QoS SCHEDULING FOR IEEE 802.16e MESH MODE USING GENETIC ALGORITHMS

Neila Krichene and Noureddine Boudriga
Communication Networks and Security Research Lab, University of the 7th of November at Carthage, Tunis, Tunisia

Keywords: IEEE 802.16e mesh mode, rtPS scheduling, Genetic algorithms.

Abstract: IEEE 802.16e amendments for the mesh mode do not specify particular QoS mechanisms. However, each Mesh Subscriber Station (MSS) transmits its own traffic and forwards the traffic of its children while each traffic flow has its particular QoS constraints. As all forwarded traffics use the same link, one MSS may experience starvation or act selfishly. Besides, the MSS's mobility adds important complexity as all the affected flows need to be re-routed, re-admitted and re-scheduled while respecting their original QoS requirements. In this paper, we adopt the concept of assigning five virtual nodes to each MSS reflecting the five QoS classes of IEEE 802.16e and we propose scheduling the uplink transmissions of the real-time Polling Service (rtPS) traffic class by adopting the Genetic Algorithms concept. We define the priority assigner component which communicates with the scheduler of the mesh BS in order to set and update flows' priority and we optimize the time spent by a flow in the queue while implementing the mobility constraints within the mutation function.

1 INTRODUCTION

The recent years have been marked by a growing need for providing advanced applications and Internet-related services at high throughput and low costs while guaranteeing the required QoS and a continuous and open access to such services. In order to address such need at the metropolitan scale, the IEEE 802.16e amendments implement service differentiation and adopt a connection-oriented philosophy within a mobility context but they left many QoS functions unspecified so that researchers and constructors can design and adopt the most suited mechanisms that fulfill particular requirements. Moreover, IEEE 802.16e amendments for the mesh mode do not specify particular QoS mechanisms as the traffic of the mobile Mesh Subscriber Station (MSS) along with the traffics of all its children use the same link which has no service or QoS parameters associated with it. Besides, the mesh Base Station (BS) schedules the transmission and all packets originating from the mobile MSS use the aggregate grant values regardless of their nature and QoS constraints, (Kuran et al., 2006). As all forwarded traffics use the same link, one mobile MSS may experience starvation or act selfishly. Last but not least, the MSS's mobility adds important complexity as all the affected flows need to be re-routed, re-admitted and re-scheduled while respecting their original QoS requirements.

Note that optimal QoS provision within the IEEE 802.16e context can not be fulfilled without the definition of optimal scheduling of space, frequency and time resources over the air interface on a frame-by-frame basis. Resources allocation and scheduling need to dynamically adapt to the bursty and unforeseeable nature of the traffic while providing a large dynamic range of throughput to specific users based on their demand without degrading the overall network performances or causing starvation to particular users or traffic flows. The implemented scheduling scheme should be priority-based in order to correctly distribute the available resources among the various flows depending on their QoS requirements. It should also be simple, efficient and fair with a low computational complexity and needs to guarantee the throughput and delay performance. Currently, many research activities are conducted to propose optimized scheduling algorithms for both the point-to-multi point mode and the mesh mode. Nevertheless, designing efficient scheduling methods for the mesh mode remains a little bit harder due to the distributed nature of the mesh mode and the constraints induced by the multi-hop communication. Regarding optimizing the scheduling scheme, we propose to adopt the

Krichene N. and Boudriga N. (2010).
QoS SCHEDULING FOR IEEE 802.16e MESH MODE USING GENETIC ALGORITHMS.
In *Proceedings of the International Conference on Wireless Information Networks and Systems*, pages 5-12
DOI: 10.5220/0002916900050012
Copyright © SciTePress

genetic algorithms concept which are relatively simple to implement and considered particularly appropriate for scheduling problems. To the best of our knowledge, genetic algorithms have not been used to solve traffic related problems particular to mesh networks.

In this paper, we aim at optimizing the scheduling of the uplink transmissions of the rtPS traffic class within the IEEE 802.16e mobile mesh context [1] while handling the selfish behavior of the nodes and the starvation issues through priority assignment. Our approach uses the Genetic Algorithms concept in order to optimize the time spent by a flow in the queue while implementing the mobility constraints within the mutation function. To the best of our knowledge, Gnetic Algorithms have not been used before for optimizing scheduling while respecting QoS constraints within the wireless mobile mesh networking context.

