International Journal of Engineering Research and Technology. ISSN 0974-3154, Volume 12, Number 9 (2019), pp. 1413-1420

Novel Error Detection Scheme in Communication Network Based on Complete Polynomial Check

Mamilus A. Ahaneku[1,*], Chibuike C. Nwonye [2], Vincent C. Chijindu[3], Udora N. Nwawelu[4] and Michael O. Ezeja[5]

[1,3] Senior Lecturers, Department of Electronic Engineering, University of Nigeria, Nsukka.

[2,4,5] Ph.D Students, Department of Electronic Engineering, University of Nigeria, Nsukka.

ORCID: 0000-0001-9427-6937 (Mamilus A. Ahaneku)

Abstract

Many communication channels are subject to channel noise which may introduce error(s) during the process of transmission of frames from the transmitter to the receiver. These errors make the receiver not to properly decode the transmitted data. It is a known fact that the present error detection schemes are inefficient in error detection, as some may be good in detecting single bit error detection, while the others may be good in another type of error detection. For that reason, there is need fashioned a new error detection scheme that could correct the shortcoming of the present error detection schemes. Consequently, this study focuses on developing a new and more robust efficient error detection scheme. In line with the above objective, an error detection scheme was developed and the scheme was validated against CRC using MATLAB Simulink. From the simulation, the results show that the new error detection technique has 16 undetected error frames per 100,000 transmitted frames compared to 330 undetected error frames per 100,000 transmitted frames for CRC. The result also shows that the new error detection scheme reduces the number of the undetected erroneous frame by 93.6% when it is subjected to the same condition with CRC-8.

Keywords*:* Complete polynomial, cyclic redundant check, Error detection, internet checksum parity check

1.1 INTRODUCTION

Reliable communication is one of the most important aspects of both wired and wireless communications.

Most often, during transmission, digital signals suffer as a result of noise introduced as error in the binary bits sent through the transmission path. The error usually occurs when the intended information sent does not match with the received one. In such a situation, it is likely that a-one (1) may change to a-zero(0), vice-versa. To protect communication systems from these errors, modern internet systems have built-in fault-tolerant designs. Outstanding fault tolerance technique is the detection of errors in communication packets. Errors in Internet packets cannot only cause the erroneous data to be transferred to a wrong destination but can also cause a fatal failure in the Internet system[1]–[3].

To guarantee error-free reception, communication systems have used many schemes to detect when erroneous data is received. Some of the schemes include: Cyclic redundancy check (CRC) scheme, Internet checksum scheme, parity check scheme [4][5], repetition and inversion scheme. These error detecting schemes are useful in the TCP/IP systems[1] [2].

The error detecting schemes such as the CRC scheme, are very effective for detecting errors that occur in the physical communication channels. On the other hand, the checksum scheme is required for detecting errors caused by system level faults [6]–[8] such as a buffer area overflow, bit errors in an internal computer bus, and so on. However, the Internet checksum scheme can only detect a single error bit that occurred in a packet[9]. For detecting multiple errors, Fletcher's checksum scheme was proposed[10].

2.1 RELATED WORKS

Cyclic Redundant Check

The cyclic redundancy check is a widely used parity bit based error detection scheme in serial data transmission applications. It sends k-bits codewords and the r-bits redundancies as its suffix [11]. Before the encoding, a fixed generator polynomial $g(x)$ is selected. The encoding rule of CRC is according to k-bits information codeword to compute the r-bits redundancies and then transmit it[12].

In the terminal, the received codeword is divided by the predefined generator polynomial. If the result has no remainder; it shows that the received word has no error. This is not conclusive as there may be an error[13].

CRC computation involves manipulating $m(x)$ and $g(x)$, where $m(x)$ is the data word and $g(x)$ is generator polynomial using modulo2 arithmetic. Suppose $x^{n-k}m(x)$ information code words and r-bits redundancies are used, then the selected $g(x)$ must own the following properties[14]:

- $g(x)$ can be divided by $x^n - 1$ with no remainder, and it's maximum number is r, constant term is 1.
- Can be divisible by all cyclic code Polynomials $C(x) = m(x)/g(x)$.
- A cyclic code group can only be generated by a generated polynomial.
- Generated polynomial must be paired. During the encoding, firstly, dividing $x^{n-k}m(x)$ by $g(x)$ results in a quotient of $Q(x)$ and a remainder of $s(x)$.

