HAL
open science

Frame Synchronization based on Robust Header Recovery and Bayesian Testing

Usman Ali, Michel Kieffer, Pierre Duhamel

To cite this version:

Usman Ali, Michel Kieffer, Pierre Duhamel. Frame Synchronization based on Robust Header Recovery and Bayesian Testing. IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, Sep 2010, Istambul, Turkey. pp.6. hal-00549251

HAL Id: hal-00549251

https://hal.science/hal-00549251v1

Submitted on 21 Dec 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Frame Synchronization based on Robust Header Recovery and Bayesian Testing

Usman ALI[1], Michel KIEFFER[1,2], and Pierre DUHAMEL[1]
[1] L2S, CNRS – SUPELEC – Univ Paris-Sud - 91192 Gif-sur-Yvette, France
[2] on sabbatical leave at LTCI, CNRS – Télécom ParisTech - 75013 Paris, France

***Abstract*—Video transmission on wireless links usually requires some frame aggregation, so that the overhead due to the headers is not a too large percentage of the bit-stream. In such a situation, header error detection is essential, because any error in the header may cause a loss of several consecutive frames. The contributions of this paper are (i) an improved reception of the length field of the header and (ii) an improved Frame Synchronization (FS) algorithm for aggregated frames. The FS algorithm has the following characteristics: (i) it makes use of the implicit redundancies which are present in the headers, thus resulting in an efficient synchronization of the variable-length frames; (ii) Bayesian hypothesis testing is used to estimate the correct synchronization; (iii) the proposed algorithm performs estimation on-the-fly and does not require reading the whole bit-stream. A comparative performance evaluation with respect to previously proposed algorithms is provided in the context of WiMAX MAC.**

I. Introduction

In several communication systems, at a given layer of the protocol stack, small packets or *frames* are aggregated into larger frames or *bursts* in order to reduce the overhead due to headers, see *e.g.*, 802.11/802.16 standards [2], [4] for Wireless Local Area Networks (WLANs). These aggregated frames are then forwarded by the transmitter to lower protocol layers. As an example, frame aggregation techniques at intermediate protocol layers have been studied recently [18] in the context of 802.11 standard. This aggregation has two effects: on one side, it increases the useful throughput in quite a large amount, but, on another side, any error upon reception of the burst header results in the loss of the complete burst, corresponding to many individual frames. When the burst header has been correctly received, efficient Frame Synchronization (FS) becomes very important, since if some frames are not correctly delineated, a large amount of bits has to be retransmitted. When the delay is constrained (situations where one cannot afford retransmission of large aggregated frames) or when retransmitting bursts in error is not possible (broadcasting), header recovery and FS remain the only options.

Initially, FS techniques, as proposed in [5], [15], [17] were using regularly spaced fixed patterns or *Sync Words* (SW) to delimit fixed-length frames. This is the situation, *e.g.*, at the PHY layer of DVB-H [11] for MPEG2-TS frames. In [5], synchronization is performed by maximizing the correlation between the SW and the received data. This has been improved in [15], where the optimal statistic for FS has been proposed for the AWGN case, taking into account the presence of data around the SW. This was further extended in [12] for more sophisticated transmission schemes. Similarly, in the presence of SW, in [8], [14], several hypothesis testing techniques have been presented to perform FS, for variable-length frames. In the absence of SW, the Header Error Control (HEC) field of the header has been employed in [19] to perform FS of variable length IP frames with a three-state automaton adapted from [1], performing bit-by-bit search for FS. A length field, assumed present in the frame header, facilitates the FS when the noise is moderate. In several situations, *e.g.*, when the HEC field is erroneous itself or when its length is too short compared to the header size, false alarms can hamper an effective FS.

The above-mentioned FS techniques work *on-the-fly*, *i.e.*, only a small portion of the aggregated frame is processed at each time to perform FS. This has the advantage of allowing early decisions, therefore introducing almost no latency.

It can be observed that none of these methods make full use of intra and inter-layer redundancies introduced by the structure of the header, redundancies which can be very useful. This usefulness was evidenced in the trellis-based approach [3] for the segmentation of MAC frames aggregated in large WiMAX PHY bursts. However, in this *hold-and-sync(hronize)* technique, the receiver must wait for whole or at least for a large part of an aggregated frame, which is sometimes undesirable in certain situations (such as framing of IP frames over ATM network), thus an unavoidable trade off between performance and delay is required.

