Multi-Protocol Header Protection (MPHP), a Way to Support Error-Resilient Multimedia Coding in Wireless Networks

F. Arnal[1, 2], L. Dairaine[1, 3], J. Lacan[1, 3], G. Maral[1, 4]

[1]TéSA : Laboratoire coopératif en Télécommunication Spatiales et Aéronautiques
2, rue Charles Camichel BP 7122 - F-31071 Toulouse
[2]Alcatel Space Industries, 26, avenue J.F. Champollion
31037 Toulouse Cedex BP 1187 France
[3]ENSICA, 1, Place Emile Blouin, 31056 Toulouse, Cedex 5, France
[4] ENST – Site de Toulouse, 10, Avenue Edouard Belin 31028 Toulouse Cedex 04 France

Fabrice.Arnal@tesa.prd.fr;{Laurent.Dairaine,Jerome.Lacan}@ensica.fr;
Maral@enst.fr

Abstract. In wireless networks, the quality of the media delivered to the user by real-time multimedia applications can be enhanced if corrupted packets are delivered to the application layer instead of being discarded at lower levels. Those erroneous packets are efficiently supported with the new error-resilient multimedia codecs and with UDP-lite. The Multi Protocol Header Protection (MPHP) is an efficient means to enable this thanks to an FEC mechanism which protects all protocol headers involved in the data transfer. MPHP improves the throughput by several orders of magnitude compared with conventional protocol stack. We estimate this efficiency in the context of satellite networks with simulations implementing error control mechanisms from physical layer to application layer.

Keywords: Wireless, FEC, Unequal Error Protection, UDP-lite

1 Introduction

Due to their delay sensitive nature, number of networking multimedia applications can not accept data that arrive too late since data are useless after a given deadline. Multimedia codecs generally do not retrieve lost (or delayed) data, and keep on delivering to the user on-time received ones. In that way, those applications are called "error tolerant" and do not require a fully reliable delivery of data. For example, applications such as Voice over IP (VoIP), multimedia conferencing or time-constrained streaming, appear more seamless to the end user when missing some video or voice samples instead of being faced with temporary service disruption due to the delay for packet recovery. The unreliability in the end-to-end delivery may originate from congestions or bit errors at physical level.

In wired networks congestion is the most important cause of unreliability. In wireless (mobile and satellite) networks, unreliability mainly originates from bit-errors due to frequent channel fades and interference. This sometimes can lead to a resulting link-layer Bit Error Rate (BER) as high as 10^{-4} or even more. With IP over a wireless connection, erroneous packets (up to several hundreds of bytes) are most of the time discarded either at link layer or at transport layer, so that none of them can reach the application layer. As a result, there is a relationship between the overall error rate as viewed by applications and the packet loss rate resulting from corrupted packet discarding mechanisms. Now, intermediate routers may also discard packets in case of congestion, and this limits one step further the end-to-end throughput. Therefore, the conventional error tolerant class of applications turns out to be "packet loss tolerant".

Recent research work has led IETF to propose a transport protocol named UDP-lite [1], whose principle is to extend the functionalities provided by the well established UDP protocol. UDP-lite allows corrupted packets at network layer level to reach application levels. This is of interest for the error tolerant applications, as it improves the quality perceived by the end-user, compared with the protocol architecture which discards all erroneous packets. This is because the packet loss at the application layer impacts the application in a worse manner than would the actual higher BER it is faced with in the absence of discarding mechanisms. Avoiding packet discarding implies that lower layers (i.e., physical and link layers) may deliver corrupted packets to the network layer. Though acceptable, this must be dealt with carefully in view of possible header corruptions. For instance, packets that carry wrong headers must not be processed so as to avoid wrong routing.

