ROBUST TRANSMISSION OF 802.11N PHYSICAL PACKET HEADERS

Zied Jaoua∗+, Anissa Mokraoui-Zergaïnough∗, Pierre Duhamel∗

∗LSS/CNRS, SUPELEC,
Plateau de Moulon, 91 192 Gif sur Yvette, France
+L2TI, Institut Galilée, Université Paris 13,
99 Avenue Jean Baptiste Clément, 93 430 Villetaneuse, France
{jaoua,duhamel}@lss.supelec.fr, anissa.mokraoui@galilee.univ-paris13.fr

ABSTRACT

This paper concerns the robust transmission of physical header frames of the IEEE 802.11n wireless network standard over a noisy channel. The proposed receiver is based on an iterative decoding scheme using the turbo principle where two blocks are serially concatenated. The outer block provides Soft Input Soft Output estimates of the headers of the physical packets based on a MAP estimation, while the inner block is some classical BCJR decoding of the channel code.

The method relies on a strategy proposed recently by the same team, making use of the inherent redundancies of the physical layer headers. This paper proposes an implementation and evaluation in which the channel coding is taken into account. Two cases are studied: (i) the fully general framework of the 802.11n physical layer, and (ii) a specific industrial implementation of the 802.11n physical layer network, which, like most implementations, does not allow all options of the standard. Simulations show that the proposed receiver allows to reduce drastically the number of errors occurring during the transmission compared to a classical hard decision technique.

1. INTRODUCTION

The Open System Interconnection (OSI) model defines and describes a network architecture where the protocol implementations on the various layers stack are specified. Each layer adds a specific header to the payload. These headers contain relevant and useful information (e.g. addresses, mode of transmission, protection codes ...) necessary to decode the transmitted stream. These headers are aimed to be decoded at a given layer and are not currently available to the other ones.

Many protection and adaptation strategies exist at different level layers of the OSI stack. The mechanism usually adopted is to retransmit erroneous frames rather than trying to use them otherwise, for example by using some joint source and channel decoding schemes. In fact, such schemes are able to process the payloads, even in the case where some errors occur. However, in an actual transmission, the payload can be as little as half the total bitstream.

In order to cope with such a situation, some solutions have been developed in the literature (see [1]). In multimedia transmission, the UDP-lite protocol has been developed [2]. The basic idea consists to protect the headers and to accept some errors in the payload so that the information is transmitted to the upper layer. This issue becomes interesting particularly for video applications. Nevertheless, the problem is not completely solved if many headers are damaged meaning that many retransmissions are required.

The concept of cross-layer has been already adopted by the Robust Header Compression (ROHC) protocol which reduces the redundancies available in the various layer overheads [3]. Usually, this strategy is applied at the higher layers, even if the process tends to reach the lower layers in recent studies. Cross layering also took many other aspects, such as in reference [4], the authors exploit the physical layer information (e.g. channel conditions or some coding parameters) to adapt the fragmentation of the MAC layer. In [5] a maximum likelihood estimation of the most important header fields is developed in order to improve throughput since it reduces the retransmission of the damaged packets. In [6], the authors extract and exploit the available redundancies to better estimate the damaged information. This work contributes to the development of the cross-layer strategy, by building on the approach in [6]. We first apply the initial strategy to the PHY layer of the IEEE 802.11n wireless standard, which requires additional work on analyzing the available redundancies, as well as a combination with the channel coding.

This paper is organized as follows. The next section describes briefly the specificities of the physical layer frame format of IEEE 802.11n wireless standard. Section 3 presents the iterative robust physical header system. Based on the strategy of cross-layer, the extraction of the relevant and redundant information available in the physical frame is discussed. Section 4 provides some simulation results. Section 5 concludes our work.

2. DESCRIPTION OF THE FRAME PHYSICAL LAYER IN 802.11N

This section introduces the useful IEEE 802.11n wireless standard information with some details. This is necessary since our robust physical header system takes into account the precise structure of the bitstream of the physical frame.

The IEEE 802.11n standard offers three different physical frame formats according to the station characteristics of the network [7]. Let describe briefly these formats illustrated by Fig.1.

The first one named ”Non HT” (HT for High Throughput) format corresponds to the basic frame of the standard 802.11 [7] which is transmitted to the stations that do not support the 802.11n standard. While the second format ”HT-mixed” is used to communicate with 802.11n stations belonging to a mixed network composed of different stations such as 802.11a, b, ... The last one named as ”HT-greenfield” format is designed for a network containing only 802.11n stations. Currently, this format is not selected in practice because this
situation is still very unlikely.

