Using Genetic Algorithms to Optimize the Behaviour of Adaptive Multimedia Applications in Wireless and Mobile Scenarios

Pedro M. Ruiz
R&D Department
Agora Systems S.A.
E-28040, Madrid (Spain)
Email: pedro.ruiz@agoratechnologies.com

Antonio F. Gomez-Skarmeta
Dept. Information Engineering Communications
Facultad de Informtica, Univ. of Murcia
E-30001, Aptdo. 4021, Murcia (Spain)
Email: skarmeta@dif.um.es

Abstract— **Adaptive applications are a key concept to take into account when dealing with multimedia internetworking in wireless and mobile environments in which abrupt changes in the network conditions may occur. In these scenarios not all the problems in the multimedia quality are due to congestion - as it happens in fixed networks - but also to interference, mobility, and many other issues. This paper presents an adaptation architecture allowing applications to use adaptation techniques at the application layer to minimize the impact of adverse and changing network conditions. This allows maintaining the QoS perceived by the user in an acceptable level. In addition, we optimize the adaptive behavior using genetic algorithms to estimate the optimal triggers for the adaptation function. We present some results comparing the fitness of using such adaptive techniques against the traditional multimedia internetworking.**

I. Introduction

There are many real-time multimedia applications which are able to distribute audio, video and many other streams. However, only a few of them have mechanisms for taking advantage of QoS and network status information like packet losses, delay variation or available bandwidth. These applications are also not able to adapt automatically to changing network conditions to offer a good user-perceived QoS. Thus, adverse network conditions - which are common in ad hoc environments - like packet losses, abrupt bandwidth changes and substantial delay variations cause excessive degradation in the quality which is perceived by the user. Some studies like [1] show that for non-adaptive applications the user-perceived audio quality starts becoming extremely bad when packet losses go over 20% even when packet retransmission techniques are used to replace lost packets.

The idea of adaptive applications was already known for fixed networks. For example, [2] applies this idea to end-to-end QoS in fixed networks. However, these results are not directly applicable to wireless and mobile networks. The reason is that in fixed networks packet losses are usually due to network congestion while in wireless networks there are many factors which influence the end-to-end QoS such as access to the radio link, interferences, fading, etc. There are also some important related works ([5], [6], [7]) identifying the need of adaptive applications. In [3] changes of data rate were enough to improve the audio or video quality. There are works focused on wireless networks like [3] and [4] which offer some interesting ideas like using speech recognition or text files at the sender which travel as plain text and then are displayed or even passed to a speech synthesizer which reproduces them. However, they need some requirements like making the user provide an estimation of the available bandwidth which may not be always met in future mobile and wireless networks.

Typical QoS provisioning solutions have been always focused on making delay-sensitive traffic to get a better treatment at the network layer. This approach has been demonstrated to work very well in fixed networks. However, in mobile and wireless networks - especially those based on multihop ad hoc nodes - the network is not always able to offer any guarantee. These networks are characterized by continuously changing network conditions and an unpredictable radio link performance. As we commented, congestion is not the only cause of packet losses in wireless networks. We will demonstrate that in such kind of networks a good QoS level can be achieved if the application tries to adapt itself to the changing network conditions rather than trying to take advantage of non-guaranteed bandwidth reservations.

What makes our approach different from the previously commented adaptive application approaches is that we make the application adapt not only audio flows but also video components. In fact we use artificial intelligence algorithms to control many other settings like codec, codec-specific parameters (e.g. Quality factor for MJPEG), frame sizes, components selection etc. In addition, our approach uses a novel signaling logic to deal with signaling-packet losses due to bad network conditions.

The reminder of the paper is organized as follows: Section II describes the architecture for adaptive applications. Section III describes the genetic algorithm which we have used. Finally, Section IV shows some empirical results derived from the use of our adaptive application approach in extreme conditions.

