2014 IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)

RELIABLE PACKET TYPE ESTIMATION VIA JOINT PROTOCOL-CHANNEL DECODING

Kai Wan[+], *Jinghui Liu*[+], *Michel Kieffer*[+,⋆,o], *and Pierre Duhamel*[+]

[+] L2S, CNRS - SUPELEC - Univ Paris-Sud, F-91192 Gif-sur-Yvette
[⋆] Institut Universitaire de France, F-75005 Paris
[o] Part-time on leave at LTCI, CNRS-Télécom ParisTech, F-75013 Paris

ABSTRACT

This paper presents a joint protocol-channel decoding techniques to improve the estimation quality of the type of a packet corrupted by transmission errors. It exploits the redundancy present in the protocol stack to determine the packet type which is the most probable *a posteriori*. An optimal and a suboptimal estimation algorithm are presented. The latter is illustrated on the type estimation of packets compressed with the Robust Header Compression algorithm in unidirectional mode. Compared to a classical packet type estimation technique, improvements of more than 3.9 dB are observed in terms of channel SNR at header error rates of 10^{-2}.[1]

1. INTRODUCTION

In the OSI model, networking tasks are partitioned between distinct protocol layers [9]. In each layer, the transmitter adds some specific information to the packets to be transmitted (receiver address, type of signal…). Independence between layers facilitates network design, since a layer has not to be aware of the information introduced by other layers, but this is sometimes at the price of redundancy in the various headers, when correlated information is necessary at various layers.

Determining the *type* of a packet is the first task performed by the protocol layer processor when it receives some packet from the physical layer, or from a lower layer in the protocol stack. The type of a packet determines its layout and part of its content. Usually, the type is encoded with few bits at the beginning of the header. It is thus very sensitive to transmission impairments and has to be estimated reliably.

This paper proposes a maximum *a posteriori* (MAP) estimator of the type of a packet in a given protocol layer. Instead of considering only the type identification field of the considered packet, the proposed technique determines the packet type which is the most consistent with the received data and some *a priori* information: the whole structure of the packet and the redundancy in the protocol stack is put at work. Soft information, *i.e.*, bit reliability information, is assumed available at the considered layer. It may either come from soft-output channel decoders at physical layer [2, 6] possibly combined with protocol layer permeability mechanisms [8, 12, 15], which allow the transmission of soft information from the physical layer through the upper layers of the protocol stack.

Redundancy in the protocol stack has been previously recognized and used in header compression techniques, such as the Robust Header Compression (RoHC) protocol [3, 16]. Alternatively, Joint Protocol and Channel Decoding (JPCD) techniques make an efficient use of the redundancy present in *uncompressed* protocol layers (including that introduced at Physical layer, *e.g.*, by channel coding) to obtain optimal performance at a global level. With this family of techniques, physical-layer synchronization can be improved [13], channel decoding is helped by *pilot symbols* from the protocol [7], aggregated packets are more efficiently delineated [1, 4, 5, 11], or packet headers are more reliably recovered [10, 14].

Usually, CRCs or checksums efficiently check the integrity of the header or data they protect. JPCD techniques are thus only used on those packets where the CRC or checksum fails. Reliable header recovery techniques [10, 14] may then be put at work to estimate the headers of noisy packets. However, these techniques, to be employed by a given receiver, rely on the hypotheses (*i*) that the structure of the header to estimate is perfectly known and (*ii*) that the packet has actually to be processed by the considered layer at this receiver. A first example is with RoHC: the type of a packet is stored in few bits at the beginning of the packet. When these bits are corrupted by noise, identifying the type becomes very difficult. A second example is with 802.11, all packets at the PHY layer are visible to all users, packet filtering is mainly done at the MAC layer, based on the receiver MAC address.

Therefore, the receiver should be aware of precise information about the location and the value of the required redundant information in order to make an efficient use of such techniques, and it turns out that the information about the type of a packet contains most of it. This paper provides some insight to evaluate how these crucial assumptions can be met, by using efficient type estimation algorithms.

The organization of packets, depending on their type is introduced in Section 2. The packet type estimator and its practical implementation are described in Section 3. This strategy is very general, and can even be applied to the robust reception of ROHC packets, which is chosen as an application. Simulation results on the identification of RoHC packets are provided in Section 4 before deriving some conclusions and future research directions in Section 5.