In this paper, we concentrate on the "HT-mixed" frame format. Some fields of this frame are identical to the "Non-HT" and "HT-greenfield" fields (see Fig.1). First analyze the different fields of "HT-mixed" frame format.

![Figure 1. Physical frame formats in the IEEE 802.11n standard](image)

The fields "L-STF", "L-LTF", "HT-STF" and "HT-LTF" are training fields. The "L-Sig" (for Legacy Signal) and "HT-Sig" (for High Throughput Signal) fields provide information on the coding mode of the physical data field frame. Fig.2 and Fig.3 provide more details.

In "L-Sig" partition, the "Rate" field is set to 6 Mbit/s. The "Length" field defines the current frame length measured in bytes using the transmission rate of the communication. The obtained value is then converted to the equivalent one assuming that the transmission rate is fixed to 6 Mbit/s.

The "MCS" field of the "HT-Sig" partition indicates the modulation type, the convolutional code with its code rate and the data rate which is used for transmitting the physical payload frame. The fields concerning the "Tail" of "L-Sig" and "HT-Sig" partitions are filled with zeroes to close trellises of the convolutional codes. The fourth bit of "L-Sig" and the second one of the "HT-Sig2" are considered as reserved bits for future use. The "P" field of the "L-Sig" is the parity bit which is computed on the 18 bits before "P". A CRC8 is used to protect the fields going from 0 to 23 bits in the "HT-Sig1" and to 0 to 9 bits in the "HT-Sig2". The "HT-Length" field contains the size of the physical frame which varies from 0 to 65535 bytes. Only one bit is used to indicate the bandwidth of the channel according to 20 MHz or 40 MHz.

![Figure 2. "L-Sig" fields of the physical header in 802.11n](image)

3. ROBUST PHYSICAL HEADER ESTIMATION

The bitstream related to the physical header contains very important information. That is why, the standard has proposed a multitude of protection and adaptation strategies. Namely, a CRC is added in order to check the consistency of the transmitted data. If the header bitstream is corrupted, no information can be transmitted to the upper layer i.e. the link layer. The receiver waits for a new retransmission. In contrast, in this section, rather than transmitting again the physical frame, we focus on the error correction of physical header frame using an iterative turbo decoding scheme. No additional redundancy is introduced, only the one already found in the headers is used. This strategy has been denoted as "permeable layer" mechanism in [6]. In this paper the term "header" concerns only the fields "L-Sig" and "HT-Sig". The preamble fields "L-STF", "L-LTF", "HT-STF" and "HT-LTF" manage the synchronization between stations. We assume that the communication between the stations has already been established, hence that these fields have been correctly received previously.

Denote \( \mathbf{d} = (d_1, ..., d_N) \) the header binary stream where \( N \) is the header length and \( d_k \) the \( k \)-th bit. The bits \( \{d_k\} \) are assumed to be uniformly distributed.

In the 802.11n standard, the binary physical header is first permuted by a pseudo-random interleaver to break the error burst at the receiver (as much as possible). The interleaved sequence is then modulated by a BPSK modulator and encoded by a convolutional encoder which has the following characteristics: a constraint length equal to 7, with a polynomial generator given by its octal representation [171,133] and a code rate equal to 1/2. The sequence is then transmitted over memoryless AWGN channel. The transmitted data stream is denoted as \( \mathbf{x}^L = (x_1, ..., x_k, ..., x_N) \) where \( x_k = (x_{k,0}, x_{k,1}, ..., x_{k,n-1}) \). The received data stream \( \mathbf{y} \) is denoted by \( \mathbf{y}^N = (y_1, ..., y_k, ..., y_N) \) where \( y_k = (y_{k,0}, y_{k,1}, ..., y_{k,n-1}) \).

Define two sets \( \mathbf{R} \) and \( \mathbf{D} \) of binary sequences \( d \) of length \( N \). The first set \( \mathbf{R} = \{d_1, d_2, ..., d_N\} \subseteq \{0,1\}^N \) contains all possible sequences, while the second one \( \mathbf{D} = \{d_1, d_2, ..., d_N\} \subseteq \{0,1\}^N \) contains only the sequences conforming the header format fields. This point will be discussed and explicitly considered in section 3.3.1.

