Cyclic Redundancy Encoder for Error Detection in Communication Channels

Othman O. Khalifa, MD Rafiqul Islam and S. Khan

International Islamic University Malaysia
Kulliyyah of Engineering
Electrical and Computer department
Jalan Gombak
E-mail: khalifa@iiu.edu.my

Abstract--Cyclic Redundancy Check is one of the most powerful methods of error detection in blocks for digital communications signals. It involves a division of the transmitted message block by a constant called the generator polynomial. The quotient is discarded, and the remainder is transmitted as the Block check Character or Frame Check Sequence. The receiving station performs the same computation on the received message block. The computed remainder, FCS is compared to the remainder received from the transmitter. If the two match, no errors have been detected in the message block. If the two do not match, either a request for retransmission is made by the receiver or the errors are corrected through use of special coding technique. This paper describes the methodology used and the implementation of the Cyclic Redundancy Check (CRC) algorithm using C++ programming. The technique gained its popularity because it combines three advantages: Extreme error detection capabilities, little overhead and ease of implementation. The CRC is calculated by performing a modulo 2 division of the data by a generator polynomial and recording the remainder after division. The most commonly used polynomials are implemented. The conclusions and analysis results were shown and presents that the Cyclic Redundancy Check Encoder is used in error detection for digital signals due to the ability to quickly determine if errors are present. The redundancy bits produced by the cyclic encoder enable the receiver to quickly determine if an error was produced and different types of polynomials are used in CRC.

1. Introduction

It is physically impossible for any data recording or transmission medium to be 100% perfect of the time over its entire expected useful life [1][2][3][4][7]. As more bits are packed onto a square centimeter of disk storage, as communications transmission speeds increase, the likelihood of error increases sometimes geometrically. Thus, error detection and correction is critical to accurate data transmission, storage and retrieval. Check digits, appended to the end of a long number can provide some protection against data input errors. Longer data streams require more economical and sophisticated error detection mechanisms. Cyclic redundancy checking (CRC) codes provide error detection for large blocks of data. Checksums and CRCs are examples of systematic error detection. It is a group of error control bits is appended to the end of the block of transmitted data. This group of bits is called a syndrome. As it is known CRCs are polynomials over the modulo 2 arithmetic field. They use mathematics to tackle the problem of error detection [5][6][7].

2. Cyclic Redundancy Encoder for Error Detection

CRC error checking is quite powerful and easily implemented. In the presence of burst transmission errors, which each begin and end with a bit error, with zero or more intervening corrupted bits, the CRC can be a useful error detection and correction scheme [5][7][8][9]. CRC codes are based upon treating bit strings as representations of polynomials with coefficients of 0's and 1's only. For example, 110001 has 6 bits and thus represents the six-term generating polynomial

$$G(x) = x^5 + x^4 + 1$$

Polynomial arithmetic is done modulo 2, according to the rules of algebraic theory. There are no carries for addition, and no borrows for subtraction. Both addition and subtraction are identical to $\oplus$, Exclusive OR. Long division is carried out the same way as it is in binary except that the subtraction is done modulo 2. A divisor is said to go into a dividend if the dividend has as many bits as the divisor. When the CRC code method is employed, the sending TCP and the receiving TCP must agree upon a Generator Polynomial G(x) in advance. Both the high-order and low-order bits of the generator must be 1. The basic idea is to append a checksum to the end of the frame M(x) in such a way that the polynomial W(x) represented by the check summed frame is divisible by

0-7803-8671-X/04/$20.00 ©2004 IEEE.

G(X), i.e., R(x) = Remainder of $W(x) \oplus G(X) = 0$. When the receiver gets the check summed frame W(x), it divides W(x) by G(x).

If the remainder of $W(x) \oplus G(x) = 0$, there were no transmission errors. Otherwise, there must be transmission error(s).