0-7803-7701-X/03/$17.00 (C) 2003 IEEE

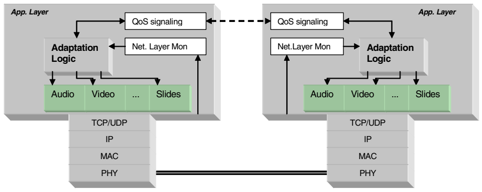

Fig. 1. Overall adaptation architecture

II. Architecture for Application Adaptation

Quality of Service (QoS) as defined in ITU-T recommendation E.800, ITU-E.800 [8] 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. Thus, the user plays an important role in QoS evaluations. We use a middleware offering additional adaptation features under the application layer. This middleware will allow applications to offer a better user-perceived QoS even in environments in which traditional multimedia applications would perform badly. The main items in this architecture are the following:

- Multimedia Application Components (audio, etc.)
- QoS signaling mechanism
- Adaptation Logic

Fig. 1 shows all these architectural parts. 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 as an additional input. Additionally, the Adaptation Logic is in charge of deciding which set of parameters is best suited to the current network conditions.

A. Adaptation Logic

The most difficult part for the Application Logic is the decision on what components to adapt when the application is exceeding the available bandwidth. The difficulty lies in taking the correct decision while taking into consideration the subjective facts about it. For example, different users may have different preferences at different moments. So, it seems reasonable - and in fact is being prepared for future work - to apply artificial intelligence techniques like fuzzy clustering to a user-created set of rules to generate a new set of rules in terms of bandwidth, loss rate, frames per second, etc.

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 facts into account. These parameters have been set according to our own experience but in this paper we improve them using a genetic algorithm.

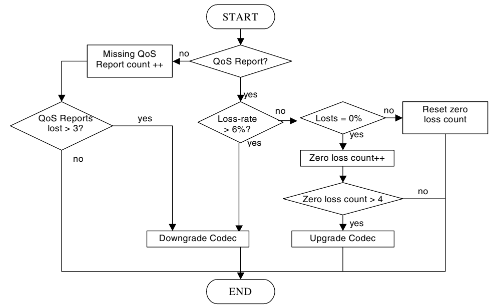

Fig. 2. Diagram of the initial adaptation logic used

Seq	% lost	Delay	User preferences	Estimated BW

Fig. 3. Components of a QoS Report message

B. QoS Signaling Mechanism

The QoS signaling is another 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 Session Initiation Protocol (SIP). In our implementation this quality will be measured in terms of percentage of packets lost and mean delay experienced by the data packets in the network. This information is carried in a special signaling packet called ”QoS Report”. A sequence number is used to deal with delayed QoS Reports. The rate at which QoS Reports are sent is 1 second which is a good tradeoff between highly dynamic adaptation and a reduced traffic load. A QoS report message presents the structure shown by Fig. 3.

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 [9] demonstrate that a UDP transport is much more appropriate to carry the feedback than a transport using TCP. To overcome the loss of a QoS Report message, the receivers send periodic reports towards the sources. In this way, whenever network problems come up, the sender can detect missing reports which are used in the heuristic of downgrading the QoS when a certain number of QoS Reports are lost.

The adaptation capabilities which we are considering have been implemented as extensions to the ISABEL [10] application. These capabilities include real-time changes in the codecs, sampling rates, size of the components, dynamic buffer management and real-time adjustment of the components to use.

0-7803-7701-X/03/$17.00 (C) 2003 IEEE

TABLE I

ADAPTIVE BEHAVIOURS TO BE TESTED

Parameter	1-Gen. Values	Human Values	GA Values
Packet loss-rate to downgrade	35	4	2
Num. of 0%-loss reports to upgrade	4	5	10
Num. of missing reports to downgrade	3	4	3

C. Adaptation Capabilities

The ISABEL [10] application has been extended to dynamically (and in real-time) adapt its behavior to the available resources. The most important adaptation capabilities implemented are as follows:

1) Coding schemes. The application may handle contents encoded in several standards such as MPEG/H.261/H.263/MJPEG for video, or GSM/G.722/G.711 for audio. The application chooses among them at user request or based on the information received from lower layers.
2) Sampling rate. Transmitting at a lower frame (or sampling) rate means saving bandwidth, and a fair quality is often achieved at less than 24 fps.
3) Component size. In environments with scarce bandwidth the user will prefer seeing smaller videos than bad quality ones in which most of the frames are lost.
4) Component use. In very constrained bandwidth scenarios the user may prefer using some components instead of using all of them with a poor quality.
5) Buffering. Intelligent and dynamically adaptable buffers help offering a better quality in adverse network conditions, alleviating the effects of jitter and delay.

