Robust MAC-Lite and Header Recovery Based Improved Permeable Protocol Layer Scheme

Cédric Marin and Yann Leprovost
Alcatel-Lucent, Research and Innovation
Route de Villejust, 91620, FRANCE
Email: {cedric.marin, yann.leprovost}@alcatel-lucent.fr

Michel Kieffer and Pierre Duhamel
L2S, CNRS–Supelec–Univ Paris-Sud
Plateau de Moulon, 91192, FRANCE
Email: {michel.kieffer, pierre.duhamel}@lss.supelec.fr

Abstract—This paper presents an improved permeable layer mechanism useful for highly robust packetized multimedia transmission. MAP estimation-based packet header recovery at various protocol layers is the cornerstone of the proposed solution. This header-correction technique exploits the available intralayer and inter-layer header correlation to define a reduced set of header configurations. The best candidate is then selected in the pre-defined set through a soft decoding based on CRC redundancy. To improve the decoding performance, we also generalize the principle of UDP-Lite and propose a robust MAC layer, called MAC-Lite. Simulation results for WiFi transmission over AWGN channels assess a substantial (12 dB) link budget improvement.

I. INTRODUCTION

In restricted bandwidth networks, high efficiency multimedia transmission is heavily dependent on the source coding mechanism at the Application (APL) layer [1]. Nevertheless, the highly compressed multimedia contents are very sensitive to transmission errors and a single corrupted bit may lead to a loss of a large amount of information at the receiver. Consequently, the encoded bitstream entering the source decoder has to be nearly error-free.

This situation is hardly satisfied when considering transmission over wireless channels. Factors such as high signal attenuation, multiple access interferences, inter-symbol interferences, and Doppler shift can heavily degrade the signal quality. As a consequence, the received information may be heavily corrupted and unusable by the source decoder. In packetized data transmission, several protection mechanisms have been introduced to ensure the error recovery at the receiver side. Two classical techniques are usually combined: (i) retransmission of damaged packets controlled by CRCs and checksums [2], (ii) packet correction based on strong error-correcting codes and packet-erasure codes [3].

Nevertheless, retransmissions may become difficult in scenarios with strong delay constraints (such as for the visiophony), or even impossible when broadcasting data (e.g. for digital multimedia delivery on satellite). Moreover, the channel code redundancy is rarely correctly dimensioned. It may be insufficient in bad channel conditions, or over-sized when channel is clear. Error-concealment techniques [4] are often used by the source decoders at APL layer when a packet is lost. 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 past, joint source-channel decoding methods have been proposed to efficiently recover corrupted packets. These techniques involve robust source decoders and exploit the residual redundancy in the received packets for correcting errors [5]–[10]. These robust decoders improve the link budget of the multimedia transmission when compared to classical schemes. These robust solutions are however not compliant with the standard protocol stacks since they require exchange of soft information from the Physical (PHY) layer to the APL layer. Nevertheless, the error-detection mechanisms included in the standard protocol stacks prevent corrupted packets to reach the APL layer. The main reason being that the errors may impact some essential information contained in the various headers at intermediate protocol layers.

To circumvent the above inconsistency, we first propose a robust protocol layer scheme where error-detection codes are applied to the header fields only. Then, we introduce an efficient header recovery technique for correcting the corrupted headers. More headers are thus correctly interpreted at each layer, increasing the number of packets reaching the APL layer. This combination allows to obtain an improved permeable [11] protocol layer mechanism having the ability of exchanging soft information between protocol layers.

The paper is organized as follows. After an introduction of the improved permeable layer scheme in Section II, Section III describes the derivation of the header recovery technique. The design of the proposed mechanism for PHY and MAC layers of WiFi is then detailed in Section IV. Finally, simulation results are presented in Section V before drawing some conclusions.

II. IMPROVED PERMEABLE LAYER MECHANISM

Real-time packetized multimedia transmission is usually based on the RTP/UDP/IP protocol stack [12]. As an example, Figure 1 illustrates segmentation and encapsulation mechanisms at each protocol layer in case of packetized multimedia transmission with the 802.11 standard (WiFi) [13]. Error-detection mechanisms are implemented at each layer and are listed below:

- At PHY layer, a CRC protects the header fields. Received packets with damaged headers are discarded.
At MAC layer, a CRC protects the header and the payload. When an error occurs, 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.

