Coding in 100VG-AnyLAN

A 5B/6B coding scheme in which five data bits are encoded into six-bit codewords is used in conjunction with offsetting the data on different channels by three bits in quartet signaling. It provides the level of error detection necessary, produces a signal balanced within narrow limits, and restricts strings of consecutive 0s or 1s to a maximum length of 6. It is also efficient.

by Simon E. C. Crouch and Jonathan Jedwab

Coding of data before transmission has three main purposes. First, it can ensure that the transmitted signals are dc balanced, that is, that there are equal numbers of 0s and 1s over an extended period of time. This is important in transformer-coupled systems, which are designed to prevent ground loops. Second, coding can aid clock synchronization by minimizing the length of strings of consecutive 0s or 1s. This ensures that there is a high density of signal transitions. Third, together with cyclic redundancy checks (CRCs), it can enable errors to be detected. There is an IEEE Project 802 functional requirement that any three bits in error within a data frame must be detected. There is also a need to detect burst errors; any coding system must not compromise the detection provided by the CRC.

The 100VG-AnyLAN team chose a 5B/6B coding scheme in which five data bits are encoded into six-bit codewords. This is used in conjunction with offsetting the data on different channels by three bits in quartet signaling. It provides the level of error detection necessary, produces a signal balanced within narrow limits, and restricts strings of consecutive 0s or 1s to a maximum length of 6. At the same time, it is efficient, adding only 20% to the data load, compared with the 100% produced by the 1B/2B Manchester coding used in 10Base-T. The 5B/6B code is also effective when the four data channels are multiplexed for transmission on STP or fiber-optic cables.

This article describes the design of the 5B/6B block code used in 100VG-AnyLAN and more generally in the IEEE 802.12 proposed local area network standard[1] and explains the reasons behind its design.

Why Code?

The method used to transmit digital data has to take account of the physical constraints that the transmission medium imposes upon the transmission system. The physical media used by 100VG-AnyLAN include unshielded twisted pair (UTP) cable, shielded twisted pair (STP) cable, and optical fiber. The demands imposed by the use of UTP cable mean that to transmit at 100 Mbits/s (even when using quartet signaling), digital data has to be coded before it is transmitted. The design of the code has to take account of various constraints imposed by implementation considerations and by the need to be compatible with other data transmission systems. In particular, the 5B/6B 100VG-AnyLAN code scheme was designed to be compatible with the packet structure and error detection capabilities defined by the IEEE 802.3 standard used in Ethernet local area networks and by the IEEE 802.5 standard used in token ring local area networks.

Block Codes and Error Detection

Generally speaking, a code is a mapping from an alphabet of symbols to a set of sequences of symbols from some other alphabet. So, for example, we could define a code from the alphabet {a,b,c,...,z} to the set of sequences of zeros and ones by: a → 010, b → 0110, c → 01110, and so on. For another example, the representation of decimal numbers as their binary equivalents can be regarded as a code. This idea of a code should not be confused with the notion of a cipher, which is a code that is specifically designed to hide the meaning of transmitted data from those not meant to understand it. The codes that we are using are designed to be easy to decode and are capable of detecting errors in transmission. For the mathematical background of coding theory, see reference 2. For an engineering perspective, see reference 3.

A block code is a specific type of code in which every codeword has the same length. In a 5B/6B block code, each element of an alphabet of 32 different symbols (which itself can be represented by a 5-bit number) is encoded as a 6-bit codeword.

The major physical constraint that affects the use of a coding scheme is the need to maintain dc balance, that is, the spectral content of the transmitted signal should have no zero-frequency component. In mathematical terms, this translates into requiring that the difference between the number of ones and the number of zeros transmitted at any time must be kept as close to zero as possible.

Clearly, a simple way to get a balanced code is to represent a data character of 0 by the codeword 01 and a data 1 by the codeword 10. The number of 1s transmitted will then equal the number of 0s transmitted at the end of each codeword. Unfortunately, this is also a very inefficient way of achieving dc balance because it requires the physical transmission of two code bits for every data bit sent. The 100VG-AnyLAN team chose to implement a 5B/6B code because it gives a suitable trade-off between efficiency and cost of implementation.

