IEEE P802.11
Wireless LANs

A Proposed Scrambling
Vector for the CCK Blockcode

Date: September 11, 1998

Author: Chris Heegard, Matthew B. Shoemake, and Stanley Ling

heegard@alantro.com, shoemake@alantro.com, sling@alantro.com

Abstract

An enhanced scrambling vector for the CCK blockcode is proposed for inclusion in the 802.11 high rate PHY standard. This scrambling vector, described by the mapping vector [5,4,7,4,0,2,3,3], provides more robust system performance and improvement in the multipath environment. The proposed scrambling vector is fully compatible with the CCK block code proposed in [1].

1   Introduction

The CCK blockcode that has been proposed for the 802.11 high rate PHY at the 2.4GHz band has a code structure that consists of two components (1) a linear block code and (2) a "cover" or "scrambling" vector. The block code has parameters: $(n = 8, k = 4)$ where n is the blocklength (in QPSK symbols), k is the information length (in 4-ary symbols); this code has 256 codewords and a minimum (normalized) Euclidean distance of 4 (the Hamming distance is also 4). The code has an asymptotic coding gain of 3dB and a realized coding gain of about 2dB. The code has a weight distribution: $1 + 24 X^4 + 16 X^6 + 174 X^8 + 16 X^{10} + 24 X^{12} + X^{16}$; this means that each codeword has 24 nearest neighbors (at distance 4), 16 neighbors at distance 6, etc.

The CCK blockcode has the following nice structural properties:

✓ A codeword that is rotated by 90 degrees in each of the eight co-ordinates is a codeword.

✓ There exists a subcode of 64 codewords that produce the complete set of codewords by the application of rotations of 0, 90, 180 and 270 degrees.

✓ The structure of the generator matrix for the subcode allows for a "fast" algorithm for determining the correlation with a received word.

These properties of the code make efficient the computation of a maximum likelihood estimate for the transmitted codeword at the receiver. The decoder:

✓ Applies a fast algorithm to determine the 64 correlations

✓ Finds the largest (in magnitude) correlation

✓ Combines the best correlation result with the phase of the correlation to estimate the data

The other aspect of the code is the scramble vector. The choice of scrambling vector has no effect on the complexity of the decoding of the blockcode but is critically important in terms of maximizing robust tolerance to channel multipath distortion. The choice of scrambling vector was left as a flexible issue in the July 802.11 meeting.

The scrambling vector controls a variation on the mapping of the symbols of the codeword onto the QPSK constellation. General collections of such mappings are shown in Figure 1. In this figure, a choice of 8 possible mapping, indexed by a symbol s is shown. In the original proposal [1], only the values of s = 0 and s = 2 are used. The scrambling vector in that case was $[0,0,0,2,0,0,2,0]$. A new vector [5,4,7,4,0,2,3,3] has been found which has more robust multipath tolerance. As shown in Figures 2 & 3, the new code shows a better noise immunity at the output of the channel. In Figure 2, 250 multipath channels were randomly generated according to [2], and the free Euclidean distance of the code, at the output of the channel, was computed. In a large majority of the cases, the new scramble vector demonstrated a larger free distance. Figure 3, which corresponds to a 200ns channel, shows a larger advantage for the new vector.

The increase in hardware complexity associated with implementing the scrambler/descrambler mapper for the proposed scrambling vector [5,4,7,4,0,2,3,3] versus [0,0,0,2,0,0,2,0] is trivial. The minimal increase in hardware is justifiable relative to the enhanced system performance in the multipath environment.

Based on this study, it is proposed that the new scrambling vector be adapted for the high rate, (TGb) PHY standard.
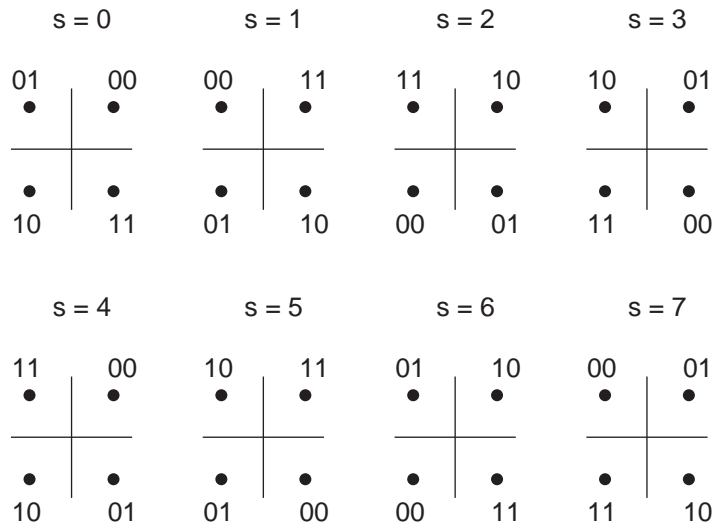
s = 0	s = 1	s = 2	s = 3
01 00	00 11	11 10	10 01
• •	• •	• •	• •
• •	• •	• •	• •
10 11	01 10	00 01	11 00

s = 4	s = 5	s = 6	s = 7
11 00	10 11	01 10	00 01
• •	• •	• •	• •
• •	• •	• •	• •
10 01	01 00	00 11	11 10