Moreover, we dissuade mobile MSSs from acting selfishly by augmenting the priority of their own flows as long as they forward the traffics of their children; thus avoiding starvation. For that aim, we define the priority assigner component which communicates with the scheduler of the mesh BS in order to set and update flows' priority and we propose a linear approach and an exponential approach to deduce the priority level that should be assigned to a flow while updating that value due to mobility. The rest of the paper is organized as follows: first, we overview the state of the art regarding scheduling in the IEEE 802.16e context. We then detail our proposed scheme. Finally, we evaluate the performances of our proposition.

2 RELATED WORK

The IEEE 802.16e amendments for the mesh mode define two modes of scheduling known as centralized scheduling and distributed scheduling, (IEEE, 2006). Distributed scheduling is adopted when a Mesh Client (MC) has data to be transmitted to a neighboring MC managed by the same mesh BS. In this case, nodes negotiate the distribution of transmission opportunities in a pairwise fashion by using a three-way-handshake. First, an MC wishing to change the transmission opportunity allocation for one of its connections should send a request for transmission opportunities to its neighbors using a *Mesh Distributed Schedule (MSH-DSCH)* packet. One or more of the neighbor correspond with a range of available transmission opportunities. The MC chooses a subset of available transmission opportunities and acknowledges that it will use them with a third *MSH-DSCH* packet. After the execution of distributed scheduling, the mesh node can transmit in the reserved timeslot without collision, (Ciao and al, 2005).

In the centralized scheduling, the mesh BS acts as a scheduler and allocates transmission and reception timeslots for each client station. First, a scheduling tree rooted at the mesh BS is established. This tree describes the routing path between each MC and the mesh BS; its is also broadcasted to all MCs. Each MC belongs to one tree layer and has a position number in that layer. The centralized scheduling operates in two stages and the time period required to perform both stages is known as the "*scheduling period*". In the first stage, the mesh BS collects the bandwidth requests from all MCs. In the second stage, the mesh BS allocates then distributes the transmission and reception schedule to all MCs within the *Mesh Centralized Schedule (MSH-CSCH)* messages. Note that the data subframe description belongs to a frame after the frame that the grant is sent,(Kuran et al., 2006). Moreover, in centralized scheduling, the mobile MSS needs to send one bandwidth request for each link it has with the neighboring stations and all the requests belonging to that MSS is sent within a unique *MSH-CSCH* message, (IEEE, 2006). The grant sent in the *MSH-CSCH* message indicates the amount of data that a node can transmit independently with the QoS requirements of the transmitted flows.

Many centralized scheduling techniques for the mesh WiMAX networks have been proposed. Those may be without spatial reuse or with spatial reuse. Spatial reuse enables the scheduler to assign the same slot to non interfering links. The proposed techniques also differ in whether they provide QoS guarantees or not, consider fairness or not and suggest routing schemes or not. For instance, authors in (Shetiya and Sharma, 2005) propose routing and centralized scheduling algorithms that guarantee per-flow QoS requirements to real-time and interactive data applications. More precisely, they separately schedule UDP and TCP connections and compute the number of slots required per flow along the path and at each node per frame while assuming an OFDM-based physical layer. The number of slots is computed with regard to the flow characteristics such as the end-to-end packet drop probability. Once the mesh BS assigns the computed number of slots to the nodes, the nodes provide the required slots to its different queues in a weighted round robin manner. For TCP traffics, slot allocation is proportionally fair to the minimum bandwidth requirements of the nodes.

