Delimiters for Wireless LAN

Francois Le Maut

CER IBM La Gaude
06610 La Gaude, France

Abstract

In this submission we propose delimiters for Wireless LAN which meet the 802 general rules by having a Hamming distance of 4 when compared to the data, provided that the data have been subjected to a zero bit insertion process before being transmitted. [1]

Introduction

In a previous submission [2] we proposed a preamble to perform antenna diversity selection, clock and data recovery, and this without any prior knowledge of the timing at which a data packet will hit the receiver. In this submission we extend the preamble to the definition of the packet structure encompassing data rate selection, start and end delimiter selection and we also address the compatibility between scrambler and the CRC used to perform error detection on the segments send over the air.

Requirement on delimiters

In the IEEE project 802 functional requirements document there is a requirement in its paragraph 5.6.3 for a Hamming distance 4. In all most recent 802 LAN standards such as Token Ring and FDDI this requirement has been met through special violation sequences which cannot be find in the payload of the frame. 802.3 standard which get standardized prior to this requirement does not fulfill this requirement and therefore a 2 byte length field has been introduced to cover partly this problem. In wireless LAN as the error rate will be several order of magnitude above what it is over wired LAN the definition of the delimiter function becomes much more critical.

The proposed method to meet that Hamming distance of 4 is through the coding of unique patterns in the decoded bit stream, in such a way that those pattern do normally not occur in the MAC data field. HDLC like framing which uses a unique flag, or idle, or abort to delimit a frame, and which prevents these code word to appear in the MAC data field through a bit stuffing/deletion algorithm is a perfect starting point to achieve this requirement. However conventional HDLC like framing cannot meet the Hamming distance requirement because a single error can cause a false delimiter detection that can lead to undetected errors.

In addition in order to provide sufficient transitions to perform data demodulation and clock recovery there is a requirement to apply a scrambling mechanism on the MAC data field. As a result of that the number of errors could get multiplied through the descrambling process. However if a whitener is used this does not occur unless extra or missed bits are experienced. And as in the 802.11 all transmissions are performed in half duplex mode it seems more appropriate to use a whitener which gets reinitialized at each transmission instead of a self synchronizing scrambler which has the drawback of error multiplication. Up to now no solution that fulfill the 802 functional requirements has been proposed, only solutions which are similar in intends with what is implemented in 802.3 have been submitted. [3]

HDLC delimiter

802.11 communication over the air will most probably be based of segmentation of a data frame into a plurality of segments. Each of those segment may then be encapsulated into a HDLC like framing. The normal High Level Data Link control (HDLC) uses a single "FLAG" character: 0 1 1 1 1 1 1 0 as a frame delimiter. This bit pattern is excluded from the message frame between a leading and a trailing FLAG character by the "bit escaping mechanism". However, a single bit error may generate a spurious FLAG within a message frame. This means that a transmitted frame of N bytes is received as two frames of length of N-1 bytes. If this error occurs and if the 16 bits ahead of the spurious FLAG are inadvertently a correct Frame Check Sequence, an undetectable message error is received. The probability of this event to occur has been calculated [4] to be in the case of 8-bit character synchronism at the receiver:

$P_{flag} = 6.1 * 10^{-7} (n - 32)p$

where n is the number of bits per air segment and p is the bit error probability.

In the extreme case of $p = 0.5$ where only noise is received, the residual error probability is 2^{-32}.

There is also an other source of error in HDLC frame called "gain and loss of bits" which may compensate each other so that the original length is maintained. This lower residual error probability is:

$P_{bit\ comp} = 2.4 * 10^{-8} (n-16)^2 p^2$

This problem, which may occur at segment level, can be overcome by having a checksum at frame level which means that the integrity of all segments of the frame are protected independently of the HDLC process, and therefore will simultaneously authenticate the frame per remote station as this checksum will cover also the 12 bits of source and destination address.

A good candidate can be the ANSI/IEEE-Standard 32-bit CRC generated by the following degree-32 primitive polynomial:

$$X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^{8} + X^{7} + X^{5} + X^{4} + X^{2} + X + 1$$

This CRC-32 will detect:

- Any one or two bit error in a message whose length does not exceed $2^{32} - 1$ bits, because CRC-32 is primitive.
- Any error pattern having length less than 33 bits.
- All but the fraction $1/2^{31}$ of possible error patterns having a length 33 bits, i.e., only one 33-bit pattern will be undetected: the error pattern corresponding exactly to CRC-32
- All but the fraction $1/2^{31}$ of possible error patterns having a length greater than 33 bits.

