Open access • Journal Article • DOI:10.1109/TCOMM.2010.03.080303

Robust MAC-lite and soft header recovery for packetized multimedia transmission

Cedric Marin, Yann Leprovost, Michel Kieffer, Pierre Duhamel

Institutions: Alcatel-Lucent, Supélec

Published on: 01 Mar 2010 - IEEE Transactions on Communications (IEEE)

Topics: Header

Robust MAC-Lite and soft header recovery for packetized multimedia transmission

Cedric Marin, Yann Leprovost, Michel Kieffer, Pierre Duhamel

Robust MAC-Lite and Soft Header Recovery for Packetized Multimedia Transmission

Cédric Marin, Yann Leprovost, Michel Kieffer *Senior Member, IEEE*,
and Pierre Duhamel *Fellow, IEEE*

***Abstract*—This paper presents an enhanced permeable layer mechanism useful for highly robust packetized multimedia transmission. Packet header recovery at various protocol layers using MAP estimation is the cornerstone of the proposed solution. The inherently available intra-layer and inter-layer header correlation proves to be very effective in selecting a reduced set of possible header configurations for further processing. The best candidate is then obtained through soft decoding of CRC protected data and CRC redundancy information itself. Simulation results for WiFi transmission using DBPSK modulated signals over AWGN channels show a substantial (4 to 12 dB) link budget improvement over classical hard decision procedures. We also introduce a sub-optimal and hardware realizable version of the proposed algorithm.**

***Index Terms*—Codes, Communication systems, Decoding, MAP estimation, Protocols**

I. Introduction

DUE to bandwidth constraints, efficient transmission of multimedia contents requires the use of some source coding scheme [1]. Nevertheless, compressed data are very sensitive to transmission errors. A single corrupted bit may lead to a loss of a large amount of multimedia data at the receiver. Consequently, the bitstream entering the source decoder has to be almost error-free.

This constraint is hardly satisfied when considering transmission over wireless channels. The data stream at receiver side may be heavily corrupted and not directly usable by the source decoder. A first solution to this problem consists in grouping data into packets protected by an error-detection code (CRC or checksum) [2], [3]. Packets, which have not been correctly received, are identified and can then be retransmitted. However, retransmissions may become difficult in scenarii with strong delay constraints, *e.g.*, for visiophony or may even become impossible when broadcasting data, *e.g.*, in satellite television.

In such situations, the standard solutions make use of very strong error-correcting codes (*e.g.*, turbo-codes, LDPC) at *Physical* (PHY) layer possibly combined with packet-erasure codes at intermediate protocol layers [4]. The redundancy introduced by these codes may however be oversized when the channel is good, reducing the bandwidth allocated for the data. In bad channel conditions, some corrupted packets still cannot be recovered and are assumed lost. Error-concealment techniques [5], [6] may then be used by the source decoders at *Application* (APL) layer. They exploit the redundancy (temporal and/or spatial) found in the multimedia data for reconstructing some information in place of the missing one.

In the recent years, *Joint Source-Channel Decoding* (JSCD) techniques have been proposed to correct damaged packets, see [7]. These methods involve robust source decoders, which exploit the inherent redundancy in the received packets for correcting errors. Several sources of redundancy have been identified. Constraints in the syntax of variable-length source codes [8]–[10] have been used first. Redundancy due to the semantic of the source coders [11], [12] improve significantly the performance of robust decoders. Further redundancy due to the packetization of compressed data has been used in [13]. Altogether, the various redundancies can attain an unexpected amount. Furthermore, redundancy introduced by channel codes at physical layer can also be used in combination with residual redundancy to build iterative decoders as in [14]. These joint decoding schemes provide improved performance when compared to classical schemes, and could be of great use in many applications. However, they are not compliant with the standard protocol stacks in several ways: (i) they require exchange of soft information (*e.g.*, likelihood ratios) between the channel decoder at PHY layer and the robust source decoder at APL layer, (ii) they are not compatible with the use of acknowledgment procedures: a packet received in error needs not be retransmitted unless the robust receiver cannot recover the error, (iii) the headers of packets at a given layer must absolutely be available without error since they contain information necessary for driving the layer in question (at the receiver).

Problem (i) above can be circumvented in some circumstances: a mobile receiver contains all the layers and can choose to forward soft values between layers. This paper assumes that it is the case. The main compatibility problem seems to be the third one: standard protocol stacks do not even allow damaged packets to reach the APL layer, the main reason being that the errors may impact some essential information contained in the headers, which is necessary even for the robust APL decoders.

This paper proposes some tools allowing to receive the various headers with an inherent robustness (even more than the robustness brought by JSCD to the payload) by using tools widely used in JSCD, and applied here to the whole protocol layers. More headers are thus correctly interpreted at each layer, increasing the number of packets reaching the APL layer. We show that robustness of the header is

C. Marin and Y. Leprovost are with the Research and Innovation Center of Alcatel-Lucent, France.

M. Kieffer and P. Duhamel are with the L2S – CNRS - SUPELEC - Univ Paris-Sud, France.

much higher than that of the corresponding payload, which is a prerequisite for implementing a fully *permeable* protocol layer mechanism [15]. The transparent network architecture presented in [16], [17] gives some insights on the way to transmit soft information between protocol layers.

The paper is organized as follows. After introducing the improved permeable layer mechanism in Section II, Section III derives the header recovery technique. Reduction of complexity is presented in Section IV. As an example, the design of the proposed mechanisms for PHY and MAC layers of WiFi is detailed in Section V. Finally, simulations are presented in Section VI.

II. Enhanced Permeable Layer Mechanism

Packetized multimedia transmission is usually based on an RTP/UDP/IP protocol stack [3]. Fig. 1 illustrates an example of segmentation and encapsulation mechanisms implemented at each protocol layer in case of a multimedia packet transmission with the 802.11 standard (WiFi) [18]. Error detection mechanisms implemented at each layer are detailed below.

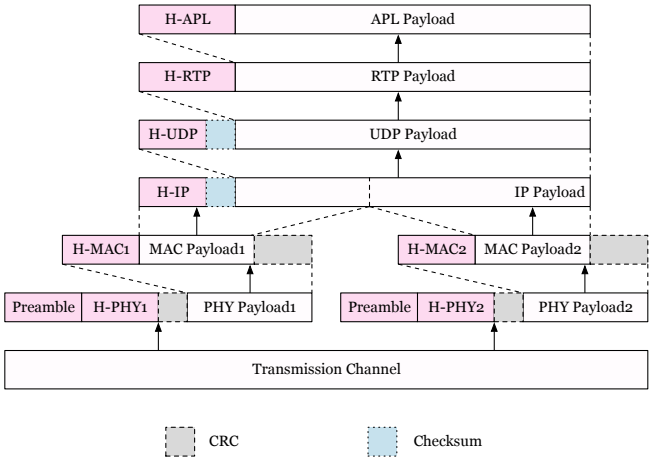

Fig. 1. Protocol stack for multimedia transmission over WiFi

At PHY layer, a known preamble allows the detection of the beginning of each PHY packet. A CRC protects the header fields (the preamble and the payload are not protected). Received packets with damaged headers are discarded. At MAC layer, a CRC protects the corresponding header and payload. When an error occurs, the packet is retransmitted. At IPv4 layer, the header fields are protected by a checksum. Received packets with damaged headers are discarded. At UDP layer, a checksum protects the header and the payload. When an error occurs, the packet is discarded. We assume in this paper that packet fragmentation only occurs at MAC layer, which is a reasonable assumption for a wireless transmission.

The error-detection mechanisms provided by CRCs and checksums, combined with the retransmission mechanism at MAC layer, allow APL layer to receive only error-free packets. The price to be paid is a reduced throughput due to MAC level retransmissions which increase when the channel conditions worsen, or frequent use of error concealment when errors are detected at IPv4 or UDP layers (generally due to time-out constraint: the limit on the number of retransmissions at MAC level has been reached).

JSCD methods allow many errors to be corrected at APL layer based on soft information provided by lower protocol layers. The recently introduced UDP-Lite [19] mechanism, combined with lower permeable protocol layers [15]–[17], allow damaged APL packets to be fed to the APL layer. With UDP-Lite, a checksum protects a limited number of bytes (generally including the UDP-Lite, RTP, and APL header fields). Thus, packets with erroneous headers are still discarded. Considering the order of magnitude of the length of the packets and that of the various headers in actual wireless communications when tuned for difficult situations [20], this may happen more than expected. The bottleneck of such permeable transmission schemes is the fact that packets are discarded due to erroneous headers.

This paper proposes a method for recovering headers based on the various sources of redundancy in the protocol stack, thus increasing the amount of packets that can be used for robust decoding at APL layer. As a result, the efficiency of the decoding at APL layer is improved in all the cases : (i) when retransmissions are allowed, only packets that were not corrected at APL layer are retransmitted, decreasing the number of retransmitted packets, (ii) when higher layer redundancy has been introduced to circumvent the problem (*e.g.*, the so-called MPE-FEC of the MAC layer in DVB-H), our strategy at least allows to reduce the amount of redundancy, and finally, (iii) when no retransmission is allowed, it improves the quality of the multimedia content, because error concealment is used less frequently.