In (Mai et al., 2009), authors designed a special

---

[1] In fact, we find that our approach is easily applied to the new standard IEEE802.16j as the multihop relay mode reduces the complexity of the mesh mode.

bandwidth requests for UGS traffic while adding the weight of delay when scheduling different Subscriber Stations (SS) with the same service type. More precisely, authors in (Mai et al., 2009) propose an Expedited Queue (EQ) scheduling scheme which considers both per-hop BW-REQ and end-to-end route path in order to provide absolute QoS guarantees for UGS traffic. When the sender SS of UGS traffic requests BW-Req, the BS assigns the slots based on the requested slots and the number of hops within the route path; thus reducing overhead and end-to-end delays. Moreover, authors in (Mai et al., 2009) assign to SSs within the same service type but with higher load higher priorities. They also reduce the access delay of real-time flows including UGS ones by giving higher priorities to data frames that have been waiting a longer time in the queue. Nevertheless, they do not differentiate between the SS's own traffic or the SS's children traffic. The performance of the proposed techniques is evaluated by considering the average delay and the delay variation (ms) versus the flow data rate of total input traffic (Mbps).

In (Ghosh et al., 2008), authors survey multiple centralized scheduling techniques for the mesh and the PMP modes. Some interesting ideas may be highlighted such as ordering the assigned slots to reduce jitter, ranking links based on their satisfaction with the schedule in the previous iteration (satisfaction s = rate achieved with the previous schedule / required bandwidth of the link), scheduling flows based on a priority value proportional to the node's load and throughput requirements or scheduling the flows while respecting the fairness constraint or the transmission power constraint. Moreover, authors of (Ghosh et al., 2007) define a metric called "Schedule Efficiency" as the proportion of the weighted measure of the admitted flows to the weighted measure of all flows seeking admission in order to compare the performance of their scheduling algorithm to other scheduling techniques.

Authors of (Belghith and Nuaymi, 2008) compare five scheduling algorithms which are the Round Robin (RR), the maximum Signal-to-Interference Ration (mSIR), the Weighted RR, the combination of the Temporary Removal Scheduler (TRS) and the RR (TRS+RR) and the combination of the Temporary Removal Scheduler and the mSIR (TRS+mSIR) by considering the number of delivered data packets versus traffic load, the number of the served SSs per frame versus traffic load. They also compared their proposed technique called modified maximum Signal-to-Interference Ration (mmSIR) and the original mSIR by considering the mean sojourn time versus the traffic load. Nevertheless, the considered algorithms are mainly designed for the PMP mode of WiMAX.

Authors of (Kuran et al., 2006) propose a Service Adaptive QoS (SAQoS) approach in which the mesh BS assigns five virtual node IDentifiers (node IDs) instead of one to each MSS. The virtual nodes IDs represent the five scheduling classes of the IEEE 802.16e standard and each of these virtual nodes requests bandwidth individually according to its requirements. Consequently, the mesh BS will handle the received requests independently. As in mesh mode, a separate request must be sent to the MBS for each hop, authors of (Kuran et al., 2006) order their mesh BS to allocate the same amount of bandwidth to each link the traffic uses to reach the mesh BS and the same allocation differentiation is valid for downlink traffic to MSSs with hop count more than one. Authors of (Kuran et al., 2006) also propose a Fair Adaptive Base Station Scheduler (FABS) that bases its scheduling decisions on each MSS's current request and the grants given to all MSSs in the network. In (Mogre et al., 2008), authors jointly optimize the routing, scheduling and bandwidth savings within the IEEE 802.16 context using network coding while reducing the computational costs.

The stated scheduling methods are highlighted and compared to our proposed method in the table 1.

3 PROPOSED SCHEME

3.1 Problematics, Assumptions and Goals

The IEEE 802.16e mesh mode have several characteristics that render complex the QoS provision. First, each MSS not only transmits its own traffic flows but also forwards the traffic flows of its children. Therefore, the flows owned by the forwarding MSS may experience starvation as the available bandwidth on the links is shared between all forwarded traffics. Besides, a MSS may act selfishly to benefit from the available bandwidth. Meanwhile, when a MSS moves, all the flows that were managed by it need to be re-routed, re-admitted and re-scheduling while compensating the induced processing delays in order to meet the original QoS requirements. Particularly, the priority assigned to the affected flows needs to be dynamically adjusted while the uplink scheduling needs to be dynamically revised in order to optimize the delays spent in the queues while meeting the QoS requirements despite mobility.