Inter-layer redundancy is present in the protocol stack of Figure 1. This type of redundancy has been exploited in the Robust Header Compression (ROHC) mechanism [15]. Here, this redundancy is used to build some a priori information on erroneous headers, which will facilitate their estimation. Second, CRCs and checksums are used as error-correcting codes [16], [17].

In such a scheme, the error-detection mechanisms provided by CRCs and checksums, combined with the retransmission device at MAC layer, allow APL layer to receive error-free packets. The price to be paid is a potentially important transmission delay due to retransmissions which may become very frequent when channel conditions worsen. To reduce this delay, the number of retransmissions is limited by a threshold. As a consequence, many MAC packets may be discarded (even if they contain only a few erroneous bits), leading to the loss of some APL packets.

Joint source-channel decoding methods are useful for correcting multiple errors at APL layer. Soft information provided by lower protocol layers is required to achieve this efficiently. The recently introduced UDP-Lite [14] mechanism allows to obtain a permeable UDP layer. In UDP-Lite, the checksum protects a limited number of bytes, generally including the UDP-Lite, RTP, and APL header fields. By analogy, the same principle may be applied to the MAC layer protocol. We introduce here a permeable MAC layer mechanism, called MAC-Lite, where the CRC covers the header fields only. So, in a protocol stack with PHY/MAC-Lite/IPV4/UDP-Lite, all received packets except the ones with corrupted header fields are transferred to the APL layer.

The aim of this paper is to use the various sources of redundancy in the protocol stack to recover corrupted headers, increasing the amount of packets reaching the APL layer. In a second aspect, this paper enables soft information delivered by the channel decoder at PHY layer to reach the APL layer, improving the performance of robust decoding.

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 of the upper layers.

---

**Figure 1. Protocol stack for multimedia transmission over WiFi**

![Protocol Stack Diagram](image)

**Figure 2. Proposed permeable layer mechanism**

![Permeable Layer Diagram](image)

---

Finally, header recovery combines soft information provided by the lower protocol layers, properties of the CRCs or checksums, as well as previously introduced a priori information. Figure 2 illustrates the general technique for sending soft information from layer $L - 1$ to layer $L + 1$ through the permeable layer $L$. This figure also summarizes the various sources of redundancy used at layer $L$ to perform the header recovery.

Additionally, the global computational complexity may be minimized by deactivating the header recovery processing when the normal CRC check is successful. It should also be deactivated when the quality of soft information provided by the lower layers is too poor, i.e., when the signal power is inferior to a pre-defined threshold. In such a situation, the packet is retransmitted or discarded.

This paper focuses on the PHY and MAC layers of WiFi, as generic examples. In fact, the proposed permeable layer mechanism could be easily applied to any other protocol layer.

**III. MAP ESTIMATOR FOR ROBUST HEADER RECOVERY**

As a general situation, at a given layer $L$, the $n$-th incoming packet includes a header, a payload, and a CRC (or a checksum). With the above presented concept, the CRC $c^L_n$ of $\ell^L_k$ bits protects the header fields only. Information contained in the header may be split in three categories:

- The constant fields, represented by the vector $k^L_n$ of $\ell^L_k$ bits, are assumed to be known.
- The predictable fields are embedded in the vector $p^L_n$ of $\ell^L_p$ bits. Contrary to the known fields, the predictable fields are estimated by exploiting the intra-layer and inter-layer redundancy represented by $R^L_n$. They are predicted by using information contained in the previous received packets. The
predictable fields are assumed to be entirely determined if the preceding headers have been correctly recovered. The unknown fields are collected in the vector $u^L_n$ of $\ell^L_n$ bits. These parameters are either completely unknown or limited to a configuration set $\Omega^L_{u,n}$, which content is determined by the values of $k^L_n$, $p^L_n$, and $R^L_n$.

All these fields are included in the vector $r^L_n = [k^L_n, p^L_n, u^L_n]$. This is only a notation for mathematical convenience, since the order of the bits in $r^L_n$ does not necessarily correspond to the order in which data are stored in the header of the packet. The CRC $c^L_n$ associated to $r^L_n$ is evaluated as $c^L_n = F^L(r^L_n)$, where $F^L$ is a generic encoding function related to layer $L$. Moreover, the $\ell^L_{x,n}$ bits of the payload are collected in $x^L_n$.