III. OPTIMIZATION OF THE ADAPTATION LOGIC

Two key decisions define the adaptive performance: when and how to do it. How to do it strongly depends on the user expectations and we are analyzing as future work the employment of more complex artificial intelligence techniques like fuzzy clustering. As the focus is on optimizing the when, in this paper, the how is covered with the simplest approach of using different audio codecs. The when is to be decided using a genetic algorithm.

In Fig. 2, we present the adaptation algorithm. As it can be seen, the real behavior of that adaptation highly depends on three different parameters: the loss percentage to downgrade, the number of consecutive 0% reports received before upgrading and finally the amount of QoS reports lost which provoke a downgrading. Given the fact that packet losses are the main issue causing the user to feel a bad quality in the multimedia flow and, provided that experiments demonstrate that 20% packet losses is when users start getting a bad audio quality, the initial values for this parameters where fixed according to our knowledge as shown in Table I.

In some other papers like [11] we have demonstrated the goodness of the adaptive applications approach as well as this concrete adaptation algorithm. As long as trying all the combinations to guess the better combination would require a lot of time, in this paper we use a genetic algorithm to calculate the optimal values for these parameters. That is, the genetic algorithm will perform thousands of executions of the our adaptive application ending up after a certain number of generations in an small group of possible solutions from which it extracts the better combination of parameters.

The most difficult part when using the genetic algorithm is to tune its parameters (like number of individuals per generation, mutation probability, crossover probability, number of generations, etc.) appropriately. After several executions we found that 30 individuals per generation, a probability of crossover of 0.6 and 0.02 as the mutation probability were good values for our problem.

The most important part of a genetic algorithm is the fitness function, which defines how good is a interim solution being found by the genetic algorithm. It must perfectly represent the problem to be solved. In our case the fitness function is derived from our knowledge about the relative effect of the packet losses and the codec in the "user perceived QoS". So, in (1) the fitness value is calculated as the weighted sum of two different functions. The codec function gives a value between 1 and 4 depending on the quality of the codec used for that packet. The better the codec the higher the value. The other term is calculated once for each reporting period. That is why it is multiplied by the number of packets per period to normalize both values. In addition, a high constant value k_1 corresponding to the maximum codec fitness that can be obtained per reporting period and the number of packets per period (k_2) are used to make the second term of the function give always positive values. Finally, the packet loss appears squared in the equation so that it has more impact in the fitness function. The constants w_1 and w_2 are the weights of the different parts of the fitness function. In our case, $w_1 = 0.11$ and $w_2 = 0.89$.

$$\begin{aligned} \phi_1 \quad &= \quad w_1\left(\frac{k_1}{maxcodec}\sum_{j=1}^{N_{packets}} codec(j)\right) \\ &+ \quad w_2\left(k_2\sum_{j=1}^{N_{periods}}(k_1-(pktloss)^2)\right) \end{aligned} \tag{1}$$

To calculate the fitness, the whole RTP [12] session is logged and processed to extract and calculate the parameters of the fitness function.

The results from the genetic algorithm are in Table I. Additionally, to demonstrate the improvements of the genetic algorithm we also use in our tests one of the individuals before being evolved. This one is also shown in Table I.

In the next section we will assess how good is this solution compared to the use of standard multimedia applications without adaptation capabilities and the other two approaches commented above.