Cyclic Code Analysis

Cyclic code can be analysed to find its capability using polynomial. Here, some parameters will be defined, and those parameters are

M(x) = Transmitted Codeword

g(x) =Generator polynomial

s(x) = Syndrome (The remainder when the codeword is divided by the generator polynomial)

e(x) =error introduced during transmission

R(x) = Received codeword.

It can be established that the transmitted codeword, received codeword and error are related to the expression in (1)

$$R(x) = M(x) + e(x) \quad (1)$$

If equation (1) is divided by the generator polynomial, then equation (2) is obtained.

$$\frac{R(x)}{g(x)} = \frac{M(x)}{g(x)} + \frac{e(x)}{g(x)} \quad (2)$$

The remainder of $\frac{R(x)}{g(x)}$ is the syndrome defined by s(x). If none of the bit of the transmitted codeword is corrupt, equation (2) will be reduced to (3).

$$\frac{R(x)}{g(x)} = \frac{M(x)}{g(x)} \quad (3)$$

Equation (3) holds when $\frac{e(x)}{g(x)} = 0$ and syndrome s(x) is zero since R(x) must be divisible by g(x).

If any of the transmitted bits is corrupt, then $\frac{e(x)}{g(x)} \neq 0$ and either that the syndrome s(x) is zero or that the syndrome is non-zero.

If the syndrome is not zero, the error is detected but if the syndrome is zero, then the error is undetected by CRC. That means there is an error but CRC did not detect the error as stated by[13]. CRC will detect all possible errors except those that change the bit value of a block of code by exactly the value of the divisor[15]. Table1 is used to demonstrate the above theory using five bits data and four bits generator code. Assuming that the generator code is 1 0 1 1 and the data code is as listed in Table1.

Table 1: Data bit and polynomial for CRC

No	Data bit	Redundant bit	No	Data bit	Redundant bit
0	00000	000	16	10000	001
1	00001	011	17	10001	010
2	00010	110	18	10010	111
3	00011	101	19	10011	100
4	00100	111	20	10100	110
5	00101	100	21	10101	101
6	00110	001	22	10110	000
7	00111	010	23	10111	011
8	01000	101	24	11000	100
9	01001	110	25	11001	111
10	01010	011	26	11010	010
11	01011	000	27	11011	001
12	01100	010	28	11100	011
13	01101	001	29	11101	000
14	01110	100	30	11110	101
15	01111	111	31	11111	110

Table 1 shows the changes that occur during transmission. If any data changes to another data of same redundant bit, the error will not be detected. In the TCP/IP systems, 32-bit cyclic redundancy check (CRC) scheme is used for the data link layer [16][17].

Internet Checksum scheme

For *Internet Checksum scheme*, the data is divided into a giving number of bits at the source, from the bits; the redundant bits called checker is calculated and appended to the message before transmission. At the receiver, the same operation is carried out as done at the source; if the new checker is zero, the message is accepted otherwise the message is erroneous and will be discarded[4].

Assuming that 11, 8, 9, 5, 6 are the messages to be transmitted, instead of transmitting it alone, negative of the sum of the numbers is sent along with the messages for error detection i.e. 11, 8, 9, 5, 6,-39 will be transmitted. If none of the numbers changes the sum will be zero but if there is any change in the number, the sum will not be zero.

When the value of one word is incremented and the value of another word is decremented by the same amount, the two errors cannot be detected because the sum and checksum remain the same. Also, if the values of several words are incremented but the sum and the checksum do not change, the errors are not detected [4]. In the TCP/IP systems, the Internet checksum scheme is used for both the network layer and the transport layer [9], [18]–[20].

Repetition Scheme

In order to transmit a message over a noisy channel that may corrupt the message in few places, the idea of the repetition code is to just repeat the message several times. The higher the number of repetition the lower the error probability but in this scheme, the redundancy bit length is greater than or equal to the message bit length. The number of repetition is inversely proportional to the rate at which useful information is transmitted[21]. So the increase in the number of repetition leads to the decrease in the rate at which useful message will be transmitted. If 10010 is the data to be transmitted and the transmitter wants to use three repetitions, then, the code word becomes 111000000111000 as each bit is transmitted three times. In this scheme, if any bit and its repeated format are corrupt, the error will not be detected. This is a drawback.

Polarity Scheme

In this scheme, the actual message is transmitted along with its inversion format. The receiver then checks if two sets represent the inverse of the transmitted messages. If they are not inverse of each other, it indicates the presence of an error. It is not popular, as the code occupies double the bandwidth for the actual message. Moreover, if the corresponding bit in the data and its inverse are erroneous, the error will not be detected[21].

