IMPROVING USER-PERCEIVED QOS IN MOBILE AND WIRELESS IP NETWORKS USING REAL-TIME ADAPTIVE MULTIMEDIA APPLICATIONS

Pedro M. Ruiz[1], **Emilio García**[2]

[1] Agora Systems, S.A., C/ Aravaca, 12 3ºB 28040 Madrid, Spain, pedro.ruiz@agoratechnologies.com
[2] Agora Systems, S.A., C/ Aravaca, 12 3ºB 28040 Madrid, Spain, emilio.garcia@agoratechnologies.com

Abstract – Adaptive applications are a key concept to take into account when dealing with multimedia internetworking in wireless and mobile environments, in which network conditions may change abruptly and not always due to congestion –as it happens in fixed networks– but also to interference, mobility and so on. This paper presents an adaptation architecture which allows applications to minimize the impact of adverse and changing network conditions, by using self-adjustment techniques at high layers, obtaining information from the network. The objective is maintaining the Quality of Service (QoS) perceived by the user at an acceptable level. Results are presented by means of some tests in a typical ad-hoc scenario.

Keywords – Multimedia, adaptability, mobility, ad-hoc, videoconferencing, QoS.

I. INTRODUCTION

There are many real-time multimedia applications which are able to distribute audio, video and many other streams. However, few of these applications 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 and mobile 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 user-perceived audio quality starts becoming extremely bad when packet loss rate goes over 20% even when packet retransmission techniques are used to replace lost packets. This demonstrates the need for adaptive applications and the appropriate signalling mechanisms.

The idea of adaptability was already known for fixed networks [2]. However, these results are not directly applicable to wireless and mobile scenarios: in fixed networks packet losses are usually due to network congestion, while in a wireless environment there are many other factors such as access to the radio link, interferences, fading, etc. In that work, changes of data rate were enough to improve the audio or video quality. There are works focused on wireless networks ([3, 4]) which offer some interesting ideas like using speech recognition or text files as an alternative. However, they handle the adaptation only by changing the sampling rate and packet size. In addition, the bandwidth value needs to be provided by the user, and the source and destination are required to be synchronized via NTP to calculate the variation delay. These requirements may not be always met in present or future mobile and wireless networks.

Typical QoS provisioning solutions have been always based on the idea of trying to reserve or assure certain network guarantees so that packets coming from delay or bandwidth sensitive applications receive a better treatment in the network. This approach has been demonstrated to work very well in fixed networks. However, in wireless networks – especially those based on multihop ad-hoc nodes– it is not always possible to offer any guarantee, due to continuously changing conditions and unpredictable radio link consumption. We will demonstrate that in such kind of networks a good QoS level can be achieved if the application tries to adapt itself by reacting to what it senses from the network rather than relying on non-guaranteed bandwidth reservations.

Our approach differs from other solutions in that many other settings like codec, codec-specific parameters (e.g. Quality factor for MJPEG or MPEG), frame sizes, selection of components, etc. are taken into account. In fact, none of the previous solutions considers video transmission, where these parameters take most of the importance. In addition, we use a novel logic to deal with signalling packet losses due to bad network conditions. In this paper we implement a sub-set of the BRAIN End-user Terminal Architecture (BRENTA [5]) functionalities, specifically the application layer QoS signalling architecture.

The reminder of the paper is organized as follows: Section II describes the architecture for adaptive applications and our implementation and Section III shows some empirical results derived from the use of our adaptive approach in extreme network conditions: an ad-hoc network with different traffic loads.

II. ARCHITECTURE FOR APPLICATION ADAPTATION

Quality of Service (QoS) is defined in ITU-T recommendation E.800 [6] as *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. Placing some additional features in the application layer would allow presenting a better QoS to the user in environments in which traditional solutions would perform badly. The main items in this architecture are shown in Fig. 1.

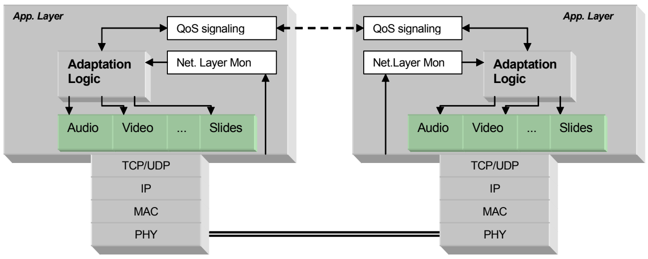

Fig. 1. Framework for adaptive applications

The QoS signalling 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 main problem at the application layer with respect to mobile and wireless networks is that of adapting data transmitted to or received from the application to the characteristics of the different networks, including throughput and delay concerns.