2. GENERIC ORGANIZATION OF PACKETS

This section identifies the various fields of bits forming the packets processed at a generic protocol layer. Usually, the organization, length, position, and/or potential content of each field may depend on the *type* of the packet in the protocol layer. This classification is inspired from [10, 16] and will be useful to build a MAP estimator of the type of a packet.

Considering a packet $\mathbf{p}_t$ of type t belonging to a finite set of types $\mathcal{T}$, one may identify the following fields. The *constant field* $\mathbf{k}$ contains all bits which value is perfectly known and independent of the type of the packet. The *type-determination* field $\mathbf{d}_t$ consists of all bits that are usually employed in a noise-free context to identify the type of the packet. The field of *unknown* bits $\mathbf{u}_t$ contains all bits that

[1]This work has been partly supported by the Newcom# NoE

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

represent useful information for processing at the considered layer. The *other* field $\mathbf{o}_t$ contains bits that are processed at upper layers of the protocol stack. The *control* field $\mathbf{c}_t$ is assumed to be evaluated from the previous fields $\mathbf{k}$, $\mathbf{d}_t$, $\mathbf{u}_t$, and $\mathbf{o}_t$, and possibly from context information kept in the memory of the processor of the considered layer. This information, which may consist of the content of previously processed packets is represented by $\mathcal{R}$. Finally, the part of the payload that is not protected by the control field is denoted $\mathbf{x}_t$. Thus, up to a permutation of the indexes of its bits, a packet of type t is organized as follows

$$\mathbf{p}_t = (\mathbf{k}, \mathbf{d}_t, \mathbf{u}_t, \mathbf{o}_t, \mathbf{c}_t, \mathbf{x}_t), \tag{1}$$

see Figure 1. The fields $\mathbf{d}_t$ and $\mathbf{u}_t$ are assumed to take only a finite number of values belonging respectively to the finite sets Ω_{d} and $\Omega_{\mathrm{u},t}$. The fields $\mathbf{o}_t$ and $\mathbf{x}_t$ are such that $\mathbf{o}_t \in \{0,1\}^{\ell(\mathbf{o}_t)}$ and $\mathbf{x}_t \in \{0,1\}^{\ell(\mathbf{x}_t)}$, where $\ell(\mathbf{x})$ denotes the length of the field $\mathbf{x}$. Each bit of $\mathbf{o}_t$ and $\mathbf{x}_t$ is independently and identically distributed according to a symmetric Bernoulli random variable. One assumes that there exists a deterministic function $\mathbf{f}_t$ such that

$$\mathbf{c}_t = \mathbf{f}_t(\mathbf{k}, \mathbf{d}_t, \mathbf{u}_t, \mathbf{o}_t, \mathcal{R}). \tag{2}$$

The set of all values which may be taken by a packet of type t is denoted as Ω_t.

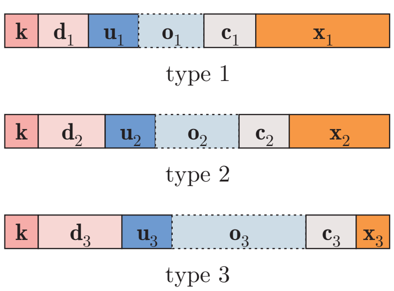

Fig. 1. Organization of the fields depending on the packet type

3. MAP PACKET TYPE ESTIMATOR

The sequence of packet types $\{T_n\}$ is assumed to be accurately modeled by a time- invariant first-order Markov process described by the transition probability matrix with entries

$$p_{i,j} = \Pr(T_n = j | T_{n-1} = i), \ (i,j) \in \mathcal{T}^2. \tag{3}$$

The main idea of the proposed type estimation approach, when the received packet is corrupted by noise, is to verify the *global consistency* of its fields with the inferred type. This is obtained by determining the type which is the most probable *a posteriori*. For that purpose, it is assumed that a vector of bit soft information $\mathbf{y}$ is available at the input of the considered layer. This soft information may consist of bit *a posteriori* probabilities or likelihood ratios provided by the channel decoders at physical layer or by lower protocol layers implementing layer permeability mechanisms [8, 12, 15].