The proposed header recovery technique detailed in the next section involves two main ideas. First, intra-layer and inter-layer redundancy is present in the protocol stack. This redundancy has been exploited in the *RObust Header Compression* (ROHC) mechanism [21], at upper protocol layers, by replacing the headers introduced by the RTP, UDP, and IP layers by a compressed version. Here, the redundancy present at all protocol layers is used to build some *a priori* information on the erroneous headers, improving their estimation. Second, CRCs and checksums are used as error-correcting codes, as proposed in [22] and [23].

Figure 2 illustrates the principle of the proposed decoding technique to build a permeable layer L. Assume that soft information associated to the payload of the n-th packet has been transmitted by layer $L-1$ to layer L. Assume also that the headers of the $n-1$-th packet at layers $L-1$, L, and $L+1$ and the header of the n-th packet at layer $L-1$ are available. The payload of the n-th packet at layer $L-1$ contains the header, the payload, and the CRC related to layer L. At layer L, the header recovery block combines the soft information provided by layer $L-1$, the properties of the CRC (or the checksum), and the *a priori* information obtained by additional sources of redundancy (corresponding to the intra-layer and inter-layer redundancy) in order to recover the header of layer L. The processing details of the header recovery are discussed in Section III. Intra-layer and inter-layer redundancy is determined by a careful examination of the network protocols and is mainly due to the correlations between the

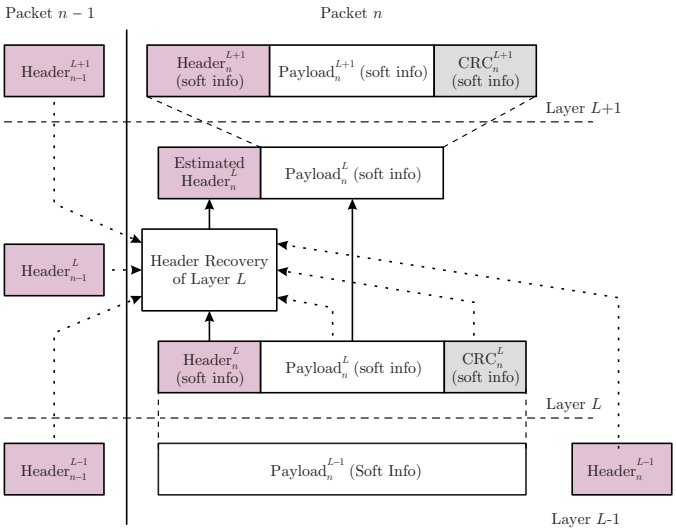

Fig. 2. Proposed permeable layer mechanism; the header at Layer L is estimated from soft information provided by Layer $L-1$ and other side information (dashed arrows); the estimated header allows then to more reliably forward the payload of Layer L to Layer $L+1$

various headers. The *a priori* information exchange due to this type of redundancy is represented in Figure 2 by dotted lines. Finally, layer L uses the decoded header to drive the soft information associated to the payload from layer L to layer $L+1$. Repeating this operation in other protocol layers makes it possible to get a complete permeable protocol stack.

This paper focuses on the PHY and MAC layers of WiFi, as generic examples. Nevertheless, the proposed permeable layer mechanism may be applied to any protocol layer.

III. MAP Estimator for Robust Header Recovery

In the sequel, $\ell(\mathbf{z})$ denotes the size of vector $\mathbf{z}$.

As a general situation, at a given layer L, the n-th incoming packet may include three items: a header, a payload, and a CRC. Information protected by the CRC $\mathbf{c}_n^L$ may have various properties, as far as the corresponding redundancy is concerned.

- The constant fields, represented by the vector $\mathbf{k}_n^L$, are assumed to be *known*.
- The *predictable* fields are embedded in the vector $\mathbf{p}_n^L$. In contrast with the known fields, the predictable fields are estimated by exploiting the *intra-layer* and *inter-layer redundancy* represented by R_n^L, which will be defined formally in what follows. They are predicted from information contained in the previously received packets. The predictable fields are assumed to be entirely determined if the previous packets have been correctly received.
- The important *unknown* fields are collected in the vector $\mathbf{u}_n^L$. These parameters are either completely unknown or limited to a configuration set $\Omega_u^L(\mathbf{k}_n^L, \mathbf{p}_n^L, R_n^L)$ the content of which is determined by the values of $\mathbf{k}_n^L$, $\mathbf{p}_n^L$, and R_n^L. This set contains the actual information on the data that the receiver must estimate.
- Finally, the vector $\mathbf{o}_n^L$ contains the *other* fields covered by the CRC. This last part contains unknown data, which are not required for the processing of the packet at layer L, but may be important at layer $L+1$.

R_n^L contains all the header information of the $n-1$-st packet (at layers $L-1$, L, and $L+1$) and that of the n-th packet at layer $L-1$

$$\begin{aligned} R_n^L \;=\; & \big\{\mathbf{k}_{n-1}^{L-1}, \mathbf{k}_{n-1}^{L}, \mathbf{k}_{n-1}^{L+1}, \mathbf{k}_{n}^{L-1}, \mathbf{p}_{n-1}^{L-1}, \mathbf{p}_{n-1}^{L}, \mathbf{p}_{n-1}^{L+1}, \mathbf{p}_{n}^{L-1}, \\ & \mathbf{u}_{n-1}^{L-1}, \mathbf{u}_{n-1}^{L}, \mathbf{u}_{n-1}^{L+1}, \mathbf{u}_{n}^{L-1}\big\}. \end{aligned}$$

In addition, data not covered by the CRC at layer L are denoted by $\mathbf{x}_n^L$.

All the bits protected by the CRC are collected in the vector $\mathbf{r}_n^L = \left[\mathbf{k}_n^L, \mathbf{p}_n^L, \mathbf{u}_n^L, \mathbf{o}_n^L\right]$ which contains the above defined fields. Note that the order of the bits in $\mathbf{r}_n^L$ does not correspond to the order in which data are actually transmitted in the n-th packet, but we use this notation for mathematical convenience. The CRC $\mathbf{c}_n^L$ associated to $\mathbf{r}_n^L$ is evaluated as $\mathbf{c}_n^L = \mathcal{F}^L\left(\mathbf{r}_n^L\right)$, where $\mathcal{F}^L$ is a generic encoding function.

When there is no ambiguity, the indices n and the exponents L are omitted in what follows.

The evaluation of $\mathbf{c}$ depends on a generator polynomial $g(x) = \sum_{i=0}^{\ell(\mathbf{c})} g_i x^i$ characterizing the CRC [2]. A systematic generator matrix $\mathbf{G} = [\mathbf{I}, \mathbf{\Pi}]$ can be associated to $g(x)$, taking into account the reordering of the bits in $\mathbf{r}$. Using $\mathbf{G}$, $\mathbf{c}$ may be obtained iteratively as follows

$$\begin{cases} \mathbf{c}^0 = \mathbf{0}, \\ \mathbf{c}^{j+1} = \mathcal{F}(\mathbf{r}^{j+1}) = \mathbf{c}^j \oplus (r_{j+1} \cdot \boldsymbol{\pi}(j+1)). \end{cases} \tag{1}$$

In (1), $\mathbf{r}^j = [r_1 \dots r_j, 0 \dots 0]$, $\oplus$ is the XOR operator, and $\boldsymbol{\pi}(j)$ represents the parity vector associated to bit r_j, which corresponds to the j-th line of $\mathbf{\Pi}$. After $\ell(\mathbf{r})$ iterations, $\mathbf{c}^{\ell(\mathbf{r})} = \mathcal{F}(\mathbf{r}) = \mathbf{c}$.

Assume that the data have been transmitted over an AWGN channel (Gaussian noise of zero mean and variance σ^2), and that soft values are forwarded inside the receiver from each layer to the next one. Noisy data and CRC coming from layer $L-1$ are denoted as $\mathbf{y} = [\mathbf{y}_k, \mathbf{y}_p, \mathbf{y}_u, \mathbf{y}_o, \mathbf{y}_c]$, which includes observations (at PHY layer) or estimations (at other layers) of $\mathbf{k}$, $\mathbf{p}$, $\mathbf{u}$, $\mathbf{o}$, and $\mathbf{c}$.