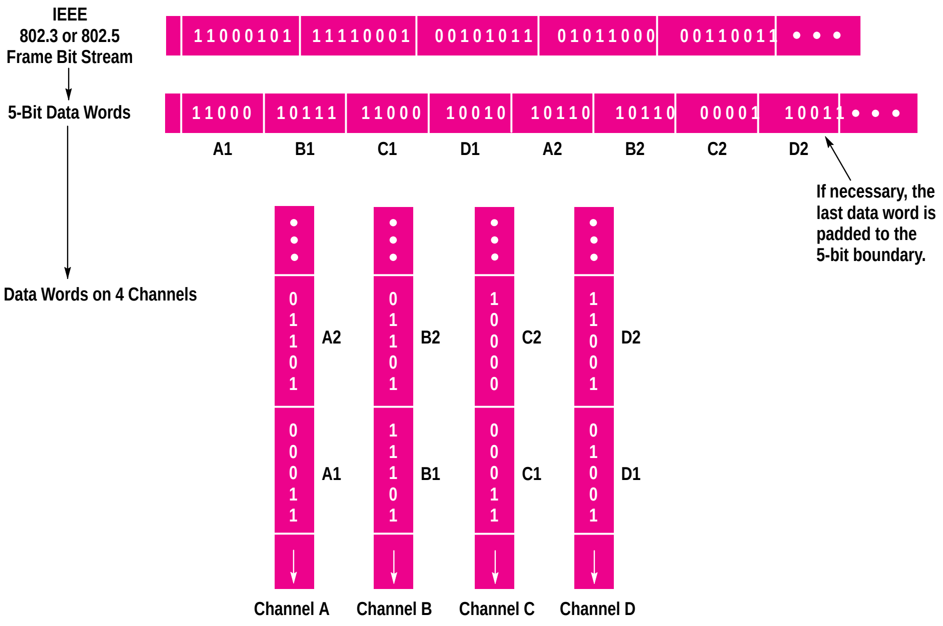


Fig. 1. Distribution of a data stream on four channels for quartet signaling.

The major compatibility constraint imposed upon the design of the 100VG-AnyLAN 5B/6B code was that it had to be compatible with the MAC frame formats defined by the IEEE 802.3 and 802.5 standards and in particular with the error-detection capabilities offered by the cyclic redundancy check (CRC) built into those formats. (See "Polynomial Arithmetic and Cyclic Redundancy Checks" on page 31). CRC-32, the CRC defined by both IEEE 802.3 and 802.5, detects up to three single-bit errors occurring anywhere in a frame, or a single burst error of up to 32 bits in the frame. The protocol will then discard that packet as flawed. (A data packet consists of the MAC frame delimited by a preamble and start and end delimiters.) The 100VG-AnyLAN team had to design the 5B/6B code so that similar error-detection characteristics will be maintained when 802.3 or 802.5 frames are coded and transmitted in parallel using quartet signaling. In particular, the requirement to detect up to three single-bit errors anywhere in a frame is a compulsory requirement of IEEE Project 802,[4] so the new IEEE 802.12 standard had to meet this demand.

In the next two sections we describe the process of coding the data frame and the error detection capabilities of 100VG-AnyLAN.

Coding a Data Packet

Data from the IEEE 802.3 or 802.5 frame is divided into 5-bit data blocks (with padding added at the end, if necessary) and is distributed between the four data streams transmitted using the quartet signaling scheme (see Fig. 1).

