JOINT SOURCE-PROTOCOL-CHANNEL DECODING: IMPROVING 802.11N RECEIVERS

Çağatay Dikici[1], *Anissa Mokraoui*[2], *Michel Kieffer*[1,3], *Pierre Duhamel*[1]

[1]L2S, CNRS – SUPELEC – Univ Paris-Sud, 3 rue Joliot-Curie, F-91192 Gif-sur-Yvette
[2]L2TI, Institut Galilée – Université Paris 13, 99 Av. J.-B. Clément, F-93 430 Villetaneuse
[3]on leave at LTCI, CNRS – Telecom ParisTech, 46 rue Barrault, F-75 013 Paris

{dikici,duhamel,kieffer}@lss.supelec.fr; {mokraoui}@univ-paris13.fr

ABSTRACT

This paper combines joint protocol-channel (JPC) and joint source-channel (JSC) decoding techniques within a receiver in the context of wireless data transmission. It assumes that demodulation and channel decoding at physical (PHY) layer can provide soft information about the transmitted bits. At each layer of the protocol stack, JPC decoding allows headers of corrupted packets to be reliably decoded and soft information on the corresponding payload to be forwarded to the correct upper layer. When reaching the application (APL) layer, packets may still contain errors and are JSC decoded, exploiting residual redundancy present in the compressed bitstream, to remove part of the residual errors. The main contribution of this paper is to show that these tools may be efficiently combined to obtain *i*) reliable protocol layers permeable to transmission errors and *ii*) improved source decoders. Performance is evaluated using an OMNET++ simulation for the transmission of compressed HTML files (HTTP 1.1) over a standard RTP/UDP-Lite/Ipv6/MAC-Lite/802.11n-PHY protocol stack, only the receiver is modified. For a given packet error rate, the proposed scheme provides gains up to 2 dB in SNR compared to a standard receiver.

1. INTRODUCTION

The widely used OSI layered model partitions networking tasks into distinct layers [16]. This facilitates network design, since each layer has not to be aware of the information introduced by other layers, allowing heterogeneous contents to be delivered via the same communication network. Moreover, each layer, assuming that the lower layers behave perfectly, attempts to provide perfect information to the upper layers. For that purpose, error-detecting codes (CRC or checksums) protecting essentially headers (and sometimes the payload) have been introduced at various places of the protocol stacks. They are combined with retransmission mechanisms (when feasible) for data packets deemed as corrupted. Moreover, since the layers work independently, but sometimes require the knowledge of identical (or correlated) information, some redundancy may be found, essentially in the packet headers that are processed at each layer. This redundancy has been recognized and used for example in ROHC [6] for reducing the header lengths. However, this redundancy can also be used for a better reception of the headers, therefore reducing the number of rejected (and possibly retransmitted) packets.

The role of Joint Protocol-Channel (JPC) decoding is to make an efficient (and joint) use of the redundancy present in the protocol layers as well as the redundancy introduced by the channel coding in order to obtain optimal receiver performance. Recently, JPC decoding techniques have been applied to perform Reliable Packet Header Estimation (RPHE) [17, 21, 19, 20]. Using JPC decoding, corrupted packets, which would be dropped at intermediate layers of the protocol stack by classical receivers, now may reach the APL layer [15, 22]. Even if they are still corrupted, the residual redundancy present in the compressed bitstream may be exploited by JSC decoding techniques to remove residual errors, see [10] and the references therein. These techniques do not need any change in the structure of the transmitted signal, since JPC and JSC decoding is performed within the receiver. The ability to use these tools in the context of existing standards makes it potentially very practical.

However, to the best of our knowledge, the performance gains obtained when implementing simultaneously JPC and JSC decoding techniques have not yet been evaluated in a realistic receiver. Several issues have to be addressed. First, soft-output demodulators and channel decoders [5] have to be used to get soft information about the transmitted packets (*a posteriori* probabilities or log-likelihood ratios). RPHE has to be performed at each layer of the protocol stack. Soft bit information has to be transmitted between layers of the protocol stack [22]. Aggregated packets have to be reliably delineated [7, 3]. Finally, JSC decoding has to be performed at APL layer.

This paper addresses several of previously mentioned issues and considers the reliable transmission of source-coded contents encapsulated in RTP/UDP-Lite/IPv6/MAC-Lite/PHY-802.11n headers over an AWGN channel with Rayleigh fading. Soft demodulation and channel decoding, RPHE in all protocol layers, transmission of soft information between layers, and JSC decoding of compressed HTML files [11] have been implemented in an OMNET++ simulator [25].