In addition to make sure that no erroneous segments (but still leading to a valid overall CRC) from an other cell or through noise is not captured a frame length indicator is also included to secure the integrity of the frame.

For HDLC framing, further sources of undetectable single bit error which corrupt either a leading or a trailing FLAG causing undetectable frame errors have been under investigation but were proven to be not true.

Several alternatives have been considered to solve the HDLC weaknesses mentioned above:

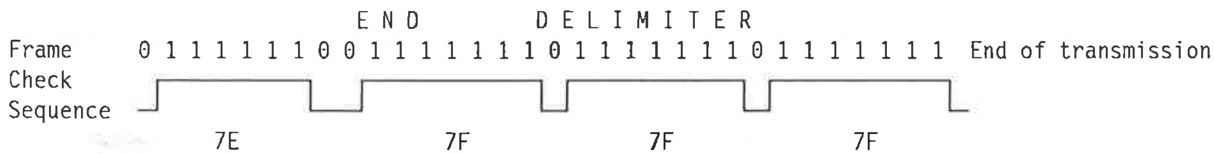
1. use only fixed frame lengths
2. use variable frame length by adding byte count information but in that case it may also happen that the counting mechanism may also be destroyed by a single error.

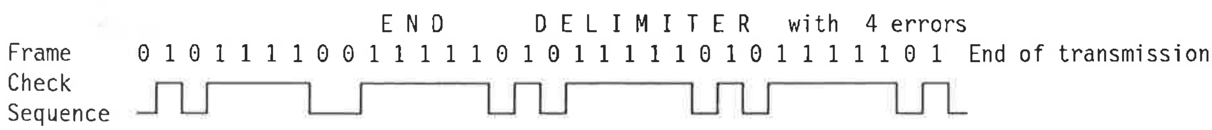
3. add and supervise additional frame delimiting characters. This last method has the capability to provide a Hamming distance of 4, as demonstrated here under.

If we assume that the basic rules of HDLC through zero insertion/deletion is performed on a MAC data field then it becomes rather trivial to put in front of it a start delimiter based on a string of FLAGs which will never be find in the data and which satisfies the Hamming distance of 4.

End delimiter

When errors in the data makes the end delimiter to be misinterpreted and that the FCS (data) happen to be good this will lead to a good frame interpretation while it is completely wrong. A way to counteract this phenomenon is through an extended end delimiter satisfying the Hamming distance of 4 which can be done very easily by extending the HDLC single byte end delimiter through 3 additional abort characters.
In addition the detection of a sequence of FLAG and ABORTs will allow to reset the radio in receive mode preparing it to receive any new transmission from any other transmitter and avoid erroneous lock on a noise.





Start delimiter

As in order to provide sufficient transitions in the data sent to the radio it is a required to whiten the data, applying the same technique for the start delimiter as to the end delimiter does not bring any advantage in term of frame delimitation, because data which look like the start delimiter can be found in the scrambled data. Therefore a bit string in front of the normal HDLC segment has been preferred because it provides the capability of even number of bit '0' and '1', which is better for the radio.

Overhead associated with bit stuffing

If the source is modeled as a random bit stream, the probability of having 5 bits at one in raw is :

$$P_5 = (.5)^5 (.5 + P_5)$$
$$P_5 = 1/62$$

which means that on average every 62-nd bit will indicate the need for stuffing thereby expanding the average segment length from 62 bits to 63 bits, giving a transmission efficiency of: 62/63 = .9841 which is equivalent to a loss of performance of about 1.5%.

However one may argue that the actual data are not random bit stream but they can be arranged to look as such through an upfront randomization process on the full MAC payload.

Compatibility between FCS and scramblers

802.11 communication over the air will most probably be based of segmentation of a data frame into a plurality of something in the range of 128 to 256-byte segments. The payload FCS characters that we propose is the one already used for normal HDLC type of protocols and which provides a 16-bit CRC generated by a polynomial whose function is:

$X^{16} + X^{12} + X^{5} + 1$

This CRC will detect:

- Any single bit errors
- Any burst not exceeding 16 bits in length
- As this polynomial sequence can be divided by (X + 1) it can detect all errors of odd weight. However this generator cannot be primitive (it is not irreducible).
- Only one 17-bit pattern will be undetected: the error pattern corresponding exactly to FCS_{16}

Therefore this 16-bit Frame Check Sequence (FCS) is designed for detecting all single, double, and three bit error patterns within a frame (Hamming distance 4).

