IEEE 802.11
Wireless Access Methods and Physical Layer Specifications

TITLE: Proposals for a Burst Protocol To be Used in Various Wireless Data Transport Protocols

DATE: Jan. 10-13th,1994

AUTHOR: Ed Geiger
Apple Computer
One Infinite Loop
Cupertino, CA 95014
edg@apple.com

Introduction

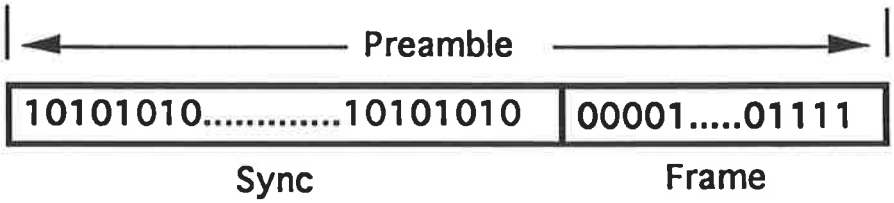
This paper purposes several burst protocols for encapsulating data to be sent over a RF Spread Spectrum PMD. These proposals give consideration to work already done in the PHY layer working group regarding preambles and headers. One proposal provides more information regarding systems employing forward error detection and correction. In addition, the paper includes work for addressing the run length problems.

1.0 Burst Format

The set of bits that is send from the transmitting node to the receiving node is called a BURST. This BURST, as shown figure 1, is composed of two parts, the PREAMBLE and the BURST Protocol Data Unit (B_PDU). The PREAMBLE is used by the receiving node to perform several functions. These functions include diversity, clock and data recovery, and identification of the start of a transmission. The B_PDU contains the information being past between the transmitter and receiver's PHY layers.

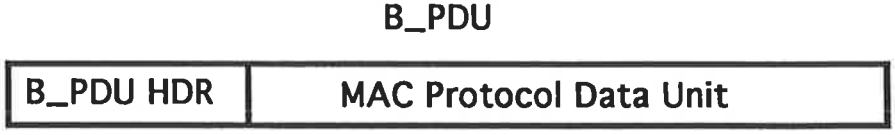
Burst Format
Figure 1

In previous meetings we have been discussing the composition of the preamble. Figure 2 shows how the preamble is divide at this time into two parts: a SYNC field and a Frame field. The SYNC field is currently defined to be 80 bit longs consisting of forty '10' patterns or symbols. The Frame field is still under study with several proposals ranging from 15-24 bits using various frame symbols.



Burst Preamble
Figure 2

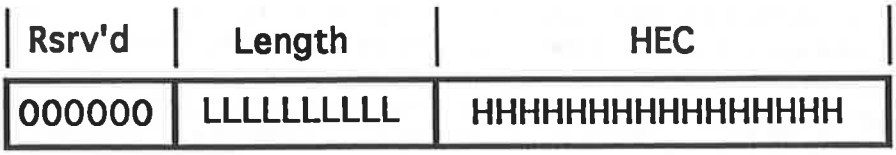
The purpose of this paper is to propose the makeup of the B_PDU. Figure 3 shows the B_PDU divided into two parts; the B_PDU Header (B_PDU HDR) and the MAC Protocol Data Unit (M_PDU). The following sections describe the makeup of the B_PDU HDR and the M_PDU.



Burst Protocol Data Unit
Figure 3

2.0 B_PDU Header

The proposed B_PDU Header is a four byte field divided into three fields as shown in figure 4. The first field is a six bit Reserved field and set to '000000'. The next field contains ten bits which is used to define the length of the user data contained in the following M_PDU field. The final field is two byte or 16 bit Header Error Check field used to protect and optional correct the B_PDU HDR.



B_PDU Header Format
Figure 4

The Reserved field is currently shown to be reserved or undefined and set to a default of all zeros. In the future, this field could be used for some of the following needs:

- Expand Length field
- Signify special handling of M_PDU
- Identify scrambling style or encrypting format
- Data rate changes

One of the advantages of using this field for indicating various B_PDU specific functions is that it is protected by the HEC thus significantly less susceptible to error in a highly erred environment.

As stated before, the Length field is used to indicate the number of bytes of data associated with the M_PDU following the header. It is assumed that this field identifies bytes but it could also be used to indicate words, long words or any other quantity. Its valid states are 000h - 3FFh, representing counts of zero to 1023. This field is formed by placing the MSB in the left most and the LSB in the right most bit positions of the field.

The Header Error Check field is a 16 bit BCH type error detection and correction field. The polynomial for this field is designed to detect any error in B_PDU HDR with an extremely high rate of success. It can also do error correction of one and two bit errors and possibly more. The polynomial F(x) used for the BCH calculation is as follows.

$$F(x) = x^{16}+x^{15}+x^{11}+x^{10}+x^{8}+x^{6}+x^{2}+x+1$$

At the transmitter, the polynomial is seeded to FFFFH prior to beginning the calculation. The remainder after performing the BCH calculation is placed in the HEC field MSB first to LSB last, just like the length field. At the receiver, the polynomial again is seeded to FFFFH prior to beginning the calculation. The BCH calculation is then performed on the received Reserved field, Length field and HEC field. If the remainder of this calculation is zero, no errors were detected in the B_PDU HDR. If the remainder is non-zero, an error exists. In the case of an error, the value in the remainder can be used to identify the what bits are erred and which might possibly be corrected.

