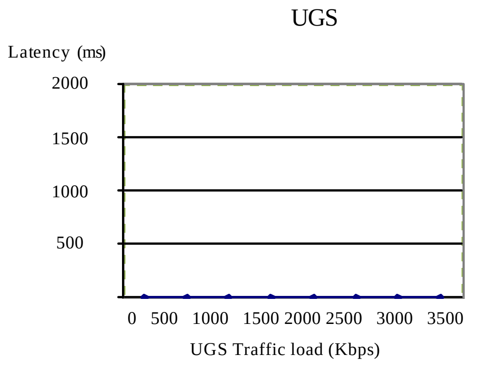

Fig. 3: Latency versus traffic

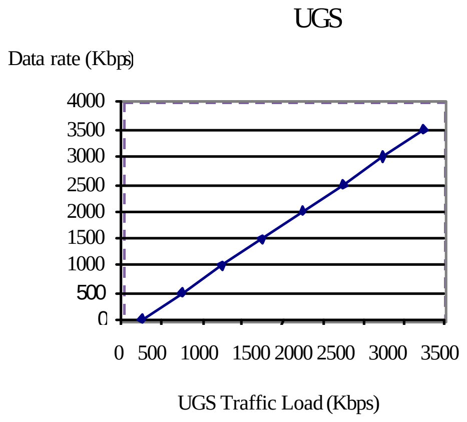

Fig. 4: Throughput versus traffic

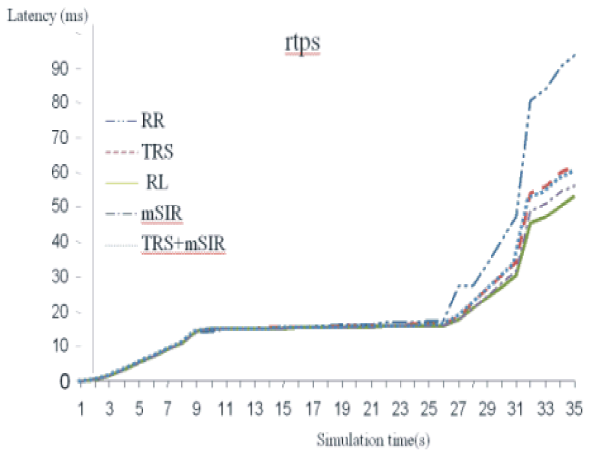

Fig. 5: Latency versus simulation time

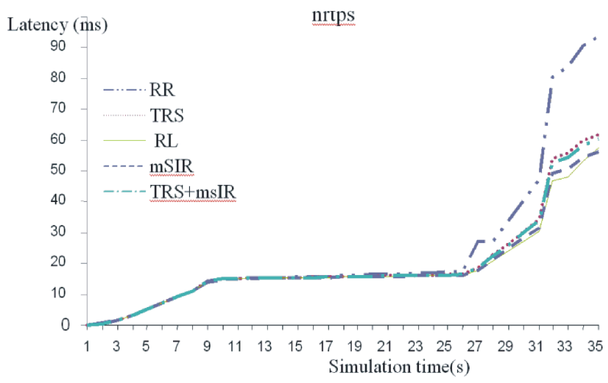

Fig. 6: Latency versus simulation time

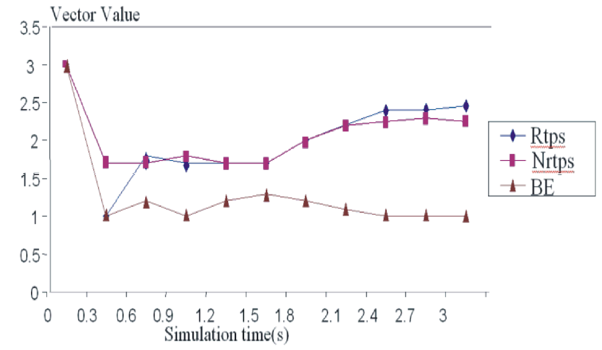

Fig. 7: Automaton Vector variation

In this section, we compare five scheduling algorithms: the NIST_RR, mSIR, RR, TRS+RR and TRS+mSIR schedulers with the proposed method.

Fig. 3 shows latency packets as a function of the traffic load submitted to the network. The data packets are generated by a streaming multimedia application. UGS scheduling algorithms considered in the round except the standard Rubin worked diagram is linear and its graph throughput is linear with a slope (Figure 4). Because of this the following are the algorithms mentioned above, the traffic request is not the highest priority if package is available in this type of traffic speed data services and will be sent.