The MAC bit stream could be whitened as proposed in [2] according to the polynomial $W(x) = X^{11} + X^{9} + 1$

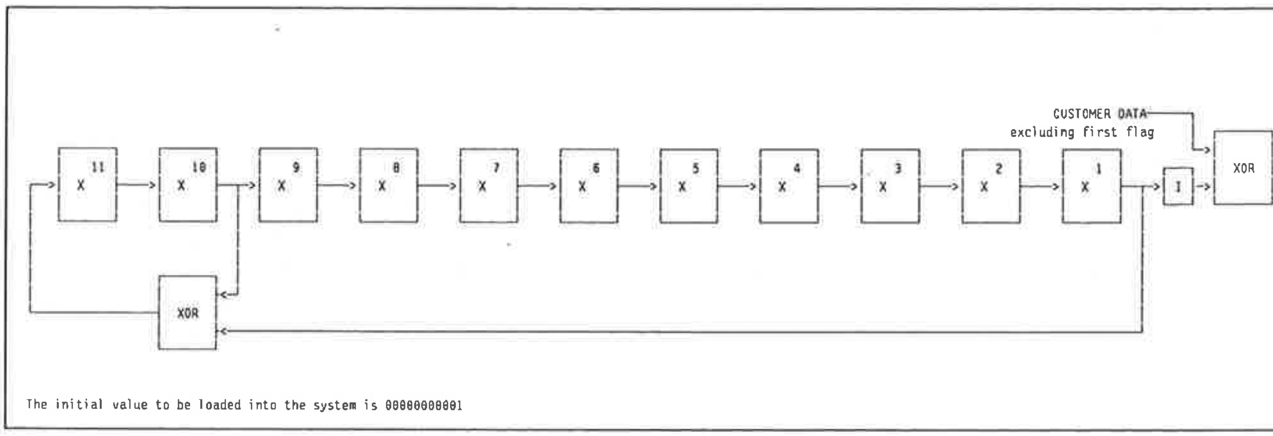


Figure 1. Withener possible implementation

It is known that choosing a whitener polynomial that has no factors in common with the CRC is the best way to avoid any interference between the two mechanisms. Error patterns that are multiples of the generator polynomial cannot be detected, but all other error patterns can, within the limits imposed by the inherent performance of the CRC generator.

By dividing CRC-16 by (X-1) we get the following quotient:

$X^{15} + X^{14} + X^{13} + X^{12} + X^{4} + X^{3} + X^{2} + X + 1$

As this quotient is irreducible and primitive as indicated in [4], this means that the CRC generator and the scrambler are prime, since the scrambler polynomial is not divisible by (X+1).

In this proposal as the scrambler process is additive (non self synchronizing) this precaution need not be observed.

PHY rate signaling field: PSRF

This field should be such that it allows future extensions to the standard as well as vendor specific extensions.

To shorten the decision process associated with the rate switching this function has to take place at the PHY level, close to the modulation and demodulation process. In addition this process should take place just after bit synch has been acquired and before equalization process if any for higher speed in order to benefit from coherent demodulation if it is made possible by the type of modulation in use.

Therefore we propose the following bit configuration before the start delimiter:

11010111 if 1 Mbps speed of operation is selected

and 11010110 if it is required to check the extended PRSF field.

PHY packet structure for Wireless LAN

```
          1    10101010
          2    10101010
          3    10101010
          4    10101010
          5    10101010     80 bits long alternating 10
          6    10101010
          7    10101010
          8    10101010
          9    10101010
          10   10101010
          1    11000100     Code Word synchronization sequence
          2    11010111    or 11010110 if there is a Rate Field for rate selection

          1    01111110    Start Frame Delimiter (at the actual data rate)

                  DA
                  SA
                  CTL
                  LGH
Whitening         D1        FCS
process           D2        Computation
                  D3
                  ...
                  Dx
                  FCS
                  FCS
          1    01111110
          2    01111111     End frame delimiter
          3    01111111
          4    01111111
```

BIBLIOGRAPHY

[1] F. Bauchot, *MAC to MAC interface* , IEEE P802.11/93-61

[2] F. Le Maut, *Preamble for frequency hopping PHY* , IEEE P802.11/93/150.

[3] W. Diepstraten *The need for MAC Data delimiters in the PHY*, IEEE P802.11/93/146.

[4] G. Funk *Message Error Detecting Properties of HDLC protocols*, IEEE Transactions on Communications, vol.Com-30, No.1, January 1982.

[5] W. Peterson, E. Weldon *Error Correcting Codes* Second Edition, MIT, 1972.