This type of header mechanism is often used in LANs and provides several advantages. The following lists some of those advantages.

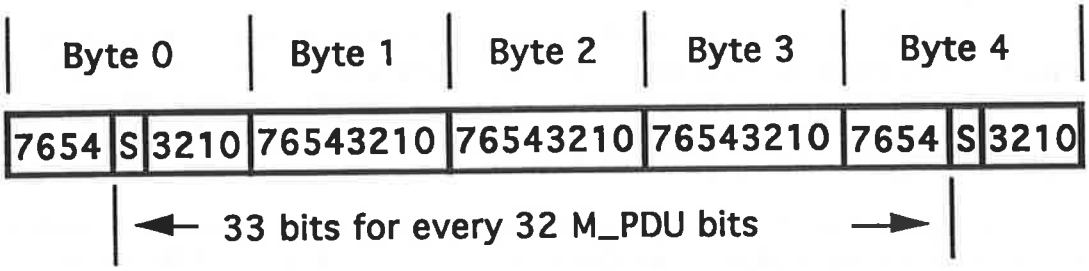
- Allows for a weaker FRAME symbol
- Strongly defines the end of a BURST
- Provides error protection for PHY layer indicators if required

The presences of the B_PDU HDR also serves as part of the framing mechanism. In this type of format, the frame symbol identifies the header and the header identifies the data field. If the frame symbol is a false positive, the B_PDU HDR will cause the false positive to become negative when the HEC is applied over the following 32 bits. Thus the B_PDU HDR extends the FRAME indication to really 48 bits, significantly decreasing the possibilities of receiving or attempting to receive a bad burst.

Another important feature of this type of header is the length field. Once a valid header is determined by a correct HEC, the end of the burst can easily be determined in time without any further uncertainty via errors in the following data stream or failure to detect an end-of-frame marker.

3.0 Addressing the Run Length Problem

As demonstrated in a previous paper (1), scrambling alone won't solve all the problems associated with reducing the run lengths of 1's or 0's in the M_PDU. Another technique for insuring a transition every so often in the bit stream is bit stuffing. Two types of bit stuffing are commonly used to insure transitions, variable rate bit stuffing and fixed rate bit stuffing. The committee has already seen some variable rate bit stuffing schemes, the proposed one here is a fixed rate bit stuffing scheme.

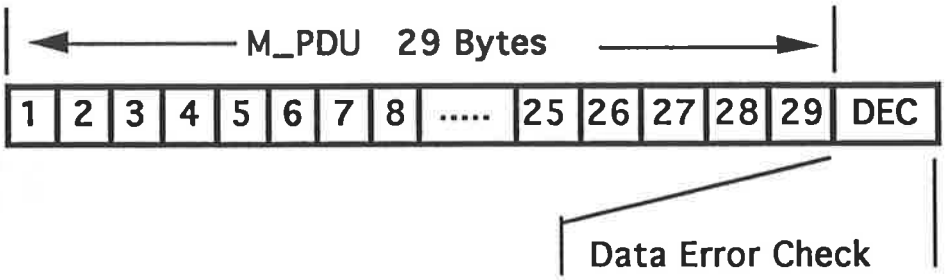


Example of Fixed Bit Stuffing
Figure 5

Figure 5 shows an example of fixed rate bit stuffing. The example shows the first 5 bytes of the M_PDU. The first stuff bit is inserted after bit 4 in byte 0 and its value is the inversion of bit 4. This process is repeated during byte 4 and continues for the full length of the burst, adding a bit to every 32 bits in the M_PDU. This guarantees a transition every 33 bits in the M_PDU whose worst case pattern is 0FFFFFFFF0000000h. It provides two transitions for strings of FFFFFFFFh or 00000000h. Radios have been designed to maintain the NRZ bit mask (avoid bit stretching) up to runs of 32 bits or less. Note: The receiver need not do any processing of the stuff bit except to remove it from the data stream before passing the data up to the MAC Layer.

4.0 Fragmentation of the M_PDU

One of the proposed Data Transport Protocols uses a fragmentation process to segment the M_PDU portion of the Burst into smaller pieces of data. These pieces are called fragments. They consist of 29 bytes of the M_PDU followed by a 2 byte Data Error Check (DEC) field as shown in figure 6. The DEC field is a 2 byte field containing the remainder of BCH calculation. The BCH polynomial is identical to the one used in the header. (See section 2.0 for information on the F(x) polynomial and how to implement it).



29 Byte Fragmentation Process
Figure 6

The DEC field can be used to perform two functions. First it can detect all single and most 2 bit errors up to burst errors of 16 bits. This could be used in schemes which only re-transmit portions of erred data rather than complete bursts. It is also useful in schemes applying forward error correction to minimize the amount of data to be corrected. Its second function can be to correct all single bit errors and most 2 bit errors. Error correction sometimes takes a lots less time to apply then re-transmission, especially in an interfered environment.

5.0 Conclusion

This paper suggest several things which might be added to the burst protocol currently be worked on in the PHY Working Group. It is hope that some of these simple-to-implement features will improve the overall performance of the a Frequency Hopping LAN by providing the rigorous type of operating functions needed to help overcome the difference between the media's expected 10^{-5} bit error rate and the 10^{-9} which might be required by some MAC protocols.

References

1. Ed Geiger, "A Look at Some Scrambling Techniques Used in Various Data Transport Protocols", DOC IEEE 802.11-93/216.