Intelligent and Adaptive Middleware to Improve User-Perceived QoS in Multimedia Applications

Pedro M. Ruiz
Agora Systems S.A
c/ Aravaca, 12 3B
E-28040 Madrid, Spain
Email: pedro.ruiz@agoratechnologies.com

Juan A. Botia, Antonio Gomez-Skarmeta, Pedro Martinez
University of Murcia
Dept. Information and Communications Engineering
Campus de Espinardo, E-30071 Murcia, Spain
Email: juanbot@um.es,{skarmeta,pma}@dif.um.es

***Abstract*— We investigate the use of adaptive multimedia applications to improve the user-perceived QoS in next-generation wireless networks. These networks will be characterized by the heterogeneity of the access technologies including ad hoc wireless network extensions. In these network scenarios in which the links are error-prone, their capacity is extremely variable and even the topology can change due to handovers and mobility of the nodes, it is unrealistic to think of traditional network-layer QoS mechanisms alone, as a means to offer strict end-to-end QoS guarantees. As a complement to the network-layer QoS protocols, this paper presents an intelligent adaptation middleware allowing multimedia applications to dynamically adapt their internal settings. This intelligence enables the applications to minimize the impact of the adverse and changing network conditions in the QoS level perceived by the user. We use a machine learning technique to model the user's perception of QoS from a large set of QoS scores given by real users. We present some results demonstrating that our proposal clearly outperforms traditional multimedia internetworking.**

I. Introduction

The vision of future 'beyond 3G' wireless and mobile networks usually includes the idea of an IP-based core network, accessed from a variety of heterogeneous wireless technologies, probably including self-organizing ad hoc networks. The reliable and high-capacity links used in today's core IP networks are unable to provide QoS in terms of guaranteed end-to-end delays or bandwidth. Thus, it is not reasonable to think that these future wireless networks, in which the wireless links have limited capacity, are error-prone, the users move and even the topology may change, are going to be able to offer this kind of guarantees. In these future network scenarios, network-layer QoS mechanisms are unable to protect multimedia applications against the variability of the end-to-end network resources which are available.

According to the definition of Quality of Service (QoS) given in the ITU-T recommendation ITU-E.800 [1], QoS is *"the collective effect of service performance, which determines the degree of satisfaction of a user of a service. It is characterized by a combination of service performance factors such as operability, accessibility, retainability and integrity"*. Therefore, the goal of any QoS mechanism in these scenarios should be maintaining a good level of user-perceived QoS even when the network conditions are constantly and unpredictably changing. One of the approaches which is gaining momentum to guarantee a good level of QoS in such network conditions, is the use of adaptive applications. The idea is to use these applications as a complement to the traditional network-layer reservations. So, whenever the network resources become scarce and the QoS guarantees are violated, the applications are informed to self-adapt their internal settings (e.g. changing coding schemes, frame rates, video sizes, etc.) to reduce their data rates to those that the network can support in that precise moment.

The idea of adaptive applications has been already used in [2] to improve end-to-end QoS in fixed networks. However, these results cannot be directly extrapolated to wireless networks, in which packet losses are not only due to congestion. In wireless links there are many other factors which are not under control and influence the instantaneous link layer capacity. In the field of wireless networks there is also much recent work ([3], [4], [5]) which shows that adaptive applications are able to perform better than traditional multimedia applications even when the network conditions are bad.

The novelty of our approach is that the adaptation mechanism has been optimized using genetic algorithms and the selection of the internal settings used by multimedia applications is performed so that the reduction of the data rate does not compromise the user-perceived QoS. This approach requires the characterization of the subjective components which define a user perception of QoS, and its relation with the different parameters which define the behaviour of the application (codecs, rates, etc.). As these subjective components are very difficult to model analytically, we use an intelligent rule induction algorithm to extract those components from evaluations given by real users. There are different combinations of settings which allow the application to reduce the bandwidth consumption to less than a required quantity. From all these combinations the application selects, using the induced rules, the proper combination of internal settings which offer a better user-perceived QoS.

The reminder of the paper is organized as follows: Section II describes the architecture for adaptive applications. Section III explains the modeling of the user perception. Finally, Section IV shows some empirical results derived from the use of our adaptive application approach over an ad hoc network, which is one of the most challenging wireless scenarios.