In some sense, this paper proposes an *on-the-fly* technique which tries to take the best of these previously proposed methods: It combines robust header recovery techniques inspired from [13] with Bayesian hypothesis testing inspired from [7], [9], [8], [14] to localize frame boundaries via a sample-by-sample search. We use the three-state automaton, derived from that of [19], but instead of hard CRC correction, a soft header recovery technique [13] for correcting the damaged headers (exploiting all known intra and interlayer redundancies) is exploited to estimate the length field of the header. Moreover, the Bayesian hypothesis testing used to search for the correct location of FS is improved based on soft channel information (the output of the channel or channel decoder) combined with *a priori* information due to the redundancy present at the header of frame (known fields, presence of CRC or checksums, *etc*).

To be fully efficient, FS techniques require soft information from the channel decoders at PHY layer to reach upper

layers, as proposed by [16], [20]. Combined with joint source-protocol-channel (JSPC) decoders (see [10] and the references therein), robust FS techniques allow to significantly reduce the amount of frames that need to be retransmitted. Moreover, the combination of transparent layer mechanisms [16] with a robust header estimation technique [13] allows erroneous frames, which would be otherwise dropped, to reach the APL layer where they may be correctly understood by JSPC decoders.

The paper is organized as follows: Structure of frames and their aggregation is presented in Section II. The proposed three-state automaton is presented in Section III. Finally, we present the results obtained for the WiMAX MAC layer over AWGN and Rayleigh channels in Section IV before drawing some conclusions.

II. Frame Aggregation

Aggregation algorithms entail the process of assembling frames with similar destination called aggregation target. Upon arrival at the target, the original small frames are recovered from the aggregate frame. This recovery process is known as fragmentation or de-aggregation, which can be performed in a robust manner using FS techniques.

A. Structure of a frame

Consider the n-th variable-length frame at a given protocol layer. This frame is assumed to contain $\lambda_n = \ell_h + \ell_{p,n}$ bits, where the leading ℓ_h bits represent the frame header, of fixed length, and the remaining $\ell_{p,n}$ bits are the variable-length payload. In the header, ℓ_c bits are some Header Error Control (HEC) bits, CRC or checksum.

In this paper, we assumed that the header $\mathbf{h}_n$ of the n-th frame may be partitioned into four fields. The *constant* field $\mathbf{k}$, contains all bits which do not change from frame to frame. It includes the sync word indicating the beginning of the frame, but also other bits which are constant [13]. The header is assumed to contain a *length* field $\mathbf{u}_n$, indicating the size of the frame in bits λ_n, including the header. The *other* field $\mathbf{o}_n$, gathers all bits of the header which are not fully determined and are not used to perform FS. Finally, the HEC field $\mathbf{c}_n$ is assumed to cover only the header (*i.e.*, $\ell_h - \ell_c$ bits) without covering the payload. Thus:

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

where $\mathbf{f}$ is some (CRC or checksum) encoding function. The payload of the n-th frame is denoted by $\mathbf{p}_n$, assumed to be generated by a binary symmetric source.

The length λ_n of the n-th frame is assumed to be a realization of a stationary memoryless process Λ characterized by

$$\pi_\lambda = \Pr\left(\Lambda = \lambda\right) \neq 0 \text{ for } \ell_{\min} \leqslant \lambda \leqslant \ell_{\max}, \tag{2}$$

where $\ell_{\min}$ and $\ell_{\max}$ are the minimum and maximum length in bits of a frame. In what follows, the length in bits of a vector $\mathbf{z}$ is denoted by $\ell\left(\mathbf{z}\right)$ and its observation (soft information) at the output of a channel, a channel decoder, or a lower protocol layer is denoted by $\mathbf{y}_z$. Moreover, $\mathbf{z}_a^b$ represents the sub-vector of $\mathbf{z}$ between indexes a (bits) and b (bits).

B. Aggregated frames within a burst

Consider a burst of L bits consisting of N aggregated frames. This burst contains either $N-1$ *data* frames and an additional *padding* frame containing only padding bits, see Figure 1, or N *data* frames and no padding bits. Assume that each of these frames, except the padding frame, contains a header and a payload and follows the same syntax, as described in Section II-A.