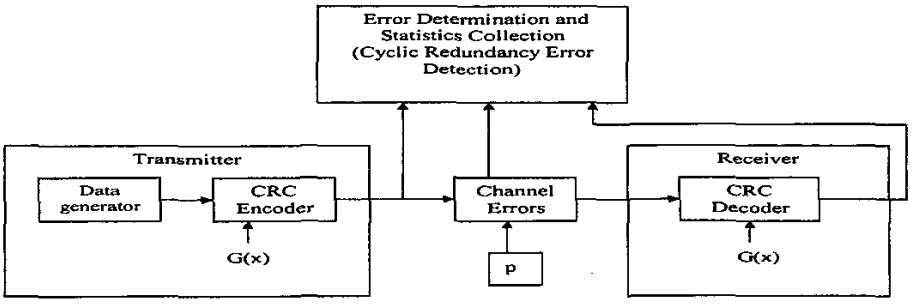

Figure 1. Diagram of the basic digital communications

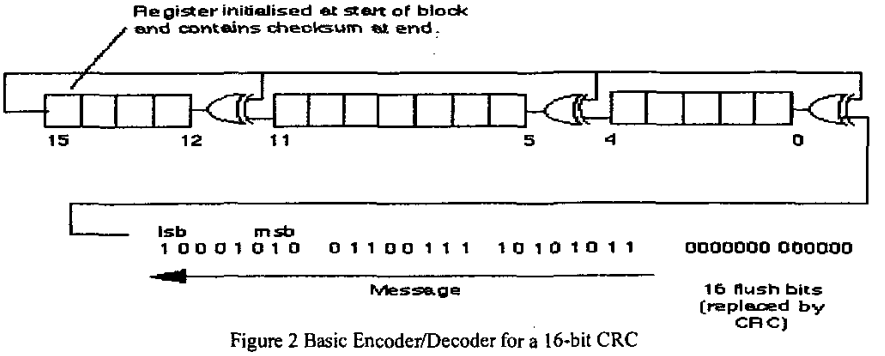

Figure 2 Basic Encoder/Decoder for a 16-bit CRC

The CRC algorithm works above the binary field. The algorithm treats all bit streams as binary polynomials. Given the original frame, the transmitter generates the FCS for that frame. The FCS is generated so that the resulting frame (the cascade of the original frame and the FCS), is exactly devisable by some pre-defined polynomial. This pre-defined polynomial is called the devisor or CRC Polynomial.

For the specific explanation we will define the following:

M - The original frame to be transmitted, before adding the FCS. It is k bits long.

F- The resulting FCS to be added to M. It is n bits long.

T - The cascading of M and F. This is the resulting frame that will be transmitted. It is k+n bits long.

P - The pre-defined CRC Polynomial. A pattern of n+1 bits.

The main idea behind the CRC algorithm is that the FCS is generated so that the reminder of T/P is zero. It is clear that

$T = M * x^n + F$ (1)

This is because by cascading F to M we have shifted T by n bits to the left and then added F to the result. We want the transmitted frame, T, to be exactly divisible by the pre-defined polynomial P, so we would have to find a suitable Frame Check Sequence (F) for every raw message (M). Suppose we divided only $M*x^n$ by P, we would get:

$M*x^n / P = Q + R/P$ (2)

There is a quotient and a reminder. We will use this reminder, R, as our FCS (F). Returning to Eq. 1:

$T = M*x^n + R$ (3)

We will now show that this selection of the FCS makes the transmitted frame (T) exactly divisible by P:

$T/P = (M*x^n + R)/P = M*x^n / P + R/P = Q + R/P + R/P = Q + (R+R)/P$ 4)

but any binary number added to itself in a modulo 2 field yields zero so:

$T/P = Q$, with no reminder. (6)

Following is a review of the CRC creation process:

1. Get the raw frame
2. Left shift the raw frame by n bits and the divide it by P.

3. The reminder of the last action is the FCS.
4. Append the FCS to the raw frame. The result is the frame to transmit

And a review of the CRC checks process:

1. Receive the frame.
2. Divide it by P.
3. Check the reminder. If not zero then there is an error in the frame.

It can be easily seen that the CRC algorithm must compute the reminder of the division of two polynomials.