3.1. Principle of the iterative system

The proposed receiver is composed as follows. The first block corresponds to the outer channel decoder and the second one concerns the inner header decoder (see Fig.4). Each decoding block is fed with soft inputs and can deliver soft outputs. This soft information is exchanged, in an iterative process, between the channel decoder and the header decoder. Classically, this soft information is the so-called extrinsic probability of each bit (i.e. the a posteriori probability of the bit, divided by its a priori probability). The iterative receiver estimates the transmitted stream \( \mathbf{d} \) so as to iteratively optimize the maximum a posteriori (MAP) criterion within each block, and considering the extrinsic information provided by the other block as an a priori probability. Thus, at a given iteration \( I \), the decoding algorithm proceeds in two steps as described below.
3.2. Outer BCJR channel decoder

In the first step, the BCJR channel decoder computes the a posteriori probability $P_N(d/y)$ for every $d \in D$ (all possible sequences) which is also equivalent to the product of the marginals a posteriori probabilities (APP) $\prod_{k=1}^{N} P_{BCJR}(d_k/y_k)$ since the channel transmission is assumed to be memoryless and the bits $\{d_k\}$ are assumed independent. At any iteration $I$, the output of the outer BCJR channel decoder is given by the extrinsic probability associated to the bit as follows:

$$E_{BCJR}^I(d_k) = K_{BCJR}^I \frac{P_{BCJR}^I(d_k/y_k)}{P_{BCJR}^I(d_k)}$$ (1)

where $K_{BCJR}$ is the normalization factor such as $E_{BCJR}^I(d_k = 0) + E_{BCJR}^I(d_k = 1) = 1$; $P_{BCJR}^I(d_k)$ is the a priori probability associated to the coded bit corresponding to the interleaved extrinsic information $E_{Robust-HD}^{I-1}(d_k)$ computed by the robust header block at the previous iteration (i.e. $I - 1$). The de-interleaved extrinsic information $E_{BCJR}^I(d_k)$ is then sent as an a priori input to the robust header block.

3.3. SISO inner robust header estimator

The second step, related to the SISO inner robust header block (see Fig. 4), consists in a projection of the distribution of the APP evaluated on $R$, on the set of the APP distribution compatible with the header distribution of the required APP, denoted by $P_{Robust-HD}^+(d)$, which is that minimizes the Kullback-Leibler distance given by:

$$P_{Robust-HD}^+(d) = \arg \min \text{dist}(P_R(d), P_{BCJR}^I(d/y))$$ (2)

Before solving this equation, the following section gives more details concerning the set $D$ of sequences which conform the physical header frame.

3.3.1. Extraction of redundant information in physical header frame

According to the new strategy based on the cross-layer procedure, the objective of this section is to extract and exploit the relevant information provided by the 802.11n wireless protocol. Based on this information, we show that it is possible to deduce other ones quite easily such as “Length” and “HT-Length” fields of the “HT-mixed” frame (see Fig.1, Fig.2 and Fig.3). Therefore, the set of valid header sequences (i.e. D) is completely provided by the standard.

First consider the CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) medium reservation procedure. Before sending information, the transmitter sends an RTS (Request To Send) control frame in order to ask the receiver if the communication is possible. If the receiver (e.g. access point) is not busy, another control frame called CTS (Clear To Send) is transmitted to (i) the transmitter in order to communicate; and (ii) all other stations of the same network in order to inform them of its unavailability for receiving. However, it is possible to select a minimum value size of the physical frame, using RTS-Threshold parameter, at which the medium reservation procedure is not triggered. The default value of the RTS-Threshold in the standard is equal to 2347 bytes. Two situations are then possible. Fig. 5 presents the structure of the RTS and CTS frames. What could be deduced about the length of the physical layer frame? Consider the first situation where the medium reservation procedure is not triggered, meaning that no RTS is sent. We deduce that the valid values of “Length” field is ranging between 0 and the fixed value given by RTS-Threshold.

In the second situation, the valid candidates of the field “Length” is deduced after some manipulations. Indeed, one can exploit the relevant information provided by the CTS frame particularly the “duration” field. This field concern the transmission time (measured in micro-seconds) of the physical frame which is used to compute the “HT-Length” according to the following equation provided in [7]:

$$\text{HT-Length} = (\text{Duration} - 3T_{SIFS} - 3T_{OVH} - 2T_{ACK} + \text{Bitrate}) \times \text{Bitrate}$$ (3)

where $T_{SIFS}$ is the duration in micro-seconds of a Short Inter-Frame Space, the $T_{OVH}$ is the duration for transmitting the overhead (the preamble and the physical header) and the $T_{ACK}$ is the one for transmitting an ACK.