Assuming that L is fixed *before* frame aggregation and that N is not determined *a priori*, the accumulated length in bits of the n first aggregated frames may be described by a Markov process, which state is denoted by S_n. If $\ell < L$, then

$$P\left(S_n = \ell \,|\, S_{n-1} = \ell'\right) = \begin{cases} \pi_{\ell-\ell'} & \text{if } \ell_{\min} \leqslant \ell - \ell' \leqslant \ell_{\max} \\ 0 & \text{else,} \end{cases} \tag{3}$$

and if $\ell = L$, then

$$P\left(S_n = L \,|\, S_{n-1} = \ell'\right) = \begin{cases} 0, \text{ if } L - \ell' > \ell_{\max} \\ 1, \text{ if } 0 < L - \ell' < \ell_{\min} \\ \sum\limits_{k=L-\ell'}^{\ell_{\max}} \pi_k, \text{ else.} \end{cases} \tag{4}$$

In (4), if $0 < L - \ell' < \ell_{\min}$, there is not enough space in the burst to put a data frame and the n-th frame is thus necessarily the (last) padding frame. If $L - \ell' > \ell_{\max}$, there is enough space to put a data frame of any allowed length, thus, the n-th frame cannot end the burst. In the other cases, only data and padding frames of $L - \ell'$ bits are allowed to finish the burst. This is possible when the n-th data frame is of $L - \ell'$ bits or if the n-th data frame generated by the source has a length strictly larger than $L - \ell'$ bits.

With this representation, the successive values taken by S_n, $n = 0, \ldots, N_{\max}$ can be described by a trellis, first proposed in [3], as shown in Figure 2. This trellis is inspired from the one proposed in [6] for the robust decoding of variable-length encoded data. In this trellis, dashed transitions correspond to padding frames and plain transitions correspond to data frames. For the last frame (when $S_n = L$), dashed and plain transitions may be parallel.

III. Three-State FS Automaton

This section proposes an adaptation of the three-state (3S) automaton proposed in [1] for the synchronization of fixed-length ATM cells and adapted by Ueda *et al.* in [19] to deal with variable-length aggregated frames. When the frames are of variable length, a single random bit error in the header may falsely signal synchronization error. Therefore, some error correction has to be used. In [19], frame headers are assumed to contain an HEC and a length field. The HEC is assumed to be a CRC, which is used as an error-correcting code for the part of the header protected by the HEC. Correction is performed before hunting for synchronization, if required. Rather long HECs are required to be able to correct errors, as error correction is less efficient when the number of HEC

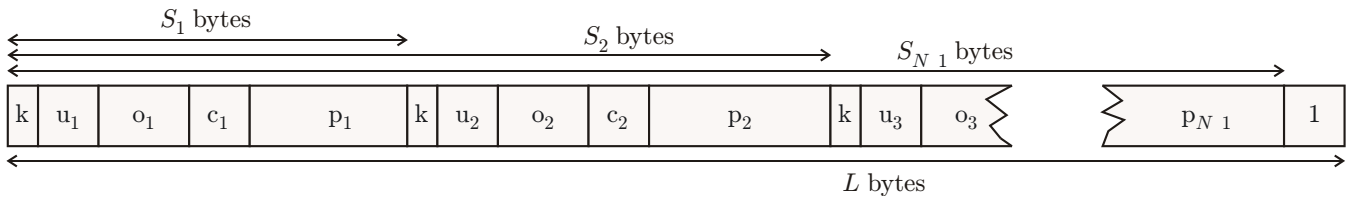

Figure 1. Aggregated frames in a WiMAX burst

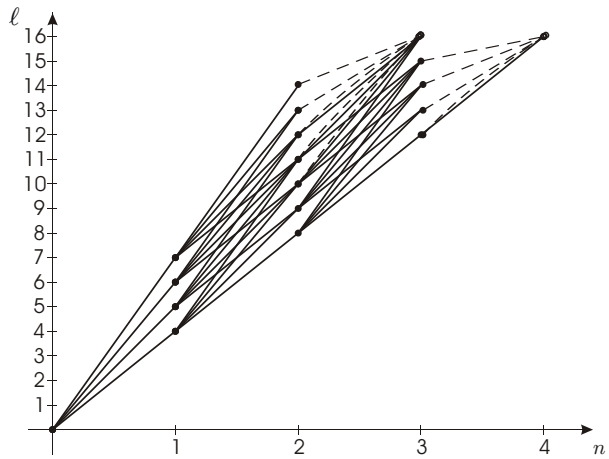

Figure 2. Trellis depicting the allowed total length in bits L vs. the number of frames n in a *burst* of $L = 16$ bits with $\ell_{\min} = 4$ bits and $\ell_{\max} = 7$ bits. Dashed lines correspond to padding bits.

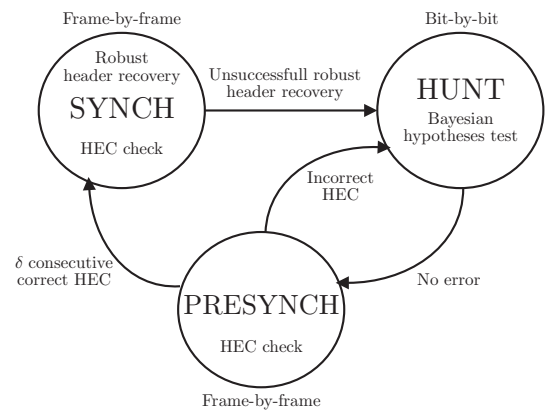

Figure 3. Three-state FS automaton

bits protecting the header is small compared to the size of the header.