Section 2 briefly recalls the principles involved in JPC decoding for RPHE and in JSC decoding. Section 3 introduces the considered protocol stack in which JPC and JSC decoding have been implemented, as described in Section 4. Simulation results are reported in Section 5 before drawing some conclusions in Section 6.

2. JOINT PROTOCOL-CHANNEL AND SOURCE-CHANNEL DECODING

This section briefly recalls estimation techniques used to perform JPC decoding for RPHE within the protocol stack and JSC decoding at APL layer. See [10] for more details.

2.1 JPC decoding

At any layer L of the protocol stack, packets have more or less the same structure. For example, the n-th packet at layer L consists of a header $\mathbf{h}_n$ and a payload $\mathbf{x}_n$. The header $\mathbf{h}_n$ may be partitioned into up to four fields (not necessarily contiguous), depending on their properties. The *constant* field $\mathbf{k}_n$ contains all bits which either do not change from one packet to the next one (they are fully determined by the standard) or which may be perfectly predicted from previously correctly decoded headers (from layer L or from other layers). The *unknown* field $\mathbf{u}_n$ contains data which cannot be predicted but are important for the correct processing at Layer L. The protocol may be such that the values taken by $\mathbf{u}_n$ are restricted to a set Ω_n, where Ω_n is determined from the various sources of redundancy of the protocol stack. All bits of the header which are not fully determined and are not instrumental for the processing of the packet at Layer L are gathered in the *other* field $\mathbf{o}_n$. Finally, an optional Header Error Check (HEC) field $\mathbf{c}_n$ is often evaluated from some or all previously mentioned bits and (possibly) from the packet payload $\mathbf{x}_n$ by some (CRC or checksum) encoding function

$$\mathbf{c}_n = \mathbf{f}\left(\mathbf{k}, \mathbf{u}_n, \mathbf{o}_n, \mathbf{x}_n\right). \tag{1}$$

Assume that soft bit information $\mathbf{y} = [\mathbf{y}_k, \mathbf{y}_u, \mathbf{y}_o, \mathbf{y}_x, \mathbf{y}_c]$ has been provided to Layer L on the header fields and on the payload by the channel decoder of the PHY layer or by Layer $L-1$. The RPHE considered in [19, 20] consists in evaluating the MAP estimate

$$\widehat{\mathbf{u}}_n = \arg\max_{\mathbf{u}\in\Omega_n} p\left(\mathbf{u}|\mathbf{k}, \mathbf{y}\right) \tag{2}$$

of $\mathbf{u}_n$ taking into account $\mathbf{k}$ and the soft information $\mathbf{y}$. Assuming that the HEC does not consider $\mathbf{x}_n$ and that all $\mathbf{u} \in \Omega_n$ are equally probable *a priori*, one gets

$$\widehat{\mathbf{u}}_n = \arg\max_{\mathbf{u}\in\Omega_n} p\left(\mathbf{y}_u|\mathbf{u}\right) \sum_{\mathbf{o}} p\left(\mathbf{o}\right) p\left(\mathbf{y}_o|\mathbf{o}\right) p\left(\mathbf{y}_c|\mathbf{f}\left(\mathbf{k}, \mathbf{u}, \mathbf{o}\right)\right) \tag{3}$$

A reduced-complexity optimal algorithm for evaluating (3) with a complexity $O(\ell(\mathbf{o}_n)\, 2^{\ell(\mathbf{c}_n)})$, where $\ell(\mathbf{z})$ is the length of the vector $\mathbf{z}$, has been proposed in [19] when the HEC is a CRC and in [20] when it is a checksum. Suboptimal algorithms achieving a complexity-efficiency trade-off have also been proposed in [19, 20].

Once $\mathbf{u}_n$ has been estimated, the header may be processed at Layer L and $\mathbf{y}_x$, containing the soft information about the bits of the packet at layer $L+1$ is forwarded to that layer. Floating-point numbers to represent soft information may be forwarded between layers. When Log-likelihood ratios (LLRs) are transmitted, quantized LLR values on 3 or 4 bits are accurate enough and significantly reduce memory requirements, see [22].

Note that no JPC decoding is performed on packets which CRC or checksum is valid. The more complex JPC decoding process is only applied to packets which header has been deemed as erroneous. With good channel conditions, JPC is then only seldom used.