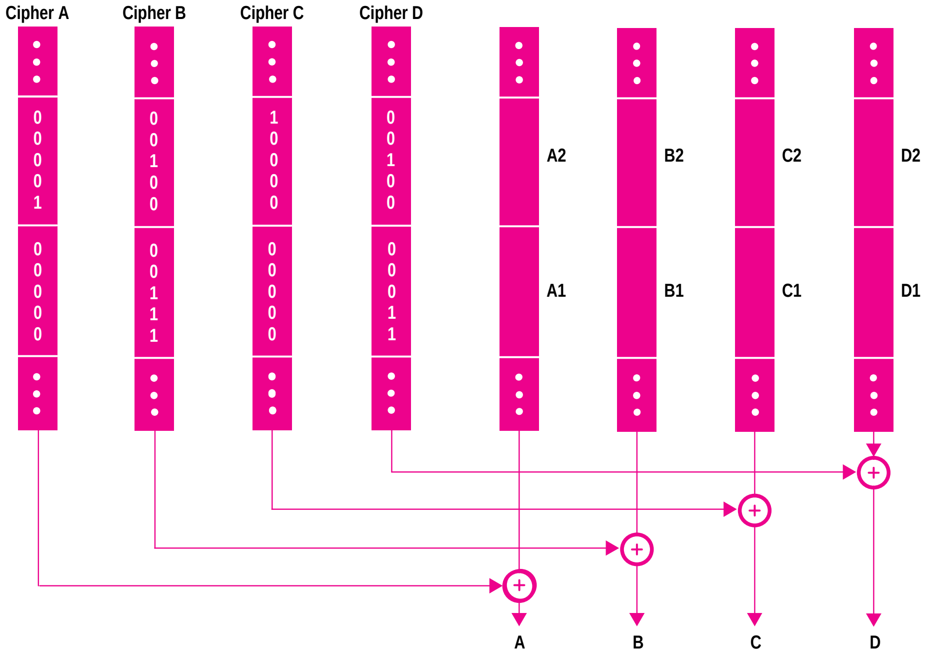


Fig. 2. The data words are XORed with pseudorandom stream ciphers.

Each data stream is exclusive-ORed with a stream of pseudorandom bits produced by stream ciphers (see Fig. 2). The ciphered data blocks on the four streams are then coded according to the following 5B/6B code table, as shown in Fig. 3.

Data	Code	Data	Code Pair
00001	101100	00000	001100 110011
00011	001101	00010	100010 101110
00101	010101	00100	001010 110101
00110	001110	01011	000110 111001
00111	001011	01100	101000 010111
01000	000111	01110	100100 011011
01001	100011	10000	000101 111010
01010	100110	10010	001001 110110
01101	011010	10101	011000 100111
01111	101001	10111	100001 011110
10001	100101	11010	010100 101011
10011	010110	11110	010010 101101
10100	111000		
10110	011001		
11000	110001		
11001	101010		
11011	110100		
11100	011100		
11101	010011		
11111	110010		

In the second column, all the 6-bit codewords are balanced, that is, there are equal numbers of 0s and 1s. There are not enough balanced 6-bit codewords to code all possible 5-bit data symbols, so twelve data symbols are coded by a choice of two different 6-bit codewords, one of weight two (two 1s and four 0s), and one of weight 4 (four 1s and two 0s). The two codewords are used in the following fashion, independently in each stream:

- For the first data symbol that codes to an unbalanced codeword, the weight-two codeword is chosen.
- When the next unbalanced codeword occurs, the weight-four codeword is used.
- Weight-two and weight-four codewords continue to alternate whenever an unbalanced codeword occurs.

The observant reader will notice that in all but one of the code pairs, the alternative unbalanced codewords are logical negatives of each other. This is not true of the second pair, for reasons having to do with the error properties of the code.

At the end of the stream, one of two end delimiters is used, independently in each stream (see Fig. 4). When the end of the stream is reached, if the next unbalanced codeword is due to have weight 2 (according to the rules above), end delimiter ED2 is used. If it is due to have weight 4, ED4 is used. The invalid packet marker (IPM) is used by repeaters to mark errored packets for disposal or further processing.

After coding, the data streams are offset by 3 bits with respect to each other before transmission (see Fig. 3). Again, the reason for this has to do with error-detection properties and will be explained below.

The actual transmitted bit sequence down each channel looks as shown in Fig. 4. The start and end of packet markers are designed to maintain error detection capabilities. The use of alternative end delimiters (effectively an extra parity check) is an essential part of the error detection scheme of 100VG-AnyLAN.