Since $\mathbf{k}$ and $\mathbf{p}$ are known or may be exactly predicted from the already received data, only $\mathbf{u}$ remains to be estimated. A MAP estimator

$$\widehat{\mathbf{u}}_{\text{MAP}} = \arg\max_{\mathbf{u}} P(\mathbf{u}|\mathbf{k}, \mathbf{p}, R, \mathbf{y}_u, \mathbf{y}_o, \mathbf{y}_c), \tag{2}$$

is thus developed, taking into account the observations $\mathbf{y}$, the knowledge of $\mathbf{k}$, $\mathbf{p}$, and R, as well as the CRC properties. After some derivations, one obtains

$$\widehat{\mathbf{u}}_{\text{MAP}} = \arg\max_{\mathbf{u}} P(\mathbf{u}, \mathbf{y}_u, \mathbf{y}_o, \mathbf{y}_c|\mathbf{k}, \mathbf{p}, R). \tag{3}$$

Given that the channel is memoryless and assuming that $\mathbf{o}$ is independent of R, one gets

$$\begin{aligned} P(\mathbf{u}, \mathbf{o}, \mathbf{y}_u, \mathbf{y}_o, \mathbf{y}_c|\mathbf{k}, \mathbf{p}, R) \;=\; & P(\mathbf{u}|\mathbf{k}, \mathbf{p}, R) \\ & P(\mathbf{y}_u|\mathbf{u}) P(\mathbf{o}, \mathbf{y}_o, \mathbf{y}_c|\mathbf{k}, \mathbf{p}, \mathbf{u}). \end{aligned} \tag{4}$$

For the sake of generality, assume that $\mathbf{u}$ does not necessarily take all the $2^{\ell(\mathbf{u})}$ values, and that a study of the protocol allows to define $\Omega_u = \Omega_u(\mathbf{k}, \mathbf{p}, R)$, the set of possible values of $\mathbf{u}$.

Further assume that these values are equally likely. Thus one may write

$$P(\mathbf{u}|\mathbf{k},\mathbf{p},R) = P(\mathbf{u}|\Omega_u) = 1/|\Omega_u|, \quad (5)$$

where $|\Omega_u|$ denotes the cardinal number of Ω_u.

Marginalizing (4) over the $2^{\ell(\mathbf{o})}$ combinations of $\mathbf{o}$, one obtains

$$\begin{aligned} P(\mathbf{u},\mathbf{y}_u,\mathbf{y}_o,\mathbf{y}_c|\mathbf{k},\mathbf{p},R) &= P(\mathbf{u}|\Omega_u)P(\mathbf{y}_u|\mathbf{u}) \\ &\quad \sum_{\mathbf{o}} P(\mathbf{o},\mathbf{y}_o,\mathbf{y}_c|\mathbf{k},\mathbf{p},\mathbf{u}), \quad (6) \end{aligned}$$

where $\sum_{\mathbf{o}} P(\mathbf{o},\mathbf{y}_o,\mathbf{y}_c|\mathbf{k},\mathbf{p},\mathbf{u})$ obviously involves the properties of the CRC. Finally, substituting (6) in (3), the MAP estimator is expressed as

$$\widehat{\mathbf{u}}_{\text{MAP}} = \arg\max_{\mathbf{u}\in\Omega_u} P(\mathbf{y}_u|\mathbf{u})\Psi(\mathbf{k},\mathbf{p},\mathbf{u},\mathbf{y}_o,\mathbf{y}_c), \quad (7)$$

with $\Psi(\mathbf{k},\mathbf{p},\mathbf{u},\mathbf{y}_o,\mathbf{y}_c) = \sum_{\mathbf{o}} P(\mathbf{o},\mathbf{y}_o,\mathbf{y}_c|\mathbf{k},\mathbf{p},\mathbf{u})$.

Being very general, the above equations encompass many different situations. For the sake of clarity, the following section details the evaluation of $\widehat{\mathbf{u}}_{\text{MAP}}$ in several practical situations.

IV. Practical Evaluation of the MAP Estimator

A. The set $\mathbf{o}$ is empty

There are many circumstances in which all the bits covered by the CRC belong only to the sets $\mathbf{k}$, $\mathbf{p}$, or $\mathbf{u}$. In these cases, there is no $\mathbf{o}$ and (3) simplifies to

$$\widehat{\mathbf{u}}_{\text{MAP}} = \arg\max_{\mathbf{u}\in\Omega_u} P(\mathbf{y}_u|\mathbf{u})P\left(\mathbf{y}_c|\mathcal{F}([\mathbf{k},\mathbf{p},\mathbf{u}])\right), \quad (8)$$

where $\mathcal{F}([\mathbf{k},\mathbf{p},\mathbf{u}])$ is directly evaluated by (1). Hence, an elementary CRC computation replaces the sum over all the possible values of $\mathbf{o}$ and the computational complexity is heavily reduced.

B. The set $\mathbf{o}$ is not empty

When $\mathbf{o}$ is present, we assume that the bits of $\mathbf{o}$ are i.i.d. and do not depend on the other parameters. This is a reasonable approximation since these bits usually depend on the whole corresponding block in the upper layer. The sum in (6) then becomes

$$\Psi(\mathbf{k},\mathbf{p},\mathbf{u},\mathbf{y}_o,\mathbf{y}_c) = \sum_{\mathbf{o}} P(\mathbf{o})P(\mathbf{y}_o|\mathbf{o})P\left(\mathbf{y}_c|\mathcal{F}([\mathbf{k},\mathbf{p},\mathbf{u},\mathbf{o}])\right). \quad (9)$$

Evaluating (9) requires the computation of the sum of probabilities related to the $2^{\ell(\mathbf{o})}$ combinations of $\mathbf{o}$ and to their corresponding CRCs. A direct evaluation has obviously a complexity exponential in $\ell(\mathbf{o})$. This section proposes two methods with reduced complexity: the first one is an exact computation while the second one provides an approximate solution.

C. Exact sum computation

The CRC can be evaluated iteratively over the data $\mathbf{r}$, as shown by (1). More precisely, the value of the CRC associated to the $j+1$ first bits of $\mathbf{r}$ (shortly, at time $j+1$) only depends on the value of the CRC at time j and on the $j+1$-st bit of $\mathbf{r}$. Each value of the CRC at time j leads to two different values of the CRC at time $j+1$. Consequently, the evolution of the CRC values according to the bits of $\mathbf{r}$ can be described by a trellis. In this trellis, states correspond to the $2^{\ell(\mathbf{c})}$ possible values of the CRC. Transitions are determined by the bits of $\mathbf{r}$. At each time $j = 1 \dots \ell(\mathbf{r})$, we study the contribution of r_j (the j-th bit of $\mathbf{r}$) over the global CRC.

In our case, $\mathbf{r} = [\mathbf{k},\mathbf{p},\mathbf{u},\mathbf{o}]$. Data contained in $\mathbf{k}$ and $\mathbf{p}$ are assumed to be known, thus have a fixed contribution to the estimate of $\mathbf{u}$. Consequently, for a given value of $\mathbf{u} \in \Omega_u$, we want to determine its probability according to the observations $\mathbf{y}$ and to the CRC properties (depending on the bits belonging to $\mathbf{o}$). Each value of $\mathbf{u}$, in conjunction with $\mathbf{k}$ and $\mathbf{p}$ defines the initial state in the trellis (there is no contribution from $\mathbf{o}$). Each new bit in $\mathbf{o}$ may provide two new possible states, thus defining a trellis. For any value of $\mathbf{o}$, one gets a path starting from the *same* state associated to $\mathcal{F}([\mathbf{k},\mathbf{p},\mathbf{u},\mathbf{0}])$ and ending in the state associated to $\mathcal{F}([\mathbf{k},\mathbf{p},\mathbf{u},\mathbf{o}])$.

This problem is clearly similar to computing the APPs of the inputs (here parts of inputs) from the measured outputs of block codes. Therefore, our method has many similarities with [24] which deals with soft decoding of block codes. In his paper, Wolf proposes a method based on a trellis, built from the parity check matrix, for the decoding of linear block codes. In our work, the computation is different since the code is divided in three parts: a known portion (vectors $\mathbf{k}$ and $\mathbf{p}$), a candidate value ($\mathbf{u}$), and an unknown part ($\mathbf{o}$ and $\mathbf{c}$). We want to find the best combination of $\mathbf{u}$ by taking into account the redundancy of the code (given by $\mathbf{c}$). The trellis is thus applied to the portions $\mathbf{o}$ and $\mathbf{c}$ for given $\mathbf{k}$, $\mathbf{p}$, and $\mathbf{u}$. Additionally, the technique does not estimate $\mathbf{o}$ and $\mathbf{c}$, but evaluates the probability associated to the $2^{\ell(\mathbf{o})}$ combinations of $[\mathbf{o},\mathbf{c}]$.

In the following, we propose a solution based on a backward construction of the trellis to directly evaluate (9) for all the values of $\mathbf{u} \in \Omega_u$. For that purpose, assume that $\bar{\mathbf{o}}^j$ represents the $\ell(\mathbf{o}) - j$ last bits of $\mathbf{o}$ ($\bar{\mathbf{o}}^j = [o_{j+1} \dots o_{\ell(\mathbf{o})}]$) and that $\mathbf{y}_o^j = [y_{o_{j+1}} \dots y_{o_{\ell(\mathbf{o})}}]$ corresponds to their respective observations. Moreover, let $V_{\mathbf{s}_i}(j)$ be the probability associated to state i at time j in the trellis. $V_{\mathbf{s}_i}(j)$ represents the sum of probabilities associated to each combination of $\bar{\mathbf{o}}^j$ and its corresponding CRC when starting from $\mathbf{s}_i$ at time j such as

$$V_{\mathbf{s}_i}(j) = \sum_{\bar{\mathbf{o}}^j} P(\bar{\mathbf{o}}^j)P(\mathbf{y}_o^j|\bar{\mathbf{o}}^j)P\left(\mathbf{y}_c|\mathbf{s}_i \oplus \mathcal{F}([\mathbf{0},\bar{\mathbf{o}}^j])\right), \quad (10)$$

for $i = 0, 1 \dots 2^{\ell(\mathbf{c})} - 1$.