2.2 JSC decoding

Assume that the n-th packet has reached the APL layer, where it has to be processed by the source decoder. This packet may still contain errors, since CRCs and checksums at lower layers have been used to help RPHE. The decoded content $\mathbf{d}_n$ has to be reliably estimated. It corresponds to the payload $\mathbf{x}_n^{\text{APL}}$ for which soft bit information $\mathbf{y}_{x,n}^{\text{APL}}$ is provided by the lower protocol layer. For that purpose, residual redundancy left by the source coder in the compressed bitstream may be exploited. This redundancy may come from the syntax of the codewords, the semantic of the bitstream, and from the packetization process, see [10]. This redundancy translates into the fact that $\mathbf{x}_n^{\text{APL}}$ can only take a limited number of values within a set denoted as Ω_{x_n}. A MAP estimate of $\mathbf{x}_n^{\text{APL}}$ may then be obtained from $\mathbf{y}_{x_n}^{APP}$ and Ω_{x_n} as

$$\widehat{\mathbf{x}}_n^{\text{APL}} = \arg\max_{\mathbf{x}\in\Omega_{x_n}} P(\mathbf{x}|\mathbf{y}_{x_n}^{APP}) \tag{4}$$

In some cases, it is possible to structure the sequences belonging to Ω_{x_n} with a multi-dimensional trellis. Efficient channel-decoding techniques such as the Viterbi or BCJR algorithms [5] may then be employed to evaluate $\widehat{\mathbf{x}}_n^{\text{APL}}$. With content-adaptive data compression techniques, such as the CABAC, the CAVLC, or the deflate algorithm [8], the number of states of the trellis may be prohibitively large. Nevertheless, in such cases, it is usually possible to determine whether a sequence $\mathbf{x}$ does not belong to Ω_{x_n} or does not correspond to a prefix of a sequence in Ω_{x_n}. This allows to use decoding algorithms such as the stack or the M-algorithm [4].

Iterative decoding may be considered when it is possible to get extrinsic information at APL layer, which is easy when the BCJR decoder is used. Soft output sequential decoders such as the soft-output M-algorithm (SOMA) [26] or the soft-output stack algorithm [23] may also be employed. The role of the outer code may be played by a CRC or a checksum, *e.g.*, at UDP layer or by the channel code at PHY layer. The price to be paid in the second case, is a large amount of additional soft information and decoded headers which have to be forwarded to the APL layer.

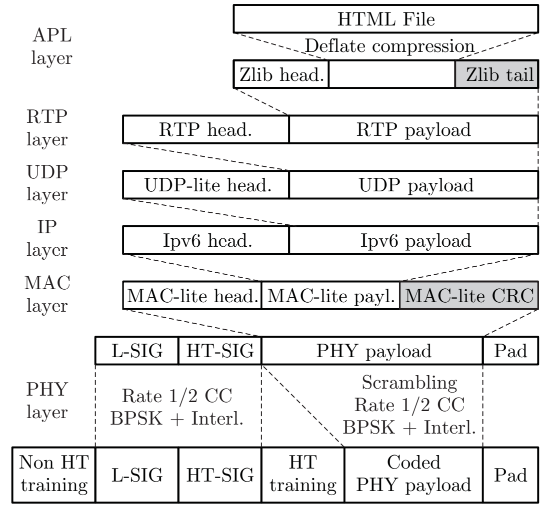

Figure 1: Encapsulations on top of 802.11n

3. CONSIDERED PROTOCOL STACK

JSC decoding of various types of data has been considered: still images, video, audio, and HTML files. In this paper, we considered the transmission of HTML files over an 802.11n WiFi network [2], with an RTP/UDP-Lite/Ipv6/MAC-Lite/802.11n-PHY encapsulation, see Fig. 1. Clearly, HTML is usually combined with TCP/IP, but RTP/UDP/IP facilitates the implementation of a permeable protocol stack (thanks to UDP-lite [18]), since no retransmission of erroneous packets has to be considered at UDP layer, contrary to the connection-oriented TCP. Moreover, this may also give some insights on the behavior of the combination of JSC decoders of video contents after JPC decoding.

3.1 Transmitter side

At transmitter side, the APL layer compresses the HTML file with deflate [8] and supplements the data with the appropriate Zlib header and tail information [9]. The packet is then delivered to the RTP layer. In order to specify the compressed HTML content, the RTP layer fixes the 7 bits of the *payload type* field value to an unassigned value in [24] (between 35 and 71).

The UDP-Lite layer [18] encapsulates the RTP packet specifying the *source port* and *destination port* fields, such that the *source port* corresponds to the HTTP port 80, and the *destination port* to a value which is not known at the receiver, The *checksum coverage* field is fixed to 24 (bytes): the 16 bits of the checksum covers the 8 bytes of the UDP-Lite header and the 16 bytes of the considered RTP header. Then, the IPv6 layer adds the appropriate addressing and header information. Here, it is assumed that the source and destination are located in the same subnetwork hence the first 12 bytes of the IPv6 source and destination addresses are identical. According to the UDP-lite specifications [18], the IPv6 pseudo-header is protected by UDP-checksum, which corresponds to the source and destination IPv6 addresses [12].

