# Error Detection Based on Generator Polynomial

Kyi Sein Linn, Lin Min Ko
Computer University (Maubin)
*kyiseinlinn@gmail.com,linnlatstar@gmail.com*

## Abstract

*In today world use high speed computers. Communication between one computer and another or between the computer and its peripheral devices must be as error free as possible. However, error free data transmission in real world communication systems does not exist. Errors, no matter how few, will always be present. To deal with this problem, a good number of error detection techniques have been developed and used extensively in data communication systems known as the Cyclic Redundancy Check (CRC). In order to investigate the CRC error, handshake data (polynomial and handshake data bit) is sent to receiver subsequently, when the receiver accept the handshake, the polynomial is accepted by receiver. While in data communication between sender and receiver, sender packed the data and polynomial and then transmitted to receiver. When the packet arrived to receiver, the received packet is divided by polynomial. The receiver catches the result data.*

## 1. Introduction

Data can be corrupted during transmission. For reliable communication, errors must be detected and corrected. Error detection uses the concept of redundancy, which means adding extra bits for detecting errors at the destination.

There are Single-Bit error, Multiple-Bit error and Burst error. Single-Bit is only one bit in the data unit has changed. Multiple-Bit is two or more nonconsecutive bits in the data unit have changed. Burst error means that two or more consecutive bits in the data unit have changed. There are many detection methods to detect this error. Now we use the cyclic redundancy check for detecting error.

Modern hardware already contains error detection techniques in integrated circuit. This system revises the characteristics of error on communication medium that CRC works at Data Link Layer of TCP/IP. And then, it is simulate to become skilled at error detection methods, why error occurs on transmission process, how does an error detection technique work?

## 2. Data Communication

Data communication consists of a data source (transmitter), a communication channel, and a destination (receiver).

The transmitter usually contains several encoders which transform the data into a form acceptable to the communication channel and consistent with a pre-determined format, One of these encoders modifies the data before transmission so as to make possible the detection of errors when the data is received, The receiver contains several decoders for putting the data back into a form that is acceptable to the user, and for determining whether or not the data has been transmitted without error.

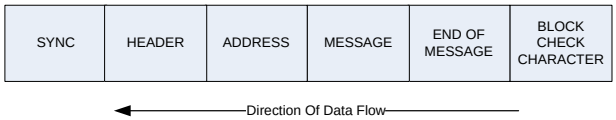


**Figure 1. Data Transmission Format**

**Sync** - there may be one or more synchronization characters to synchronize the receiver with the transmitter.
**Header** - contains the control information for the message, e.g., destination, priority, or message type.
**Address** - identification code for a specific device such that only that device will decode the message.
**Message** - the information (data) being transferred.
**End of Message** - a single character which lets the receiver know that the message has ended.
**Block Check Character** - one or more characters added to the message for error detection purposes.

## 3. Error Detection Methods

There are three main errors in data communication, they are single bit error, multiple bit error and burst error because of noise, crosstalk, line outages. The error detection methods are:

1. VRC (Vertical Redundancy Check) or Parity
2. LRC (Longitudinal Redundancy)
3. Checksum
4. CRC (Cyclic redundancy Check)

### 3.1. VRC (Vertical Redundancy Check) or Parity

A parity bit is added to every data unit so that the total number of 1s (including the parity bit) becomes even for even-parity check or odd for odd-parity check.

### 3.2. LRC (Longitudinal Redundancy Check)

Parity bits of all the positions are assembled into a new data unit, which is added to the end of the data block.

### 3.3. Checksum

Checksum is used by the higher layer protocols.

### 3.4. CRC (Cyclic redundancy Check)

CRC is a technique for detecting errors in digital data, but not for making corrections when errors are detected. It is used primarily in data transmission. CRC is based on binary division. Uses modular-2 division. The cyclic redundancy check, or CRC, is a technique for detecting errors in digital data, but not for making corrections when errors are detected. It is used primarily in data transmission from one computer to another or internal data transmission.

In the CRC method, a certain number of check bits, often called a checksum, are appended to the message being transmitted. The receiver can determine whether or not the check bits agree with the data, to ascertain with a certain degree of probability whether or not an error occurred in transmission is more effective and more efficient than either VRC or LRC. A **cyclic redundancy check** (CRC) is a non-secure hash function designed to detect accidental changes to raw computer data, and commonly used in digital networks and storage devices.

#### 3.4.1. Properties of CRC

1. Single and double errors
2. Odd number of bit errors
3. Bursts of length 16 or less
4. 99.997% 17 bits error bursts
5. 99.998% 18 bits and long error bursts.