In this paper, we propose a new approach that provides an efficient support for UDP-lite based communications by making utterly useful the received information. In mobile and satellite wireless networks the transmission techniques are complex due to Forward Error Correction (FEC), synchronization and signalling procedures, while bandwidth is a costly resource. Hence the overall transmission efficiency must be optimized. Therefore we adopt a cross-layer point of view that coordinates the behaviour of the low-level layers to the real needs of the higher layers, with the support of UDP-lite. For that, we introduce two joint mechanisms at link-layer.

First, corrupted link-layer frames are allowed to be passed on to the next upper level (IP). This is simply done by having the classical Cyclic Redundancy Check (CRC) error checking covering the header only. Though appealing, this approach actually performs poorly with high BER, since header corruption probability may remain high (e.g. 15% for 20 bytes headers when BER equals 10^{-3})

We propose therefore, in addition, a selective FEC mechanism over the wireless link. We name it *Multi Protocol Header Protection* (MPHP). It specifically protects the headers of protocols at several levels simultaneously. For instance, results show that this scheme allows a multimedia flow to be delivered with as high as 40% throughput while without MPHP, the same flow would not be delivered at all.

The remaining of this paper is organised as follows. Section 2 provides an overview of some related works dealing with corrupted data delivery. Section 3 details the MPHP proposal and the limited packet discarding mechanism. Section 4 presents a performance evaluation suited for satellite networks with MPEG-2. Concluding remarks and future work are discussed in Section 5.

2 Background

The idea of allowing damaged data units to be delivered to the application-level in an IP stack is not a new one. However, the development of this idea is made difficult because most of existing link layers technologies is based on a common service model where all corrupted frames must be discarded. This is the case for example in classical Ethernet networks. Note that in those networks, due to the very low bit error rate of the underlying wired technology, the benefits of allowing damaged frame to be delivered to upper layer would not be really important (i.e., due to the high quality of cables, most of the Ethernet frames are delivered at receiver side without any bit error). Then, the effort needed to upper layers (e.g., in terms of coding) to take into account the possibility to get damaged frames would be, in the case of wired network, not acceptable. As a result of the use of this service model, most of classical applications have been designed to take into account the packet discarding mechanism at lower layers.

However, the emergence of the wireless error-prone links networks notably modifies those communication hypotheses. As a result, the deployment of error-resilient multimedia flows with specific toolsets suited for wireless networks (e.g. ISO MPEG-4, see [2], the latest versions of ITU H.263 for video integrating error resilience [3], Adaptive Multi-Rate (AMR) for audio), the UDP-lite protocol [1] has recently been proposed at the IETF and is under standardisation.

Conversely to UDP, UDP-lite implements a special delivery from the transport to the application levels, where the packet protection coverage is chosen by the application developer. The checksum minimally covers the UDP-lite header, and can extend to any offset within the UDP-lite packet. If the checksum covers the whole packet, UDP-lite is semantically equivalent to UDP. The indication of the coverage range is piggybacked in the UDP-lite packet header. Note that this delivery can also be done with UDP, but would come at the high risk of having the checksum completely switched off.

The main relevant question that has not been clearly answered up to now is about the real benefits that such a concept can provide, in particular in an error-prone environment. The reason that makes it difficult to give an answer, is that the benefit strongly relies on the nature of the multimedia flows and parameterisation related to codecs. However this has been tentatively addressed in some papers: one of the first papers concerning UDP-lite [4] shows promising packet loss rates comparisons between UDP and UDP-lite but does not deal with the benefits incurred at higher level. In [5] the authors have integrated a wireless error simulator with an experimental video over GSM wireless transmission, using the MPEG-4 or H.263 error-resilient features of the codecs. Thanks to the transparent delivery mode of GSM enabling corrupted radio frames to be delivered, they have also implemented a modified PPP layer (PPP-lite) that acts as UDP-lite, here in delivering damaged data units to the IP layer. Their major point is that UDP-lite shows a significant reduction in end-to-end delays, a constant packet inter-arrival time, and a loss rate halved in respect with the UDP delivery. They also conclude on a better perceived video quality.