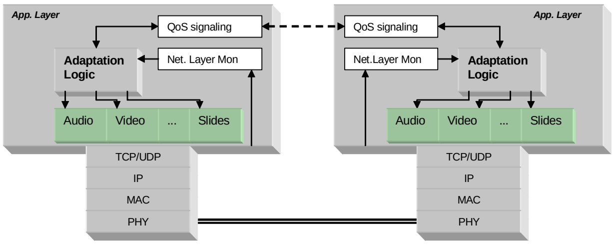

Fig. 1. Overall adaptation architecture

II. ADAPTIVE APPLICATION ARCHITECTURE

Traditional multimedia applications are a combination of one or more components (e.g. audio, video, etc.) which use the RTP[6] protocol to transport multimedia data. These applications can use the RTCP[6] protocol to receive some feedback from the applications at the other end, but their adaptation capabilities are very limited and usually they cannot adapt at all. In order to support real adaptive applications, we add some components to the architecture of a traditional real-time multimedia applications. These new components are in charge of the signaling of QoS information, and the provision of the intelligence to keep the user-perceived QoS at an acceptable level. So, the overall architecture, as shown in Fig. 1, consists of the following components:

- Multimedia Application Components
- QoS signaling mechanism
- Adaptation Logic

The QoS signaling mechanism is the protocol in charge of sending and receiving reports describing the network conditions from the other end. When such a report is received, it is passed to the Adaptation Logic so that it can decide which internal settings the application has to use to adapt to the current network conditions while providing a good user-perceived QoS.

A. QoS Signaling Mechanism

The QoS signaling is a key point of the adaptation architecture as it is the only feedback that the source has from the other end. It is basically an end-to-end transport mechanism for signaling data; no special protocol is needed. In fact, it may be enough with a TCP/UDP socket between sides, or even a standard protocol like the Session Initiation Protocol (SIP). The QoS signaling module calculates the loss-rate and mean delay experienced by the data packets in the network. This information is carried in a special signaling packet called ”QoS Report”, which is sent back to the source. A sequence number is used to deal with delayed QoS Reports. The rate at which QoS Reports are sent has to represent a good tradeoff between highly dynamic adaptation and a reduced traffic load. At the moment in our implementation it is fixed at one QoS report per second. However, the use of adaptive rates is being investigated.

An additional issue is that the QoS report packet itself has to traverse the network back to the server, and the probability of it actually making it there on time is inversely proportional to its importance. That is, a feedback packet is most important when it carries information about a congested network and it is not important when it is just saying that all is going well. Some experiments performed in [7] demonstrate that a UDP transport is much more appropriate to carry the feedback than a transport using TCP. The loss of QoS report messages may cause the applications not to have updated information about the network status. There are basically two approaches to overcome this problem: prioritizing QoS reports or allowing the applications to detect such losses. As the former approach requires a tight coupling with the network layer, we have found easier to make the receivers send periodic reports towards the sources. In this way, whenever network problems come up, the adaptation logic at the sender can detect missing reports. This information can be used by the adaptation logic to implement some heuristics based on that information (e.g. downgrading the bandwidth consumption when a certain number of QoS Reports are lost).

B. Adaptation Logic

The adaptation logic can be seen as a function which uses the QoS Reports and additional local information to decide which settings need to be configured in each of the different multimedia components. The adaptation logic solves the problem of adapting multimedia flows to the characteristics of the different networks or terminals including throughput and delay concerns.

The most difficult part for the Application Logic is the decision on what components need to be reconfigured and the concrete settings that each component has to use whenever it detects that the application is exceeding the available bandwidth. The difficulty lies in taking the correct decision to reduce the data rate while not compromising the user-perceived QoS. The most difficult part is dealing with the subjective user’s perception. For example, different users may have different preferences at different moments. So, it seems reasonable to apply artificial intelligence techniques like rule induction algorithms to a user-created set of scores to generate a rule set representing the user perception (objective and subjective). In the next section, we present how we model this perception in terms of bandwidth, loss rate, and the different settings supported by the multimedia application.

In general, most of the bad effects perceived by the user are due to packet losses. The most important input for the Adaptation Logic will be the end-to-end percentage of packet losses per reporting period. End-to-end delay problems may also make the user feel a bad quality but this problem can usually be avoided with a proper buffer management. Our adaptation algorithm, which is presented in Fig. 2, takes these objective facts into account, while the subjective ones are characterized in the next section and we consider them to be introduced inside the ’downgrade QoS’ and ’upgrade QoS’ primitives.