Our proposed scheme adopts the idea of the five virtual nodes ID assigned to each MSS, (Kuran et al.,

Table 1: Comparing some scheduling techniques.

Scheduling Method	Scheduling Policy	Priority Policy	Layer of scheduling Implementation	Performance criteria addressed	Mobility Involvement
Routing and scheduling for IEEE 802.16d mesh mode (Shetiya and Sharma, 2005)	Centralized scheduling for real-time and non real-time flows	-Traffic based: priority to UDP traffic over TCP traffic -Does not treat selfish behavior and starvation issues	Physical: calculates the number of slots to be allocated	-Average bandwidth provided for each flow versus required average bandwidth	No mobility considerations
Combined scheduling QoS framework for IEEE mesh mode (Mai et al., 2009)	Combines centralized scheduling and distributed scheduling	-Within the same service type, the SS with higher load has higher priority -Traffic based: Highest priority for UGS traffic -Higher priority is given to real-time data-frames that waited a longer time in the queue	Cross layer approach involving the MAC layer and the IP layer	-Average delay and delay variation (ms) versus the flow data rate of total input traffic (Mbps)	No mobility considerations
Enhancement of of the maximum Signal-to-Interference Ratio (mSIR)(Belghith and Nuaymi, 2008)	Schedules rtPS flows for the Point to Multi-Point (PMP) mode	If it does not serve an SS having unicast request opportunities, it gives priority to other SSs having higher SIR	Physical: involving the quality of the link	Mean sojourn time versus traffic load	No mobility considerations
Service Adaptive QoS approach (Kuran et al., 2006)	-Assigns 5 virtual node identifiers for each QoS class - Proposes a BS scheduler for the centralized scheduling of the mesh mode	Not addressed	Physical: grants bandwidth to links	- Service delays of 5 flows of involved SSs	Partially considered
Our method	Minimizes the sojourn time in the queue using genetic algorithms for rtPS traffic	- Traffic based and history based: Own rtPS flows' priority depends on the amount of children's forwarded traffic -Addresses selfishness and starvation - The flow's priority is incremented when the managing node leaves the route due to mobility	MAC layer: adds a new set of functions	-Number of rescued flows in each round -Number of genomes that persist until the last round -Delay is reduced since waiting time is optimized	Flows affected by mobility are rescued and their QoS constraints are revised in order to meet the original QoS requirements

2006), and considers the rtPS traffic class. Each rtPS flow is characterized by its *minimum reserved traffic rate* (bits/s), its *maximum sustained traffic rate* called also *peak rate* (bits/s), its *maximum latency* (s), and its *priority*, (IEEE, 2006). According to the mobile WiMAX specifications, rtPS flows are generated by the third class of applications that includes streaming media, (Forum, 2006). Guidelines specify bandwidth values ranging between 5 kbits/s and 2 Mbits/s while latency values are not specified, (Forum, 2006). Priority ranges between 0 to 7 where higher numbers indicate higher priority, (IEEE, 2006). rtPS flows that we consider in our scheme have a size ranging from 20 Mbits to 340 Mbits, a minimum reserved traffic rate raging between 500 kbits/s and 2 Mbits/s, a maximum sustained traffic rate ranging from 500 kbits/s to 2 Mbits/s and a maximum latency value of 680 seconds. We also assume that a route has always been established to the mesh BS so that intermediate MSSs are known to the source and that every intermediate MSS issues its own request in order to forward the correspondent flow. We also assume that the original QoS values are set by the source and are integrity protected. Besides, each intermediate MSS should update the maximum latency value (i.e., the updated value is specified in a different field) when issuing its request while taking into consideration the time spent by the flow before reaching that MSS; the updated value should always be smaller than the original one.

Our proposed scheme intends to schedule the uplink transmission of rtPS traffic while a priority will be set for every flow in order to encourage the MSS forwarding the traffic of their children. That priority value is managed by a new component that we call "***Priority Assigner***" which communicates with the scheduler of the mesh BS. We intend to optimize the time spent by a flow in the queue while respecting the flow's QoS requirements. We adopt the Genetic

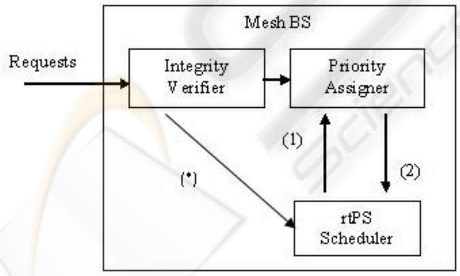
Algorithms concept in order to solve the scheduling problem.