Figures 5 and 6 show that the RL has a very good behavior in packets mean latency, because this scheduler controls rtPS and nrtps classes latency in both inter and intra-class mechanism by considering latency values of subscribers.

TRS+mSIR has good behavior than the RR scheduler because the channel quality of different SSs is not taken into consideration in RR.

RR has low efficiency because this scheduler allocates all the symbols to SS even if it has not data to send.

Fig. 7 shows all the three queues decrease significantly that is due to delay satisfaction. Then rtps value gently ascends until rtps and nrtps numbers tend to a fixed numbers.

## CONCLUSION

In this article, the behavior of some scheduling algorithms and the proposed algorithm based on the delay parameters and simulation were compared. Algorithm proposed algorithms and Maximum signal to interference ratio hvea the best results. RL schemes are able to satisfy QoS requirements. For multiple traffic classes, without starving resources from best-effort traffic. Furthermore, our RL schemes can adapt to changing traffic statistics and QoS requirements. Simulation results show that the proposed Scheduler Find the best solution for nrtPS and rtPS traffic.

For future work, we will use our proposed method to build up a flexible and intelligent system based on mSIR scheduler. In this system we will train network with minimize packet latency and signal noise ratio for improving QoS and increasing system performance.

## REFERENCES

1. IEEE 802.16-2004, IEEE Standard for local and metropolitan area Networks, Air Interface for Fixed Broadband Wireless Access Systems, Oct 2004.
2. Borin, J.F. and N.L.S. Da Fonseca, 2008. Simulator for WiMAX networks. Elsevier journal, Simulation Modelling Practice and Theory, 15(7): 817-833.
3. Ball, C.F., E. Humburg, K. Ivanov and F. Treml, 2005. Comparison of IEEE802.16 WiMAX Scenarios with Fixed and Mobile Subscribers in Tight Reuse, the 14th IST Mobile & Wireless Communication Summit, Dresden, 19-23 June 2005.
4. Borin, J.F. and N.L.S. Da Fonseca, 2008. Simulator for WiMAX networks. Elsevier journal, Simulation Modelling Practice and Theory, 15(7): 817-833.
5. IEEE 802.16e-2005. 2006. IEEE Standard for local and metropolitan area networks, Air Interface for Fixed Broadband Wireless Access Systems Amendment 2: Physical and Medium Access Control Layers for combined Fixed and Mobile Operation in Licensed Bands and Corrigendum. February 2006.
6. Tsai, T., C. Jiang and C. Wang, 2006. CAC and Packet scheduling Using Token Bucket for IEEE 802.16 Networks. J. Communications, 1(2).

7. Belghith, A. and L. Nuaymi, 2008. Design and Implementation of a QoS-included WiMAX Module for NS-2 Simulator. First International Conference on Simulation Tools and Techniques for Communications Networks and Systems, SIMUTools 2008, Marseille, France, March 3-7,2008.
8. Ball, C.F., F. Treml, X. Gaube and A. Klein, 2005. Performance Analysis of Temporary Removal Scheduling applied to mobile WiMAX Scenarios in Tight Frequency Reuse", the 16th Annual IEEE International Symposium On Personal Indoor and Mobile Radio Communications, PIMRC' 2005, Berlin, 11-14 September 2005.
9. Rath, H.K., A. Bhorkar and V. Sharma, 2006. An Opportunistic DRR (O-DRR) Uplink Scheduling Scheme for IEEE 802.16-based Broadband Wireless Networks, IETE, International Conference on Next Generation Networks (ICNGN), Mumbai.
10. Vinay, K., N. Sreenivasulu, D. Jayaram and D. Das, 2006. Performance Evaluation of End-to-end Delay by Hybrid Scheduling Algorithm for QoS in IEEE 802.16 Network, Wireless and Optical Communications Networks, 2006 IFIP International Conference.
11. Hall, J. and P. Mars, 1998. Satisfying QoS with a Learning Based Scheduling Algorithm. School of Engineering, University of Durham.
12. Meybodi, M.R. and S. Lakshmivarahn, 1983. A Learning Approach to Priority Assignment in aTwo Class M/M/1 Queuing System with Unknown Parameters, Proceedings of the Third Yale Workshop on Applications of Adaptive Systems Theory, Center for Systems Science, Yale University, Yale, USA, pp: 106-109.
13. The network simulator ns-2. http:// www.isi.edu/ nsnam/ ns/, September 2007.