Another paper [6] studies the improvements for real-time VoIP transmissions with Adaptive Multi-Rate (AMR) speech codec that has been chosen for example in UMTS. This codec provides some error concealment mechanisms thanks to Unequal Error Protection (UEP) with Data Partitioning (DP). It consists in separating AMR data packets in 3 classes, each one having a different sensitivity to bit-errors. The first one, referred as class A bits, has a high sensitivity to bit-errors while the third one, class C bits, only has a low sensitivity against bit errors. The speech quality is evaluated with a RTP/UDP-lite/IP transmission scenario, thanks to the ITU Perceptual Evaluation of Speech Quality (PESQ) standardized quality metric [7]. In making the UDP-lite checksum covering or not class A, B or C bits, the evaluation quality is provided for different bit error rates, and a significant improvement is shown for the Mean Opinion Score (MOS) factor [7]). One step beyond, the additional use in this work of RObust Header Compression (ROHC) [8] allows to prevent some packet losses due to headers corruption. As a result, the MOS is improved by even more. Here, when bandwidth savings is at stake, ROHC appears as the best candidate, considering that short VoIP packets imply a high overhead with RTP/UDP-lite/IP headers (40 bytes). At last, according to [9], UDP-lite usage in 802.11b wireless networks “fails to provide significant enhancement in perceived media quality” though a higher throughput can be expected if a MAC-lite link-layer is supported to achieve corrupted frames to be delivered to IP. The authors nevertheless demonstrate the benefit of using FEC at applicative-level with UDP-lite, as they show that the overhead redundancy that FEC requires is minimized with respect to UDP.

At this point, one can see that UDP-lite can provide appealing enhancements under suitable situations. However, its full exploitation still remains a challenge, since an efficient support of the underlying layer is to be provided to deliver erroneous packets. We describe in the next section our proposed solutions for this support.

3 Limited link-level frame discarding and Multi-Protocol Header Protection (MPHP)

To counteract the transmission inefficiencies inherent in wireless networks where a CRC policing detects and drops all the corrupted link-level frames, we consider a more efficient protocol architecture that enables the service of UDP-lite. For this, damaged link-level frames must be allowed to be delivered to IP (see Figure 1). This may easily be done by having the link CRC only covering the link-layer header. Moreover, the basic transmission data units, referred to as BTDUs in Figure 1 (e.g. RLC PDU in UMTS [10], MPEG2-Transport Stream packets in IP over DVB satellite networks [11]) should not be discarded by the receiver unless their headers are corrupted. Such functionalities may be obtained by using some confidence information given by the FEC decoder, and/or with a classic CRC error detection. A second mechanism is now needed, since the reassembly layer misses the discarded BTDUs. Instead of simply concatenating the undiscarded BTDUs to form the actual frame, reassembly must detect the sequence break while keeping the undiscarded BTDUs in the frame with their original sequence number. In this case, any missing BTDU has to be replaced by an equivalent length blank unit (e.g. all binary 1s) before the reassembled frame is delivered to the data link layer.

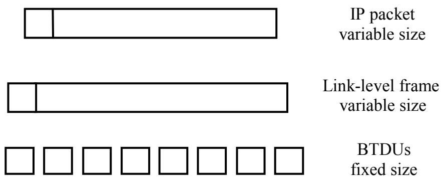

Figure 1 : IP encapsulation over generic wireless networks

Given the BER at the output of the FEC decoder, we have modelled such a scheme with a limited frame discarding, assuming perfect reassembly mechanisms (i.e. the link-level frame size is always recovered).

In this model, a single remaining bit-error in any of the headers (BTDU, link-level frame or IP) leads to discard the entire data unit. The model inputs are the resulting Symbol Error Rate (SER), i.e. the byte error rate, after FEC decoding, header and payload lengths for the BTDU, the header length sum between the link-layer and the IP level, and the IP PDU length. Note that this model could also cover any additional layer (e.g. transport with RTP/UDP protocols) only by changing the header length sum parameter. The output of the model is the mean Packet Loss Rate.