We have extended the ISABEL [8] application to dynamically (and in real-time) adapt its behavior to the available

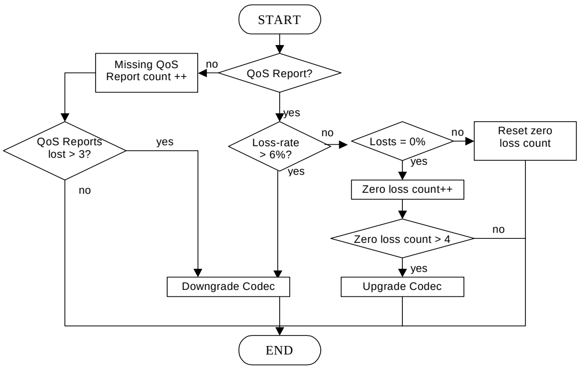

Fig. 2. Diagram of the adaptation logic

resources. The most important adaptation capabilities which we have implemented are re-configuration of the audio and video coding schemes, sampling rates, video size, video rates and video codec quantization.

III. Modeling QoS by rule induction

We optimized the timings which drive when to upgrade or downgrade the quality using a genetic algorithm in [9]. In this paper, as an extension to the baseline adaptation approach which we showed in [10], we focus on doing these upgrades and downgrades of the quality, providing the higher perceptual quality to the user. In general, there are different combinations of settings (e.g. codecs, frame rate, video size, etc.) which allow real-time adaptive multimedia applications to adapt their bandwidth consumption to the network conditions. However, all the different combinations are not perceived with the same quality by an user. For example, given some specific network conditions, it might be more desirable for an user having a bigger video size rather than a higher frame rate. This evaluation of user-perceived QoS depends on many aspects and it is very complex to model. In this section we present an approach to model the user perception of QoS, using a machine learning approach.

We use a rule induction algorithm called SLIPPER (Simple Learner with Iterative Pruning to Produce Error Reduction) [11] for the induction of a model representing the user perception of quality delivered by a multimedia application. Thus, SLIPPER works by generalization. It takes as an input a set of tuples with evaluations that the users have made about a concrete combination of settings in a concrete network scenario, and generates a set of rules which models the QoS perception of the user. Each tuple contains the network bandwidth, the audio and video codecs, the quantization of the video codec, the frame rate used, the video size, the loss-rate which produces this combination and the user's score in the range 1..5 (as recommended by ITU recomendation ITU-P.800 [12]) representing the perceived quality. In order to produce all this input tuples, we have setup a scenario using our adaptive video conference application, over a link emulator which emulates different network conditions. A total of 864

TABLE I

Parameters changed to generate user's valorations

Parameter	Values	Explanation
BW	33, 56, 88, 128, 384	Limit of network bandwidth
LOSS	0..100	% loss packets
AUDCOD	PCM, G711-u, G722, GSM	Audio codec
VIDCOD	MJPEG, H.263	Video codec
FSIZE	CIF, QCIF, 160x128	Size of video frames
QFVIDEO	5, 10, 15, 30, 60	Quantization factor of video codec
FPS	2, 6, 12	Frames by second sent
QoS	1, 2, 3, 4, 5	User perceived quality

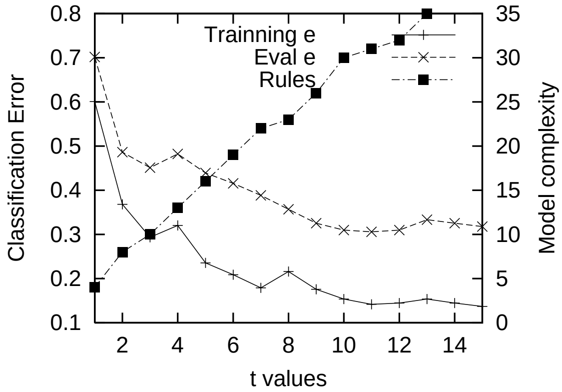

Fig. 3. Model accuracy vs. complexity of the model

different combinations of applications settings (in the ranges shown in Table I) have been tested, and scored by users.

The SLIPPER algorithm makes use of a configuration parameter, let it be denoted as t, taking values on the natural numbers, which corresponds to successive refinements of the rule set. The higher the value, the more accurate the model and the bigger its complexity (in number of resulting rules). We applied SLIPPER to the aforementioned data set, using values of t in the range 1..15. In figure 3 we have represented the evolution of the classification error during learning and during testing and the complexity of the rule set in number of rules. It is clear that there is a tradeoff between classification error and complexity.