3.2 Priority Assignment

We define the *Priority Assigner* component at the mesh BS level as depicted by Figure 1. The sending MSS generates a rtPS transmission request while indicating the minimum reserved traffic rate, the maximum sustained traffic rate and the maximum latency values without specifying a priority value. The *Priority Assigner* component will get the request and update it by setting the priority value depending on the history of the issuer. More precisely, the *Priority Assigner* component maintains a table with these entries $(NID, T_0, AmountofForwardedTraffic, Priority)$ where NID refers to the issuer's node Identifier, T_0 refers to the instant when the issuer has entered under the coverage of the managing mesh BS and $AmountofForwardedTraffic$ refers to the amount of the traffic that the issuer has forwarded so far for its children. The *Priority Assigner* component assigns a priority value to the issuing MSS which is computed by the following

$$\frac{amountofForwardedTraffic}{t - T_0} \tag{1}$$

then adjusted to be in the interval [0,7] depending on 8 thresholds to be defined in order to cope with the standard amendments. The priority value of that MSS is then communicated to the scheduler so that it can update the priority of the rtPS flow lastly generated by that MSS then schedule it with regard to the assigned value.



(*) The integrity verifiers communicates the CID of a forwarded flow

Figure 1: The priority assigner interactions.

Note that when the mesh BS scheduler schedules the transmission of a flow (identified by a *CID*) by a MSS (identified by a *NID*) and that flow is not the own flow of the forwarding MSS, the scheduler communicates with the *Priority Assigner* component to update the $AmountofForwardedTraffic$ value. When one flow is affected by the mobility of an intermediate node on the route, the priority of that flow is incremented independently of its issuer.

Mobile WiMAX parameters are stated in (Forum, 2006). For instance, the mobile nodes can have a speed reaching 120 km/h. Moreover, the distance between two BSs is about 2.8 km. That means that a mobile WiMAX node moving at 120 km/h needs 84 seconds to cross the distance between two BSs. In order to adjust the priority value to be in the interval [0,7], we propose a linear approach and an exponential approach. Regarding the linear approach, we assume that a mesh node may stay under the coverage of the mesh BS for a certain period of time during which its should transmit a certain amount of traffic for its children in order to increase its own traffic's priority. The time and traffic size are augmented linearly by adding chosen values and the obtained values are mapped to 8 levels of priority. The numerical scenario that we propose assumes that after 85 seconds, the mesh node should forward $85/2$ Mbits for its children in order to increase the priority of its own traffic by one. The assigned priority values are given by the table 2. For illustration purpose, when a node stays for 85 seconds under the coverage of the same mesh BS and forwards less than 42.5 Mega bits for all its children, independently of their number, all the flows issued by that node will have 0 as priority. When that node stays 100 seconds and forwards less than 85 Mega bits for its children, the priority remains 0 but when the node stays 100 seconds and forwards between from 42.5 to 85 Mega bits, the priority is set to 1.

Table 2: Priority assignment in case of a linear approach.

$AmountofForwarded$ $Traffic(Mega\ bits)$	$thet - T_0$ $(seconds)$	Assigned Priority
340	680	7
297.5	595	6
255	510	5
212.5	425	4
170	340	3
127.5	255	2
85	170	1
42.5	85	0

Regarding the exponential approach, we assume that a mobile mesh node which transmits less than 42.5 Mega bits for its children during 85 seconds implies that its own flows will be assigned 0 as priority. These forwarded traffic and time values will be doubled in order to increase the assigned priority as depicted by the table 3.

We also define the ***"Integrity Verifier"*** component shown by the Figure 1 which role is to verify whether a request is issued by its owner or by an intermediate MSS on the route. The CID of a request that is not

Table 3: Priority assignment in case of an exponential approach.