More precisely, the automaton presented in [19] consists of three states: *SYNCH*, *HUNT*, and *PRESYNCH*. Assume that the automaton is in the *SYNCH* state. It remains in this state as long as no synchronization error is detected using HEC. If HEC detects a synchronization error, one first tries to correct errors in the header. In case of failure, the automaton switches to the *HUNT* state. In *HUNT* state, the automaton hunts for the correct synchronization by searching bit-by-bit for the correct HEC and for the assumed header fields. Once an agreement is found, the automaton switches to the *PRESYNCH* state, an intermediate state. In this state, a frame-by-frame checking is performed. In the *PRESYNCH* state, one makes sure that the synchronization retrieved in the *HUNT* state is indeed the correct synchronization by verifying that HEC is correct for $\delta > 0$ consecutive frames. Once δ consecutive correct HECs have been obtained, the automaton returns to the *SYNCH* state and in case of failure it again switches to *HUNT* state. This method is denoted in what follows as the *Ueda's method*.

This section proposes several improvements to Ueda's method, see Figure 3. First, instead of performing error correction in the *SYNCH* state, the robust header estimation presented in [13] and briefly recalled in Section III-A is employed to estimate the length field of the frame. In case of failure to verify HEC of header with estimated length field (HEC verification is performed after replacing the received noisy length field with the estimated length field), the automaton switches to the *HUNT* state, where Bayesian hypotheses testing is performed to search for the correct FS, see Section III-B. The operation performed in the *PRESYNCH* state remains unchanged.

Note that alternatively, an automation with a single *HUNT* State (*HUNT* State Alone, HSA) may be considered to perform FS. This may be useful when the frame length is not included in the header.

A. SYNCH State: Header Recovery

In [13], a maximum *a posteriori* estimator is proposed to determine some fields in the headers of the aggregated frames. In case of FS with frames of variable lengths, one is mainly interested in the length field denoted by $\mathbf{u}_n$ for the n-th frame. With the notations of Section II-A, one obtains

$$\hat{\mathbf{u}}_n = \arg \max_{\mathbf{u}_n \in \Omega_u} P\left(\mathbf{y}_u | \mathbf{u}_n\right) \sum_{\mathbf{o}} P\left(\mathbf{y}_o | \mathbf{o}\right) P\left(\mathbf{o}\right) P\left(\mathbf{y}_c | \mathbf{c} = \mathbf{f}\left(\mathbf{k}, \mathbf{u}_n, \mathbf{o}\right)\right), \quad (5)$$

where $\Omega_u = \{\ell_{\min}, ..., \ell_{\max}\}$ is the set of lengths which may be taken by the length field. The evaluation of (5) may be done optimally with a complexity $\mathcal{O}(\ell\left(\mathbf{o}\right) 2^{\ell(\mathbf{c})})$, or suboptimally with a reduced-complexity algorithm, see [13] for more details.

B. HUNT State: Bayesian hypotheses test

In [7], [9], [8], [14], several hypotheses tests based on a Neyman-Pearson (NP) criterion are introduced to determine whether a frame starts at a given bit index. Only the presence

of a SW is assumed. Evaluations have been performed for balanced data symbols in [8] and for unbalanced ones in [14]. This technique is efficient when the SW is long, but suffers limitations when it is short.

This section is devoted to the construction of Bayesian hypotheses tests exploiting all information present in the header as proposed in [3], [13]. This allows to build more efficient tests, especially when the SW is short.

Consider a given bit index ℓ of the burst. Under the hypothesis H_h that a frame header $\mathbf{h} = [\mathbf{k}, \mathbf{u}, \mathbf{o}, \mathbf{c}]$ starts at ℓ, one may interpret the corresponding channel output as $\mathbf{y} = [\mathbf{y}_k, \mathbf{y}_u, \mathbf{y}_o, \mathbf{y}_c]$ and

$$P(\mathbf{y}|H_h) = \sum_{\mathbf{h}} P(\mathbf{y}|\mathbf{h}, H_h) P(\mathbf{h}|H_h). \quad (6)$$