These parameters are fixed by the standard and are then known by the transmitter and the receiver. The valid candidates “Bitrate” values are deduced from the “MCS” and “Bandwidth” values imposed by the standard. Only 77 “MCS” values are valid although 7 bits are reserved (see [8]). While 2 values are valid candidates for the field “Bandwidth”. The forbidden “Bitrate” values are then discarded.

For the options in “HT-sign” we must consider all configurations associated to 9 bits. For each configuration of headers, the parity bit “P” and the associated CRCs are computed. Therefore, all valid candidate fields are considered as the D valid headers.

3.3.2. Header extrinsic information

Since the D valid header sequences are identified, this section focuses on the computation of the header extrinsic information. To
solve equation (2), it has been shown in [9] that the required APP distribution is given by:

\[
P_{\text{Robust-Header}}^+(d) = \frac{P_k(d/\gamma)}{\sum_{a \in B} P_a(d/\gamma)} \quad \text{if } d \in D
\]

\[
P_{\text{Robust-Header}}^+(d) = 0 \quad \text{if } d \in R \setminus D
\]

(4)

At any iteration \(I\), the output of the inner (Robust-Header) decoder is given by the extrinsic probability associated to the header bit as follows:

\[
E^I_{\text{Robust-Header}}(d_k) = K_{\text{Robust-Header}} P^I_{\text{Robust-Header}}(d_k/y_k) P^I_{\text{Robust-Header}}(d_k) \quad \text{(5)}
\]

where \(K_{\text{Robust-Header}}\) is the normalization factor such as

\[
E^I_{\text{Robust-Header}}(d_k = 0) + E^I_{\text{Robust-Header}}(d_k = 1) = 1;
\]

\(P^I_{\text{Robust-Header}}(d_k)\) is the a priori probability associated to the header bit corresponding to the interleaved extrinsic information \(E^{I-1}_{\text{BCJR}}(d_k)\) computed by the BCJR decoder at the previous iteration (i.e. \(I - 1\)).

The marginal APP is deduced from equation (2) as follows:

\[
P^I_{\text{Robust-Header}}(d_k/y_k) = \sum_{d_k \in \{0,1\}} P^I_{\text{Robust-Header}}(d_k) \quad \text{(6)}
\]

The de-interleaved extrinsic information \(E^I_{\text{Robust-Header}}(d_k)\) is then sent to the BCJR decoder. The Robust-Header focuses on the distribution of the APP that maximizes \(P^I_{\text{Robust-Header}}(d_k)\) which is equivalent to preserve the optimal solutions with respect to the complete sequences and the individual bits as follows:

\[
\widehat{d}_k = \arg \max_{d_k \in \{0,1\}} \log \left( \sum_{d \in \{0,1\}} P^I_{\text{Robust-Header}}(d) \right) \quad \text{(7)}
\]

Iterations between soft robust header reception and the BCJR algorithm used for channel decoding are then performed classically, as in BICM iterative reception.

### 4. SIMULATION RESULTS

This section presents and discusses the implementation results of our robust header system. The results are carried out on the "HT-mixed" ("L-Sig" and "HT-Sig" parts) physical frame of IEEE 802.11n standard as described above.

In the proposed approach, we assume that the synchronisation is established, which means that the preamble and training fields (i.e. "L-STF", "L-LTF", "HT-STF" and "HT-LTF") of the physical frame are correct.

Two different situations are studied. The first one considers the "HT-mixed" frame in its full generality. In the second situation, the "HT-mixed" frame is designed according to the requirements of our industrial partner (COMSIS) in the project "DITEMOF" for the design of the 802.11n card ([10]). No options and only the mandatory "MCS" values (i.e. 0 to 15) of the 802.11n standard are implemented. One can deduce that only 2347\times 16 different physical header configurations must be considered when the frames are smaller than 2347 bytes and only 16 valid configurations when receiving a CTS frame control.

Moreover in each situation, described above, we have considered the case where the length of the physical frame is smaller and greater than the RTS-Threshold leading to two working conditions: (i) no transmitted CTS; or (ii) with transmitted CTS.