We consider that the data have been transmitted over an AWGN channel, introducing a noise distributed according to $N(0, \sigma^2)$. Noisy header and CRC, associated to the $n$-th packet and coming from layer $L-1$, are denoted by $y^L_n = [y^L_{k,n}, y^L_{p,n}, y^L_{c,n}, y^L_{c,n}]$, which includes observations of $k^L_n$, $p^L_n$, $u^L_n$, and $c^L_n$.

Since $k^L_n$ is known and $p^L_n$ may be predicted according to $R^L_n$, only $u^L_n$ remains to be estimated. The MAP estimator

$$\hat{u}^L_n = \arg \max_{u^L_n} P(u^L_n|y^L_{u,n}, y^L_{c,n}, k^L_n, p^L_n, R^L_n)$$

$$= \arg \max_{u^L_n} P(u^L_n|y^L_{u,n}, y^L_{c,n}|k^L_n, p^L_n, R^L_n),$$

(1)

takes into account the observations $y^L_{u,n}$, the knowledge of $k^L_n$, $p^L_n$, and $R^L_n$ as well as the CRC properties useful to estimate $u^L_n$.

The channel is memoryless and one may write

$$P(u^L_n|y^L_{u,n}, y^L_{c,n}, k^L_n, p^L_n, R^L_n) = \frac{P(u^L_n|y^L_{u,n}, y^L_{c,n}|k^L_n, p^L_n, R^L_n)P(y^L_{u,n}|u^L_n)P(y^L_{c,n}|c^L_n)}{P(y^L_{u,n}, y^L_{c,n}|k^L_n, p^L_n, R^L_n)},$$

(2)

where $c^L_n = F^L([k^L_n, p^L_n, u^L_n])$ corresponds to the direct evaluation of the CRC. In (2), $P(y^L_{u,n}|u^L_n) \sim N(u^L_n, \sigma^2 I_{\ell^L_n})$ and $P(y^L_{c,n}|c^L_n) \sim N(c^L_n, \sigma^2 I_{\ell^L_n})$. $P(u^L_n|k^L_n, p^L_n, R^L_n)$ represents the a priori probability of $u^L_n$. Assuming that all combinations of $u^L_n \in \Omega^L_{u,n}$ are equally likely, one gets

$$P(u^L_n|k^L_n, p^L_n, R^L_n) = P(u^L_n|\Omega^L_{u,n}) = \frac{1}{|\Omega^L_{u,n}|},$$

where $|\Omega^L_{u,n}|$ denotes the cardinal number of $\Omega^L_{u,n}$.

Finally, using (2) in (1), the MAP estimator becomes

$$\hat{u}^L_n = \arg \max_{u^L_n \in \Omega^L_{u,n}} P(y^L_{u,n}|u^L_n)P(y^L_{c,n}|c^L_n)$$

$$= \arg \max_{u^L_n \in \Omega^L_{u,n}} \|y^L_{u,n} - u^L_n\|^2 + \|y^L_{c,n} - F^L([k^L_n, p^L_n, u^L_n])\|^2,$$

(3)

where $\| \cdot \|$ denotes the euclidian distance.

IV. APPLICATION TO THE 802.11 STANDARD

Figure 3 presents the studied network architecture. In this paper, we focus on the downlink multimedia transmission over WiFi and we apply our method to the two lowest layers of the protocol stack. First, the format of PHY and MAC-Lite packets are described in Sections IV-A and IV-B. Intra-layer and inter-layer redundancy are then identified in Section IV-C. The processing used for PHY and MAC-Lite header recovery are presented in Sections IV-D and IV-E. A global scheme is finally proposed in Section IV-F.

A. DSSS PHY Layer Description

At PHY layer, the 802.11 standard provides 1 or 2 Mbps transmission rates in the 2.4 GHz when using the Direct Sequence Spread Spectrum (DSSS) mode. The DSSS PHY packet format is illustrated in Figure 4. Contrary to the payload, the preamble and the header are always transmitted by using the 1 Mbps bitrate. The SYNC and SFD fields are used to perform the synchronization operation.