From our point of view, the model with $t = 5$, represents an example of a good tradeoff which produces following rule set:

```
if matchConfidence {
 [QFVIDEO >= 60, VIDCOD = MJPEG,
  FSIZE = QCIF, LOSS <= 10, FPS >= 6] -> 2.8792
 [AUDCOD = GSM, BW >= 80,
  QFVIDEO >= 30, FSIZE = QCIF, FPS <= 6] -> 1.4357
 [AUDCOD = GSM, BW >= 128, LOSS = 0,
  QFVIDEO >= 30, FPS >= 3, VIDCOD = MJPEG] -> 1.7013
 [] -> -2.4188
} > 0 then 5 else if matchConfidence {
 [BW >= 384, QFVIDEO >= 40, FSIZE <= 2] -> 2.7121
 [QFVIDEO >= 30, VIDCOD = MJPEG,
  LOSS <= 3, AUDCOD = G722, BW >=80] -> 1.1756
 [FSIZE = CIF, QFVIDEO >= 30,
  LOSS <= 3, AUDCOD = G722, BW >= 80] -> 1.4437
 [] -> -1.5044
} > 0 then 4 else if matchConfidence {
 [LOSS >= 30] -> 2.1188
 [QFVIDEO <= 5] -> 1.4142
```

TABLE II

QUALITY STEPS FOR THE REAL-TIME ADAPTIVE APPLICATION

Step	A. Codec	V. Codec	V. Rate	V. Size	Q	Est. BW
0	GSM	-	0 fps	-	-	20 Kbps
1	GSM	MJPEG	4 fps	160x128	50	80 Kbps
2	G.722	MJPEG	8 fps	160x128	40	140 Kbps
3	G.722	H.263	6 fps	QCIF	50	190 Kbps
4	G.722	MJPEG	4 fps	CIF	30	230 Kbps
5	G.722	MJPEG	6 fps	CIF	50	350 Kbps

```
 [LOSS >= 16, FPS <= 3] -> 1.5438
 [] -> -1.0984207275826066
} > 0 then 1 else if matchConfidence {
 [LOSS >= 16] -> 1.9109
 [QFVIDEO <= 10, FSIZE = QCIF] -> 1.5861
 [FSIZE = 160X128, QFVIDEO <= 40, VIDCOD = H.263] -> 1.2546
 [] -> -0.3953
} > 0 then 2 else 3
```

The rule set extracted by SLIPPER allows us to identify which individual settings are most important for the user-perception of QoS and which is the relation between them. For example, the rules imply that the higher the frame rate the better the quality, but the user prefers changing from a higher frame rate to a lower one, provided that the video size is increased. Although there are big differences in bandwidth consumption between the different audio codecs, the rule set has successfully identified the proportional but non-linear relation between audio bandwidth and audio quality perception. The information given by the rule set, has been used to produce a concrete combination of settings among which the application will change depending on the network conditions. The derivation of these settings (see Table II) from the rules representing the user's QoS perceptions guarantees a good user-perceived QoS.

IV. EMPIRICAL RESULTS

In order to evaluate the effectiveness of our proposal, we have set up a real ad hoc testbed, on which we will compare the performance of real-time videoconferencing using traditional applications against our QoS-aware adaptive applications. The testbed has been deployed in the basement of the CS Faculty at the Univ. of Murcia (see Fig. 4). We use the Multicast MANET Ad hoc Routing Protocol (MMARP[13]), which is our new Multicast ad hoc routing protocol supporting both ad hoc nodes as well as standard IP Multicast nodes. MMARP nodes are numbered from 1 to 4, R is a standard-IP multicast receiver and S is a standard-IP multicast source. The source (S) follows (at walking speed) the path which is shown in Fig. 4 while it runs a videoconference session with node (R).

The route has been specifically selected so that link breaks and changes in MMARP routes occur during the videoconferencing session. Furthermore, we provoke signal strength changes due to the variation of the distance to MMARP nodes and the number of intermediate walls to traverse. This makes the available bandwidth vary during the session.