Applying (10) to state i at time $j-1$ results in

$$\begin{aligned} V_{\mathbf{s}_i}(j-1) = & \sum_{\bar{\mathbf{o}}^{j-1}} P(\bar{\mathbf{o}}^{j-1})P(\mathbf{y}_o^{j-1}|\bar{\mathbf{o}}^{j-1}) \\ & P\left(\mathbf{y}_c|\mathbf{s}_i \oplus \mathcal{F}([\mathbf{0},\bar{\mathbf{o}}^{j-1}])\right) \quad (11) \end{aligned}$$

which leads to

$$\begin{aligned} V_{\mathbf{s}_i}(j-1) &= P(o_j=0)P(y_{o_j}|o_j=0) \\ &\quad \sum_{\bar{\mathbf{o}}^j} P(\bar{\mathbf{o}}^j)P(\mathbf{y}_{\bar{o}}^j|\bar{\mathbf{o}}^j)P\left(\mathbf{y}_c|\mathbf{s}_i \oplus \mathcal{F}([\mathbf{0},\bar{\mathbf{o}}^j])\right) \\ &\quad +P(o_j=1)P(y_{o_j}|o_j=1) \\ &\quad \sum_{\bar{\mathbf{o}}^j} P(\bar{\mathbf{o}}^j)P(\mathbf{y}_{\bar{o}}^j|\bar{\mathbf{o}}^j)P\left(\mathbf{y}_c|\mathbf{s}_q \oplus \mathcal{F}([\mathbf{0},\bar{\mathbf{o}}^j])\right) \\ &\quad = P(o_j=0)P(y_{o_j}|o_j=0)V_{\mathbf{s}_i}(j) \\ &\quad +P(o_j=1)P(y_{o_j}|o_j=1)V_{\mathbf{s}_q}(j), \end{aligned} \tag{12}$$

where $\mathbf{s}_q = \mathbf{s}_i \oplus \boldsymbol{\pi}(j)$. (12) above is the key for computing $V_{\mathbf{s}_i}(j)$ through a backward iteration over the bits of $\mathbf{o}$.

After $\ell(\mathbf{o})$ iterations, $V_{\mathbf{s}_i}(0)$ may be expressed as

$$V_{\mathbf{s}_i}(0) = \sum_{\mathbf{o}} P(\mathbf{o})P(\mathbf{y}_o|\mathbf{o})P\left(\mathbf{y}_c|\mathbf{s}_i \oplus \mathcal{F}([\mathbf{0},\mathbf{o}])\right), \tag{13}$$

for $i = 0, 1 \ldots 2^{\ell(\mathbf{c})}-1$. This method allows to simultaneously compute (9) for *all* values of $\mathbf{u} \in \Omega_u$. It is no more necessary to construct a new trellis for each value of $\mathbf{u}$. For a given $\mathbf{u}$, the corresponding probability is given by

$$\begin{aligned} V_{\mathcal{F}([\mathbf{k},\mathbf{p},\mathbf{u},\mathbf{0}])}(0) &= \sum_{\mathbf{o}} P(\mathbf{o})P(\mathbf{y}_o|\mathbf{o}) \\ &\quad P\left(\mathbf{y}_c|\mathcal{F}([\mathbf{k},\mathbf{p},\mathbf{u},\mathbf{0}]) \oplus \mathcal{F}([\mathbf{0},\mathbf{o}])\right) \\ &= \Psi(\mathbf{k},\mathbf{p},\mathbf{u},\mathbf{y}_o,\mathbf{y}_c). \end{aligned}$$

The steps for evaluating (9) backwards are summarized below. The trellis is constructed by starting from $j = \ell(\mathbf{o})$ and going backwards to $j = 0$.

Step 1 - At time $j = \ell(\mathbf{o})$, $V_{\mathbf{s}_i}(\ell(\mathbf{o})) = P(\mathbf{y}_c|\mathbf{s}_i)$ for $i = 0, 1 \ldots 2^{\ell(\mathbf{c})}-1$.

Step 2 - For $j = \ell(\mathbf{o})-1 \ldots 1, 0$, $V_{\mathbf{s}_i}(j)$ is updated according to (12) as

$$\begin{aligned} V_{\mathbf{s}_i}(j) &= P(o_{j+1}=0)P(y_{o_{j+1}}|o_{j+1}=0)V_{\mathbf{s}_i}(j+1) \\ &\quad +P(o_{j+1}=1)P(y_{o_{j+1}}|o_{j+1}=1)V_{\mathbf{s}_q}(j+1), \end{aligned}$$

where $i = 0, 1 \ldots 2^{\ell(\mathbf{c})}-1$ and $\mathbf{s}_q = \mathbf{s}_i \oplus \boldsymbol{\pi}(j+1)$.

Step 3 - After $\ell(\mathbf{o})$ iterations, for any value $\mathbf{u} \in \Omega_u$, $\Psi(\mathbf{k},\mathbf{p},\mathbf{u},\mathbf{y}_o,\mathbf{y}_c) = V_{\mathcal{F}([\mathbf{k},\mathbf{p},\mathbf{u},\mathbf{0}])}(0)$.

With this method, we can directly evaluate (9) for each state i such that $\mathbf{s}_i = \mathcal{F}([\mathbf{k},\mathbf{p},\mathbf{u},\mathbf{0}])$ with $\mathbf{u} \in \Omega_u$. The global complexity of the process is thus $\mathcal{O}\left(\ell(\mathbf{o})2^{\ell(\mathbf{c})}\right)$. Example 1 illustrates the trellis constructed for a backward evaluation of $\Psi(\mathbf{k},\mathbf{p},\mathbf{u},\mathbf{y}_o,\mathbf{y}_c)$.

Example 1: The trellis obtained for a systematic binary Hamming$(7,4)$ code with

$$\boldsymbol{\Pi} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix},$$

is represented in Fig. 3. In this example, we assume that $\ell([\mathbf{k},\mathbf{p},\mathbf{u}]) = 1$ bit and $\ell(\mathbf{o}) = 3$ bits. The sum in (9) is then simultaneously computed for the two possible initial states $\mathcal{F}([\mathbf{k},\mathbf{p},\mathbf{u},\mathbf{0}]) = [0,0,0]$ and $\mathcal{F}([\mathbf{k},\mathbf{p},\mathbf{u},\mathbf{0}]) = [1,1,0]$ obtained when $[\mathbf{k},\mathbf{p},\mathbf{u}] = [0]$ and $[\mathbf{k},\mathbf{p},\mathbf{u}] = [1]$ respectively. $\diamondsuit$

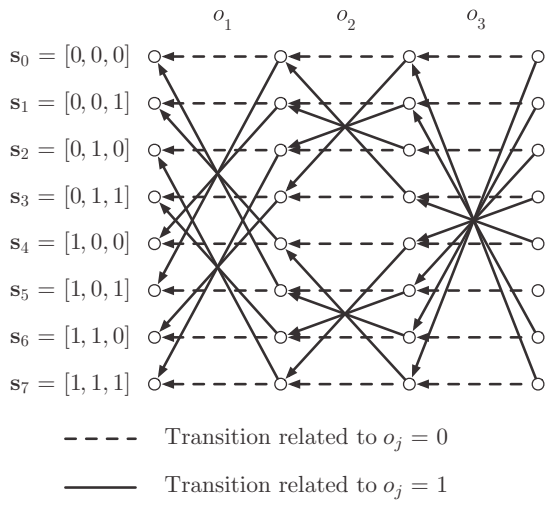

Fig. 3. Trellis obtained with the backward construction

D. Approximate sum computation

Most CRCs are larger than 16 bits and the complexity $\mathcal{O}\left(\ell(\mathbf{o})2^{\ell(\mathbf{c})}\right)$ is too large to allow a real-time implementation of the method presented in Section IV-C. An approximate computation consists in splitting the CRC into M_b blocks of $\ell(\mathbf{c})/M_b$ bits, each block being assumed statistically independent from the others. Thus, $\mathbf{y}_c$ may be written as $\mathbf{y}_c = [\mathbf{y}_{c_1}, \mathbf{y}_{c_2} \ldots \mathbf{y}_{c_{M_b}}]$. Using the independence approximation, the sum in (9) becomes

$$\Psi(\mathbf{k},\mathbf{p},\mathbf{u},\mathbf{y}_o,\mathbf{y}_c) \approx \prod_{m=1}^{M_b} \Psi_m(\mathbf{k},\mathbf{p},\mathbf{u},\mathbf{y}_o,\mathbf{y}_{c_m}), \tag{14}$$

with

$$\begin{aligned} \Psi_m(\mathbf{k},\mathbf{p},\mathbf{u},\mathbf{y}_o,\mathbf{y}_{c_m}) &= \sum_{\mathbf{o}} P(\mathbf{o})P(\mathbf{y}_o|\mathbf{o}) \\ &\quad P\left(\mathbf{y}_{c_m}|\mathcal{F}_m([\mathbf{k},\mathbf{p},\mathbf{u},\mathbf{o}])\right), \end{aligned} \tag{15}$$

where $\mathcal{F}_m$ is the encoding function associated to the columns $(m-1)\cdot\frac{\ell(\mathbf{c})}{M_b}+1$ to $m\cdot\frac{\ell(\mathbf{c})}{M_b}$ of $\boldsymbol{\Pi}$, corresponding to a partial CRC of $\ell(\mathbf{c})/M_b$ bits.