We compare the two approaches where BTDU and data link-layer level drop any erroneous unit (conventional approach), and BTDU and data link-layer always deliver frames provided that its header has not been corrupted, thanks to the mentioned reassembly mechanism (modified approach). The configuration for UMTS and MPEG-2 TS is specified in Table 1.

	Value	
Parameter	MPEG-2 TS like profile[1]	UMTS like profile[2]
Header length for BTDU	4 bytes	4 bytes
Payload length for BTDU	184 bytes	40 bytes
Header lengths (link-level and IP)	8+20 bytes	1+4 bytes
Payload length (IP)	576 bytes	576 bytes

Table 1 : Model parameters

Results are shown on Figure 2. This scheme shows that for high SER, IP packet loss rates are in a factor 10 lower in the proposed UMTS architecture with respect to the same proposal with MPEG-2 TS. This is explained by the high probability that each of the uncompressed 20 bytes of IP header is corrupted, leading to the packet loss. Note that this statement is not only true for IP level, but can also be applied at link, transport, and possibly applicative level layers.

[1] Assuming use of Ultra Lightweight Encapsulation (ULE) scheme [12]

[2] Using DSCH transport channels at 2048 kbps, and RLC unacknowledged mode, taking into account RLC, MAC headers and layer 1 CRC. For simplicity we assume PDCP level always compress 20-byte IP header in 4 bytes.

Since headers are the key elements on which decisions are taken by the various protocols implied in the communication process, we propose to design specific actions onto these headers, in order to render them more reliable compared to the rest of the packet. Two ways appearing different, are interesting tracks. On one hand, a first solution consists in reducing the amount of headers data sent over the air interface so as to decrease the overall header corruption probability. This method can be performed with header compressions, like in ROHC. On the other hand, increasing the headers size by using FEC coding also achieves the same result. This second solution is the next step of our proposal and we call this *Multi Protocol Header Protection* (MPHP).

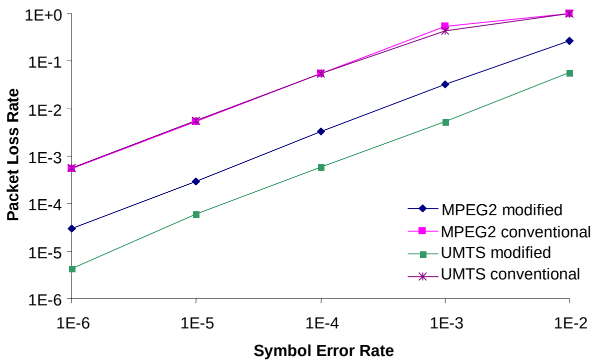

Figure 2 : IP Packet Loss Rate vs. Symbol Error Rate

MPHP is implemented at the wireless link only. Its principle consists in selectively adding a strong FEC protection to headers in order to prevent packet losses induced by header corruptions. This action can be seen as a form of Unequal Error Protection and does not introduce an expensive bandwidth overhead since it only protects the few header bytes. So as to prevent header corruption even for high error rates, we choose a robust code rate, e.g. 1/3, and we implement it through a Reed-Solomon (RS) systematic code. MPHP works in a per-flow way and may define different profiles, like ROHC, depending on the protocol stack (e.g. IP, UDP/IP, RTP/UDP/IP). According to the profile, the appropriate shortened code is applied to the conformant flow. For example in the IP profile with link level headers of X bytes, the RS code (n=3*X+60, k=X+20) is applied to cover X+20 bytes of link-level and IP headers. Furthermore, the simultaneous use of MPHP and ROHC or any similar mechanism to compress the headers is also feasible and could be of interest, for the same reasons as explained in Section 2.