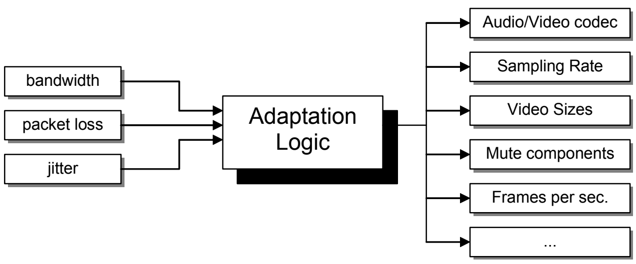

Fig. 2. Operation of the adaptation logic.

Therefore, as Fig. 2 shows, the Adaptation Logic can be seen like a somewhat complex function, where output has to take into consideration the subjective aspects of user preferences. 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 the standard inputs.

Most of the bad effects perceived by the user are due to packet losses, and therefore 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 without needing to reduce the used bandwidth.

The detailed description of the Adaptation Algorithm which is applied to every QoS report received is presented in Fig. 3. As shown, a downgrade in the quality will only be performed when the end-to-end packet losses percentage exceeds 5% or 3 consecutive QoS Reports are lost – possibly by a network congestion or too much interference. Additionally, the quality is upgraded to the estimated bandwidth value whenever 4 consecutive QoS reports indicating 0% packet loss arrive. These parameters have been set according to our own experience. However, they could be tuned according to the testbed trials and they could also be dynamically calculated according to the concrete network scenario. Finally, intelligent techniques like supervised learning could also be applied to calculate these values. The difficulty of applying such techniques derives mainly from the existence of subjective components in the user-perceived QoS which may vary from person to person and which are very difficult to model mathematically.

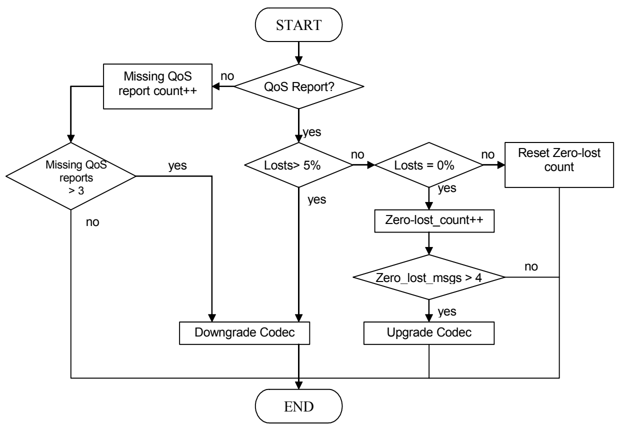

Fig. 3. Schematic diagram of the adaptation logic.

B. QoS Signalling Mechanism

The QoS signalling 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 signalling data; no special protocol is needed. In fact, it may be enough with a TCP/UDP socket in both sides, and 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 signalling packet called “QoS Report”. A QoS report message presents the structure shown in Fig. 4.

Type	% Lost	Delay	User Preferences	Estimated BW

Fig. 4. Format of a QoS report message.

Some experiments performed in [7] demonstrate that a UDP transport is much more appropriate to carry the feedback than a transport using TCP. TCP retransmissions result in stale QoS information especially on a congested network.

An additional issue is that the feedback 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. There are at least two approaches for solving that: prioritization of QoS reports, and periodical sending of reports. The first one is related to the idea of giving a higher priority in every router within the intermediate network to these signalling messages. In this way, when intermediate routers have data to send, they will firstly forward signalling messages and then the rest of the data packets. Thus signalling messages are given a higher probability to reach the destination. This mechanism works very well but its main drawbacks are related to its difficult implementation – especially in ad-hoc networks. We have used the second approach which is based on the idea of the clients sending periodic reports towards the sources. This way, whenever network problems come up, the sender can detect missing reports because it is expecting them at periodical intervals. The sender uses the heuristic of downgrading the quality when a certain number of QoS Reports are lost.

C. Bandwidth Signalling Mechanism

Apart from the discussed end-to-end QoS signalling, applications can benefit from a local feedback on actual bandwidth capabilities of the network in use:

- The local terminal can quickly adapt the characteristics of its outgoing data flow to match the bandwidth of which it has been informed. It can also send this information to its peer, not only referring to the outgoing bandwidth, but also the incoming capabilities.
- The peer terminal can use the information received, and check it against its own capabilities, to select the best quality available to send.

This signalling mechanism is complementary to the first one in the sense that the locally sensed bandwidth of the network cannot always be assumed to be a good guess on the real bandwidth available. Thus, data streams are shaped not only based on bandwidth, but in all the rest of network parameters.