## 4. Background Theory

When generating the CRC character (Block Check Character), the entire serial block of data (the message) is treated, not as a string of l's and 0's, but as a binary polynomial where the l's and 0's are the coefficients of that polynomial, M(X), known as the message polynomial. For instance, if 10110101 was the message being transmitted, it would be treated as the polynomial $(l)X^7 + (0)X^6 + (l)X^5 + (l)X^4 + (0)X^3 + (1)X^2 + (0)X + l$, or simply $X^7 + X^5 + X^4 + X^2 + 1$. The highest power of X is attached to the most significant bit **(MSB)** of the message.

M(X) is then pre-scaled and divided by a fixed polynomial, G(X), known as the generator polynomial. The division will yield a quotient, Q(X), and a remainder, R(X). R(X) is the CRC character and it is appended to the end of the message before transmission. The pre-scaling of M(X) is done to insure that the degree of M(X) is always greater than the degree of G(X), so that the remainder, R(X), is always different from the message itself.

The process is carried out as follows:

**1.** Multiply (pre-scale) M(X) by $X^{(n-k)}$, where:

**n** is the total number of bits being transmitted
**k** is the number of information (message) bits
**(n-k)** is the number of bits of the CRC character.

$X^{(n-k)} M(X) = Q(X) G(X) + R(X)$ -------- (eq. 1)

2. Divide $X^{(n-k)} M(X)$ by G(X) to determine R(X).

$$\frac{X^{(n-k)} M(X)}{G(X)} = \frac{R(X)}{G(X)} + Q(X)$$ --------- (eq. 2)

3. Add R(x) to $X^{(n-k)} M(X)$ to form the code message polynomial, F(X).

$X^{(n-k)} M(X) - R(X) = Q(X) G(X)$ ------ (eq. 3)
In modulo-2 arithmetic, R(X) = -R(X),
Therefore,
$X^{(n-k)} M(X) - R(X) = X^{(n-k)} M(X) + R(X)$ ------ (eq. 4)

$F(X) = X^{(n-k)} M(x) + R(X) = Q(x) G(x)$ ------ (eq. 5)

For example,

Consider a message 110010 represented by the polynomial,

$M(x) = x^5 + x^4 + x$

Consider a generating polynomial

$G(x) = x^3 + x^2 + 1(1101)$

This is used to generate a 3 bit CRC = C(x) to be appended to M(x), where G(x) is prime.

Steps:

**1.** Multiply M(x) by $x^3$ (highest power in G(x)). Add 3 zeros. 110010000

2. Divide the result by G(x). The remainder is C(x). 1101 long division into 110010000 is equal to 100100 remainder 100. If x div y gives remainder c that means: x= ny + c. Hence (x-c) = ny and (x-c) div y gives remainder 0. Here (x-c) = (x + c), (x = c) div y gives remainder 0
3. Transmit 110010000 + 100
   To be precise, transmit: $T(x) = x^3M(x) + C(x)$ = 110010100
4. Receiver end: Receive T(x). Divide by G(x), should have remainder 0.

**Table 1. Useful CRC Polynomials**

| Common Name | Generator | |
|---|---|---|
| | Polynomial | Hex |
| CRC-12 | $x^{12}+x^{11}+x^3+x^2+x+1$ | 80F |
| CRC-16 | $x^{16}+x^{15}+x^2+1$ | 8005 |
| CRC-CCITT | $x^{16}+x^{12}+x^5+1$ | 1021 |
| CRC-32 | $x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$ | 04C11DB7 |

## 5. System Implementation

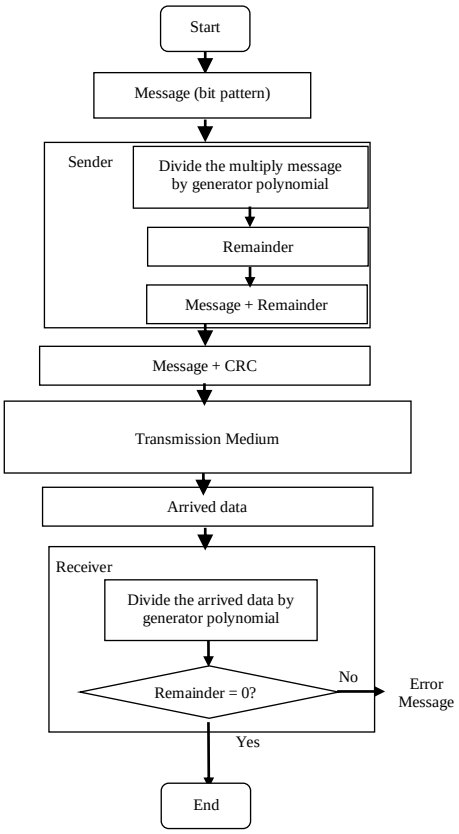


**Figure 2. System Flow Diagram**

In my system, the messages which want to send to another computer via TCP/IP, a form of bit pattern are added the number of CRC bits. And the transmitter divides the message by generator polynomial before send message to receiver. After dividing the message, the remainder left. This remainder is CRC bit appends to the message for error check.