The MAC-Lite layer is a MAC layer [2] where the 32 bit MAC CRC covers only the 802.11n MAC header.

At PHY layer, IEEE 802.11n defines three PHY frame formats. The HT-mixed (HT for *High Throughput*) format has been chosen since the packets are transmitted with a preamble which is compatible with the legacy 802.11a/g. In HT-mixed format, the PHY

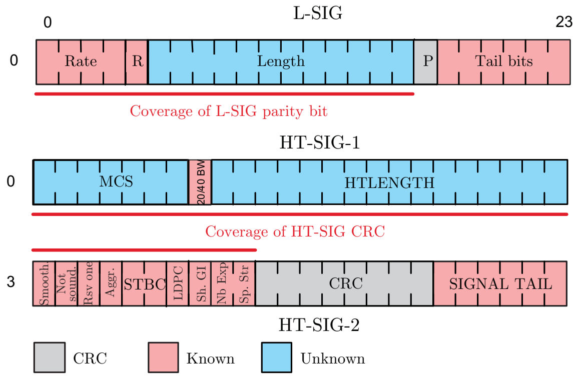

Figure 2: PHY header format in 802.11n standard

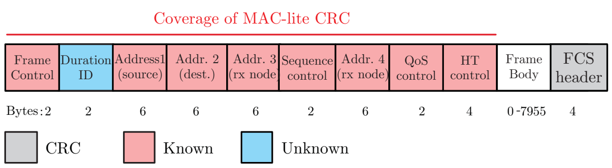

Figure 3: MAC-Lite packet format

payload is first supplemented with 3 bytes *Legacy Signal* (L-Sig) header and 6 byte HT-Sig header [2, S-20.3.9.2].

The L-Sig and HT-Sig headers are encoded separately using a convolutional code with generator polynomials $g_0 = (133)_o$, $g_1 = (171)_o$, and a block interleaving follows as described in [2, S-17.3.5].

The PHY payload is first scrambled [2, S-20.3.10.2]. Then a *Modulation and Coding Scheme* (MCS) is chosen. Here it is for 6 *Mbps* corresponding to a rate $1/2$ convolutional coding scheme with BPSK modulation. The PHY packet is finally interleaved. A *Multiple Input Multiple Output* (MIMO) model with 2 transmitter and 2 receiver antennas is used to transmit the PHY packets over a Rayleigh fading channel.

In this paper, we assume that medium reservation is performed via Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) [2, S-9.2.5]. The transmitter sends a *Request To Send* (RTS) packet before sending information. If the receiver is not busy, it responds with a *Clear To Send* (CTS) packet to notify that: *i*) the transmitter can send the information and *ii*) all other stations of the same network are informed that the receiver is unavailable.

4. REDUNDANCY

The aim of this section is to specify the transmission conditions in order to identify the fields in the various headers which content is known, predictable, or unknown.

Assuming that the CSMA/CA medium reservation succeeds, the receiver has many information about the access point to which it is connected, the duration of the packet it will receive, etc.

4.1 802.11n Physical Layer

A 1 bit *parity* field covers the L-SIG header, whereas the HT-SIG header has a 7 bit *CRC* field. In L-SIG, the *Rate* field and the *Reserved* bit are known to the receiver. The receiver has then to estimate the 12 bits of *Length* in L-SIG, the 7 bits of *MCS* and 16 bits of *HTLength* in HT-SIG (see Fig.-3). Since the three unknown fields are related, RPHE is performed jointly on L-SIG and HT-SIG headers using a CRC-based correction algorithm as described in [19, 13].

4.2 MAC-Lite Layer

As explained in Section 3, the MAC-Lite CRC covers only the MAC header. Considering a non-encrypted downlink transmission of ordered MAC data packets with deactivated retransmission and power-save mode, the 2-byte *Frame Control* field is assumed to be known. The 6-byte *Address2* field contains the MAC address of the receiver and is thus known. The 6-byte *Address1* and *Address3* are transmitted during the medium reservation procedure (RTS-CTS) and may be totally deduced by the receiver. Assuming that the access point is connected to a single router and that the router address has been already received in other information packets, the 6-byte *Address4* field may also be predicted by the receiver. The 2-byte *Sequence Control* field contains two parameters: a sequence number and a fragment number. The sequence number represents the value of the current IP packet counter. The fragment number indicates the value of the current MAC data packet counter. In this study, packets are transmitted in order and these parameters can be easily determined: the sequence number is incremented by one for each RTS-CTS and the fragment number is incremented by one for each received MAC data packet, hence can be estimated by the receiver. The last field of the MAC header is reserved for local wireless networks and is composed of 6 bytes of zeros in this study.