Multiplexing

Another design constraint of the 5B/6B coding scheme was that it should behave well when the four data streams are multiplexed onto fewer channels for networks using STP or fiber-optic cables. As noted in the article on page 18, STP and fiber-optic PMDs pass four parallel streams of data through a multiplexer, which combines the four codeword streams into one stream, codeword by codeword (Fig. 5).

When combined in this way, the physical and error protection capabilities of the 5B/6B code are maintained, as explained in the next section.

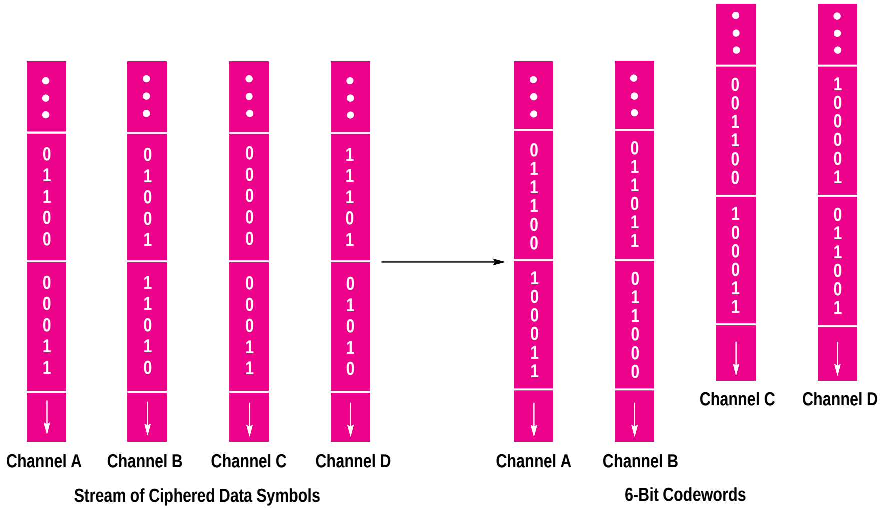


Fig. 3. Conversion of 5-bit data symbols to 6-bit codewords. The three-bit offset of channels C and D is for error detection.

IEEE 802.3 and 802.5 Frame Formats

The IEEE 802.3 frame (Fig. 1) consists of a 48-bit destination address, a 48-bit source address, a 16-bit length field, and then a data field ranging between 368 and 12,000 bits in length and consisting of data organized into octets. This is followed by a 32-bit cyclic redundancy check (CRC).

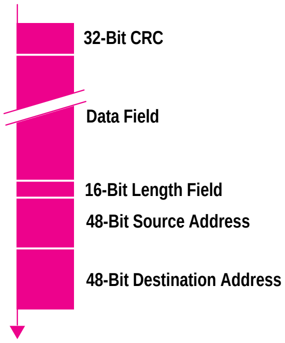


Fig. 1. IEEE 802.3 frame structure.

The IEEE 802.5 token-ring frame (Fig. 2) consists of an 8-bit access control field (not used in the 802.12 standard), an 8-bit frame control field, a 48-bit destination address, a 48-bit source address, between 0 and 240 bits of routing information, and then a data field ranging between 0 and 36,016 bits in length and consisting of data organized into octets. This data field is followed by a 32-bit CRC.

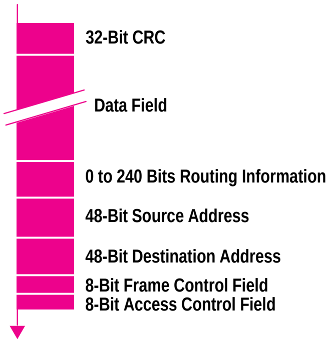

32-Bit CRC
Data Field
0 to 240 Bits Routing Information
48-Bit Source Address
48-Bit Destination Address
8-Bit Frame Control Field
8-Bit Access Control Field

Fig. 2. IEEE 802.5 frame structure.