The evaluation of (15) is similar to that of Ψ described in Section IV-C. The only difference lies in the size of the trellis: $2^{\ell(\mathbf{c})/M_b}$ states have to be considered at any depth (instead of $2^{\ell(\mathbf{c})}$ states without splitting the CRC). The total complexity for evaluating (14) is now $\mathcal{O}\left(M_b\ell(\mathbf{o})2^{\ell(\mathbf{c})/M_b}\right)$, at the cost of a slightly suboptimal performance.

V. Application to 802.11 Standard

In this paper, we focus on the downlink multimedia transmission over the 802.11 radio interface [18]. First, the format of packets at PHY and MAC layers are briefly recalled in Sections V-A and V-B. Intra-layer and inter-layer redundancy are then described in Section V-C. The resulting processing details for the enhanced permeable layer mechanism are finally proposed in Sections V-D, V-E, and V-F.

A. DSSS PHY layer description

At PHY layer, the 802.11 standard provides 1 or 2 Mbps transmission rate in the 2.4 GHz band using either *Frequency Hopping Spread Spectrum* (FHSS) or *Direct Sequence Spread Spectrum* (DSSS). In DSSS, an 11-chip Barker code sequence is used for spreading the 1 Mbps bitstream. The coded flow thus represents an 11 MHz baseband signal. A DBPSK or DQPSK modulation is applied depending on the required bitrate.

The DSSS PHY packet format is illustrated in Fig. 4. The preamble and the header are transmitted by using the 1 Mbps DBPSK modulation while the payload is modulated either in 1 Mbps DBPSK or 2 Mbps DQPSK. In such PHY packets, the *SYNC* and *SFD* fields consist of 144 known bits, which are not protected by the CRC. These fields are used to estimate the variance of the channel noise (see Section V-F).

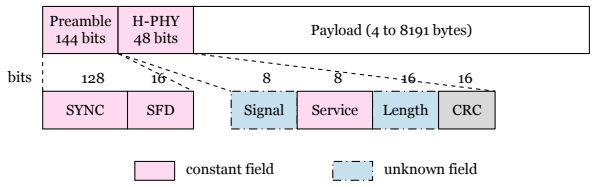

Fig. 4. PHY packet format in 802.11 standard

The CCITT CRC-16 $\mathbf{c}^{\text{PHY}}$ of 2 bytes protects the *Signal*, *Service*, and *Length* fields; its associated encoding function is denoted by $\mathcal{F}^{\text{PHY}}$. The payload, assumed to contain only one MAC packet, is not protected at this layer. *Service* is reserved for future recommendation. It is set to 00_{16}, and included in $\mathbf{k}^{\text{PHY}}$, according to the notations of Section III. *Signal* indicates the payload modulation and is equal to $0A_{16}$ or 14_{16} for 1 or 2 Mbps bitrate respectively. *Length* indicates on 2 bytes the number of microseconds required to transmit the payload. It depends on both the bitrate and the payload size. It ranges from 16 to $2^{16}-1$. *Signal* and *Length* form thus $\mathbf{u}^{\text{PHY}}$. At this layer, $\mathbf{p}^{\text{PHY}} = \mathbf{o}^{\text{PHY}} = \emptyset$ and $\mathbf{x}^{\text{PHY}}$ contains the $\ell(\mathbf{x}^{\text{PHY}})$ bits of payload.

B. MAC layer description

The MAC packet format is depicted in Fig. 5. In this packet, the CRC $\mathbf{c}^{\text{MAC}}$ of 4 bytes protects both the header fields and the payload; its encoding function is $\mathcal{F}^{\text{MAC}}$.

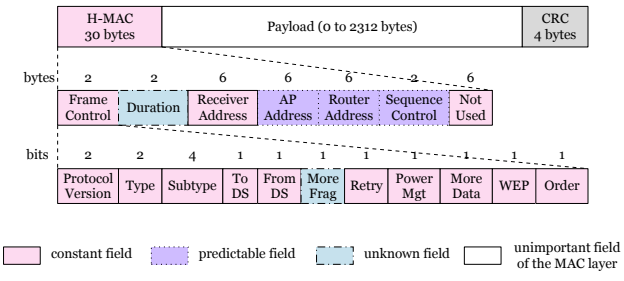

Fig. 5. MAC packet format in 802.11 standard

Considering a non-encrypted downlink transmission of ordered MAC data packets with deactivated retransmission and power-save mode, the 2-byte *Frame Control* field except the *More Frag* flag are assumed to be known. The 6-byte *Receiver Address* field contains the MAC address of the receiver and is thus known. The last field of the MAC header is reserved for local wireless networks and is composed of 6 bytes of zeros in this study. Using the notations of Section III, all the previously mentioned fields may thus be embedded in $\mathbf{k}^{\text{MAC}}$.

The 6-byte *AP Address* field contains the MAC address of the access point AP. This address is transmitted during the medium reservation procedure (RTS-CTS) and may be totally deduced by the receiver. The 6-byte *Router Address* field corresponds to the MAC address of the router. Assuming that the AP is connected to a single router and that the router address has been already received in other information packets, *Router Address* 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. *Sequence Control* can be estimated by the receiver. All these predictable fields are represented by $\mathbf{p}^{\text{MAC}}$.

The *More Frag* flag specifies if the current MAC data packet is the last fragment composing an IP packet. The 2 bytes of *Duration* indicate the number of microseconds required to transmit the next MAC fragment and some control packets. Its value depends on the current modulation and the size of the coming MAC data packet. These two fields are embedded in $\mathbf{u}^{\text{MAC}}$. Finally, the payload contains the data to be transmitted and its size is between 0 and 2312 bytes. It is represented by $\mathbf{o}^{\text{MAC}}$.

C. Identifying intra-layer and inter-layer correlations

To evidence these correlations, the transactions at MAC layer have to be described.

In the 802.11 standard, transmission of each IP packet is initialized by a medium reservation procedure at MAC layer consisting of an RTS-CTS exchange. MAC fragments composing the IP packet are then transmitted to the receiver, which acknowledges them (ACK). In this work, control packets such as RTS, CTS, and ACK are assumed to be correctly received. This assumption is reasonable since these packets are small and DBPSK-modulated. Only errors in data packets (or fragments) will be considered. A *Short Inter-Frame Space* (SIFS) of 10 μs separates two packets successively transmitted over the channel. A *Duration* field is included in each packet and its value indicates the number of microseconds required to transmit the next fragment and some specific packets (CTS and ACK). *Duration* allows to adjust the *Network Allocation Vector* (NAV) for the other terminals. The other stations cannot communicate during the NAV period to avoid interferences.

Assume that D_n^{MAC} and B_n^{PHY} represent the value of *Duration* and the transmission bitrate (coded in *Signal*) associated

to the n-th packet transmitted by the AP (either RTS or data packets). Following the MAC layer specifications of 802.11 standard, D_n^{MAC} is defined as

$$D_n^{\text{MAC}} = 3T_{\text{SIFS}} + 3T_{\text{OVH}} + 2\ell_{\text{C-A}}/B_n^{\text{PHY}} + \ell(\mathbf{x}_{n+1}^{\text{PHY}})/B_n^{\text{PHY}}, \quad (16)$$

except for the last fragment of an IP packet, *i.e.*, when the value of *More Frag* $M_n^{\text{MAC}} = 0$. In this case, one has

$$D_n^{\text{MAC}} = T_{\text{SIFS}} + T_{\text{OVH}} + \ell_{\text{C-A}}/B_n^{\text{PHY}}. \quad (17)$$

In (16) and (17), T_{SIFS} denotes the duration of a SIFS and T_{OVH} represents the duration for transmitting at 1 Mbps the PHY overhead (composed of the preamble and the header of constant size). The other terms depend on the current bitrate B_n^{PHY}. CTS and ACK have the same constant size $\ell_{\text{C-A}}$ and $\ell_{\text{C-A}}/B_n^{\text{PHY}}$ thus corresponds to the duration for sending one of these packets. Finally, $\ell(\mathbf{x}_{n+1}^{\text{PHY}})/B_n^{\text{PHY}}$ refers to the transmission duration of the next PHY payload of $\ell(\mathbf{x}_{n+1}^{\text{PHY}})$ bits.