Requirements for implementing the MPHP scheme are minor. At the air interface sender side, it is done by a packet filtering on some packet fields (e.g. IP addresses, port numbers). The relevant parameters could be well-known values set at the content server's side. Conformant packets are processed according to their headers, while other packets are transparently forwarded. The RS, implemented in fast dedicated hardware, encodes the conformant packets. The parity data is appended at the end of the original packet in order to prevent short bursts of bit-errors occurring both in the header and in the MPHP parity. However, this method will slightly slow down the coding and decoding because of discontinuous data access. Decoding operations work in a same way as coding. When a MPHP-protected packet is received, the decoder tries to correct the protected headers. If it fails, the whole packet must be dropped. Otherwise, the packet is delivered. When no errors are detected, the systematic form of the RS codewords allows fast decoding and delivery. Thus, at the MPHP transmitter and receiver level, delays due to filtering and coding/decoding are so low that the scheme does not introduce any penalty for time-constrained transmissions. The full MPHP receiving process is shown on Figure 3.

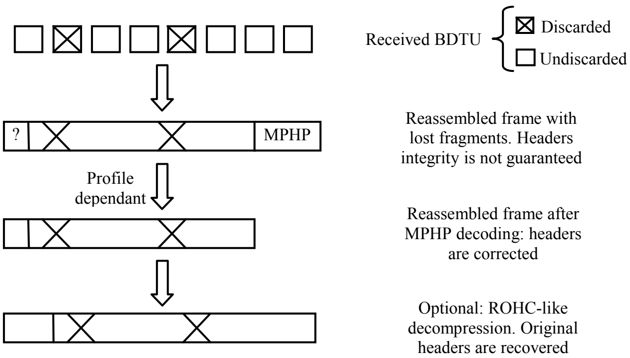

Figure 3 : Receiving operations with MPHP

The MPHP scheme performs well with multimedia codecs that place a known fixed-size header at the beginning of the video packet. For example, MPEG-4 with Data Partitioning [13] meets this condition. In such a case, headers (containing for example synchronization information, quantization parameter, etc…) are clearly the most sensitive part of the video packet. Thus to protect this header by MPHP seems relevant. Then, when an error occurs with our scheme, it often impacts the motion or texture data, and does not impact the media playback during a long time.

To limit packet discarding at physical and link layers, and to integrate error resilience tools in the applicative layer are both necessary conditions to offer useful erroneous data packets to the receiver, while MPHP is used to reduce the header corruption probability. We now give a performance evaluation of the complete scheme using a throughput analysis within satellite networks.

4 Performance study with IP over DVB-S

In this section we show that the MPHP-based architecture, possibly extended with header compression, may greatly enhance the error robustness. For that, throughput performance is chosen to be the evaluation criterion. We define this throughput as being the product between the percentage of correctly received symbols at the end-user and the encapsulation efficiency from the physical to the transport layer protocols. Efficiency is defined as the ratio of the transport service data unit length to the link-level frame length. When MPHP is activated, the extra data (parity) is taken into account in the efficiency parameter. At last, the throughput is computed at the interface between the transport and the applicative level.

The throughput analysis is performed thanks to a coding and decoding chain simulator compliant with DVB-S standards [11]. Our simulator implements all reliability mechanisms from physical to application levels, and channel transmission modelling:

- Outer coding: Reed-Solomon (n=204, k=188, d=16);
- Convolutional interleaving (interleaving depth I=12);
- Inner coding: punctured convolutional coding (R_c= 3/4) with Viterbi decoding (constraint length K=7);
- BPSK modulation;
- Transmission through an AWGN channel;
- MPEG-2 TS, a simplification of the ULE layer [12] currently proposed from IP over DVB Working Group [14], IP and transport-level (e.g. UDP-lite) formatting, as well as their corresponding error detection and packet discarding mechanism. Reassembly and discarding policy are done according to the studied architecture.

This simulator has been implemented using IT++, a library of mathematical, signal processing, speech processing, and communications classes and functions [15].