$Amount of Forwarded Traffic(Mega\ bits)$	$thet - T_0$ $(seconds)$	Assigned Priority
5440	10880	7
2720	5440	6
1360	2720	5
680	1360	4
340	680	3
170	340	2
85	170	1
42.5	85	0

issued by the owner of the flow along with the node ID of the forwarding node are communicated to the scheduler so that the latter can update the correspondent $AmountofForwardedTraffic$ value.

3.3 Genome Modeling, Fitness, Mutation and Cross-over

We model a genome as a list of flows. Each flow is characterized by its size, its minimum reserved traffic rate, its maximum sustained traffic rate, its maximum latency, its priority value and the delay that the flow spends in the queue before being scheduled for transmission $d_{CID,NID}$. We assume that we have N flows to be scheduled and we use the Genetic Algorithms concept to minimize the delay that a flow spends in the queue before being scheduled for transmission.

We calculate the fitness of a genome by

$$\sum d_{CID,NID} \tag{2}$$

and we define the optimal genome among M ones as the one having

$$Min_M(\sum_{i=1}^{N} d_{CID,NID})$$

and we intend to optimize

$$d_{CID,NID} = T_s - T_e \tag{3}$$

where T_s is the instant at which the request is served and T_e is the instant at which the request was issued subject to constraints:

$$maximumSustainedRate < available\,BW$$

and

$$MinimumReservedRate < available\,BW$$

We always verify that

$$d_{CID,NID} < ML - \lambda$$

where λ is the maximum propagation delay needed for the node to receive the schedule then transmit the flow till the mesh BS.

When a MSS leaves the network, its own flows are no longer scheduled for uplink while the flows of its children should be rescued by neighboring MSSs. Rescuing a flow by a neighboring MSS induces a delay that should be taken into consideration when the rescuing MSS issues an uplink transmission request regarding that flow. Therefore, the rescuing MSS should decrement the ML value and increment the priority of the rescued flow. The mutation operator reflects the mobility of MSSs. More precisely, the mobility of a MC implies

$$T_{sMSS} = \infty, T_{sj} = T_{sj} - 10 \tag{4}$$

(where j is the index of a child of the quitting MSS) and

$$p_j = p_j + 1 \tag{5}$$

(where p is the priority and j is the index of a child of the quitting MSS). The number of chromosomes of an individual is given by the Formula

$$\frac{Total\,BW\,of\,meshBS}{minimal\,BW\,required\,by\,a\,flow} \tag{6}$$

4 PERFORMANCE EVALUATION

We implemented our scheduling genetic algorithm by generating a 500 genomes population where each genome is composed by 250 flows and a fitness. Every node has 5 parent nodes and and each parent node ID is comprised between 1 and 25. In every round of the algorithm, the cross-over procedure is applied followed by the mutation procedure which randomly affects 100 genomes. The mutation consists in eliminating 100 nodes of the mutant genomes; that is 100 nodes quit the network due to mobility. The 100 eliminated nodes are randomly chosen and their flows have a $T_s$value equal to 999999 which is the synonym for an infinite value. After that, the resulting new population is sorted in order to keep the best 500 genomes having the best 500 fitness values. The pre-described round is executed 100 times and for each round we keep the fitness of the best genome in population. It is worth noticing that an arbitrary number of simulations may be considered as we need just to launch the genetic algorithm. However, we opted for considering three simulations only as we have noticed that the difference between the obtained curves is tiny and in order of 1.25%. This means that even if we launch a greater number of executions, the obtained results will be similar and that the conclusions that we can have in the light of three simulations are relevant.

The results obtained after executing the genetic algorithm while adopting the numerical scenario stated in 3.2 are plotted in the Figures 2 and 3 below. We may notice that the genetic algorithm converges after maximum 10 executions when assigning the priority of the flows using the linear approach as depicted by Figure 2. When adopting the exponential approach, the algorithm converges also after maximum 10 executions as depicted by Figure 3. Note that the chosen numerical scenario fixing the maximum latency, the minimum reserved traffic rate and the size of the forwarded traffic values influences the number of rounds needed for convergence. However, that number has not exceeded 20 rounds for more than 10 different numerical scenarios that we probed. Note also that we have tried other numerical scenarios and we obtained curves having nearly the same shape with a rapid convergence. We think that the rapid convergence of the genetic algorithm is mainly due to the fact that we perform multiple verifications on the generated flows and we state conditions on the values characterizing the scheduled flows. We also conclude that the fitness values of the genomes in both Figures 2 and 3 are very close and that they converge to nearly the same value.