3.1 ERROR DETECTION SCHEME BASED ON COMPLETE POLYNOMIAL CHECK (CPC)

This section presents an error detection scheme proposed in this paper. The aim is to develop an error detection scheme that makes use of variable length redundant bits, unlike the existing error detection schemes that use constant length redundant bits. The performance of the proposed scheme with variable length redundant bit will be compared with the existing scheme with constant length redundant bit.

To start this analysis, we generate a binary number from a particular data bits such that the number will change when the data bit changes. This binary number is what will be appended to the data bits for error detection. The transmitter should have the ability to perform the calculation to get the binary number and append it to the data bits. If the data bit corrupts during transmission (bit 1 changes to 0 or bit 0 changes to 1) the resultant effect is to cause the transmitted data bit to change. When this transmitted data bit gets to the receiver, the receiver performs the same calculation the transmitter did at the source. If a data bit is corrupt, the binary number which the receiver gets will be different from what the transmitter has appended. It was stated earlier that binary number should have the ability to change when the data bit of the same number of bits changes.

3.1.1. MODELING OF DATA BIT POLYNOMIAL

Modelling of the polynomial function that represents data bit requires the use of a complete polynomial. A complete polynomial is a polynomial that none of the polynomial coefficients is zero and the power starts from n and ends in 1. In a complete polynomial, a sequence with a common difference of -1 is formed and the power of each element of the polynomial is determined by the position of the bit, specifically; the coefficient must be 1or -1. The general polynomial that models n-digit message is given as in (4).

$$f(x) = x^n + x^{n-1} + x^{n-2} + \cdots + x^2 + x^1 \qquad (4)$$

If a binary data 00110101 is transmitted, we need a polynomial function that will model this data bits such that when any of the bits changes the equation will change, equation (4) reduces to (5) since the data is in binary.

$$f(2) = 2^n + 2^{n-1} + 2^{n-2} + \cdots + 2^2 + 2^1 \qquad (5)$$

Using another illustration, if we have 10111011101 as a data word, the power of each element of the polynomial that will represent the data will be 11,10 ,9,8,7,6,5,4,3,2,1 i.e. position of the bits starting from the right. If assume bit 1 to be positive and bit 0 to be negative, the polynomial function in (4) becomes

$$x^{11} - x^{10} + x^9 + x + x^7 - x^6 + x + x + x^3 - x^2 + x$$

Table 2: Data bits and their polynomial, using four bits.

0000	$-x^4-x^3-x^2-x$	1000	$x^4-x^3-x^2-x$
0001	$--x^4-x^3-x^2+x$	1001	$x^4-x^3-x^2+x$
0010	$-x^4-x^3+x^2-x$	1010	$x^4-x^3+x^2-x$
0011	$-x^4-x^3+x^2+x$	1011	$x^4-x^3+x^2+x$
0100	$-x^4+x^3-x^2-x$	1100	$x^4+x^3-x^2-x$
0101	$-x^4+x^3-x^2+x$	1101	$x^4+x^3-x^2+x$
0110	$-x^4+x^3+x^2-x$	1110	$x^4+x^3+x^2-x$
0111	$-x^4+x^3+x^2+x$	1111	$x^4+x^3+x^2+x$

Table 2 shows more examples on how to generate polynomial function using data bits.

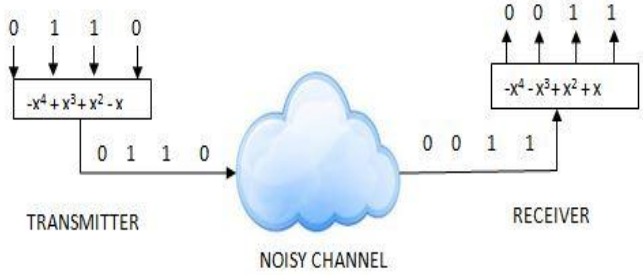


Figure 1: Effect of communication error on data and its polynomial

Figure 1 illustrates how the transmitter computes the polynomial of the data, 0110, when transmitted through a noisy communication channel or when there is a poor synchronisation between the receiver and the transmitter.