The evaluation is provided for different IP PDU lengths that are applicable to various sorts of media up to 1500 bytes and for different noise levels. We assume that the header compression, when activated, reduces the link-layer and IP headers down to 16 bytes, and the transport level headers (e.g. RTP/UDP) are supposed to be 20 bytes long.

In Figure 4, throughputs are plotted as a function of the IP PDU lengths for different E_c/N_0 values (ratio of the energy per transmitted (encoded) bit to the single-sided noise power density), with or without header compression (ROHC-like), each with or without MPHP.

In Figure 4.a, the conventional architecture is used with classical physical and link-level discarding mechanisms. One may note that in this context, choosing a high PDU length not always gives the best throughput, depending on the noise level. This is due to the increasing of packet loss probability when the packet size increases. With ROHC (Figure 4.b), a significant improvement is achieved, specially for short packets. When MPHP is implemented (Figures 4.c and 4.d), the previous trade-off disappears, so that having the higher PDU length simply assures a better throughput. ROHC benefits are still present with MPHP architecture and the combination of both gives by far the best results.

Figure 5 compares the throughput, for a fixed IP PDU length of 576 bytes, between conventional and MPHP and ROHC architectures, as a function of the E_c/N_0 value. This figure displays the four previous architectures along with the limited CRC policy architecture, when MPHP is not implemented. Results prove that our solution still allows fair delivery for E_c/N_0 as low as 1 dB, and that MPHP FEC overhead is amortized as long as E_c/N_0 is less than 2.0 dB if the ROHC-like mechanism is deployed.

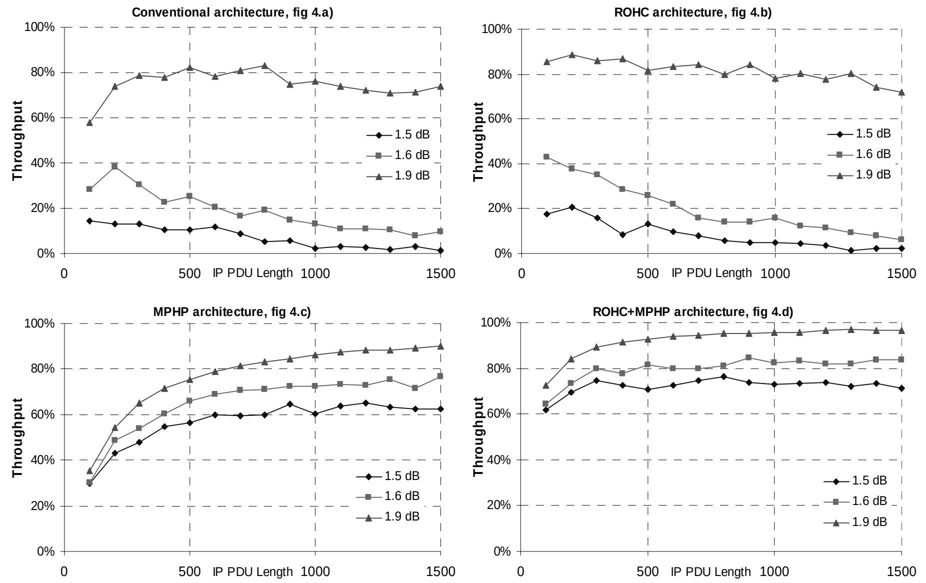

Figure 4: Throughput vs. IP PDU length (for E_c/N_0 = 1.5 dB, 1.6 dB, 1.9 dB)