3.1. Optimal estimator

Using $\mathbf{y}$ and the knowledge of $\mathcal{R}$, the MAP estimator for the type t_n of the n-th packet is

$$\begin{aligned}\widehat{t}_n &= \arg\max_{t\in\mathcal{T}} \Pr(T_n = t \,|\, \mathbf{y}, \mathcal{R}) \\ &= \arg\max_{t\in\mathcal{T}} \sum_{\mathbf{p}\in\Omega_t} \Pr(\mathbf{p} \,|\, \mathbf{y}, \mathcal{R}) \\ &= \arg\max_{t\in\mathcal{T}} \sum_{\mathbf{p}\in\Omega_t} p(\mathbf{y} \,|\, \mathbf{p}, \mathcal{R}) \Pr(\mathbf{p} \,|\, \mathcal{R}). \end{aligned} \tag{4}$$

Assuming that the channel is memoryless, that the fields $\mathbf{u}$, $\mathbf{o}$, and $\mathbf{x}$ are independent two-by-two, and that only the locations of the fields $\mathbf{o}_t$ and $\mathbf{x}_t$ is influenced by the packet type, (4) becomes

$$\widehat{t} = \arg\max_{t\in\mathcal{T}} \Phi_t(\mathbf{y}),$$

with

$$\begin{aligned}\Phi_t(\mathbf{y}) = {} & p(\mathbf{y}_{\mathrm{k}} \,|\, \mathbf{k}) \Pr(\mathbf{k})\, p(\mathbf{y}_{\mathrm{d},t}|\mathbf{d}_t) \Pr(\mathbf{d}_t|\mathcal{R}) \\ & \sum_{\mathbf{x}_t} p(\mathbf{y}_{\mathrm{x},t}|\mathbf{x}_t) \Pr(\mathbf{x}_t|\mathbf{d}_t) \sum_{\mathbf{u}_t\in\Omega_{\mathrm{u},t}} p(\mathbf{y}_{\mathrm{u},t}|\mathbf{u}_t) \Pr(\mathbf{u}_t|\mathbf{d}_t, \mathcal{R}) \\ & \sum_{\mathbf{o}_t} p(\mathbf{y}_{\mathrm{o},t}|\mathbf{o}_t) \Pr(\mathbf{o}_t|\mathbf{d}_t)\, p(\mathbf{y}_{\mathrm{c},t}|\mathbf{f}_t(\mathbf{k}, \mathbf{d}_t, \mathbf{u}_t, \mathbf{o}_t, \mathcal{R})). \end{aligned} \tag{5}$$

The subvectors $\mathbf{y}_{\mathrm{k}}$, $\mathbf{y}_{\mathrm{d},t}$, $\mathbf{y}_{\mathrm{u},t}$, $\mathbf{y}_{\mathrm{o},t}$, $\mathbf{y}_{\mathrm{c},t}$, and $\mathbf{y}_{\mathrm{x},t}$ form a partition of the vector $\mathbf{y}$. The type subscript t indicates that the bits considered in each subvector may depend on the considered type, see Figure 1.

The expression (5) may be simplified by removing all parts that are common to all types. For each type t, one may partition $\mathbf{x}_t$ into a *common* part $\mathbf{x}_c$, shared in all types, *i.e.*, bits which belong to the payload, whatever the type, and a *specific* part $\mathbf{x}'_t$. Then, $\Phi_t(\mathbf{y})$ becomes

$$\begin{aligned}\Phi_t(\mathbf{y}) = {} & p(\mathbf{y}_{\mathrm{d},t}|\mathbf{d}_t) \Pr(\mathbf{d}_t|\mathcal{R})\, 2^{-\ell(\mathbf{x}'_t)-\ell(\mathbf{o}_t)} \sum_{\mathbf{x}'_t} p(\mathbf{y}_{\mathrm{x},t}|\mathbf{x}'_t) \\ & \sum_{\mathbf{u}_t\in\Omega_{\mathrm{u},t}} p(\mathbf{y}_{\mathrm{u},t}|\mathbf{u}_t) \Pr(\mathbf{u}_t|\mathbf{d}_t, \mathcal{R}) \\ & \sum_{\mathbf{o}_t} p(\mathbf{y}_{\mathrm{o},t}|\mathbf{o}_t)\, p(\mathbf{y}_{\mathrm{c},t}|\mathbf{f}_t(\mathbf{k}, \mathbf{d}_t, \mathbf{u}_t, \mathbf{o}_t, \mathcal{R})). \end{aligned} \tag{6}$$