The CRC provides two types of protection for an IEEE 802.3 or 802.5 frame:

- Any three single-bit errors occurring anywhere in the frame are detected.
- Any burst of errors for which the distance between the first corrupted bit and the last corrupted bit is less than or equal to 32 bits is detected.

IEEE Project 802 requires that any transmission scheme developed under its aegis can perform the first type of error detection, that is, that at least three single-bit errors can always be detected.

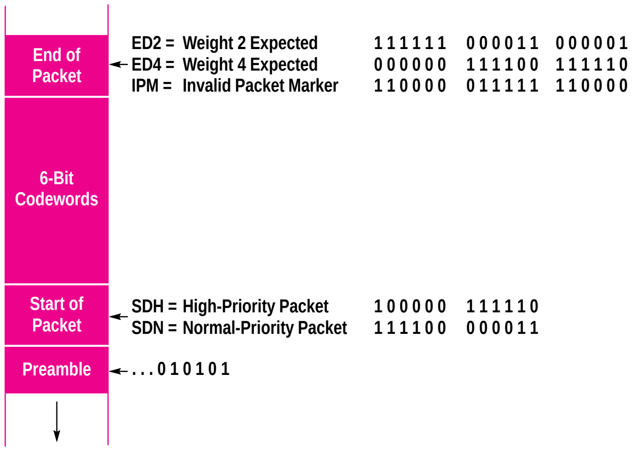

End of Packet
ED2 = Weight 2 Expected 111111 000011 000001
ED4 = Weight 4 Expected 000000 111100 111110
IPM = Invalid Packet Marker 110000 011111 110000
6-Bit Codewords
Start of Packet
SDH = High-Priority Packet 100000 111110
SDN = Normal-Priority Packet 111100 000011
Preamble
...010101

Fig. 4. Transmitted bit sequence on each channel.

Properties of the Coding Scheme

The choice of 5B/6B code, together with the alternation of unbalanced codewords and the use of the alternative end delimiters, ensures that IEEE 802's requirement is met: up to three single-bit errors occurring anywhere on the four channels within a data frame are detected. This is not a trivial matter to confirm, because the use of a code means that a single-bit error in the transmitted bit stream may cause many more than one single bit to be in error in the data stream after decoding. A substantial portion of the code design effort was dedicated to limiting the number of data bits affected by a single code bit in error.

It is not difficult to see how the use of the alternation rule for choosing unbalanced codewords together with the use of the alternative end delimiters leads to the detection of all triples of single-bit errors, when those errors occur in distinct codewords on a single channel. Fig. 6 shows several cases in which three single errors are detected by the alternation rules and the use of end delimiters.

A second major design thrust was to make sure that burst error-detection properties inherited from CRC-32 are not compromised in 100VG-AnyLAN. Because we are now talking about a system that can transmit over four parallel streams as well as a single stream, we have to be careful to define what we mean by a burst error. We decided that the clearest case could be made by considering a burst error to be any error caused by arbitrary corruption across all the streams for a certain number of bit periods (see Fig. 7). So, for example, for a burst error of seven code bit periods, there is a block seven code bits long by four code bits wide (corresponding to the four channels), within which any given code bit may or may not be inverted.

The first thing we realized was that if all the codewords were transmitted synchronously on all four channels, a burst error of length 2 (corrupting eight code bits) could cause trouble if it occurred at a codeword boundary (see Fig. 8). The error could corrupt eight code words and thus eight data blocks, corresponding to forty data bits. This is well above the 32 protected by the CRC. It would therefore require a remarkably good coding scheme to protect against even such a short error burst.