Thus, the receiver only estimates the following unknown field (See Fig. 3): 16 bits *DurationID* using RPHE based on the CRC field, as in [19].

4.3 RTP, UDP-Lite and IPv6 layers

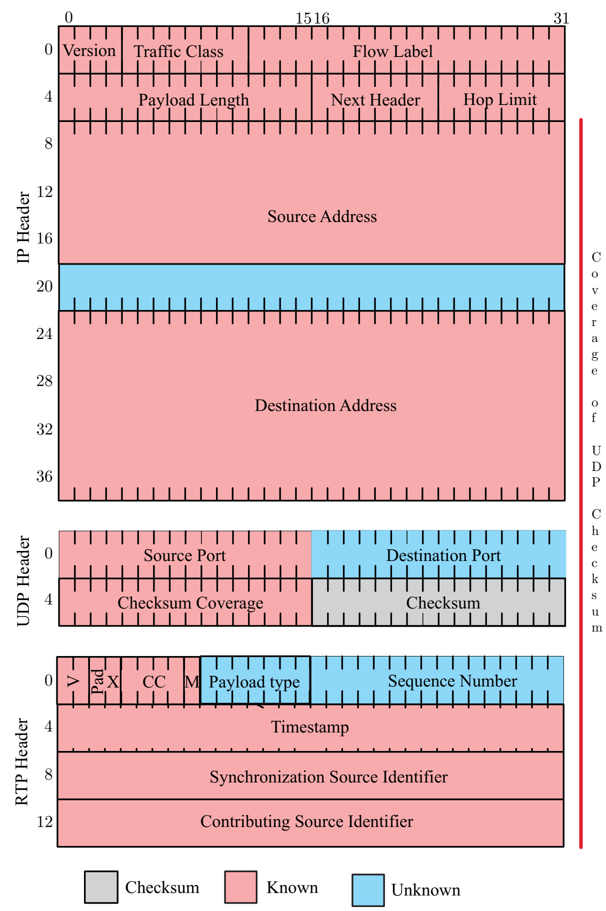

Figure 4: IPv6, UDP-Lite, and RTP header formats

Since in the encoding scheme, the concatenation of the RTP, UDP-Lite, and pseudo-IPv6 headers is protected with a 16 bit checksum introduced by the UDP-Lite header [18]. RPHE of the headers introduced by the three layers is done jointly. The receiver determines the known fields using inter and intra-layer redundancy and estimates the following unknown fields (see Fig. 4): the last 32 bits of the *Source Address* field of IPv6, the 16 bits of *Destination Port* of UDP-Lite, the 7 bits of *Payload type*, and the 16 bits of *Sequence number* introduced by the RTP layer. The decoding algorithm based on the checksum field may be found in [20].

4.4 Application Layer

From the RTP layer, the receiver knows that the packet it receives contains a compressed HTML file. The decoded file has to be compliant with the HTML syntax as specified in [1]. The set Ω_{x_n} introduced in Section 2.2 contains thus all encoded sequences which could be produced by a compliant HTML page provided in [1]. Based on this *a priori* information and on the soft information provided by the lower layers of the protocol stack, a JSC decoder involving a modified soft-output M-algorithm is employed. Iterative decoding with the channel decoder at APP layer is performed, see [14] for more details.

5. SIMULATION RESULTS

This section evaluates the performance of the JPC and JSC decoding schemes previously described via a simulation using OMNET++ [25]. JPC decoding is only performed on packets which CRC or checksum is not valid.

5.1 Channel model

We assume that if the transmitted signal is $\mathbf{p}$, the received signal is $\mathbf{y} = \mathbf{H}\mathbf{p} + \mathbf{n}$, where $\mathbf{H}$ is the complex channel response matrix which is assumed to be known at the receiver, and $\mathbf{n}$ is a complex-value Gaussian noise distributed as $\mathscr{CN}(0, \sigma^2 \mathrm{I})$.

5.2 Performance of JPC decoding

The RPHE performance at each layer of the protocol stack is evaluated for several values of E_b/N_0. The header error rate as a function of E_b/N_0 is given for the PHY, MAC-Lite, IPv6, UDP-Lite, and RTP layers. For each value of E_b/N_0, enough simulations have been performed to get 100 erroneous headers.