With the same hypotheses as in [3], only the $\mathbf{h}$ starting[1] with $\mathbf{k}$ have to be considered since this field is fully determined. Moreover, assuming that the channel is memoryless, taking into account the fact that $\mathbf{k}, \mathbf{u}, \mathbf{o}$, and $\mathbf{c}$ do not depend on $\mathbf{p}$, and the fact that the HEC $\mathbf{c}$ is fully determined by $\mathbf{k}$, $\mathbf{u}$, and $\mathbf{o}$, (6) becomes

$$P(\mathbf{y}|H_h) = P(\mathbf{y}_k|\mathbf{k}) \sum_{\mathbf{u}\in\Omega_u} \left(P(\mathbf{y}_u|\mathbf{u}) P(\mathbf{u}) \sum_{\mathbf{o}} P(\mathbf{y}_o|\mathbf{o}) P(\mathbf{o}) P(\mathbf{y}_c|\mathbf{c} = \mathbf{f}(\mathbf{k}, \mathbf{u}, \mathbf{o}))\right). \quad (7)$$

Under the hypothesis H_d that ℓ does not correspond to the beginning of a frame, $\mathbf{y}$ is the channel output when data bits $\mathbf{d}$ are transmitted. Thus, assuming balanced data symbols, one gets

$$\begin{aligned} P(\mathbf{y}|H_d) &= \sum_{\mathbf{d}} P(\mathbf{y}|\mathbf{d}, H_d) P(\mathbf{d}|H_d) \\ &= \sum_{\mathbf{d}} P(\mathbf{y}|\mathbf{d}, H_d) 2^{-\ell(\mathbf{d})}. \end{aligned} \quad (8)$$

Bayesian hypotheses test can now be represented as

$$\Lambda(\mathbf{y}) = \frac{P(\mathbf{y}|H_d)}{P(\mathbf{y}|H_h)} \underset{D_d}{\overset{D_h}{\lessgtr}} \frac{P_a(\ell, H_h)}{P_a(\ell, H_d)}, \quad (9)$$

where D_h or D_d correspond to deciding H_h or H_d respectively. $P_a(\ell, H_h)$ and $P_a(\ell, H_d)$ are the *a priori* probabilities of the hypotheses at the bit index ℓ .

When $L - \ell < \ell_{\max}$, ℓ may also represent the start of a padding frame. Thus, an additional hypothesis H_p, corresponding to the presence of a padding frame, has to be considered. The Bayesian hypotheses test for deciding between H_p and H_d is given by

[1]Here, to facilitate presentation, we have assumed that the bits of the known field $\mathbf{k}$ are at the beginning of the header. Any other organization of the header may be considered.

$$\Lambda(\mathbf{y}) = \frac{P(\mathbf{y}|H_d)}{P(\mathbf{y}|H_p)} \underset{D_d}{\overset{D_p}{\lessgtr}} \frac{P_a(\ell, H_p)}{P_a(\ell, H_d)}. \quad (10)$$

Under H_p, we have $P(\mathbf{y}|H_p) = P(\mathbf{y}|\mathbf{1})$, where $\mathbf{1}$ is a vector of ones of the same size as of $\mathbf{y}$. $P(\mathbf{y}|H_d)$ is given by (8).

First (9) is applied to choose between header and data. Then, if data has been decided, (10) is considered to see whether the data corresponds to a padding frame or not.

1) A priori probabilities: To determine the *a priori* probability $P(\ell)$ that a frame (data or padding) starts at a bit index ℓ of a burst of L bits, consider again the trellis representation as shown in Figure 2. One may write

$$P(\ell) = \sum_{1\leqslant n\leqslant \lceil \ell/\ell_{\min}\rceil} P(S_n = \ell), \quad (11)$$

where $P(S_n = \ell)$ is the *a priori* probability that the n-th frame ends at bit index $\ell - 1$ (or that the $n+1$-th frame starts at bit index ℓ). $P(S_n = \ell)$ satisfies

$$P(S_n = \ell) = \sum_{\ell'} P(S_n = \ell | S_{n-1} = \ell') P(S_{n-1} = \ell'), \quad (12)$$

which may be evaluated iteratively with the help of (3) and (4), starting from $n = 1$ till $n = \lceil \ell/\ell_{\min}\rceil$, with initial condition $P(S_0 = 0) = 1$ and $P(S_0 \neq 0) = 0$.