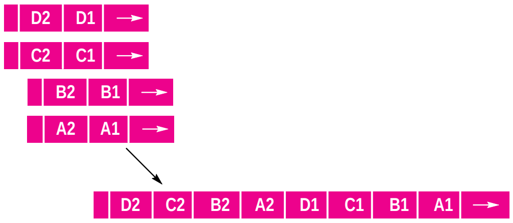
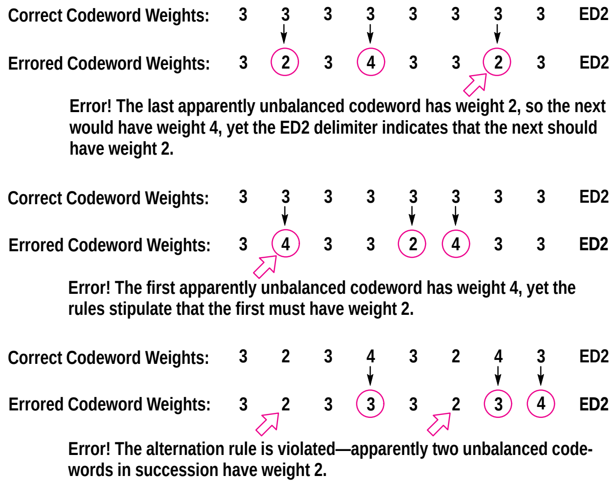


Fig. 5. Multiplexing four streams into one.


Correct Codeword Weights: 3 3 3 3 3 3 3 3 ED2
Errored Codeword Weights: 3 2 3 4 3 3 2 3 ED2
Correct Codeword Weights: 3 3 3 3 3 3 3 3 ED2
Errored Codeword Weights: 3 4 3 3 2 4 3 3 ED2
Correct Codeword Weights: 3 2 3 4 3 2 4 3 ED2
Errored Codeword Weights: 3 2 3 3 3 2 3 4 ED2

Error! The last apparently unbalanced codeword has weight 2, so the next would have weight 4, yet the ED2 delimiter indicates that the next should have weight 2.

Error! The first apparently unbalanced codeword has weight 4, yet the rules stipulate that the first must have weight 2.

Error! The alternation rule is violated—apparently two unbalanced codewords in succession have weight 2.

Fig. 6. Detection of errors through the alternation rule and end delimiters.

The solution to this problem is to offset the codewords on two of the channels relative to the other two. Fig. 9 clearly shows that with such an offset, even an error burst of length four (corrupting 16 code bits) can only corrupt the code bits of at most 6 codewords. This corresponds to 6 data blocks (30 data bits) and so will be detected automatically by the burst error properties of the CRC. Thus, even with an arbitrary code, any burst error of length four is detected.

Substantial theoretical work led us to design a code on top of this offset scheme that extends the burst error-detection capability to seven code bit periods (arbitrarily corrupting 28 code bits), while still meeting the requirement to detect three separate single-bit errors (see Fig. 10). The triumph of theory over trial and error should not be underestimated here—there are approximately 10^{45} choices of 5B/6B code!

When multiplexed onto a single channel for STP or fiber-optic solutions, the error detection capabilities of the 5B/6B code are enhanced. The single-bit error protection remains at three while the burst error-detection capabilities are increased to 34 code bits.

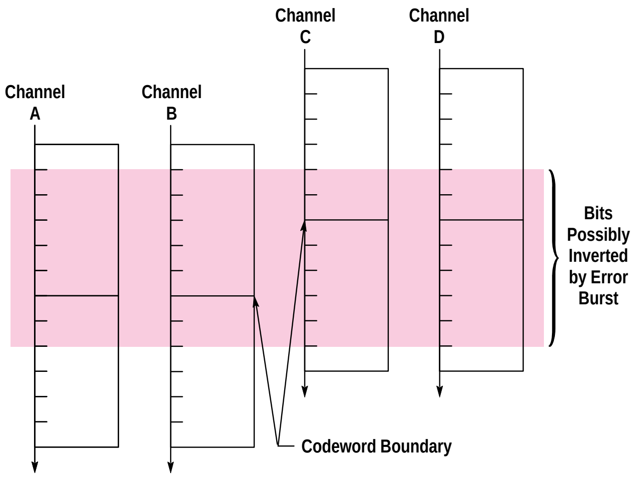


Fig. 7. Illustration of a burst error with a length of seven code bit periods.