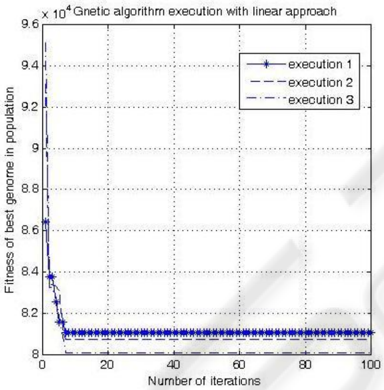


Figure 2: Execution of the genetic algorithm when adopting the linear approach.

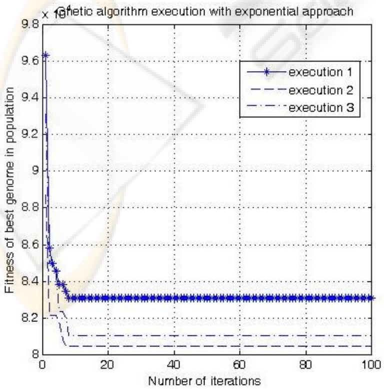


Figure 3: Execution of the genetic algorithm when adopting the exponential approach.

In order to evaluate the performances of our scheduling technique, it is interesting to evaluate the number of rescued flows in each round, that is those affected by mobility, and to consider the number of initial genomes that persist until the last round. We notice that the number of rescued flows when adopting the linear approach is very reduced in the first 10 rounds as depicted by Figure 4, then it raises exponentially until the 23^{th} round, after that it converges after the 45^{th} round. This behavior persists when adopting the exponential approach as depicted by Figure 5. However, the number of rescued flows stabilizes after the 38^{th} round in this case. We notice that adopting either the linear or the exponential approach for the priority assignment does not seriously affect the number of rescued flows; nevertheless, the convergence is obtained rapidly with the exponential approach. Moreover, we conclude that after a particular x number of rounds, which is also equivalent to executing the mutation function on $100*x$ genomes of the retained population, it becomes difficult to rescue more flows while respecting their QoS constraints. This conclusion is relevant even after only three simulations for the same reasons stated earlier.

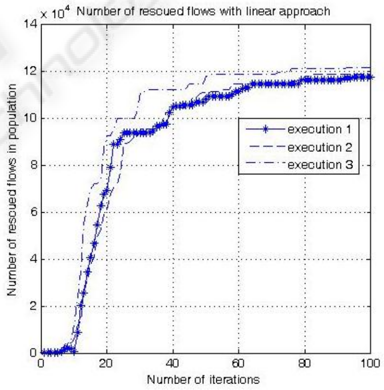


Figure 4: Number of rescued flows in each round when adopting the linear approach.

Regarding the number of persistent genomes, we notice that after 6 to 8 rounds, it becomes null whether we adopt the linear approach or the exponential approach as depicted by Figures 6 and 7. This means that the genetic algorithm rapidly produces better genomes and that it completely renews the population leading to the creation of a better scheduling scheme that optimizes the sojourn time of the rtPS flows. The obtained results regarding the number of persistent genomes confirm the rapid convergence of the algorithm.

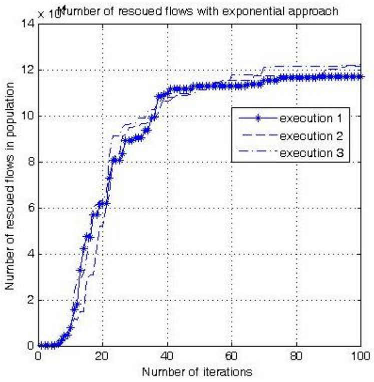


Figure 5: Number of rescued flows in each round when adopting the exponential approach.