A priori probability $\overline{P}(\ell) = P_a(\ell, H_d)$ corresponding to an absence of the start of frame at bit index ℓ is calculated by $\overline{P}(\ell) = 1 - P(\ell)$. $P(\ell)$ represents the *a priori* probability of a frame start, be it a data and/or a padding frame. Thus,

$$\begin{aligned} P(\ell) &= P_a(\ell, H_h) + P_a(\ell, H_p) \\ &= P(\ell) P(H_h|\ell) + P(\ell) P(H_p|\ell), \end{aligned}$$

where

$$P(H_h|\ell) = \begin{cases} 0, \text{ if } 0 < L - \ell < \ell_{\min} \\ 1, \text{ if } L - \ell > \ell_{\max} \\ \sum_{\lambda=\ell_{\min}}^{L-\ell} \pi_\lambda, \text{ else,} \end{cases} \quad (13)$$

and $P(H_p|\ell) = 1 - P(H_h|\ell)$ are the conditional *a priori* probabilities of H_h (header hypothesis) and H_p (padding hypothesis), respectively.

IV. Simulation results

Simulation results for WiMAX MAC layer are provided in this section. In the WiMAX standard [4], downlink (DL) sub-frames begin with a frame control section that contains the down-link map (DL-MAP) for the current down-link frame as well as the up-link map (UL-MAP) for a frame in future. The DL sub-frame is divided into *bursts*, each containing multiple concatenated fixed-length or variable-length MAC frames received from the higher layers. Each burst is modulated and coded at the PHY layer before transmission in the PHY DL sub-frame. Lengths of burst are assigned by the scheduler and

are communicated to users in the DL-MAP. In what follows, we consider only the DL scenario and thus only the DL subframe.

MAC PDU begins with a fixed-length header of $\ell_h = 48$ bits, followed by a variable-length payload and an optional CRC [4]. The header can be of two types, one is the generic header containing either MAC Management messages or Convergence Sublayer (CS) data and the other is the bandwidth request header used to request additional bandwidth. For DL with already established connections MAC PDUs always have a generic header as shown in Figure 4. The LEN field represents the length in bytes of the MAC PDU including the MAC header and Header Check Sequence (HCS). The HCS is a function of the content of all header fields, and is stored in an 8-bit field. Some assumptions are made for the sake of simplicity. CRC, ARQ, packing, fragmentation, and encryption are not used. Furthermore, no extended subheader is present. Some fields are already fixed in a MAC header, but with the considered assumptions, fields such as Header Type (HT), Encryption Control (EC), sub-headers, and special payload types (Type), Reserved (Rsv), CRC Indicator (CI), and Encryption Key Sequence (EKS) remain constant. All these fields are merged together in a constant field $\mathbf{k}$ of 13 bits, which can be used as a SW. A simulator consisting of a burst generator, a channel, and a receiver has been implemented. Simulations are carried over both Additive White Gaussian Noise (AWGN) and Rayleigh fading channel. In the case of Rayleigh fading channel, the modulated signal is subject to zero mean and unit variance fast (bit) Rayleigh fading plus zero-mean AWGN noise.

Bits	1	1	6	1	1	2	1	11	16	8
	HT	EC	Type	Rsv	CI	EKS	Rsv	LEN	CID	HCS

Figure 4. Generic MAC Header

In these simulations, we assume that all frames are byte aligned and that byte alignment has been already performed. Then in the *HUNT* state, the search is done byte-by-byte. This assumption is reasonable when FS is considered at intermediate layers of the protocol stack. Moreover, the burst size L is received without any error as it is transmitted in DL-MAP, which is protected with a more robust modulation and coding scheme. We have chosen $L = 1800$ bytes, as by ignoring the header overhead, 1800 bytes in every frame of 20 ms can provide a raw data rate of 703.125 kbps, which, if fully dedicated, is enough for a real-time video service. Random sized data frames with $\ell_{\min} = 50$ bytes and $\ell_{\max} = 200$ bytes are concatenated in a burst. If the generated frame is not insertable in the remaining space of a *burst*, a padding frame is inserted to fill the burst, which is then BPSK modulated before being sent over the channel.

For performance analysis, Erroneous Frame Location Rate (EFLR) is evaluated as a function of the channel Signal-to-Noise Ratio (SNR). A frame is deemed correctly synchronized when both of its ends have been correctly determined.

The proposed *on-the-fly* 3S FS method is compared with several other FS techniques. First, *on-the-fly* methods such as Ueda's method [19][2] and the method presented in [8], denoted as NP FS in what follows, serve as reference. Second, the *hold-and-sync* trellis-based technique presented in [3] is also considered, mostly as a lower bound for the EFLR, since the corresponding complexity and delay are not of the same order of magnitude. Since the MAC header uses an HEC of 8 bits only, a modified version[3] of Ueda's method, denoted MU, is applied.