The terms of the first line in $\Phi_t(\mathbf{y})$ are easily evaluated from the channel model, using the characteristics of the soft information, or *a priori* probabilities. With a memoryless channel, the complexity for evaluating $\sum_{\mathbf{x}'_t} p(\mathbf{y}_{\mathrm{x},t}|\mathbf{x}'_t)$ is linear in $\ell(\mathbf{x}'_t)$. The last term

$$\begin{aligned}\Psi(\mathbf{y}, \mathbf{d}_t) = {} & \sum_{\mathbf{u}_t\in\Omega_{\mathrm{u},t}} p(\mathbf{y}_{\mathrm{u},t}|\mathbf{u}_t) \Pr(\mathbf{u}_t|\mathbf{d}_t, \mathcal{R}) \\ & \sum_{\mathbf{o}_t} p(\mathbf{y}_{\mathrm{o},t}|\mathbf{o}_t)\, p(\mathbf{y}_{\mathrm{c},t}|\mathbf{f}_t(\mathbf{k}, \mathbf{d}_t, \mathbf{u}_t, \mathbf{o}_t, \mathcal{R})) \end{aligned} \tag{7}$$

may be evaluated with a worst-case complexity of $O\left(|\Omega_{\mathrm{u},t}|\, 2^{\ell(\mathbf{o}_t)}\right)$, where $|\Omega_{\mathrm{u},t}|$ is the cardinal number of $\Omega_{\mathrm{u},t}$. This complexity is unmanageable for large values of $\mathbf{o}_t$. To address this issue, [10] has proposed an optimal trellis-based evaluation algorithm for $\Psi(\mathbf{y}, \mathbf{d}_t)$ with a complexity $O\left(|\Omega_{\mathrm{u},t}|\, \ell(\mathbf{o}_t)\, 2^{\ell(\mathbf{y}_{\mathrm{c},t})}\right)$, exponential in the length of the check field. By partitionning the check field into m parts assumed independent, [10] gets an approximate evaluation of $\Psi(\mathbf{y}, \mathbf{d}_t)$ with a complexity $O\left(m\, |\Omega_{\mathrm{u},t}|\, \ell(\mathbf{o}_t)\, 2^{\ell(\mathbf{y}_{\mathrm{c},t})/m}\right)$.

3.2. Suboptimal estimator

In some cases, due to delay or complexity constraints, a decision regarding the type of a packet has to be taken considering a vector $\mathbf{y}$ providing soft information on only a part of the packet header. For each type, only a part of the fields identified in Section 2 may be considered. For example, when no measurements related to the check field are present in $\mathbf{y}$, and (6) becomes

$$\Phi_t(\mathbf{y}) = p(\mathbf{y}_{\mathrm{d},t}|\mathbf{d}_t)\Pr(\mathbf{d}_t|\mathcal{R})\,2^{-\ell(\mathbf{x}'_t)-\ell(\mathbf{o}_t)}\sum_{\mathbf{x}'_t} p(\mathbf{y}_{\mathrm{x},t}|\mathbf{x}'_t)$$
$$\sum_{\mathbf{u}_t\in\Omega_{\mathrm{u},t}} p(\mathbf{y}_{\mathrm{u},t}|\mathbf{u}_t)\Pr(\mathbf{u}_t|\mathbf{d}_t,\mathcal{R})\sum_{\mathbf{o}_t} p(\mathbf{y}_{\mathrm{o},t}|\mathbf{o}_t)\,. \qquad (8)$$

Similarly, when measurements on only a part of the check field are available, the partial check field may be treated as an other field, leading to the same expression as (8).

4. SIMULATIONS

In what follows, we will present simulation results for the suboptimal packet type classification algorithm of Section 3.2 applied to RoHC packets. This section focuses on the Unidirectional mode (U-mode) employed, *e.g.*, in broadcast applications, where feedback is difficult to get from the receivers [3]. The main packet types employed in the U-mode are recalled first before describing the simulation conditions and providing results.