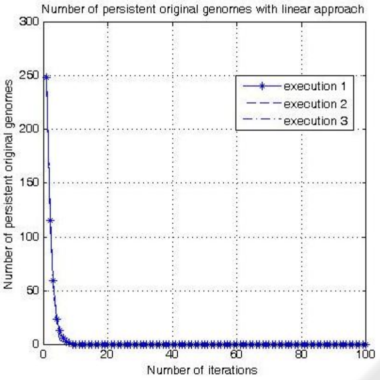


Figure 6: Number of persistent genomes in each round when adopting the linear approach.

5 CONCLUSIONS

In this paper, we addressed the scheduling problem of the rtPS flows within the IEEE 802.16e mesh context while minimizing the sojourn time of such flows in the queues; thus leading to a reduction in the transmission delays and to a better QoS provision. To that aim, we built an ad hoc genetic algorithm and we showed, through the simulations, that our genetic algorithm converges rapidly. In other words, we rapidly reach an optimized scheduling scheme that minimizes the transmission delays of rtPS flows. Moreover, we proposed a novel scheme for assigning priorities to the transmitted flows then dynamically updating such values in case of mobility in order to encourage the mesh nodes forwarding the traffic of their children while favoring flows that need to be rescued when the managing parent node quits the network. That scheme may adopt a linear approach or a linear approach which both lead to comparable performances. The results provided in this paper are encouraging; therefore we plan to further analyze the performance in the network by adapting additional metrics and we also plan to investigate the effects of mobility independently from the effects of the prioritization scheme.

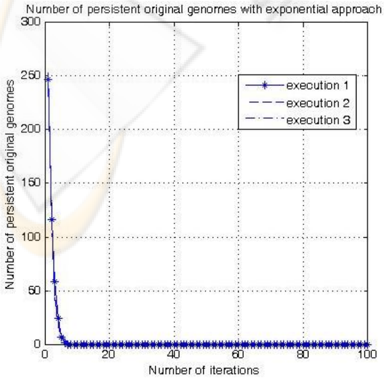


Figure 7: Number of persistent genomes in each round when adopting the exponential approach.

REFERENCES

Belghith, A. and Nuaymi, L. (2008). Comparison of wimax scheduling algorithms and proposal for the rtps qos class. Technical report, TELECOM Bretagne.

Ciao, M. and al (2005). Modeling and performance analysis of the distributed scheduler in ieee 802.16 mesh mode. In *MobiHoc'05*, Urbana-Champaign, Illinois, USA.

Forum, W. (2006). Mobile wimax- part i: A technical overview and performance evaluation. Technical report, Wimax Forum.

Ghosh, D., Gupta, A., and Mohapatra, P. (2007). Admission control and interference-aware scheduling in multihop wimax networks. In *IEEE MASS*.

Ghosh, D., Gupta, A., and Mohapatra, P. (2008). Scheduling in multihop wimax networks. *Mobile Computing and Communication Review*, 12(2).

IEEE (2006). Ieee standard for local and metropolitan area networks part 16: Air interface for fixed and mobile broadband wireless access systems amendment 2: Physical and medium access control layers for combined fixed and mobile operation in licensed bands and corrigenda 1. IEEE.

Kuran, M. S., Y1lmaz, H. B., Alagz, F., and Tucu, T. (2006). Quality of service in mesh mode ieee 802.16 networks. In *International Conference on Software in Telecommunications and Computer Networks*, pages 107–111, Split-Dubrovnik, Croatia.

Mai, Y.-T., C-C.Yang, and Wen, C.-J. (2009). A new cut-through mechanism in ieee 802.16 networks. In *World Academy of Science, Engineering and Technology*.

Mogre, S., D'Heureuse, P., Hollick, N., and Steinmetz, M. (2008). Core: Centrally optimized routing extensions for the ieee 802.16 mesh mode. In *33rd IEEE Conference on Local Computer Networks*.

Shetiya, H. and Sharma, V. (2005). Algorithms for routing and centralized scheduling to provide qos in ieee 802.16 mesh networks. In *WMuNeP*.