In the $n$-th PHY packet, a CCITT CRC-16 $c^P_n$ protects the Signal, Service, and Length fields only. Its associated encoding function is denoted by $F^P$. Service is reserved for future recommendation and is composed of zeros. It is included in $k^P_n$ according to the notations of Section III. Signal defines the payload bitrate and may take two values. Length specifies the duration in microseconds required to transmit the payload. It depends on both the bitrate and the PHY payload size. Signal and Length are thus collected in $u^P_n$. At this layer, $p^P_n = \emptyset$ and $x^P_n$ contains the $\ell^P_{x,n}$ bits of payload.

B. MAC-Lite Layer Description

The introduced MAC-Lite data packet (or MAC-Lite fragment) format is depicted in Figure 5. In the $n$-th incoming MAC-Lite fragment, a CRC $c^M_n$ of 4 bytes protects the header fields only. Its encoding function is denoted by $F^M$. This modification represents the only difference between the standard MAC fragment and the proposed one.

Considering a non-encrypted downlink transmission of ordered MAC-Lite data packets with deactivated retransmission
and power-save mode, the Frame Control field except the More Fragment flag are assumed to be known. Receiver Address specifies the MAC address of the receiver (T1) and is also known. The last field of the MAC-Lite header is reserved for local wireless networks and is composed of zeros in our scenario. Using the notations defined in Section III, all the previously mentioned fields are embedded in $k_{n}^{MAC-L}$.

AP Address indicates the MAC address of the access point (AP). During the medium reservation procedure (RTS-CTS), this address is transmitted to the receiver and may be totally predicted in the next packets. Router Address corresponds to the MAC address of the router (R1). Assuming that the access point is connected to a single router and that the router address has been already received in other information packets, Router Address may also be deduced by the receiver. Sequence Control contains two parameters: a sequence counter and a fragment counter. Considering that the data packets are transmitted in order, these parameters may be readily determined by the receiver. All these predictable fields are represented by $p_{n}^{MAC-L}$.

More Frag specifies if the current MAC-Lite data packet is the last fragment of an IP packet. Duration indicates the number of microseconds required to transmit the coming MAC-Lite fragment and some control parameters. These two fields are included in $u_{n}^{MAC-L}$. Additionally, $\alpha_{n}^{MAC-L}$ represents the $\alpha_{x,n}^{MAC-L}$ bits of the MAC-Lite payload which ranges from 0 to 2312 bytes.

### C. Identifying Redundancy

Intra-layer and inter-layer correlations facilitate the recovery of noisy headers, since they allow the prediction of $p_{n}^{MAC-L}$ fields and the construction of the $\Omega_{x,n}$ sets defined in Section III. To evidence these correlations, the transactions at MAC layer have to be described.

#### D. PHY Header Recovery

For a given packet at PHY layer, observations associated to $k_{n}^{PHY}$, $u_{n}^{PHY}$, and $\alpha_{n}^{PHY}$ defined in Section IV-A are collected in

$$y_{n}^{PHY} = [y_{x,n}^{PHY}, y_{u,n}^{PHY}, y_{c,n}^{PHY}]$$

In addition, $y_{x,n}^{PHY}$ represents the observations related to the $\ell_{x,n}^{PHY}$ bits of payload $x_{n}^{PHY}$.

The number of values that $u_{n}^{PHY}$ may take is significantly reduced when exploiting the Duration field contained in the previously received MAC-Lite packet (RTS or data packet). Using $B_{n}^{PHY}$ and $D_{n}^{MAC-L}$, one may deduce $\ell_{x,n}^{PHY}$ from (4) as

$$\ell_{x,n}^{PHY} = \left(D_{n}^{MAC-L} - 3T_{SIFS} - 3T_{OVH} - 2\frac{\ell_{C,A}}{B_{n}^{PHY}}\right)B_{n-1}^{PHY}. \quad (6)$$