D. PHY header recovery

For the n-th packet at PHY layer, the observations associated to $\mathbf{k}_n^{\text{PHY}}$, $\mathbf{u}_n^{\text{PHY}}$, and $\mathbf{c}_n^{\text{PHY}}$ defined in Section V-A are collected in $\mathbf{y}_n^{\text{PHY}} = [\mathbf{y}_{k,n}^{\text{PHY}}, \mathbf{y}_{u,n}^{\text{PHY}}, \mathbf{y}_{c,n}^{\text{PHY}}]$. In addition, $\mathbf{y}_{x,n}^{\text{PHY}}$ denotes the observations associated to the $\ell(\mathbf{x}_n^{\text{PHY}})$ bits of the payload $\mathbf{x}_n^{\text{PHY}}$, which is not protected by the CRC.

The number of possible values taken by $\mathbf{u}^{\text{PHY}}$ is significantly reduced when exploiting the *Duration* field contained in the previously received MAC packet (either an RTS or a data packet). Using B_{n-1}^{PHY} and D_{n-1}^{MAC}, one may deduce $\ell(\mathbf{x}_n^{\text{PHY}})$ from (16) as

$$\ell(\mathbf{x}_n^{\text{PHY}}) = \left(D_{n-1}^{\text{MAC}} - 3T_{\text{SIFS}} - 3T_{\text{OVH}} - 2\ell_{\text{C-A}}/B_{n-1}^{\text{PHY}}\right) B_{n-1}^{\text{PHY}}. \quad (18)$$

Then, the duration L_n^{PHY} coded in the *Length* field of the current PHY packet is computed as

$$L_n^{\text{PHY}} = \ell(\mathbf{x}_n^{\text{PHY}})/B_n^{\text{PHY}}. \quad (19)$$

In (18), $\ell(\mathbf{x}_n^{\text{PHY}})$ is totally determined assuming correct estimation of the header of the previous packet. Then, according to (19), L_n^{PHY} may only take two values depending on B_n^{PHY}, which are stored in $\Omega_{u,n}^{\text{PHY}}$. Integrating these properties in (8), one obtains

$$\widehat{\mathbf{u}}_n^{\text{PHY}} = \arg \max_{\mathbf{u}_n^{\text{PHY}} \in \Omega_{u,n}^{\text{PHY}}} P(\mathbf{y}_{u,n}^{\text{PHY}} | \mathbf{u}_n^{\text{PHY}}) P(\mathbf{y}_{c,n}^{\text{PHY}} | \mathbf{c}_n^{\text{PHY}}), \quad (20)$$

with $\mathbf{c}_n^{\text{PHY}} = \mathcal{F}^{\text{PHY}}([\mathbf{k}_n^{\text{PHY}}, \mathbf{u}_n^{\text{PHY}}])$.

E. MAC header recovery

The PHY layer provides

$$\mathbf{y}_n^{\text{MAC}} = \mathbf{y}_{x,n}^{\text{PHY}} = [\mathbf{y}_{k,n}^{\text{MAC}}, \mathbf{y}_{p,n}^{\text{MAC}}, \mathbf{y}_{u,n}^{\text{MAC}}, \mathbf{y}_{o,n}^{\text{MAC}}, \mathbf{y}_{c,n}^{\text{MAC}}]$$

at the input of MAC layer [1]. It contains the observations associated to $\mathbf{k}_n^{\text{MAC}}$, $\mathbf{p}_n^{\text{MAC}}$, $\mathbf{u}_n^{\text{MAC}}$, $\mathbf{o}_n^{\text{MAC}}$, and $\mathbf{c}_n^{\text{MAC}}$ specified in Section V-B.

[1]When encryption is activated, the *WEP* flag in the MAC header is set to 1. In addition, $\mathbf{y}_{o,n}^{\text{MAC}}$ and $\mathbf{y}_{c,n}^{\text{MAC}}$ are the observations of the encrypted bits (plainstream XORed with a pseudo-random keystream). Decryption may easily be performed at receiver side by inverting some LLRs in $\mathbf{y}_{o,n}^{\text{MAC}}$ and $\mathbf{y}_{c,n}^{\text{MAC}}$ according to the known keystream.

The number of possible combinations for $\mathbf{u}_n^{\text{MAC}}$ may be significantly reduced when exploiting (16) and (17). Note that D_n^{MAC} is fully determined when $M_n^{\text{MAC}} = 0$. When $M_n^{\text{MAC}} = 1$, the value of *Duration* depends on the next PHY payload size. The number of combinations is associated to the range of MAC payload size. Considering that the payload contains an entire number of bytes, the possible values of $\ell(\mathbf{x}_{n+1}^{\text{PHY}})$ in (16) are given by

$$\ell(\mathbf{x}_{n+1}^{\text{PHY}}) = \ell_{\text{HDR}} + 8i, \quad (21)$$

where $i = 1, 2 \ldots 2312$. In (21), ℓ_{HDR} specifies the known size of the header in a MAC data packet. Then, using (16), (17), and (21), one may show that $\mathbf{u}_n^{\text{MAC}}$ is limited to 2313 combinations which are inserted in $\Omega_{u,n}^{\text{MAC}}$. Combining these properties in (7), one obtains

$$\begin{aligned} \widehat{\mathbf{u}}_n^{\text{MAC}} &= \arg \max_{\mathbf{u}_n^{\text{MAC}} \in \Omega_{u,n}^{\text{MAC}}} P(\mathbf{y}_{u,n}^{\text{MAC}} | \mathbf{u}_n^{\text{MAC}}) \\ &\quad \Psi(\mathbf{k}_n^{\text{MAC}}, \mathbf{p}_n^{\text{MAC}}, \mathbf{u}_n^{\text{MAC}}, \mathbf{y}_{o,n}^{\text{MAC}}, \mathbf{y}_{c,n}^{\text{MAC}}), \end{aligned} \quad (22)$$

where the second term can be computed with methods presented in Sections IV-C and IV-D.

F. Global scheme

Fig. 6 illustrates the improved permeable layer mechanism applied to the PHY and MAC layers at the receiver, emphasizing on the exchange of information between layers and between consecutive packets, as presented in Sections V-D and V-E. At PHY layer, for the n-th packet, the header recovery block first exploits the knowledge of the fields *Duration* and *Signal* (decoded in the $n-1$-st PHY and MAC packets) to construct a reduced set of combination Ω_u^{PHY} associated to the unknown part $\mathbf{u}_n^{\text{PHY}}$. The PHY header is then estimated by taking into account the observations of the PHY header along with the properties of the CRC. The recovered header is then be used as reference to decode the header of the next PHY packet. At MAC layer, the header recovery block combines the soft information transmitted by the PHY layer, the knowledge of the decoded field *Signal*, together with the CRC of the MAC layer to find an estimate of the MAC header. The estimated field *Duration* is exploited at the PHY layer to decode the header of the $n+1$-st PHY packet.

In addition, we consider that $\mathbf{y}_{s,n}^{\text{PHY}}$ represents the observation vector of the known preamble $\mathbf{s}^{\text{PHY}}$. As explained in Section V-A, the receiver synchronization is performed with $\mathbf{s}^{\text{PHY}}$. We simultaneously estimate σ^2 from $\mathbf{s}^{\text{PHY}}$ and $\mathbf{y}_{s,n}^{\text{PHY}}$. This measure is essential for working with soft information, as it allows the evaluation of all the likelihoods. The estimator $\widehat{\sigma}^2$ is given by

$$\widehat{\sigma}^2 = \|\mathbf{y}_{s,n}^{\text{PHY}} - \mathbf{s}^{\text{PHY}}\|^2 / \ell(\mathbf{s}^{\text{PHY}}). \quad (23)$$

Computational complexity is minimized by deactivating the robust header recovery processing when the normal CRC check is successful. It should also be deactivated when the quality of the soft information provided by the lower layer is too poor, *i.e.*, when the signal power is lower than a predefined threshold. In such a case, the packet is retransmitted.

VI. Simulation Results

The improved permeable scheme for 802.11 PHY and MAC layers has been implemented (see Fig. 6). A transmission device consisting of a transmitter (AP), an AWGN channel, and a receiver has been simulated using a C program. The AP generates PHY and MAC packets following the format defined in Section V. The MAC payloads consist of a variable amount of randomly generated bytes. The transmitter modulates data in DBPSK for all the simulations. The channel coding relative to the 802.11 DSSS standard is deactivated in these simulations. Spreading the data stream by using an 11-chip Barker code sequence would only shift the obtained curves towards lower SNRs, but relative gains are preserved.