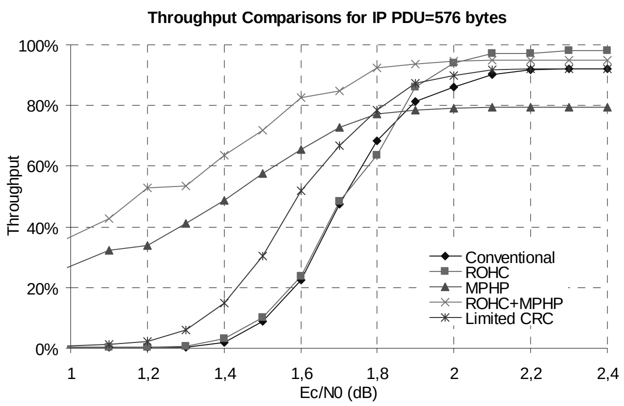

Figure 5: Throughput vs. E_c/N_0

5 Conclusion and Future Work

In this paper we have proposed a scheme that introduces in wireless environments some mechanisms at physical and link layer, enabling an efficient support of applications on top of UDP-lite. First the delivery of damaged data units from physical to network layer level is performed thanks to limited packet discarding and adaptation of the reassembly process. Second, implementing MPHP enhances the headers protection. Combined with header compression, this scheme supports time-constrained data delivery to the end-user with high link utilization, specially at low signal-to-noise ratio. Thanks to the deployment of new features of performant codecs such as in MPEG-4 or AMR, damaged data units at applicative levels are really useful, instead of being unrecoverable, which highly reduces the impact of bit-errors on the media playback. Our future work will consist in giving an applicative-level evaluation of the scheme with real measurements of PSNR for noisy MPEG-4 transmission. We will also investigate the use of FEC at transport level with real-time deliveries in order to ensure a specified level of partial reliability at the application level. Finally simulations will be extended to the case of a mobile channel.

References

1. L-A. Larzon, M. Degermark, S. Pink, L-E. Jonsson, G. Fairhurst, “The UDP-Lite Protocol”, draft-ietf-tsvwg-udp-lite-02.txt, August 2003.
2. R. Talluri, “Error-resilient video coding in ISO MPEG-4 standard”, IEEE Communication Mag., 36(6):112–119, June 1998.
3. Recommandation ITU-T H.263, February 1998.
4. L-A. Larzon, M. Degermark, S. Pink, “Efficient Use of Wireless Bandwidth for Multimedia Applications”, Proceedings of the IEEE Mobile Multimedia Communications (MOMUC) Workshop, 2000.
5. A. Singh et al, “Performance Evaluation of UDP Lite for Cellular Video”, NOSSDAV'01, June 2001.
6. F. Hammer et al, “Corrupted Speech Data Considered Useful”, Proc. First ISCA Tutorial and Research Workshop on Auditory Quality of Systems, Mont Cenis, Germany, April 2003.
7. Recommendation ITU-T P.862, February 2001.
8. C. Bormann et al, RFC 3095 (Standard Track), Robust Header Compression, July 2001.
9. S.A. Khayam et al, “Cross-Layer Protocol Design For Real-Time Multimedia Applications Over 802.11b Networks”, IEEE International Conference on Multimedia and Expo (ICME), July 2003.
10. ETSI TS 100 946, “Digital cellular telecommunications system (Phase 2+); Radio Link Protocol (RLP) for data and telematic services on the Mobile Station - Base Station System (MS - BSS) interface and the Base Station System - Mobile-services Switching Centre (BSS - MSC) interface”, V8.0.0, July 1999.
11. ETSI EN 301 192, “Digital Video Broadcasting (DVB); DVB Specification for Data Broacasting”, January 2003.
12. G. Fairhurst, B. Collini-Nocker, “Ultra Lightweight Encapsulation (ULE) for transmission of IP datagrams over MPEG-2/DVB networks”, Internet Draft, draft-fair-ipdvb-ule-02.txt, November 2003.
13. S. Fabri et al, “Real-Time Video Communications over GPRS”, IEE 3G2000, London, pp. 426-430, March 2000.
14. IETF IP over DVB Working Group, http://www.ietf.org/html.charters/ipdvb-charter.html
15. http://itpp.sourceforge.net/