For a target packet error rate of 10^{-3}, JPC decoding for RPHE provides a gain of 1.1 dB for the PHY layer header (see Fig. 5), 2.1 dB for MAC-Lite layer (see Fig. 6), 2.3 dB for IPv6 layer (see Fig. 7), 1.8 dB for UDP-Lite layer (see Fig. 8), and 1.9 dB for RTP layer (see Fig. 9) all compared to a classical decoding scheme.

5.3 Performance of JSC at APL layer

The `worldcup.html` html file test is available at www.forum.nokia.com/tools. In a first step, as in [14], we assume that all headers of the lower protocol layers were without errors. Therefore, the performance of the HTML file recovery alone is determined, using a plot of the HTML symbol error rate (SER) as a function of E_b/N_0, see Fig. 10. For a target SER of 10^{-4}, the JSC decoding algorithm achieves 0.8 dB gain compared to a standard decoder, see Fig. 10.

Fig. 11 provides the SER at APL layer when JPC has been employed on the noisy packet headers at all layers.

1. RPHE and JSC decoding at APL layer;
2. RPHE and classical decoding at APL layer;
3. Classical header processing and JSC decoding at APL layer;
4. Classical header and APL layer decoding.

In average, JPC and JSC decoding provides an improvement of 0.6 dB compared to classical header and APL layer decoding. Fig. 11 shows that for E_b/N_0 between 5 dB and 7 dB, the performance of the robust receiver (RPHE and JSC decoding at APL layer) is better then that obtained by the other three methods. For E_b/N_0 above 7 dB, RPHE and classical header processing perform similarly when combined with JSC decoding. Moreover, the results in terms of SER are similar to those in Fig. 10 where all headers were assumed error-free.

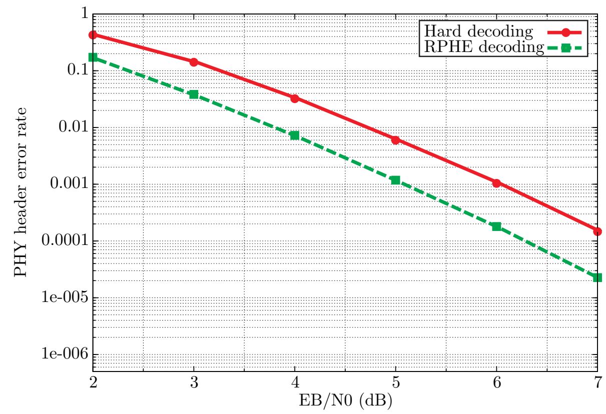

Figure 5: PHY header decoding performance

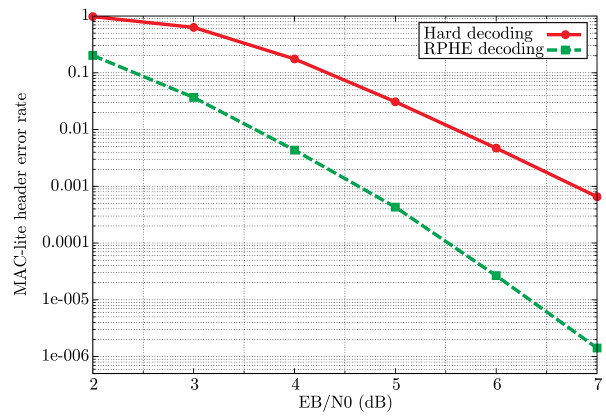

Figure 6: MAC-Lite header decoding performance

6. CONCLUSION

This paper combines JPC and JSC decoding techniques in a realistic communication context. Within the protocol stack, gain of up to 2 dB may be obtained with JPC decoding techniques compared to classical processing. At APL layer, JSC decoders provide a gain of about 0.8 dB in channel SNR. The gain is moderate due to the little redundancy left in compressed HTML files.

These results are directly applicable to other types of contents: speech, music, or video. JPC and JSC decoding reduces the need for retransmission of damaged packets. Moreover, they are compatible with the existing layered structure of a wireless communication network avoiding therefore modifications of the transmitters.