The binary physical header is first permuted by a pseudo-random interleaver which is then encoded by a convolutional encoder having the following characteristics: a constraint length equal to 7, with a polynomial generator presented by its octal representation [171,133] and a code rate equal to 1/2. The BPSK modulated sequence is then transmitted over memoryless AWGN channel as described in Fig. 4.

The performance of the developed system is measured in terms of Header Error Rate (HER) versus \(E_b/N_0\). The results of our system are compared with (i) the traditional BCJR channel decoding; and (ii) the SOVA channel decoding [11] since it is the classical implementation choice.

For the situation where an RTS is sent, an analysis of the computational complexity of our algorithm shows that, the algorithm has to apply the CRC8 on the number of valid "Bitrate" candidates \(2^9\) (related to the 9th options) valid sequences. Then, for each valid sequence, the metric computation is required. If no RTS is sent, 2347 \times the number of valid "Bitrate" candidates \(2^9\) metrics must be computed. This method is too complex to fit in our iterative decoding scheme. To reduce the computational load (especially when the RTS is not transmitted), we propose to modify slightly our original system. A hard decision is taken on the soft values at the output of the BCJR block on the 9 option bits of the physical header. This strategy allows to reduce the number of metrics by a factor of \(2^9\).

Moreover 9 extrinsic values out of 51 ones (corresponding to the following computation: header size — signal Tail size — Reserved bit size — Rate size) are not computed. At any iteration, a new hard decision is again taken on these soft values. Hence, if an RTS is sent the number of metrics is given by the number of valid "Bitrate" candidates, otherwise by 2347 \times the number of valid "Bitrate" candidates. This iterative physical header estimation is named IPHE. Simulation results are summarized by the graphs presented by Fig. 6.

For a given HER, equal for example to 10^{-2} and in the general physical header case, the proposed iterative decoding system improves the performance at the second iteration by 1.5 dB when an RTS is sent and by 1 dB without RTS compared to the BCJR channel decoding with hard decision (see Fig. 6).

Fig. 7 presents the simulation results in the "DITEMOF" project framework where the 9 options are fixed by the industrial partner. Therefore only one valid configuration is clearly identified. The problem is less complex than the general implemented situation (described above). At the same HER, equal for example to 10^{-2} and at the second iteration, the proposed system improves significantly the performance by a factor of 6 dB when the CSMA/CA strategy is used and by slightly more than 2 dB if CTS is not transmitted compared to the SOVA followed by hard decision.

In order to achieve a good tradeoff between computational cost and performance of the iterative approach, a CRC cross-layer method introduced in [6] is implemented in our system. In reference [6], the authors concentrate on the MAC layer of 802.11 standard where they proposed a backward MAP decoding algorithm (Exact Sum computation Backward method). A trellis is built using all the possible values of the CRC as states. A marginalization is then applied on the payload of the MAC frame. Adapted to our situation, i.e. only \(9 \times 2^{|CRC|\text{size}}\) operations added to the number of valid "Bitrate" candidates are required to construct the backward trellis. This method (named IESB for iterative exact sum computation backward method) is implemented in our iterative robust physical header system. Simulation results of this method are given by Fig. 8 using the CSMA/AC strategy. A comparison with the results of Fig. 6 show that at the first iteration the performance of the two algorithms (IPHE and IESB) are almost equivalent. However at the second iteration, for the same HER, saving of about 0.5 dB is achieved by IESB compared to the IPHE approach against a high computational cost.
5. CONCLUSION

This paper proposed a robust physical header system based on an iterative joint source channel decoder using a MAP estimator. The soft header estimator exploits the redundant information available in the physical header frame of the IEEE 802.11n standard. Among other, the proposed technique avoids packet retransmission. The performance analysis showed that the developed receiver allows to reduce significantly the number of errors occurring during the transmission compared to a classical hard decision technique. Current investigations are being conducted in order to design a complete system based on a cross-layer architecture.

![Graph 6: HER performance of the IPHE receiver using the general physical header in 802.11n](image)

![Graph 7: HER performance of the iterative receiver using the “DITE-MO1” physical header in 802.11n](image)

Fig. 6. HER performance of the IPHE receiver using the general physical header in 802.11n

Fig. 7. HER performance of the iterative receiver using the “DITE-MO1” physical header in 802.11n

6. REFERENCES