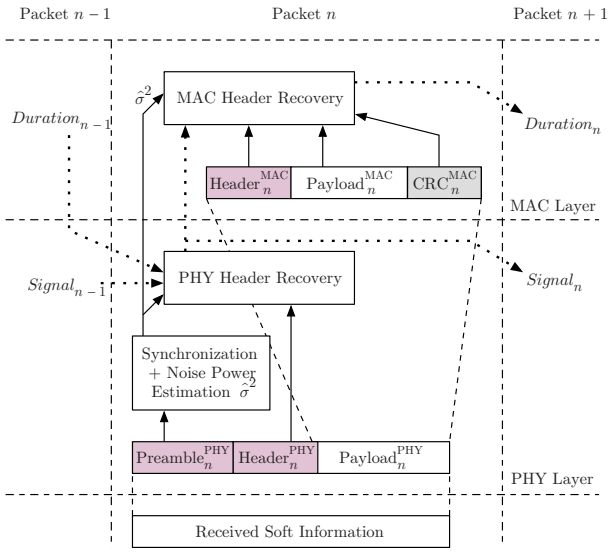

Fig. 6. Proposed scheme for PHY and MAC layers

Three types of header recovery methods are considered at each layer of the receiver. The *standard* decoder performs hard decisions on the data at the channel output. The *robust* decoder exploits only the intra-layer and inter-layer redundancy through a soft decoding algorithm, neglecting the information provided by the CRC. Finally, the *CRC-robust* decoder combines the intra-layer and inter-layer redundancy together with the information provided by the CRC through the soft decoding algorithm presented in Sections V-D and V-E. Performance analysis is done in terms of *Header Error Rate* (HER) versus SNR.

In Fig. 7, the standard, robust, and CRC-robust (see Section V-D) PHY decoders are compared under the assumption that the *Duration* field of the previous MAC packet has been correctly received. Obviously, robust decoders outperform the standard one. An HER of less than 10^{-5} is obtained with the robust decoder for an SNR of 4 dB and with the CRC-robust decoder for an SNR of 2 dB. With the standard decoder, an SNR of at least 15 dB is required to get a comparable HER. At PHY layer, considerable coding gains for a relatively low additional complexity are thus observed, since (8) is used to perform the decoding.

Fig. 8 compares the coding gains obtained by the standard, robust, and CRC-robust (see Section V-E) MAC decoders.

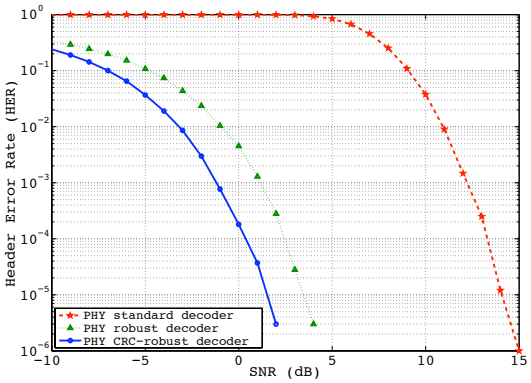

Fig. 7. Header Error Rate (HER) vs. SNR for the standard, robust, and CRC-robust decoders used at the 802.11 PHY layer.

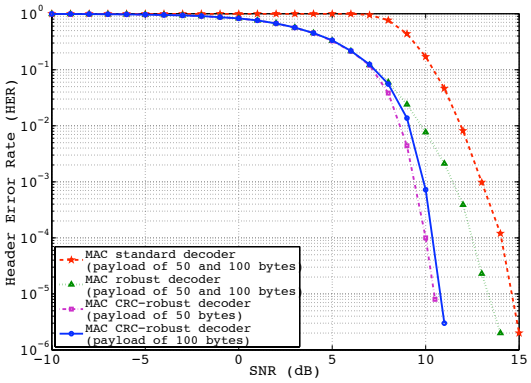

Fig. 8. Header Error Rate (HER) vs. SNR for the standard, robust, and CRC-robust decoders used at the 802.11 MAC layer. Two payload sizes (50 and 100 bytes) have been considered.

Here, the *Bitrate* field of the current PHY packet is assumed to be correctly decoded. Two payload sizes (50 and 100 bytes) have been considered. Moreover, the suboptimal method presented in Section IV-D has been used, dividing the CRC in four blocks of 1 byte each. The shape of the curves is very similar to the results obtained at PHY layer, but with significantly smaller gains. Gains due to the MAC CRC information improve with increasing SNR. With payloads of 100 bytes, HER lower than 10^{-5} are achieved for SNRs of 11 dB, 14 dB, and 15 dB when using CRC-robust, robust, and standard decoders respectively.

Note that the above numbers were obtained under some assumptions (correctly received *Duration* field of the previous MAC packet or *Bitrate* field of the current PHY packet), which allows to study the header recovery mechanism independently at each layer. Our motivation here is to show the large potential interest of such a method.

The MAC processing is more complex than the one done at PHY layer due to the marginalization operation required in (9). The larger the payload, the more complex the decoding process. To reduce the complexity and improve the MAC header recovery performance, the principle of UDP-Lite has been applied at the MAC layer, resulting in a permeable MAC

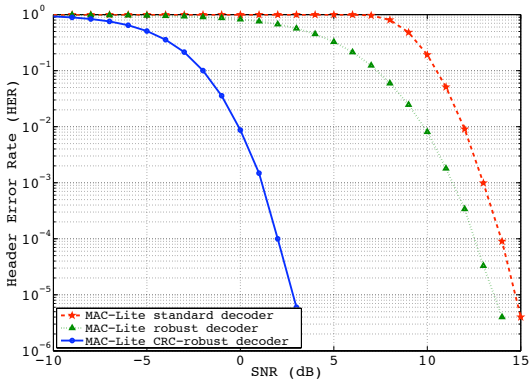

Fig. 9. Header Error Rate (HER) vs. SNR for the standard, robust, and CRC-robust decoders of the MAC-Lite layer.

layer (called *MAC-Lite*) where the CRC protects the MAC header field only, see [25]. In this case, $\mathbf{o}_n^{\text{MAC-L}} = \emptyset$ and (22) becomes

$$\widehat{\mathbf{u}}_n^{\text{MAC-L}} = \arg \max_{\mathbf{u}_n^{\text{MAC-L}} \in \Omega_{u,n}^{\text{MAC-L}}} P(\mathbf{y}_{u,n}^{\text{MAC-L}} | \mathbf{u}_n^{\text{MAC-L}}) P(\mathbf{y}_{c,n}^{\text{MAC-L}} | \mathbf{c}_n^{\text{MAC-L}}), \quad (24)$$

where $\mathbf{c}_n^{\text{MAC-L}} = \mathcal{F}^{\text{MAC-L}}([\mathbf{k}_n^{\text{MAC-L}}, \mathbf{p}_n^{\text{MAC-L}}, \mathbf{u}_n^{\text{MAC-L}}])$.

Standard, robust, and CRC-robust MAC-Lite decoders are depicted in Fig. 9. Comparison with Fig. 8 does not show any difference between MAC and MAC-Lite situations for the standard and robust decoders. This is normal, since the information provided by the CRC is not used by these decoders. However, Fig. 9 demonstrates that the CRC-robust decoder is now significantly more efficient for decoding MAC-lite headers than for decoding classical MAC headers. HER is lower than 10^{-5} for SNRs larger than 3 dB when exploiting the CRC redundancy whereas the two other methods need at least 14 dB. Additionally, the CRC-robust decoder is significantly less complex when processing MAC-Lite headers instead of classical MAC headers, since (24) does not require any marginalization.

Consequently, the combination of the proposed permeable PHY and MAC-Lite layer mechanisms recovers eventually all the PHY and MAC headers from 3 dB SNR onwards. The combination of the proposed permeable PHY and MAC layers reaches this result when the SNR is about 11 dB for an increased complexity. This result demonstrates the potential of replacing the classical MAC layer by the proposed MAC-Lite layer.

VII. Conclusion

A robust header estimation technique has been proposed and has been applied to PHY and MAC layers of WiFi. The main tool of this mechanism consists of a MAP header estimator exploiting jointly the structural properties of the protocol stack along with the CRC redundancy through a soft decoding algorithm. This technique may readily be applied to other layers for various transmission protocols. The estimation technique allows an enhanced permeable layer mechanism (compared, *e.g.*, to UDP-lite) to be defined. This mechanism is particularly well-suited when combined with joint source-channel decoding techniques at Application layer. Simulations with PHY and MAC layers of WiFi illustrate the significant performance gains achieved with the proposed decoding technique. As a result, such techniques allow the headers to be much more robust to channel impairments than the payload, thus avoiding the necessity of packet retransmission in most cases. Adaptation of the proposed technique to IP, UDP-Lite, and RTP layers will be studied in a future work. For these three remaining layers, the CRC is replaced by a checksum. Additionally, at these upper layers, ROHC is an alternative to which the proposed technique has to be compared. One may also try to perform a soft ROHC decoding, thus combining the advantages of a smaller header provided by ROHC and of an increased resilience to errors obtained by the proposed solution.