According to (6), $\ell_{x,n}^{PHY}$ is thus totally determined assuming correct estimation of the previous packet header fields. Then, the duration $L_{n}^{PHY}$ coded in the Length field of the current PHY packet may be computed by using $B_{n}^{PHY}$ as

$$L_{n}^{PHY} = \frac{\ell_{x,n}^{PHY}}{B_{n}^{PHY}}. \quad (7)$$

Figure 6 illustrates the 802.11 MAC transmission protocol for an IP packet fragmented in two MAC data packets. Transmission is initialized by a medium reservation procedure consisting in an RTS-CTS exchange between the AP and the receiver T1. Data packets are then transmitted to T1 which acknowledges them (ACK). In this work, we assume that control packets such as RTS, CTS, and ACK are correctly received. This assumption is reasonable since these packets are small and DBPSK modulated. Only errors in data packets are considered. A Short Inter-Frame Space (SIFS) of 10 $\mu$s separates each packet. The Duration field, included in each MAC packet, specifies the duration scheduled for sending information related to the next MAC fragment. During this period, the other receivers cannot communicate to avoid collisions.

We consider below that $D_{n}^{MAC}$ and $B_{n}^{PHY}$ represent the value of Duration and Signal in the $n$-th packet transmitted by the AP (corresponding to an RTS or a data packet). Following the MAC layer specifications of the 802.11 standard, $D_{n}^{MAC}$ is defined as

$$D_{n}^{MAC} = 3T_{SIFS} + 3T_{OVH} + 2\frac{\ell_{C,A}}{B_{n}^{PHY}} + \frac{\ell_{PHY}}{B_{n}^{PHY}}, \quad (4)$$

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

$$D_{n}^{MAC} = T_{SIFS} + T_{OVH} + \frac{\ell_{C,A}}{B_{n}^{PHY}}. \quad (5)$$

In (4) and (5), $T_{SIFS}$ denotes the duration of a SIFS and $T_{OVH}$ represents the duration for transmitting the PHY overhead (composed of the preamble and the header of constant size) at 1 Mbps. The other terms of (4) depend on the current bitrate $B_{n}^{PHY}$. CTS and ACK have the same constant size $\ell_{C,A}$, and $\ell_{PHY}$ corresponds to the duration for sending one of these packets. Finally, $\ell_{PHY}^{x,n+1}$ refers to the transmission duration of the next PHY payload of $\ell_{x,n+1}^{PHY}$ bits.
According to (7), $L_{\text{PHY}}$ is limited to only two combinations depending on $B_{\text{PHY}}$. These values are stored in $\Omega_{\text{PHY}}^n$.

Integrating these structural redundancy in (3), one obtains the PHY estimator

$$u_{\text{PHY}}^n = \arg \min_{u_{\text{PHY}}^n \in \Omega_{\text{PHY}}^n} \|y_{\text{PHY}}^n - u_{\text{PHY}}^n\|^2 + \|y_{\text{PHY}}^n - c_{\text{PHY}}^n\|^2,$$

where $c_{\text{PHY}}^n = f_{\text{PHY}}^n(k_{\text{PHY}}^n, u_{\text{PHY}}^n)$.\n
\section{MAC-Lite Header Recovery}

Soft information of the PHY payload enters the MAC-Lite layer and represents the observations associated to $k_{\text{MAC-L}}^n$, $p_{\text{MAC-L}}^n$, $u_{\text{MAC-L}}^n$, $x_{\text{MAC-L}}^n$, and $c_{\text{MAC-L}}^n$ specified in Section IV-B. One may write

$$y_{\text{MAC-L}}^n = y_{\text{PHY}}^n = \left[y_{k_{\text{MAC-L}}^n}^n, y_{x_{\text{MAC-L}}^n}^n, y_{p_{\text{MAC-L}}^n}^n, y_{u_{\text{MAC-L}}^n}^n, y_{x_{\text{MAC-L}}^n}^n, y_{c_{\text{MAC-L}}^n}^n\right].$$

$u_{\text{MAC-L}}^n$ the MAC-Lite may be associated to a reduced set of combinations when exploiting the properties defined in (4) and (5). Assuming that $B_{\text{PHY}}$ is known from the PHY layer, $D_{\text{MAC-L}}$ is completely determined when $M_{\text{MAC-L}}^n = 0$. When $M_{\text{MAC-L}}^n = 1$, the value of $\text{Duration}$ depends on the next PHY payload size. The number of combinations is thus associated to the range of MAC-Lite payload size. The number of combinations is thus associated to the range of MAC-Lite payload size. The number of combinations is thus associated to the range of MAC-Lite payload size. The number of combinations is thus associated to the range of MAC-Lite payload size.