The trials have been performed using our own MMARP implementation for Linux and our RTP-based ISABEL-lite [8]

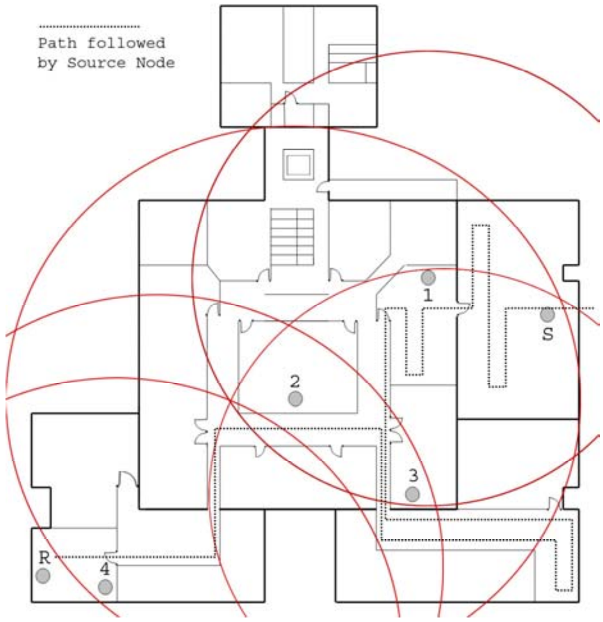

Fig. 4. Map of the testbed scenario

videoconferencing application to use our adaptive application framework. The settings which are used by the application are those extracted from the rule set and shown in Table II. This guarantees that the user-perceived QoS will be maintained as a consequence of keeping the loss-rate at a minimum. In order to guarantee a fair comparison of both approaches, the adaptive application starts with the quality step number 3, which is the only one used by the non-adaptive application, and is around the mean bandwidth which we calculated during the whole session. The results which we present are extracted from the RTP traces which are generated by the videoconferencing application. We have used the same route, at the same speed and in the same network conditions for the adaptive and non-adaptive trials.

The results presented in Fig. 5 show that the use of adaptive applications is able to reduce the overall packet losses both for audio and video to approximately one third. As expected, the differences are higher in the periods in which there is less bandwidth available.

The overall packet losses is a good reference to identify the points of the trial in which the network conditions are most critical. This is identified by an increase in the slope of the total packet loss curve. However, what really affects the user perception of QoS is the instantaneous loss-rate, which is what causes the service disruptions.

In Fig. 6, we compare the statistical histogram for the distribution of the audio loss-rate for both approaches. The same statistical analysis is performed for the video flow in Fig. 7. For example, for the audio flow, the adaptive application approach is able to keep the loss-rate below 10% all the time. In fact, it keeps the loss-rate below 5% during the 91% of the time. For the video flow, the loss-rate is kept under the 5% the 64% of the time, and its has been under the 10% the 78%

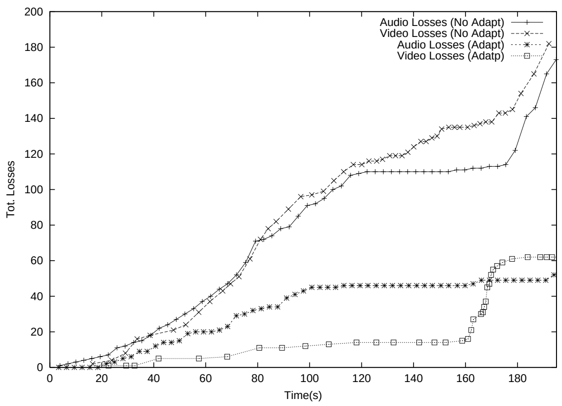

Fig. 5. Variation of the total losses over time

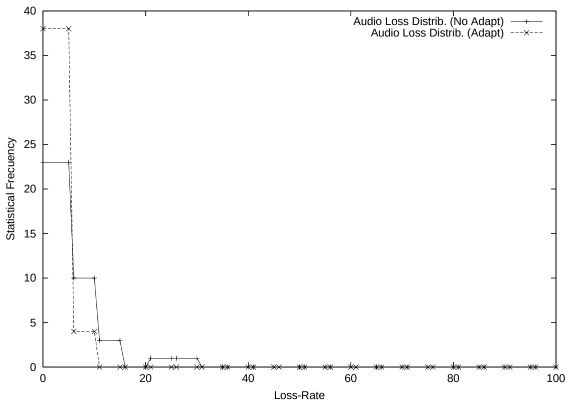

Fig. 6. Histogram for audio loss-rate

of the time.

These figures clearly demonstrate that the adaptive application approach on top of our MMARP implementation has been able to offer a very good user-perceived QoS in a scenario in which traditional multimedia applications offer a highly variable quality. Furthermore, both for audio and video flows the adaptive applications approach highly reduces the loss-rate, demonstrating that adaptive applications are a good approach for dealing with the changing network conditions by which ad hoc networks are characterized.

