I move that the FH PHY sub-group adopts the FH packet formatting method presented in submission 94/069 with 32-bit block sizes and stuffing/inversion beginning with the first block following the PLCP header.

Details of Algorithm

![Packet Formatting Illustration]

Figure 1. Stuffing/Inversion Packet Formatting Illustration
Stuffing Algorithm: If stuff bit = 1 = next block is inverted; 0 = not inverted

Input parameter: number_of_MPDU_bytes; /* can be even or odd */

number_of_blocks_in_packet = truncate{(number_of_MPDU_bytes + 3) / 2}; /* no padding is necessary for number of bytes not multiple of 4 */

Nhdr = 32; /* Number of bits in header */

Read in header {b(1),...,b(Nhdr)}; /* b(1) is first bit in */

accum = 2 [Sum{b(1),...,b(Nhdr)}] - (Nhdr); /* calculate bias in header */

Transmit {b(1),...,b(Nhdr)}; /* no stuffing on header */

For n = 1 to number_of_blocks_in_packet
    
    {b(0) = 0; /* b(0) is the stuff bit */
     N = min(4, # of bytes remaining) * 8; /* N= block size in bits */
     Read in next block {b(1),...,b(N)}; /* b(n) = 0, 1 */
     bias_next_block = 2 [Sum{b(0),...,b(N)}] - (N+1); /* calculate bias with b(0) */
     If {[accum * bias_next_block > 0] then /* if accum and bias of next block has the same sign, then invert block */
         
         Invert {b(0),...,b(N)};
         bias_next_block = - bias_next_block;
         
     }
     accum = accum + bias_next_block;
     transmit {b(0),...,b(N)}; /* b(0) is first bit out */
    }

Figure 2. Stuffing/Inversion Packet Formatting Pseudo-Code