D. Adaptation Capabilities

The ISABEL [8] application has been extended to dynamically (and in real-time) adapt its behaviour to the available resources. It handles different RTP [9] flows for each service, but our main focus will be in audio and video. The most important adaptation capabilities implemented are as follows:

- **Codecs.** The application may handle contents encoded in several standards such as H.263/MJPEG for video, or GSM/G.722/G.711 for audio, choosing among them at user request or based on information lower from layers. Each standard has its own restrictions and parameters, and the application may choose among them at user request or based on information from the lower layers. The key feature which we have added is that the parameters of the codec are controlled by the Adaptation Logic and codecs can be changed dynamically avoiding any session re-establishment delay.
- **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.
- **Component size.** The bandwidth requirements of a video transmission or a slide presentation depend heavily on the size of the videos or images. In addition, this adaptability allows for terminal heterogeneity regarding the display capabilities of each device. The bigger the video size, the higher the bandwidth consumption. In scarce bandwidth environments the user will prefer seeing smaller videos than bad quality ones in which most of the frames are lost.
- **Component use.** In very constrained bandwidth scenarios the user may prefer using some components instead of using all of them with a poor quality. For example, in a 30 Kbps scenario the user may prefer just receiving a GSM audio flow without any losses rather than receiving a low-quality video and an unintelligible audio stream.
- **Buffering.** Intelligent and dynamically adaptable buffers help offering a better quality in adverse network conditions, lessening the effects of jitter and delay.

III. RESULTS

To demonstrate how adaptive multimedia applications can improve the user-perceived QoS in the case of ad-hoc networks, we have deployed a network testbed in which the source and destination of multimedia data are connected by a 1-hop ad-hoc network. As Fig. 5 shows, an additional host is introduced in the same radio range in order to generate interferences – both by taking part of the bandwidth and also accessing to the shared media. The idea is not to simulate a real environment but to test the application reaction under different interference level. Therefore, four interference periods of increasing load, 100s long each, have been planned for the tests, but also we have introduced periods of 50s without interference so that the different tests can start from the same initial conditions.

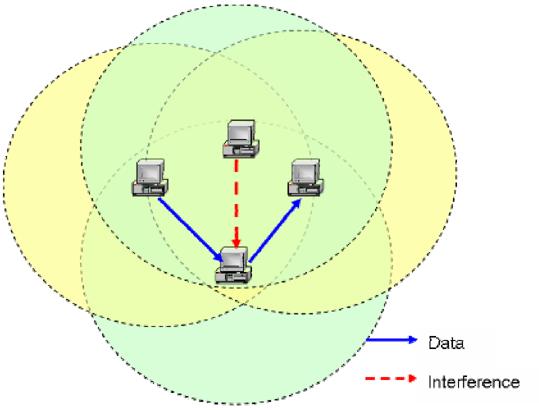

Fig. 5. Multihop ad-hoc network.

For these tests we have selected the audio part of ISABEL, which represents the most sensitive component when losses appear. A set of four different codecs has been used, as shown in Table 1.

Table 1
Audio codecs available.

Id	Codec	S. Rate	S. Depth	Bandwidth
1	GSM	8000 Hz	16 bit	~13 Kbps
2	G.722	8000 Hz	16 bit	~32 Kbps
3	G.711	8000 Hz	16 bit	~64 Kbps
4	PCM	8000 Hz	16 bit	128 Kbps

The adaptation strategy is based on the one shown in Fig. 3, having the statistics generated by RTCP reports. To make the application dynamically switch among the codecs, we have tested it under different levels of network congestions, with adaptive capabilities deactivated first, and then turned on. When no adaptation is programmed, the application uses only the higher bandwidth codec (PCM).

A traditional application in such kind of environment starts losing packets as the interference periods are introducing a higher load (Fig. 6). When adaptive behaviour is used, the overall number of packets is strongly reduced in a 66% (from 6981 packets to 2387). The packet losses which are produced in the adaptation part are due to the losses not exceeding the loss threshold (5%) that the application requires to trigger the adaptation process. As we show in the next figures, these small losses are not enough for damaging the user perception of the audio because the loss percentage in most of the cases is maintained below the 20% which many studies [1] have identified as the point from which the audio quality starts getting noticeably poor.

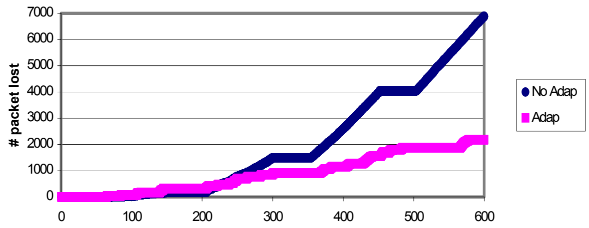

Fig. 6. Packet losses with and without adaptation.