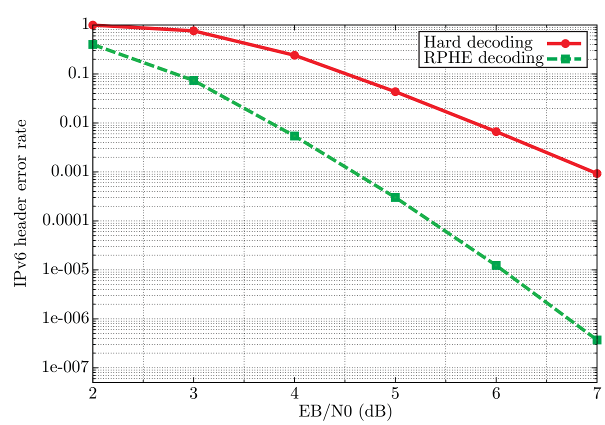

Figure 7: IPv6 header decoding performance

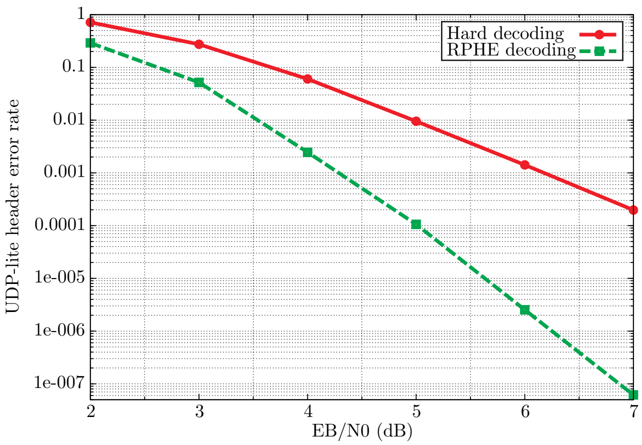

Figure 8: UDP-Lite header decoding performance

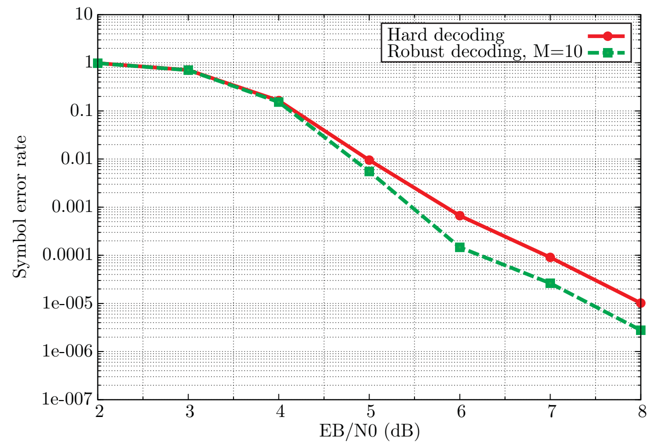

Figure 10: SER at APL layer with clean headers

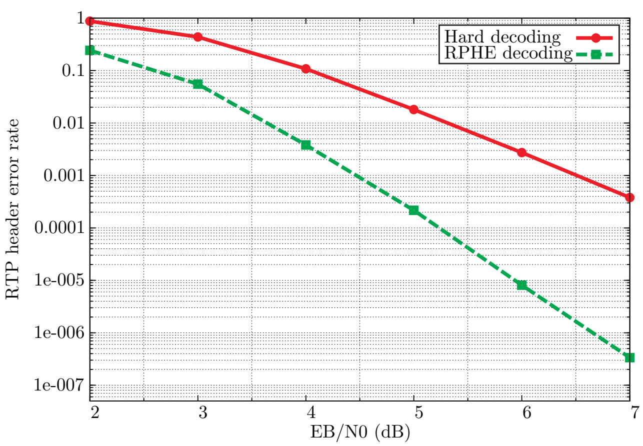

Figure 9: RTP header decoding performance

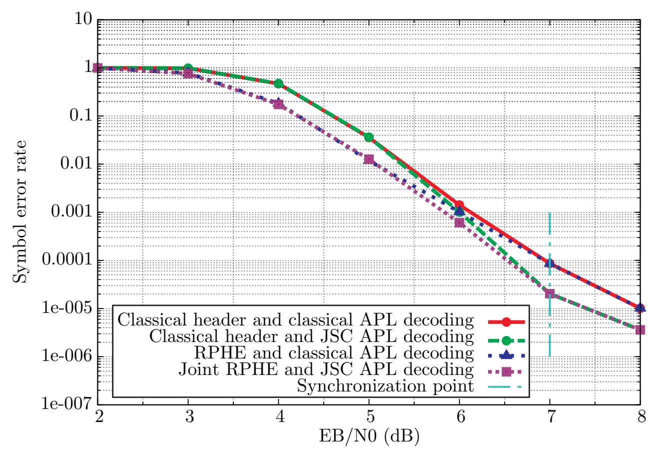

Figure 11: SER at APL layer with JPC decoding

REFERENCES

[1] HTML 4.01 specification.