Simulation results for transmission over an AWGN channel are provided in Figure 5. For the NP FS method, two curves

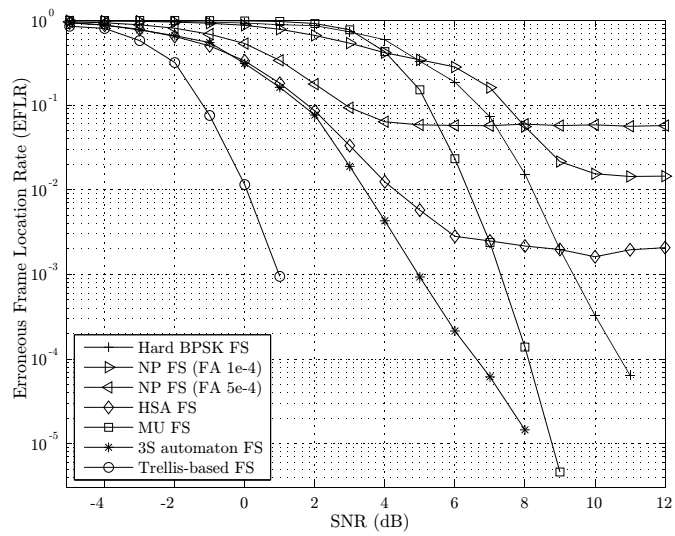

Figure 5. FS methods over an AWGN channel, with $\ell_{\min} = 50$ bytes, $\ell_{\max} = 200$ bytes, and $L = 1800$ bytes

with two probability of False Alarm (FA) have been provided. The NP FS method with such a short SW of 13 bits has some FS error floor at high SNR, as expected, due to unavoidable and persistent high false alarm (emulation), as the payload data can simulate the SW more often due to the short size of the SW. The HSA method, since it uses more redundancy from the header, gives a much lower floor.

[2]In case of two-bit error syndromes, the frame length is verified by evaluating the HEC over the next header, at the position indicated by the corresponding candidate. If no candidate is correct, *i.e.*, corresponds to correct HEC at the potential location of the next header, then the candidate that gives one bit error syndrome at the next header is selected. In case of failure, the automaton switches to the *HUNT* state.

[3]**Modified UEDA's (MU) method:** UEDA's Method [19] is well adapted for rather long HEC like CRC-16, because in this case there are no more than two candidates for two-bit error syndromes. For low order HEC, *e.g.*, CRC-8, many more candidates can be found for syndromes with two bits in error, thus UEDA's method needs some modification. We propose to search for the best candidate by shifting the bit stream by the potential frame length and then calculating HEC over the next header sequence at the position indicated by corresponding candidate. If no candidate is correct (*i.e.*, corresponds to correct HEC) at the potential location of the next header sequence, then the candidate that gives one-bit error syndrome at the next header sequence is selected as the best candidate. In case of failure, the FS automaton switches to *HUNT* state. Unfortunately, in case of low order HEC, meeting a configuration in which random bits emulate the header are more likely, making the method of hunting using HECs unreliable at low SNR.

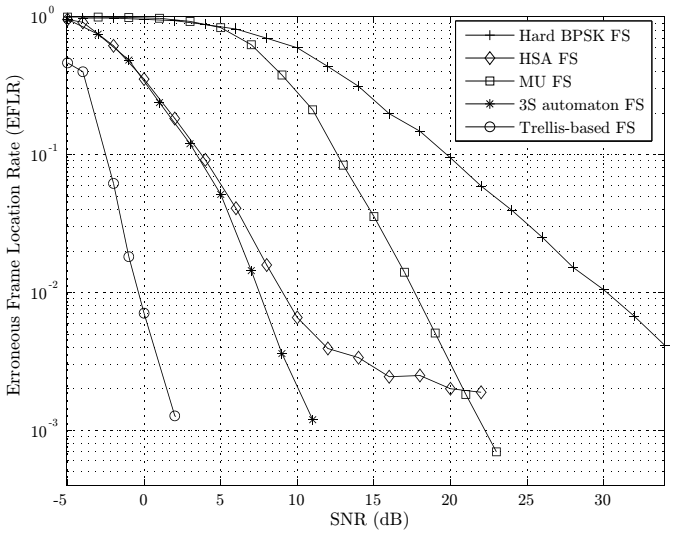

Figure 6. FS methods over a Rayleigh channel, with $\ell_{\min} = 50$ bytes, $\ell_{\max} = 200$ bytes, and $L = 1800$ bytes

The proposed 3S FS method performs better than MU method, due to the use of soft hunt operation and to the robust header recovery technique. The MU method performs poorly, specifically at low SNR, due to the fact that it uses hard HEC detection/correction and the size of HCS (8 bits) is small compared to the size of the header (48 bits), thus more than 10 candidates for two bit error syndrome have to be considered. The 3S FS automaton performs better at low SNR due to the effectiveness of Bayesian hypothesis testing in the *HUNT* state, which can retrieve synchronization quickly. The difference between 3S FS and MU decreases at high SNR, since even though header recovery performs well, erroneous FS by Bayesian hypothesis testing degrades the performance because even in a good channel condition one can wrongly assume misalignment due to the simulation of header by random data. Trellis-based technique provide the best performance as expected, but at the cost of a delay equivalent to the length of an aggregated frame. The proposed 3S FS technique clearly provides improved FS compared to other *on-the-fly* algorithms thus providing reasonable compromise between performance and latency.