If we observe the variation of the instantaneous packet loss percentage over time –which is shown in Fig. 7– we can see that, as expected, the adaptive application tends to use the lower-bandwidth codecs in highly loaded periods (350-450 sec and 500-600 sec). This is represented in the graph as a number of steps in the codec "number" (see "Id" column in Table 1), which goes from 4 (PCM) to 2 (G.722). Conditions are never so bad for GSM (1) to be used.

This way, even in such heavy loaded situations, most of the packet loss percentages are successfully kept at 0%.

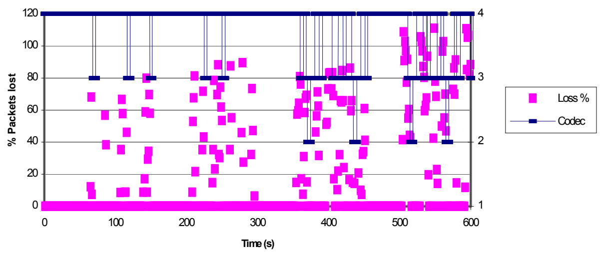

Fig. 7. Losses distribution and codec changes.

These results are also supported by the figures of the packet loss distribution during time. Its mean value has been calculated for both approaches and we have found that using adaptive applications, the mean value is 14% versus the 34% in the case of no adaptation. However, this value is not really significant if it is not taken in conjunction with the distribution. As we present in Fig. 8, in the period of higher load (500 to 600 sec), when no adaptation is used, although the mean value is 34%, most of the values are really in the range 75%-100%. However, in our case although the mean value is 14%, most of the values are in the range 0%-14% being 0% the most repeated value.

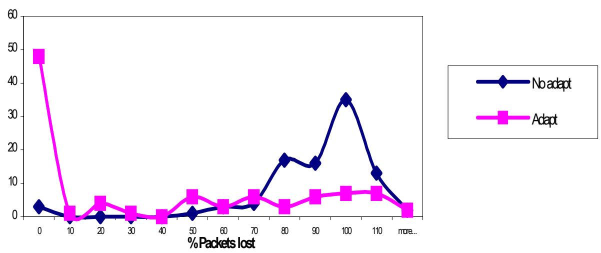

Fig. 8. Packet loss distribution with high interference.

This demonstrates that except for very punctual peaks, the adaptive approach offer a good quality audio (under 20% losses) in situations in which traditional applications are unable to offer less than 75% packet losses.

IV. CONCLUSIONS AND FUTURE WORK

We have defined and implemented an adaptive multimedia application framework for mobile and wireless applications. Its performance has been shown in wireless routing scenarios. Through a sample testbed and some experiments, this paper has demonstrated the benefits that such a technology can provide to enhance the experience of an end user in a typical changing conditions environment, ensuring an adequate service level even in dramatic bandwidth changes. It has been shown how applications can help in offering a good QoS even in situations in which the network is not able to guarantee anything.

The behaviour for other types of streams, such as video or full audio/videoconference can be easily obtained by extrapolation from the described tests for the single audio flow, and are to be confirmed in the future in similar scenarios.

Another trend for the future is working in improving the reactions of the application, related to the use of intelligent techniques for the selection of the application parameters which better suit the current network conditions. In addition, total integration with lower layer QoS control facilitates the final objective to reach a quite fair level of service quality.

AKNOWLEDGEMENTS

This work has been performed in the framework of the IST project IST-2000-28584 MIND, which is partly funded by the European Union. The authors would like to acknowledge the contributions of their colleagues from Siemens AG, British Telecommunications PLC, Agora Systems S.A., Ericsson AB, France Télécom S.A., King's College London, Nokia Corporation, NTT DoCoMo Inc, Sony International (Europe) GmbH, T-Systems Nova GmbH, University of Madrid, and Infineon Technologies AG.

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 Fifth International Workshop on Quality of Service, 1997.

[3] M. Kazantzidis, S.J. Lee, M. Gerla. "Permisible throughput Network Feedback in AODV MANETs". Proceedings of ICC 2001, Helsinki, Finland, June 2001.

[4] T.-W. Chen, M. Gerla, M. Kazantzidis, Y. Romanenko, and I. Slain. "Experiments on QoS Adaptation for Improving Enduser Speech Perception over Multihop Wireless Networks". Proceedings of QoS Mini Conference in conjunction with IEEE ICC'99, Vancouver, Canada, Jun. 1999.

[5] A. Kassler et al. BRENTA - Supporting Mobility and Quality of Service for Adaptable Multimedia Communication, in Proceedings of IST Mobile Summit 2000, Galway, Ireland, October 2000.

[6] ITU-T Recommendation E.800 (08/94) "Terms and definitions related to quality of service and network performance including dependability"

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

[8] ISABEL CSCW application. http://www.agora-2000.com/productos/isabel

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