4.1. RoHC in U-mode

For the transmission of data using header compression, RoHC introduces dedicated *contexts* for each packet stream at the encoder and decoder, which contain relevant information from previous headers in the packet stream and possible reference values for compression and decompression. Contexts have to be initialized and regularly refreshed to allow recovery from errors.

In each mode, including U-mode, for each context, there are three possible states of the RoHC compressor and of the RoHC decompressor. For the compressor, one may distinguish the *Initialization and Refresh* (IR), *First Order* (FO), and *Second Order* (SO) states. In the IR state, the headers are sent uncompressed, which helps setting up the context at the decompressor. In the FO state, the context is only partly updated. In the SO state, headers are the most efficiently compressed. In the U-mode, the transitions between states depend on the variability of header contents and on the confidence the compressor has in the context of the decompressor. In other modes, transitions are triggered from feedback information. For the decompressor, one may identify the *No context* (NC), *Static context* (SC), and *Full context* (FC) states. Initially, the decompressor works in the NC state and switches to the SC and then to the FC state upon reception of initialization packets with static and dynamic information. The decompressor transits to the lower states upon repeated failures. To compress the value of slowly-varying quantities in original headers, several lossless compression techniques have been defined in [3]. The main idea is to only store the compression residuals.

In what follows, we assume that they are more than 15 contexts, but that the context identifier (CID) can be represented in compressed form over a single byte. The three first bytes of the six packet types that may be generated by the compressor are summarized in Figure 2. To limit the complexity of the type estimation algorithm, the estimation procedure will be based on these three bytes.

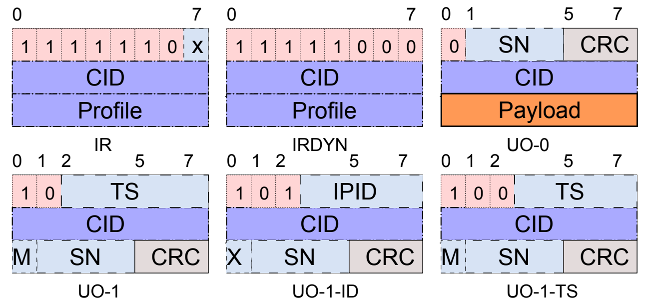

Fig. 2. Fields of the 3 first bytes of the 6 different RoHC packet types considered in the U-mode

IR packets are generated in the IR state, IRDYN packets in the FO state, and other packets in the SO state. For all these packet types, the value of the known bits is indicated. The CID bits is assumed to belong to the $\mathbf{u}$ field, since it takes a limited number of values. Similarly, Profile, as defined in [3], is between 0 and 3, and thus belong to the $\mathbf{u}$ field. In the IR packet type, X is a profile-specific information interpreted according to the profile indicated in the Profile field. It is considered as an $\mathbf{o}$ field. UO-0 packets are used to update the sequence number (SN) of RTP headers. Its third byte is assumed to belong to the payload and to take any possible value. UO-1 and UO-1-TS packets are employed to update the RTP Timestamp (TS), the RTP Marker bit (M) and the RTP SN. UO-1-ID packets are used to update the IPv4 Identification (IP-ID) field and the Extension bit (X). The SN, TS, M, and X fields all belong to the $\mathbf{o}$ field. The three-bit CRC present in the UO packet types is evaluated on the *original* headers, *before compression* with the polynomial $C(x) = 1 + x + x^3$.

The generation of packets of types UO-1, UO-1-ID, and UO-1-TS depends on the context variable RND, set up in IR and IRDYN packets. When an IPv4 header, for which the corresponding RND flag has not been set to 1, is present in the context, the packet type UO-1 cannot be generated. When no IPv4 header is present in the context, or the RND flags for all IPv4 headers in the context are set to 1, the packet types UO-1-ID, and UO-1-TS cannot be generated.

4.2. Simulation conditions

We consider RTP/UDP/IPv4 packets of 20 ms compressed voice sampled at 8 kHz. RoHC in U-mode is employed for header compression. We consider a scenario where the compressor returns to the IR state every 100 packets. When in the SO state, it returns to the FO state with a probability 0.07. In the IR and FO states, three consecutive IR or IRDYN packets are transmitted to increase robustness to transmission errors. A typical evolution of the state of the compressor is represented in Figure 3. The value of RND is chosen at random every series of three IR or IRDYN packets (to artificially increase the packet type diversity).