IV. NUMERICAL RESULTS

In order to assess the goodness of the adaptation driven by the genetic algorithm we have prepared a testbed scenario

TABLE II

AUDIO CODECS USED IN THE TRIALS

Codec	Sample Depth	Sampling Rate	Estimated BW
PCM	16 bit	8000 Hz	$\sim$ 128 Kb/s
G.711	16 bit	8000 Hz	$\sim$ 64 Kb/s
G.722	16 bit	8000 Hz	$\sim$ 40 Kb/s
GSM	16 bit	8000 Hz	$\sim$ 15 Kb/s

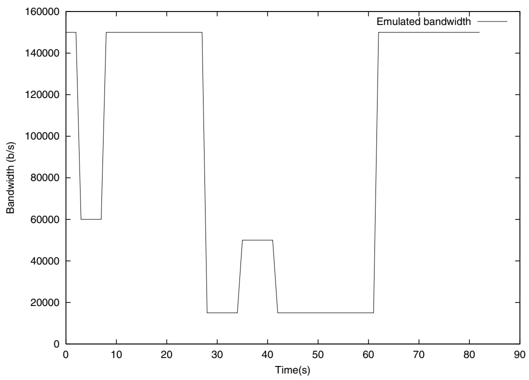

Fig. 4. Bandwidth changes emulated in the virtual link

including different kind of bandwidth changes not only in data rate but also in duration. On this testbed we will test four different types of applications:

- Traditional applications which are not able to adapt to the network conditions
- Adaptive application with human-set parameters
- Adaptive application with non-evolved parameters
- Adaptive application with parameters from the GA

To be sure that we are fair comparing the different kinds of adaptive behaviors, instead of performing the trials over real radio links or ad hoc networks we have developed a link emulation tool which is able to reproduce specific link properties in user-configured periods of time. Additionally, we focus our trials in audio transmission which is much more delay-sensitive than other media. To perform the trials we use the ISABEL-Lite application which is a reduced version of ISABEL [10] which we have specially designed for mobile scenarios and we have extended with adaptive behavior. The audio codecs used in our trials are summarized in Table II.

The bandwidth changes and timings which are emulated in order to assess the goodness of the different approaches are shown in Fig. 4.

The previously commented applications have been tested in this same scenario under the same emulated link conditions. The trials have been repeated many times to be sure that the emulation is correct and no anomalous values are produced among different simulations. Fig. 5 shows the packet losses which are experienced under the three different adaptation schemes. As can be noticed the adaptation driven by the parameters found by the genetic algorithm outperforms all the other approaches. The amount of packet losses is reduced more than 10 times. In fact, the GA-based approach has 45% less packet losses than the human-configured adaptation

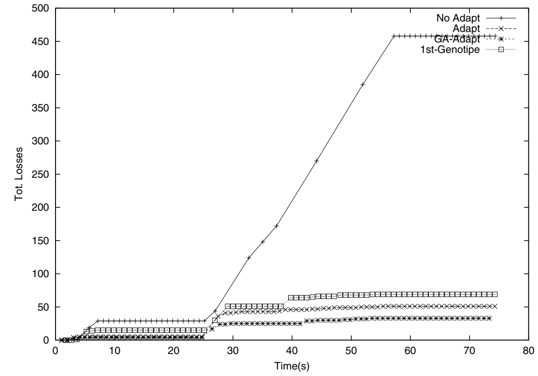

Fig. 5. Total packet losses with the different behaviours

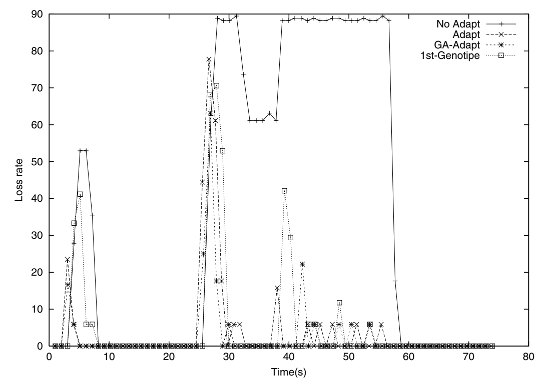

Fig. 6. Loss rate for the different approaches

These results are also corroborated by the loss percentages which are shown in Fig. 6. As it can be noted, in low bandwidth periods most of the packets sent by the non-adaptive application are lost. However, for adaptive applications only the first strong bandwidth reduction seems to have an important impact. In any case, this impact is reduced to less than a couple of seconds. Again, the GA-adaptation approach demonstrates that it is able to adapt better than the other approaches. In fact, rarely the packet losses go over 20% which is when the quality start being bad.