Acknowledgements

This work has been partly supported by the ANR DIVINE Project, and by the European Commission in the framework of the FP7 Network of Excellence in Wireless Communications NEWCOM++.

References

[1] K. Sayood. *Introduction to Data Compression, Second Edition*. Morgan Kaufmann, San Francisco, 2000.

[2] R. E. Blahut. *Theory and Practice of Error Control Codes*. Addison-Wesley, Reading, MA, 1984.

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

[4] T. Richardson and U. Urbanke. *Modern Coding Theory*. Cambridge University Press, 2008.

[5] M. C. Hong, H. Schwab, L. P. Kondi, and A. K. Katsaggelos. Error concealment algorithms for compressed video. *Signal Processing: Image Communication*, 14(6-8):473–492, 1999.

[6] W.-Y. Kung, C.-S. Kim, and C.-C. J. Kuo. Spatial and temporal error concealment techniques for video transmission over noisy channels. *IEEE trans. on circuits and systems for video technology*, 16(7):789–803, 2006.

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

[8] V. Buttigieg and P.G. Farrell. A MAP decoding algorithm for variable-length error-correcting codes. In *Codes and Cyphers: Cryptography and Coding IV*, pages 103–119, Essex, England, 1995. The Inst. of Mathematics and its Appl.

[9] S. Kaiser and M. Bystrom. Soft decoding of variable-length codes. In *Proc. of ICC*, volume 3, pages 1203–1207, New Orleans, 2000.

[10] R. Thobanen and J. Kliewer. Robust decoding of variable-length encoded markov sources using a three-dimensional trellis. *IEEE Communications Letters*, 7(7):320–322, 2003.

[11] H. Nguyen, P. Duhamel, J. Brouet, and D. Rouffet. Robust VLC sequence decoding exploiting additional video stream properties with reduced complexity. In *Proc. IEEE ICME*, pages 375–378, June 2004. Taipei, Taiwan.

[12] C. Bergeron and C. Lamy-Bergot. Soft-input decoding of variable-length codes applied to the H.264 standard. In *Proc. IEEE 6th Workshop on Multimedia Signal Processing*, pages 87–90, 29 Sept.-1 Oct. 2004.

[13] C.M. Lee, M. Kieffer, and P. Duhamel. Soft decoding of VLC encoded data for robust transmission of packetized video. In *Proc. IEEE ICASSP*, pages 737–740, 2005.

[14] R. Bauer and J. Hagenauer. On variable length codes for iterative source/channel decoding. In *Proc. IEEE Data Compression Conference*, pages 272–282, Snowbird, UT, 1998.

[15] H. Jenkac, T. Stockhammer, and W. Xu. Permeable-layer receiver for reliable multicast transmission in wireless systems. In *Proc. IEEE Wireless Communications and Networking Conference*, volume 3, pages 1805–1811, 13-17 March 2005.

[16] M. G. Martini, M. Mazzoti, C. Lamy-Bergot, J. Huusko, and P. Amon. Content adaptive network aware joint optimization of wireless video transmission. *IEEE Communications Magazine*, 45(1):84–90, 2007.
[17] 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.
[18] ANSI/IEEE. 802.11, part 11 : Wireless LAN medium access control (MAC) and physical layer (PHY) specifications. Technical report, 1999.
[19] L. A. Larzon, M. Degermark, L. E. Jonsson, and G. Fairhurst. The lightweight user datagram protocol (UDP-Lite). Technical Report RFC 3828, The Internet Society, 2004.
[20] M. Van der Schaar and S. Shankar. Cross-layer wireless multimedia transmission: Challenges, principles, and new paradigms. *IEEE Wireless Communications Magazine*, 12(4):50–58, 2005.
[21] C. Bormann (Ed.). Robust header compression (ROHC): Framework and four profiles. Technical Report RFC 3095, 2001.
[22] 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.
[23] C. Marin, P. Duhamel, K. Bouchireb, and M. Kieffer. Robust video decoding through simultaneous usage of residual source information and MAC layer CRC redundancy. In *Proc. IEEE Globecom 07*, pages 2070–2074, 2007.
[24] J. K. Wolf. Efficient maximum-likelihood decoding of linear block codes using a trellis. *IEEE Trans. Inform. Theory*, 24(1):76–80, 1978.
[25] J. Huusko, J. Vehkapera, P. Amon, C. Lamy-Bergot, G. Panza, J. Peltola, and M. G. Martini. Cross-layer architecture for scalable video transmission in wireless network. *IEEE Signal Processing: Image Communication*, 22(3):317–330, 2007.

PLACE PHOTO HERE

Cédric Marin Cédric Marin was born in Chaumont, France, in 1980. He received the Ing. degree in electrical engineering from the University Paris-Sud, Orsay, in 2003, and a master degree in signal processing in 2004 from the same institution. In 2009, he receives a PhD degree in electrical and computer engineering at the University of Paris-Sud. This PhD was prepared jointly with the Alcatel-Lucent Bell Labs center.

His interests are in the areas of video compression, joint source-channel decoding, iterative decoding, wireless transmission protocol, and cross-layer-design. His thesis focuses on the problematics of video transmission over wireless channels.

PLACE PHOTO HERE

Yann Leprovost Yann Leprovost is a research-engineer in the Triple-Play Wireless Networks group, Networking and Networks Research Domain, at Bell Labs in Villarceaux, France. He holds a degree in electronic embedded systems from the Ecole Polytechnique d'OrlÈans, France. He began his carrier as a consultant in embedded software development working for EADS, Thales and Wavecom. He joined Alcatel's Research and Innovation department in 2006 to work on end to end technologies for efficient compression and delivery of video streams over wireless networks. His current research interests are video compression, adaptable and robust streaming platforms, video quality monitoring.

PLACE PHOTO HERE

Michel Kieffer (M'02 – SM'07) was born in Sarreguemines, France in 1972. In 1995, he received the Agrégation in Applied Physics at the Ecole Normale Supérieure de Cachan, France. He received a PhD degree in Control and Signal Processing in 1999, and the Habilitation à Diriger des Recherches degree in 2005, both from the Univ Paris-Sud, Orsay, France. Michel Kieffer is an assistant professor in signal processing for communications at the Univ Paris-Sud and a researcher at the Laboratoire des Signaux et Systèmes, Gif-sur-Yvette, France. His research interests are in joint source-channel coding and decoding techniques for the reliable transmission of multimedia contents. He is also interested in guaranteed parameter and state estimation for systems described by non-linear models.

Michel Kieffer is co-author of more than 90 contributions in journals, conference proceedings, or books. He is one of the co-author of the book *Applied Interval Analysis* published by Springer-Verlag in 2001 and of the book *Joint source-channel decoding: A crosslayer perspective with applications in video broadcasting* published by Academic Press in 2009. He is associate editor of Signal Processing since 2008.

PLACE PHOTO HERE

Pierre Duhamel (F'98) was born in France in 1953. He received the Eng. Degree in Electrical Engineering from the National Institute for Applied Sciences (INSA) Rennes, France in 1975, the Dr. Eng. Degree in 1978, and the Doctorat ès sciences degree in 1986, both from Orsay University, Orsay, France.

From 1975 to 1980, he was with Thomson-CSF, Paris, France, where his research interests were in circuit theory and signal processing, including digital filtering and analog fault diagnosis. In 1980, he joined the National Research Center in Telecommunications (CNET), Issy les Moulineaux, France, where his research activities were first concerned with the design of recursive CCD filters. Later, he worked on fast algorithms for computing Fourier transforms and convolutions, and applied similar techniques to adaptive filtering, spectral analysis and wavelet transforms. From 1993 to Sept. 2000, he has been professor at ENST, Paris (National School of Engineering in Telecommunications) with research activities focused on Signal processing for Communications. He was head of the Signal and Image processing Department from 1997 to 2000. He is now with CNRS/LSS (Laboratoire de Signaux et Systemes, Gif sur Yvette, France), where he is developing studies in Signal processing for communications (including equalization, iterative decoding, multicarrier systems, cooperation) and signal/image processing for multimedia applications, including source coding, joint source/channel coding, watermarking, and audio processing. He is currently investigating the application of recent information theory results to communication theory.

Dr. Duhamel was chairman of the DSP committee from 1996 to 1998, and a member of the SP for Com committee until 2001. He was an associate Editor of the IEEE Transactions on Signal Processing from 1989 to 1991, an associate Editor for the IEEE Signal Processing Letters, and a guest editor for the special issue of the IEEE Trans. on SP on wavelets.

He was Distiguished lecturer, IEEE, for 1999, and was co-general chair of the 2001 International Workshop on Multimedia Signal Processing, Cannes, France. He was also co-technical chair of ICASSP 06, Toulouse, France. The paper on subspace-based methods for blind equalization, which he co-authored, received the "Best paper award" from the IEEE transactions on SP in 1998. He was awarded the "grand prix France Telecom" by the French Science Academy in 2000. He is the co-author of the book *Joint source-channel decoding: A crosslayer perspective with applications in video broadcasting* published by Academic Press in 2009.