Polynomial Arithmetic and Cyclic Redundancy Checks

The calculation of cyclic redundancy checks (CRCs) depends upon the arithmetic of modulo 2 polynomials. A modulo 2 polynomial is an expression of the form $a_0 + a_1x + a_2x^2 + ... + a_nx^n$, where the coefficients a_0, a_1, ... ,a_n are integers modulo 2, that is, they can take values of 0 or 1. The coefficients obey regular modulo 2 arithmetic as follows:

Addition

+	0	1
0	0	1
1	1	0

Multiplication

×	0	1
0	0	0
1	0	1

Electrical engineers will recognize the addition operation as the XOR operation defined on binary states.

The following are examples:

- $(1+x^2+x^4) + (1+x+x^4) = x+x^2$
- $(1+x^2+x^4)\ \ (1+x+x^4) = 1+x+x^2+x^3+x^5+x^6+x^8$.

Division of modulo 2 polynomials is done in exactly the same way as it is for ordinary polynomials, remembering that the coefficients obey modulo 2 arithmetic as defined above.

A very important operation in the calculation of CRCs is the calculation of the remainder R(x) when one polynomial M(x) is divided by another G(x). R(x) is uniquely defined by M(x) = G(x)K(x) + R(x), where the degree of R(x) is less than the degree of G(x). The degree of a polynomial is defined as the largest value of n for which the coefficient a_n is nonzero, so, for example, $1+x^2+x^5$ has degree 5.

Although this sort of polynomial division may look formidable, there are very efficient means of calculation based on shift registers.

CRCs are calculated on a stream of data by assuming that the data represents the coefficients of some modulo 2 polynomial. So, given a stream of data n bits long, the first bit can be considered as the coefficient a_{n-1} of x^{n-1}, the second bit as the coefficient a_{n-2} of x^{n-2}, the (n – 1)th bit as the coefficient a_1 of x and the nth bit as the constant term a_0.

Roughly speaking, the 32 bits of the CRC are defined to be the polynomial remainder R(x) when the polynomial defined by the data M(x) is divided by a standard polynomial:

$$G(x)= 1+x+x^2+x^4+x^5+x^7+x^8+x^{10}+x^{11}+x^{12}+x^{16}+x^{22}+x^{23}+x^{26}+x^{32}.$$

Slight modifications are made for implementation reasons, but as far as the error properties are concerned, this is what is calculated.

The CRC bits are appended onto the data. When the data is received, the CRC is calculated and compared with the received CRC. If there is a difference, the data is known to have been corrupted.

When a corruption occurs in transmission, a number of bits are inverted. Let the errored bits define the coefficients of a polynomial E(x); for example, if errors occur at bit positions p and q, the polynomial will be $x^{n-p} + x^{n-q}$. Since the operation of calculating CRCs is linear, the error is detected if and only if E(x) is not exactly divisible by G(x). CRC-generating polynomials such as G(x) are chosen precisely to detect as many polynomials like E(x) as possible.

In addition to the error detection, the 5B/6B code has other properties that are highly pertinent to physical transmission. The first of these is run length—the maximum number of consecutive zeros or consecutive ones transmitted on any code stream. This is important in transmission systems where some form of clock information is recovered from the data stream, since clock recovery usually depends on receiving a

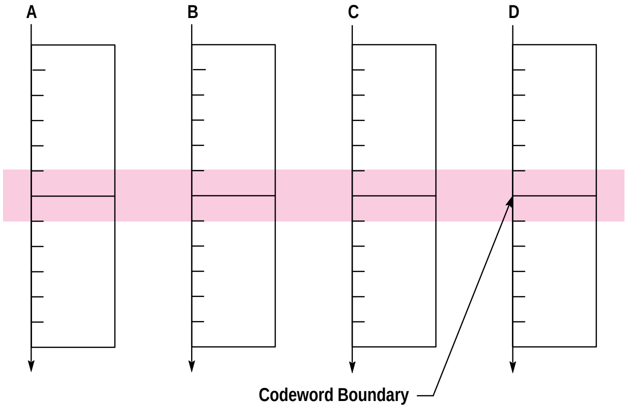


Fig. 8. A burst error across synchronous channels. A burst of length 2 corrupts up to eight code bits.

reasonable density of signal transitions. For the 100VG-AnyLAN 5B/6B code this maximum run length is 6, whether on four channels or on one.