Simulation results for transmission over a Rayleigh channel are shown in Figure 6. MU method performance is highly impacted by Rayleigh channel, while the results for the 3S automaton clearly show its effectiveness in this context.

V. Conclusion

Header Error Control detection used as an indicator of correct frame synchronization does not perform efficiently at low SNR for variable-length frames. This paper proposes a FS technique exploiting structural properties of the frame headers to perform efficient segmentation. This technique makes use of header recovery and Bayesian hypothesis testing, while exploiting soft information provided at the output of the channel (or channel decoder) in conjunction with *a priori* information due to the redundancy present at the frame header. Simulations clearly demonstrate an improvement in terms of erroneous frame localization rate compared to methods performing computations *on-the-fly*. Practically, this improvement will result in reducing the amount of frames that need to be retransmitted.

Acknowledgment

This work was partly supported by Microsoft Research through its PhD European Scholarship program and by the NoE NEWCOM++.

References

[1] B - ISDN user-network interface - physical layer specification: General characteristics, February 1999.

[2] IEEE 802.11n part 11: Wireless LAN Medium Access Control (MAC) and physical layer (PHY) specifications: Enhancements for higher throughput, March 2006.

[3] U. Ali, M. Kieffer, and P. Duhamel. Joint protocol-channel decoding for robust aggregated packet recovery at WiMAX MAC layer. In *Proc. IEEE SPAWC*, 21-24 June, 2009.

[4] ANSI/IEEE. 802.16: IEEE standard for local and metropolitan area networks, air interface for fixed broadband wireless access systems. Technical report, 2004.

[5] R. H. Barker. *Group synchronization of binary digital systems in Communication Theory.* Butterworth, London, 1953.

[6] R. Bauer and J. Hagenauer. Symbol-by-symbol MAP decoding of variable length codes. In *Proc. 3rd ITG Conference Source and Channel Coding*, pages 111–116, München, 2000.

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

[8] M. Chiani and M. G. Martini. On sequential frame synchronization in AWGN channels. *IEEE Transactions on Communications*, 54(2):339 – 348, February 2006.

[9] 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.

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

[11] ETSI. Digital video broadcasting (DVB); framing structure, channel coding and modulation for digital terrestrial television. Technical report, ETSI EN 300 744 v1.5.1, jun. 2004.

[12] G. L. Lui and H. H. Tan. Frame synchronization for gaussian channels. *IEEE Transactions on Communications*, 35(8):818–829, 1987.

[13] C. Marin, Y. Leprovost, M. Kieffer, and P. Duhamel. Robust header recovery based enhanced permeable protocol layer mechanism. In *IEEE 9th Workshop on Signal Processing Advances in Wireless Communications, 2008. SPAWC*, pages 91–95, 6-9 July 2008.

[14] M. G. Martini and M. Chiani. Optimum metric for frame synchronization with gaussian noise and unequally distributed data symbols. In *IEEE 10th Workshop on Signal Processing Advances in Wireless Communications, 2009. SPAWC2009*, 21-24 June 2009.

[15] J. L. Massey. Optimum frame synchronization. *IEEE Transactions on Communications*, 20(4):115 – 119, 1972.

[16] 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.

[17] R. A. Scholtz. Frame synchronization techniques. *IEEE Transactions on Communications*, 28(8):1204 – 1213, 1980.

[18] A. Sidelnikov, J. Yu, and S. Choi. Fragmentation/aggregation scheme for throughput enhancement of IEEE 802.11n WLAN. In *IEEE APWCS 2006*, August 2006.

[19] H. Ueda, U. Yamaguchi, and R. Watanabe. Reducing misframe frequency for HEC-based variable length frame suitable for IP services. In *Proc. IEEE ICC*, volume 4, pages 1196 – 1200, 2001.

[20] R. G Woo, P Kheradpour, D. Shen, and D. Katabi. Beyond the bits: cooperative packet recovery using physical layer information. In *Proceedings of the 13th annual ACM international conference on Mobile computing and networking*, pages 147 – 158, 2007.