[2] IEEE standard for information technology–telecommunications and information exchange between systems–local and metropolitan area networks–specific requirements part 11: Wireless lan medium access control (mac) and physical layer (PHY) spec. amendt 5: Enhancements for higher throughput, 29 2009.

[3] U. Ali, M. Kieffer, and P. Duhamel. Joint protocol-channel decoding for robust aggregated packet recovery at WiMAX MAC layer. In *Proc. IEEE SPAWC*, pages 672 – 676, Perugia, Italy, 2009.

[4] J. B. Anderson and S. Mohan. *Source and Channel Coding: An Algorithmic Approach*. Kluwer, 1991.

[5] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv. Optimal decoding of linear codes for minimizing symbol error rate. *IEEE Trans. Info. Theory*, 20:284–287, 1974.

[6] C. Bormann, C. Burmeister, M. Degermark, H. Fukushima, H. Hannu, L.-E. Jonsson, R. Hakenberg, T. Koren, K. Le, Z. Liu, A. Martensson, A. Miyazaki, K. Svanbro, T. Wiebke, T. Yoshimura, and H. Zheng. Robust header compression (ROHC): Framework and four profiles. RFC 3095, 2001.

[7] M. Chiani and M. G. Martini. Optimum synchronization of frames with unknown, variable lengths on Gaussian channels. In *Proc. IEEE GLOBECOM*, pages 4087 – 4091, 2004.

[8] P. Deutsch. Deflate compressed data format specification version 1.3, 1996. `http://www.ietf.org/rfc/rfc1951.txt`.

[9] P. Deutsch and J.-L. Gailly. Zlib compressed data format specification version 3.3, 1996. `http://www.gzip.org/zlib/rfc-zlib.html`.

[10] P. Duhamel and M. Kieffer. *Joint source-channel decoding: A cross-layer perspective with applications in video broadcasting*. Academic Press, 2009.

[11] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. Hypertext transfer protocol – HTTP 1.1. RFC 2616, The Internet Society, 1999.

[12] R. Hinden and S. Deering. Ip version 6 addressing architecture, 1998. `http://tools.ietf.org/html/rfc2373`.

[13] Z. Jaoua, A. Mokraoui-Zergainoh, and P. Duhamel. Robust transmission of 802.11n physical packet headers. In Proc. SPAWC, pages 369–373, 2009.

[14] Z. Jaoua, A. Zergainoh-Mokraoui, and P. Duhamel. Robust transmission of html files: iterative joint source-channel decoding of Deflate codes. In Proc. EUSIPCO, 2008.

[15] H. Jenkac, T. Stockhammer, and W. Xu. Permeable-layer receiver for reliable multicast transmission in wireless systems. In Proc. IEEE WCNC, pages 1805–1811, 2005.

[16] J. F. Kurose and K. W. Ross. *Computer Networking: A Top-Down Approach Featuring the Internet*. Addison Wesley, Boston, third edition, 2005.

[17] C. Lamy and S. Mérigeault. Method of correcting an erroneous frame by a receiver. patent WO03101028, Dec. 2003.

[18] L. A. Larzon, M. Degermark, L. E. Jonsson, and G. Fairhurst. The lightweight user datagram protocol (UDP-Lite). RFC 3828, The Internet Society, 2004.

[19] C. Marin, Y. Leprovost, M. Kieffer, and P. Duhamel. Robust mac-lite and soft header recovery for packetized multimedia transmission. *IEEE Trans. on Comm.*, 58(3):775–784, 2010.

[20] F. Mériaux and M. Kieffer. Robust IP and UDP-lite header recovery for packetized multimedia transmission. In *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, 2010.

[21] J. W. Nieto and W. N. Furman. Cyclic redundancy check (CRC) based error method and device. US Patent US 2007/0192667 A1, Aug. 16 2007.

[22] G. Panza, E. Balatti, G. Vavassori, C. Lamy-Bergot, and F. Sidoti. Supporting network transparency in 4G networks. In *Proc. IST Mobile and Wireless Communication Summit*, 2005.

[23] L. Perros-Meilhac and C. Lamy. Huffman tree based metric derivation for a low-complexity soft VLC decoding. In *Proc. IEEE ICC*, pages 783–787, 2002.

[24] H. Schulzrinne and S. Casner. RTP profile for audio and video conferences with minimal control, 2003. `http://tools.ietf.org/html/rfc3551`.

[25] A. Varga. *OMNeT++*, chapter Modeling and Tools for Network Simulation. Springer Verlag, 2010.

[26] K. K. Y. Wong. *The Soft-Output M-Algorithm And Its Applications*. PhD thesis, Queen's University, Kingston, Canada, 2006.