From the evolution of the states of the compressor, one may deduce the values of two state transition matrices introduced in (3),

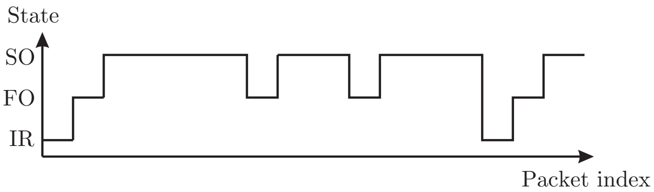

Fig. 3. Evolution of the state of the compressor in U-mode

depending on the value of RND as

$$\Pi_0 = \begin{array}{cc} & \begin{array}{ccccc} \text{IR} & \text{IRDYN} & \text{UO-0} & \text{UO-1-ID} & \text{UO-1-TS} \end{array} \\ \begin{array}{c} \text{IR} \\ \text{IRDYN} \\ \text{UO-0} \\ \text{UO-1-ID} \\ \text{UO-1-TS} \end{array} & \begin{pmatrix} 0.67 & 0.33 & 0 & 0 & 0 \\ 0.01 & 0.66 & 0.165 & 0.0825 & 0.0825 \\ 0.03 & 0.07 & 0.3 & 0.15 & 0.15 \\ 0.03 & 0.07 & 0.45 & 0.225 & 0.225 \\ 0.03 & 0.07 & 0.45 & 0.225 & 0.225 \end{pmatrix} \end{array}$$

when RND $= 0$ and

$$\Pi_1 = \begin{array}{cc} & \begin{array}{cccc} \text{IR} & \text{IRDYN} & \text{UO-0} & \text{UO-1} \end{array} \\ \begin{array}{c} \text{IR} \\ \text{IRDYN} \\ \text{UO-0} \\ \text{UO-1} \end{array} & \begin{pmatrix} 0.67 & 0.33 & 0 & 0 \\ 0.01 & 0.66 & 0.165 & 0.165 \\ 0.03 & 0.07 & 0.45 & 0.45 \\ 0.03 & 0.07 & 0.45 & 0.45 \end{pmatrix} \end{array}.$$

when RND $= 1$.

Remark 1 *For the evaluation of* $\Psi(\mathbf{y}, \mathbf{d}_t)$ *in* (7)*, the method presented in [10] has to be slightly adapted to take into account the fact that the CRC is evaluated on the original uncompressed header. Assume that at time* n*, the part of the uncompressed packet* $\mathbf{p}_n$ *on which the CRC is evaluated in* $\mathbf{h}_n$*. The resulting CRC is thus* $\mathbf{c}_n = \mathbf{f}(\mathbf{h}_n)$*. At time* $n+1$*, one gets* $\mathbf{c}_{n+1} = \mathbf{f}(\mathbf{h}_{n+1})$*. The difference* $\Delta\mathbf{h}_{n+1} = \mathbf{h}_{n+1} - \mathbf{h}_n$ *may be directly deduced from the type and the content of the* $n+1$*-th RoHC compressed packet and using the linearity of the CRC, one has*

$$\mathbf{c}_{n+1} = \mathbf{f}(\Delta\mathbf{h}_{n+1}) + \mathbf{c}_n. \tag{9}$$

When evaluating the sums over the $\mathbf{u}_t$*s and over the* $\mathbf{o}_t$*s in* (7)*, instead of working on* $\mathbf{h}_{n+1}$*, one may consider* $\Delta\mathbf{h}_{n+1}$*, which contains far less non-zero bits and allows an evaluation of* (7) *with a complexity* $O\left(|\Omega_{\mathrm{u},t}|\, \ell(\mathbf{o}_t)\, 2^{\ell(\mathbf{y}_{\mathrm{c},t})}\right)$*, which is very manageable here, since* $\ell(\mathbf{y}_{\mathrm{c},t}) = 3$*.*

Packets with RoHC compressed headers are directly transmitted over an AWGN channel. Since further encapsulations of the RoHC packets is performed at the MAC and PHY layers, such channel model may be reasonable only assuming that other joint protocol-channel decoding techniques and layer permeability mecanisms have been put at work at lower layers of the protocol stack.