After message and CRC bits are packed, transmitter send data to the receiver through the transmission medium. The receiver divides the arrived data by generator polynomial. Then the receiver checks that the remainder is equal to zero. If the remainder is equal to zero, accept the message. Otherwise the receiver display error message to the user.

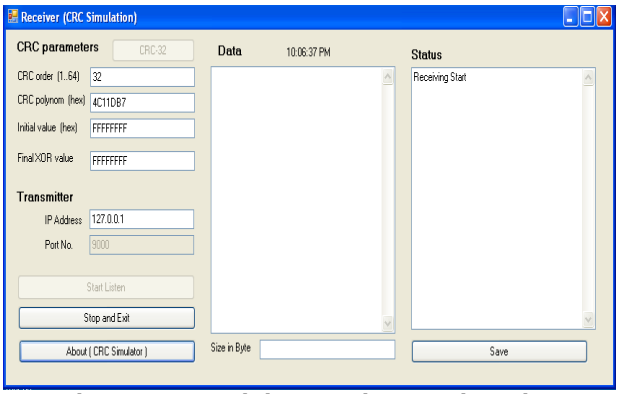


**Figure 3. Receiving Station or Listening Receiver**

First, the receiver establishes to listen from the transmitter and must wait till the time the message arrived.

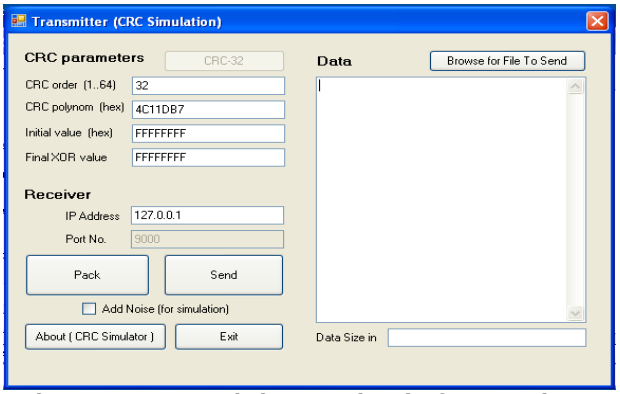


**Figure 4. Transmitting Station before Package**

Transmitter chooses data to send and pack with CRC polynomial. All the detail processes are described as following figures.