3. Modulo-2 Arithmetic

The multiplication process is merely a series of logical ANDs and XORs , the vector [1001] is multiplied by the identity matrix I. This serves to demonstrate the technique of modulo-2 matrix multiplication, as well as to prove that dI = d.
Each row of the matrix corresponds to a bit in the data vector, with the top row being the most significant bit, and the bottom row being the least significant.

5. Cyclic Redundancy Check Codes in comparison to Hamming Codes

The goal in creating a CRC code is to select a generating polynomial G(x) that covers the statistically likely errors for a given fault model [7][9]. Common generating polynomials that give good error coverage include:

$$\text{CRC-16} = x^{16} + x^{15} + x^2 + 1$$

$$\text{CRC-CCITT} = x^{16} + x^{12} + x^5 + 1$$

$$\text{CRC-32} = x^{32} + x^{26} + x^{23} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

On the other hand, the Hamming codes can not only detect but also correct any 1-bit transmission errors. More research is needed for Hamming Codes that will correct transmission errors of 2 or more bits, even though the likelihood of such transmission errors are very small (less than 0.003%/32-bit transmission). The use of $\oplus$ (Exclusive OR) in constructing Cyclic Redundancy Check codes (packets) appears to be computationally more efficient than the complex method by which Hamming codes are built. But, the number of check bits (checksums) is exact (no more than needed) in Hamming codes, while they are generally longer and even arbitrary in Cyclic Redundancy Check codes.

6. Results and Discussion

The analysis results demonstrate that significant gains in error detection capability can be obtained by using CRC polynomials other than the standard polynomials that are in use worldwide. While there have been a few publications that indicated the standard CRCs were not optimal. Finding such a large opportunity for improvement in widely used standard approaches is a somewhat startling result, and so merits some discussion as to why it is so and speculation as to why it apparently has not been found.

7. Conclusions

Data transmission errors are easy to fix once an error is detected. Just ask the sender to transmit the data again. Thus, to provide data integrity over the long term, error correcting codes are required. Computers store data in the form of bits, bytes, and words using the binary numbering system. Error detecting and correcting codes are necessary because it expects no transmission or storage medium to be perfect. This work presents a methodology and example calculations of how to determine the optimal CRC polynomial. As a result, optimal polynomials that outperform the divisible-by-(x+1) class were found that can substantially improve error detection performance.

References

[1] A. B., Carlson P. B., Crilly ands J. C., Rutledge, *Communication Systems: An introduction to signals and Voise in Electrical Communication*, 4th edition, McGraw Hill, 2002.

[2] B. Sklar, *Digital Communications: Fundamentals and Applications*, 2nd Edition, 2001.

[3] A. F., Behrouz, *Data communication and Networking*, 2nd Edition, McGraw Hill International Edition, 2001

[4] H. Warren, *Telecommunications*, 4th Edition, Prentice Hall,2001.

[5]http://www2.rad.com/networks/1994/ercon/crc_how.htm

[6]http://www.tml.hut.fi/Studies/Tik110.300/1999/Wireless/channel_1.html

[7] E. B., Richard, *Theory and Practice of Error Control Codes*, Addison-Wesley,1984.

[8] J. Costello Jr., J. Hagenauer, H. Imai, and S. B. Wicker, "Applications of Error-Control Coding ", *IEEE Transactions on Information Theory*, Vol. 44 number 6, October 1998.

[9] D., Chun and J. K. Wolf, "Special Hardware for Computing the Probability of Undetected Error for Certain Binary CRC Codes and Test Results", *IEEE Transactions on Communications*, Vol. 42, No. 10, p. 2769, October 1994.