Another physical attribute of the 5B/6B code is the running digital sum (RDS)—the difference between the number of zeros and the number of ones in the transmitted bit stream since the code transmission began. A bounded RDS ensures bounded baseline wander. For the four-pair scheme, the RDS is bounded between −5 and +3. For the single-stream scheme, the bounds are −11 and +3. The bounds are not symmetrical because of the alternation rule—a weight-2 codeword is always sent first on a channel when there is a choice of unbalanced codewords

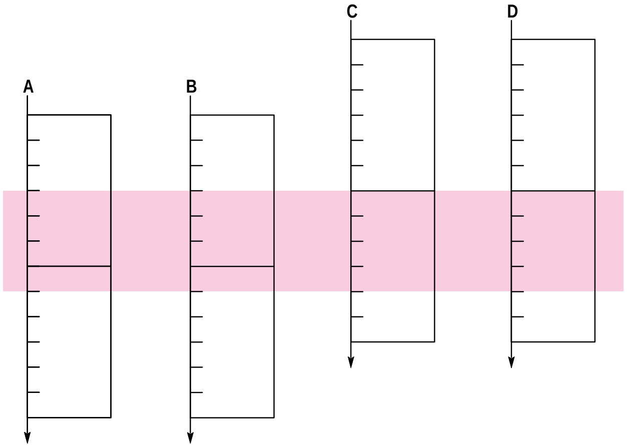


Fig. 9. Detection of a burst of length 4. A burst of length 4 will always be detected because of the offset between channels.

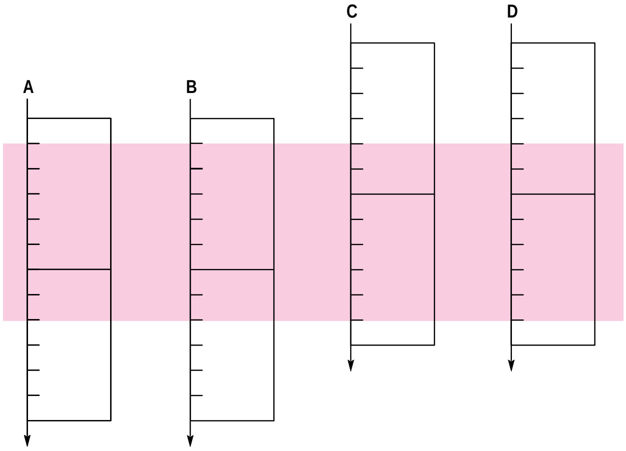


Fig. 10. Detection of a burst of length 7. A burst of length 7 is always detected because of the offset and the coding scheme.

In summary then, the choice of 5B/6B code, with the alternative end delimiters working on top of the offset transmission scheme, allows 100VG-AnyLAN to meet IEEE 802's requirements. It will detect three separate single-bit errors anywhere in the frame and is capable of detecting substantial burst errors for all the media choices while maintaining favorable physical transmission attributes.

Acknowledgments

The authors would like to thank all of the team members at HP Laboratories Bristol and the Roseville Networks Division for their individual contributions, which made the team effort so fulfilling. Special thanks should go to Pat Thaler and Dan Dove at RND and to David Cunningham, Alistair Coles, Miranda Mowbray, David Coker, and Michael Spratt at HP Laboratories, who all provided valuable advice in their own special ways.

References

1. *Information Technology—Local and Metropolitan Area Networks—Part 12: Demand Priority Access Method and Physical Layer Specification*, IEEE, 1994.
2. J.H. van Lint, *Introduction to Coding Theory*, GTM 86, Springer Verlag, 1982.
3. R.E. Blahut, *Theory and Practice of Error Correcting Codes*, Addison Wesley, 1983.
4. *Functional Requirements, IEEE Project 802*, Local and Metropolitan Area Networks Standards Committee, Draft 6.10.