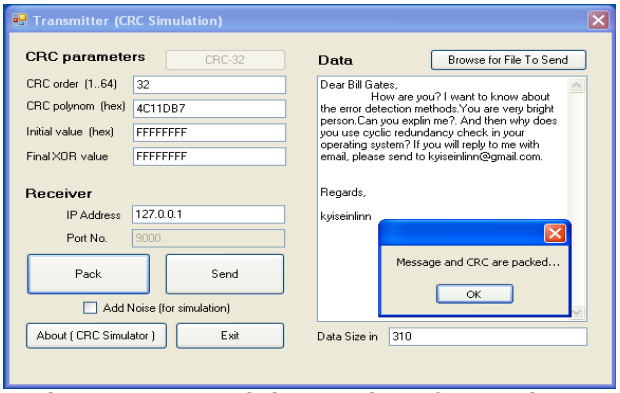


Figure 5. Transmitting Station after Package

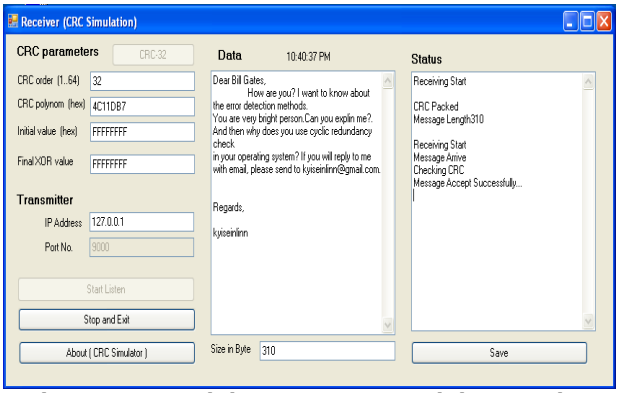


Figure 8. Receiving Data at Receiving Station

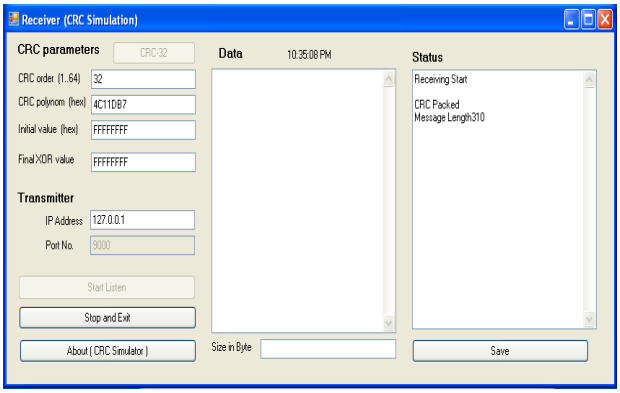


Figure 6. Receiving Data at Receiving Station

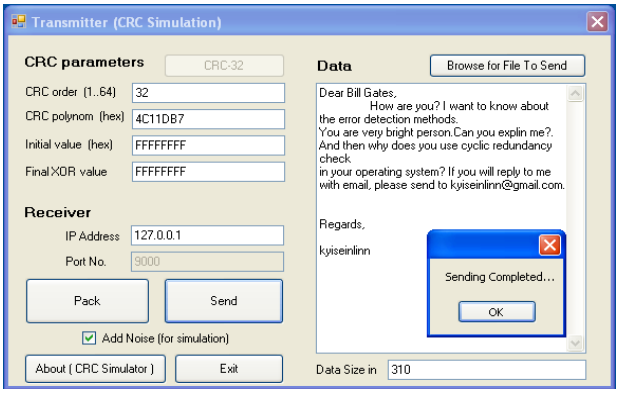


Figure 9. Sending Data at Transmitting Station after Package with Error (user)

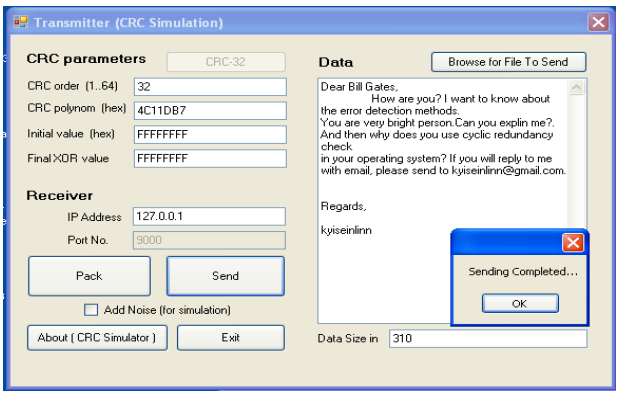


Figure 7. Sending Data at Transmitting Station after Package

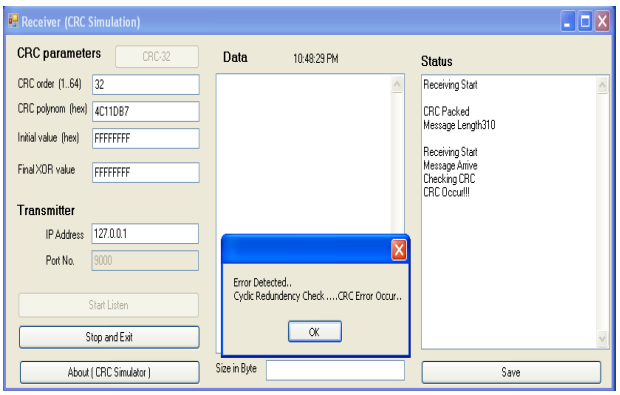


Figure 10. Detecting Error at Receiving Station

## 6. Conclusion

The intent of this system was to describe the problems plaguing the data communications field, and to present one of a number of possible solutions. The Cyclic Redundancy Check (CRC) was chosen because it is a viable solution to the difficult task of providing accurate and reliable data transmission. It has, as of this writing, become one of the most widely implemented error detection schemes, being used extensively, not only in data communication systems, but also in data storage systems, such as in disk controllers for use with mini and microcomputers.

The proposed system will implement 32bit CRC error detection system and simulate in ethernet network.

## 7. References


[1] Peterson, W. W. and Brown, D.T. "Cyclic Codes for Error Detection." In *Proceedings of the IRE*, January 1961, 228–235.

[2] Black, Richard. University of Cambridge Computer Laboratory Systems Research Group, February 1994. Website: www.cl.cam.ac.uk/Research/SRG/bluebook/21/crc/crc.html.

[3] Tanenbaum, Andrew S. *Computer Networks*, Second Edition. Prentice Hall, 1988.

[4] Forouzan, Data Communication and networking (fourth edition) Chapter 10 Error Detection and Correction.

[5] http://en.wikipedia.org/wiki/Cyclic_redundancy_check.html

[6] http://www.educypedia.be

[7] http://www.ee.unb.ca/tervo/
Steps in CRC generation are from Freeman,12 OCT 01 - tervo@unb.ca , University of New Brunswick, Department of Electrical and Computer Engineering

[8] http://www.cs.waikato.ac.nz/~312/crc.txt

[9] http://www.mindprod.com

[10] http:// www.rad.com

[11] William Stallings, Data & Computer Communications (sixth edition), Prentice Hall International Editions,
ISBN 0-13- 086388-2