Combining these properties in (3), one obtains the MAC-Lite estimator

$$u_{\text{MAC-L}}^n = \arg \min_{u_{\text{MAC-L}}^n \in \Omega_{\text{MAC-L}}^n} \|y_{\text{MAC-L}}^n - u_{\text{MAC-L}}^n\|^2 + \|y_{\text{MAC-L}}^n - c_{\text{MAC-L}}^n\|^2,$$

where $c_{\text{MAC-L}}^n = f_{\text{MAC-L}}^n(k_{\text{MAC-L}}^n, p_{\text{MAC-L}}^n, u_{\text{MAC-L}}^n)$.\n
\section{Global Scheme}

Figure 7 illustrates the robust header recovery mechanisms for PHY and MAC-Lite layers at the receiver, emphasizing on the exchange of information between layers and between consecutive packets, as presented in Sections IV-D and IV-E.

\section{Simulations Results}

The improved permeable scheme (depicted in Figure 7) for the 802.11 PHY and MAC-Lite layers has been implemented in C language. A transmission system consisting of a transmitter (AP), an AWGN channel, and a receiver (T1) has been simulated. The transmitter generates the MAC-Lite and PHY packets following the format defined in Section IV. The MAC-Lite payloads contain a variable amount of randomly generated bytes. The transmitter sends data at 1 Mbps for all the simulations.

Three types of header recovery methods are considered at each layer of the receiver. The \textit{standard} decoder takes hard decision based on the received soft data. The \textit{robust} decoder exploits only the intra-layer and inter-layer redundancy through a soft decoding algorithm, neglecting the CRC properties. Finally, the \textit{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 IV-D and IV-E. Performance analysis in terms of \textit{Erroneous header Rate (EHR)} versus SNR is studied.

In Figure 8, the standard, robust, CRC-robust PHY decoders are compared. To study the performance of PHY and MAC-Lite layers independently, we assume that the \textit{Duration} field contained in the preceding MAC-Lite packet is received without error. Obviously, the standard method is outperformed by the two robust decoders. We may notice that a large part of the coding gains results of intra-layer and inter-layer properties.
An EHR of less than $10^{-5}$ is reached for SNRs above 2 and 4 dB with the CRC-robust and robust decoders respectively. With the standard decoder, an SNR larger than 14 dB is required to get a comparable EHR. At PHY layer, considerable coding gains for a relatively low complexity are thus observed when using the two robust decoders.

Figure 9 compares the coding gains obtained with standard, robust, and CRC-robust MAC-Lite decoders. To only analyze the performance at the MAC-Lite layer, the Signal field included in the current PHY packet is assumed to be correctly decoded by the receiver. At MAC-Lite layer, most of the proposed decoding technique performance comes from the CRC redundancy. An EHR lower than $10^{-5}$ is achieved for SNRs above 3, 14, and 15 dB when using CRC-robust, robust, and standard decoders respectively. At MAC-Lite layer, the CRC-robust decoder thus allows to reach considerable coding gains for a reduced complexity.

Consequently, the combination of the proposed permeable PHY and MAC-Lite layer mechanisms recovers eventually all the PHY and MAC-Lite headers from 3 dB SNR onwards. This result demonstrates the potential of replacing the classical MAC layer by the proposed MAC-Lite layer.

VI. Conclusions

In this paper, we have presented an enhanced permeable layer scheme based on a header recovery mechanism. This mechanism is particularly well-suited when combined with joint source-channel decoding techniques at APL layer. The header-correcting technique consists of a MAP header estimator jointly exploiting the structural properties of the protocol stack along with the CRC redundancy through a soft decoding algorithm. To improve the processing performance, we have extended the principle of UDP-Lite into the global protocol stack and introduced a robust 802.11 MAC layer, named MAC-Lite. Simulation results, associated to PHY and MAC-Lite layers of WiFi, show significant link budget improvement for a very low computational complexity. This mechanism is readily applicable to other layers for various transmission protocols. Adaptation to IP and UDP-Lite layers will be provided in future works.

ACKNOWLEDGMENT

The authors would like to thank Vinod Kumar for his useful comments during this work.

REFERENCES