V. Conclusions and Future Work

We have used a mobile ad hoc network testbed, to demonstrate the suitability of adaptive applications to operate in mobile and wireless scenarios in which the QoS cannot be guaranteed at the network layer. In such scenarios, adaptive applications play an important role in improving the user-perceived QoS. In addition to the adaptation of the data rates generated by the applications to the network conditions, our intelligent adaptation approach takes into account the user's perception of quality. The user-perceived QoS has been modeled using a rule induction algorithm, allowing real-time adaptive applications to offer a good level of quality in situations in which traditional multimedia applications perform badly.

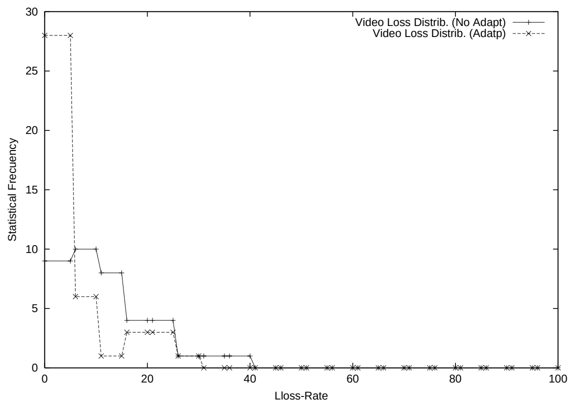

Fig. 7. Histogram for video loss-rate

As a future work, we are extending this platform to support real-time adaptation not only to the network conditions, but also to the user's context, preferences, terminal resources, terminal capabilities, etc.

Acknowledgement

Part of this work has been funded by the ISAIAS project (TIC2000-0198-P4-04) which is partially funded by the Spanish Education and Culture Ministry.

References

[1] ITU-T Recomendation E.800 (0894): *Terms and definitions related to quality of service and network performance including dependability*

[2] D. Sisalem, *"End-to-end Quality of Service Control using Adaptive Applications"*. IFIP International Workshop on Quality of Service, 1997.

[3] A. Alwan, R. Bagrodia, N. Bambos, M. Gerla, L. Kleinrock, J. Short, and J. Villasenor, *"Adaptive Mobile Multimedia Networks"* IEEE Personal Communications, April 1996, pp. 34-51.

[4] M. Mirhakkak, N. Schult, and D. Thomsom, *"Dynamic Bandwidth Management and Adaptive Applications for a Variable Bandwidth Wireless Environment"* IEEE JSAC, October 2001, pp. 1985-1997.

[5] R. Ramanathan, and R. Hain, *"An Ad Hoc Wireless Testbed for Scalable, Adaptive QoS Support"* IEEE WCNC, November 2000, pp. 998-1002.

[6] H. Schulzrinne, S. Casner, R. Frederik, V. Jacobson, *"RTP: A Transport Protocol for Real Time Applications"*, IETF, RFC 1889, January 1996.

[7] M. Kazantzidis, L. Wang, M. Gerla, *"On Fairness and Efficiency of Adaptive Audio Application Layers for Multihop Wireles Networks"*. Proceedings of IEEE MOMUC'99, San Diego, CA, November, 1999.

[8] *The ISABEL CSCW application*.[On line], http://www.agora-2000.com/productos/isabel.html.

[9] P.-M. Ruiz, A. Gomez-Skarmeta, *"Using Genetic Algorithms to Optimize the Behaviour of Adaptive Multimedia Applications in Wireless and Mobile Scenarios"*. IEEE WCNC'2003, March, 2003. New Orleans.

[10] P.-M. Ruiz, E.-J. Garcia, *"Improving User-perceived QoS in Mobile and Wireless IP Networks Using Real-Time Adaptive Multimedia Applications"*. Proceedings of (PIMRC'2002). Lisbon, September, 2002. Vol. 3, pp. 1467–1471.

[11] W.-W. Cohen and Y. Singer, *"A Simple, Fast, and Effective Rule Learner"*. Proceedings of the Conference of American Asociation for Artificial Ingelligence, 1999.

[12] ITU-T Recomendation P.800 (1996): *Subjective Quality test based on Mean Oppinion Scores (MOS)*

[13] P.-M. Ruiz, A. Gomez-Skarmeta, I. Groves, *"The MMARP Protocol for Efficient Support of Standard IP Multicast Communications in Mobile Ad hoc Access Networks"*. In proc. of the IST Mobile & Wireless Summit 2003, Vol II, pp.478–483. Aveiro, June, 2003.