4.3. Results

For each value of the channel SNR, enough packets have been generated to get 100 header type estimation errors. The suboptimal estimator presented in Section 3.2 is compared to a standard estimator performing first a hard decision on the soft information provided by the channel output and deciding afterwards the packet type. Figure 4 represents the Header Type Error Rate (HTER) as a function of the channel SNR. For an HTER of 10^{-2}, the proposed method provides a gain in channel SNR of about 3.9 dB. At a channel SNR of 10 dB, the HTER is divided by 20 whith the proposed appraoch compared to a standard approach.

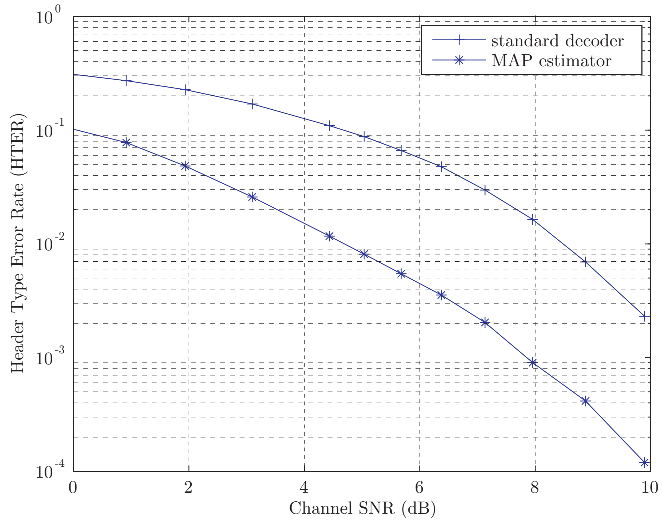

Fig. 4. Evolution of the header type error rate as a function of the channel SNR

5. CONCLUSIONS

This paper presents a JPCD techniques for reliable packet type estimation of packets which content has been corrupted by transmission impairments. This type estimation problem is encountered in several situations, such as MAC packet classification, reliable processing of RoHC packets. It has been illustrated on the estimation of the type of packets in the unidirectional mode of RoHC, where no feedback is allowed. This is a particularly interesting context for JPCD techniques, since the receiver has really to take the best out of noisy packets, if it does not want to drop too many out of them. Further experiments have been considered in the estimation of the type of MAC packets (RTS, CTS, ACK, Data packets) in the context of 802.11.

6. REFERENCES

[1] U. Ali, M. Kieffer, and P. Duhamel. Frame synchronization based on robust header recovery and bayesian testing. In *Proc. IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2010.

[2] 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.

[3] 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. Technical Report RFC 3095, 2001.

[4] M. Chiani and M. G. Martini. On sequential frame synchronization in AWGN channels. *IEEE Trans. Comm.*, 54:339 – 348, 2006.

[5] M Chiani and M.G Martini. Practical frame synchronization for data with unknown distribution on AWGN channels. *IEEE Communication Letter*, 9(5):456 – 458, May 2005.

[6] J. Hagenauer and P. Hoeher. A Viterbi algorithm with soft-decision outputs and its applications. In *Proc. Globecom*, pages 1680–1686, Dallas, TX, 1989.

[7] R. Hu, M. Kieffer, and P. Duhamel. Protocol-assisted channel decoding. *IEEE Signal Processing Letters*, 19(8):483–486, 2012.

[8] 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.

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

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

[11] M. G. Martini and M. Chiani. Optimum metric for frame synchronization with Gaussian noise and unequally distributed data symbols. In *Proc. IEEE SPAWC*, Perugia, Italy, 21-24 June 2009.

[12] M.G. Martini, M. Mazzotti, C. Lamy-Bergot, J. Huusko, and P. Amon. Content adaptive network aware joint optimisation of wireless video transmission. *IEEE Communications Magazine*, 45(1):84 – 90, 2007.

[13] C. L. Nguyen, A. Mokraoui, and N. Duhamel, P. Linh-Trung. Time synchronization algorithm in IEEE 802.11a communication system. In *Proc. European Signal Processing Conference*, 2012. submitted.

[14] 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.

[15] 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.

[16] M. West and S. McCann. TCP/IP field behavior. Technical Report RFC 4413 (Informational), March 2006.