Figure 1: QPSK Mapping for Each Scrambling Value s

Histogram of Relative Multipath Loss
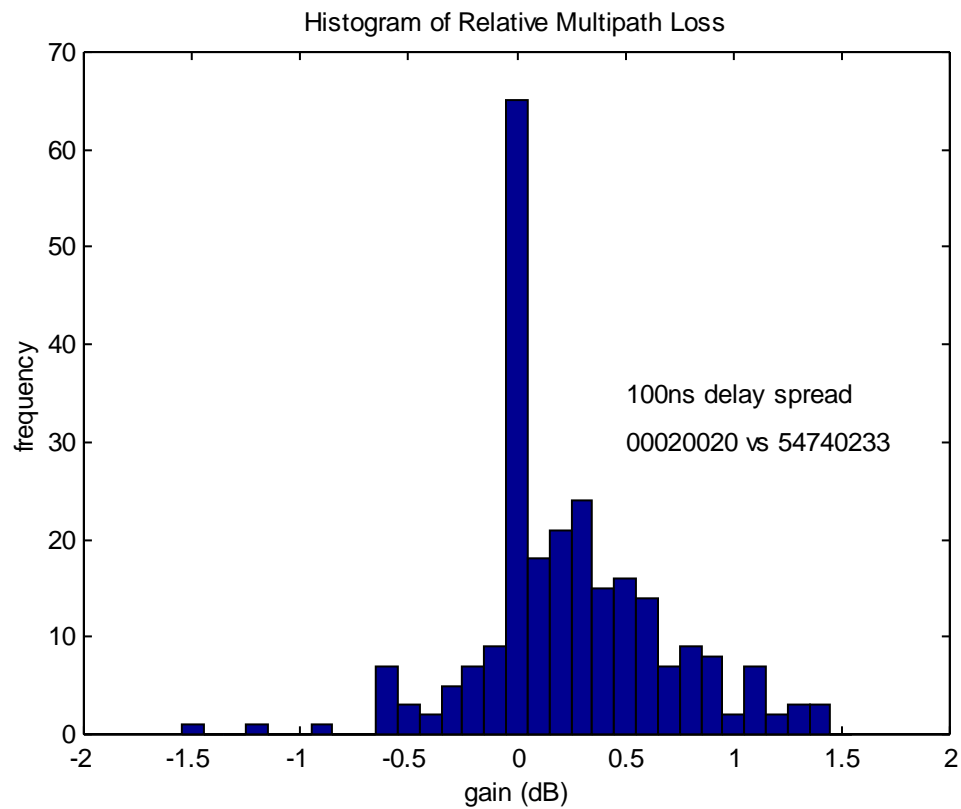
100ns delay spread
00020020 vs 54740233

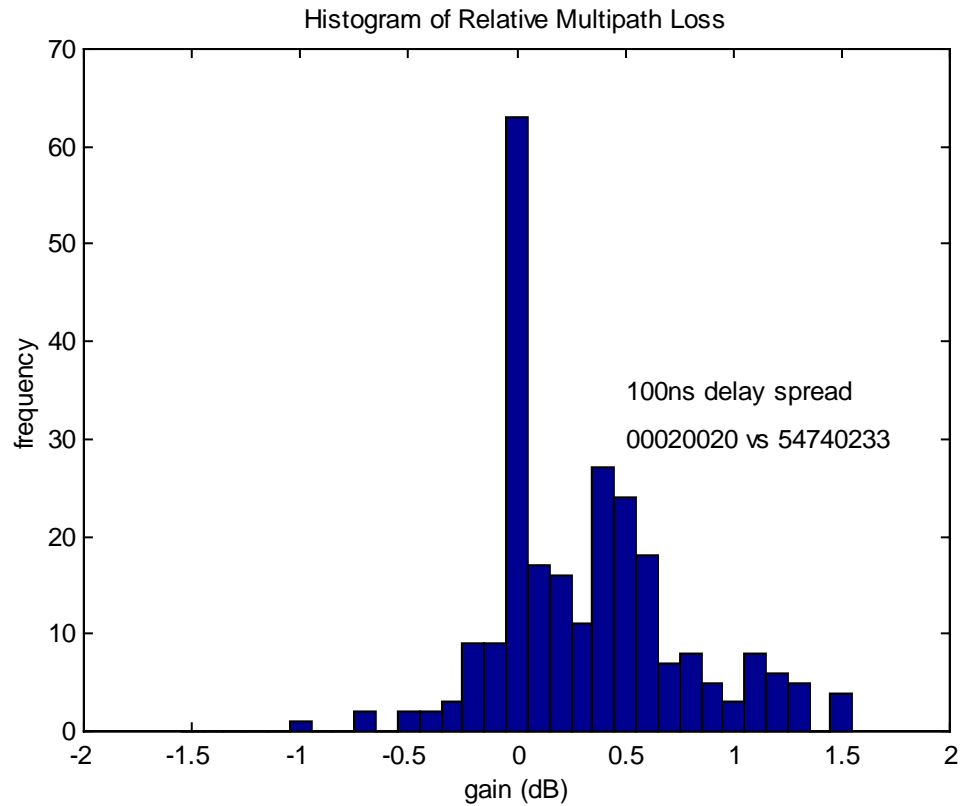Figure 2: Multpath Loss Comparison at 100ns

Figure 3: Multpath Loss Comparison at 200ns

2   References

1   C. Andren/M. Webster and J. Boer/R. van Nee, "Introducing the Harris-Lucent Compromise Proposal for TGb," Harris Semiconductor and Lucent Technologies, IEEE P802.11-98/246, July 1998.

2   IEEE P802.11-97/157r1, Nov. 1997.