The latter proposition gets clearly demonstrated in the histogram for the loss percentage distribution, which is shown in Fig. 7. This figure shows how most of the values for the adaptive case are fewer than 20% of packet losses - in fact the most of the values are actually zero - while the non-adaptive application has many points going beyond 70% and most of them in the 90-100% frame.

V. CONCLUSION

We have defined and implemented an adaptive multimedia application framework for mobile and wireless applications. Its performance has been shown in an emulated link scenario. Through extensive emulation and experiments, this paper has demonstrated the benefits that adaptive multimedia applications can provide for achieving a much better fulfillment of the

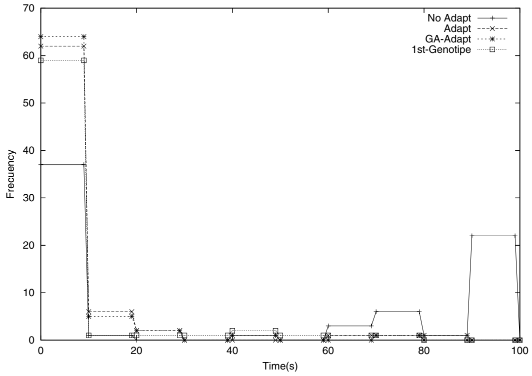

Fig. 7. Statistic histogram of loss-rate values

user expectations - especially in wireless and mobile networks in which end to end QoS cannot be always assured. In addition, it has also been demonstrated that the use advanced problem solving methods like genetic algorithms allow us to improve even more the good results offered by adaptive application approaches.

As a future work, improvements in reaction are related to the use of intelligent techniques not only for the selection of when to trigger the adaptation but also to decide the specific multimedia parameters (like codecs, codec's quality factors, etc.) which provide a better user-perceived QoS.

ACKNOWLEDGMENT

This work has been partially funded in the framework of the 6POWER IST-project (IST-2001-37613), which is partially funded by the European Union. The authors would like to thank the contributions from their colleagues Emilio Garcia and Juan A. Sanchez.

REFERENCES

[1] J.-C. Bolot and A. Vega-Garcia, *The case for FEC-Based Error Control for Packet Audio in the Internet*, ACM Multimedia Systems, 1998.
[2] D. Sisalem, *End-to-end Quality of Service Control using Adaptive Applications*. IFIP International Workshop on Quality of Service, 1997.
[3] M. Kazantzidis, S.-J. Lee , M. Gerla, *Permisible Throughput Network Feedback in AODV MANETSs*. Proceedings of ICC 2001, Helsinki, Finland, June 2001.
[4] T.-W. Chen, M. Gerla, M. Kazantzidis, Y. Romanenko, I. Slain, *Experiments on QoS Adaptation for Improving Enduser Speech Perception over Multihop Wireless Networks*. Proceedings of QoS Mini Conference in conjuntion with IEEE ICC'99, Vancouver, Canada, June 1999.
[5] 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.
[6] 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.
[7] R. Ramanathan, and R. Hain, *"An Ad Hoc Wireless Testbed for Scalable, Adaptive QoS Support"* IEEE WCNC, November 2000, pp. 998-1002.
[8] ITU-T Recomendation E.800 (0894): Terms and definitions related to quality of service and network performance including dependability.
[9] 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.
[10] *The ISABEL CSCW application*.[On line], http://www.agora-2000.com/productos/isabel.html.
[11] 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.
[12] H. Schulzrinne, S. Casner, R. Frederik, V. Jacobson, *RTP: A Transport Protocol for Real Time Applications*, IETF, RFC 1889, January 1996.