3.1.2.MODELING REDUNDANTBIT POLYNOMIAL

For the polynomial in (5) not to have the same value when the sign(s) change(s) (6) will hold.

$$2^n > 2^{n-1} + 2^{n-2} + \cdots + 2^2 + 2^1 \qquad (6)$$

If we assume that 2^n is 2 greater than $2^{n-1} + 2^{n-2} + 2^{n-3}$

$$2^n - 2 = 2^{n-1} + 2^{n-2} + \cdots + 2^2 + 2^1 \qquad (7)$$

If we prove equation (7), it means that any number gotten from equation (5) will be peculiar to that polynomial. This implies that when the sign of the polynomial changes, the number must change irrespective of the highest power of the polynomial.

To prove this, we will invoke the principle of mathematical induction (PMI). The principle of mathematical induction may be stated as follows: suppose n is a variable with range Z_+ (set of a positive integer) and p(n) is a statement which is either true or false. If it can be proved that (i) P(1) is true and (ii) whenever p(k) is true, then p(k+1) is also true, otherwise, it follows that p(n) is true for all n in Z_+[22] .

Let us further illustrate the use of this principle by the following example.

Prove that $1 + 2 + 3 + 4 + 5 + \cdots + n = \frac{n(n+1)}{2}$ for all n in Z_+

Let p(n) be the statement $1 + 2 + 3 + 4 + 5 + \cdots + n = \frac{n(n+1)}{2}$

P(1) is true since $\frac{1(1+1)}{2} = 1$

Suppose P (k) is true, that is

$$1 + 2 + 3 + 4 + 5 + \cdots + k = \frac{k(k+1)}{2} \qquad (8)$$

Then we want to prove that p(k+1) is true

$$1 + 2 + 3 + 4 + 5 + \cdots K + k + 1 = \frac{(k+1)(k+2)}{2} \qquad (9)$$

Now

$1 + 2 + 3 + 4 + 5 + k + (k+1) = (1 + 2 + 3 + 4 + 5 + \cdots + k) + K + (k+1)$

From equation (8)

$1 + 2 + 3 + 4 + 5 + \cdots + k + (k+1) = \frac{k(k+1)}{2} + k + 1,$

factorizing out $(k+1)$ from the right hand side, $(k+1)\left(\frac{k}{2} + 1\right) = \frac{(k+1)(k+2)}{2}$ which establishes equation (9)

Then standing on this principle to prove that $2^n - 2 = 2^{n-1} + 2^{n-2} + 2^{n-3} + \cdots + 2^2 + 2$ for all n set of integer greater than 1

$P(n)$ is the statement which is $2^n - 2 = 2^{n-1} + 2^{n-2} + 2^{n-3} + \cdots + 2^2 + 2$

$P(2) = 2^2 - 2 = 2^1$

Assuming that $P(K)$ is true, meaning that

$$2^k - 2 = 2^{k-1} + 2^{k-2} + 2^{k-3} + \cdots + 2^2 + 2 \qquad (10)$$

Then we want to prove that $P(K+1)$ is true

$$2^{k+1} - 2 = 2^k + 2^{k-1} + 2^{k-2} + \cdots + 2^2 + 2 \qquad (11)$$

$2^k + (2^{k-1} + 2^{k-2} + \ldots + 2^2 + 2) = 2^k + 2^k - 2 = 2*2^k - 2 = 2^{k+1} - 2$ which establishes equation (11)

Since we have established that $2^n - 2 = 2^{n-1} + 2^{n-2} + 2^{n-3} + \cdots + 2^2 + 2$

The general polynomial function can be compressed to be;

$$\mathrm{F(x)} = \sum_{i=1}^{n} \left[\delta(bi-1) - \delta(bi) \right] x^i \qquad (12)$$

Where δ is delta function, b is the bit, i is the position of the bit and n is the number of bits. Delta function[23] specified that

$$\delta(x) = \begin{cases} 1 & x = 0 \\ 0 & x \neq 0 \end{cases} \qquad (13)$$

$$\text{Then } f(2) = \sum_{i=1}^{n} \left[\delta(bi-1) - \delta(bi) \right] 2^i \qquad (14)$$

The redundant bit will be calculated from equation 15

$$\frac{F(x)}{2} - C = 0, \qquad (15)$$

Where C is the redundant bits in decimal, since C is in decimal, after calculating C, the value should be converted to binary to get the redundant bit in binary. If C is a negative number the complement should be taken.

Since x=2, the equation (15) reduces to

$$\frac{F(2)}{2} - C = 0 \qquad (16)$$

Making C the subject, equation (16) leads to equation (17).

$$C = \frac{F(2)}{2} \qquad (17)$$

Since one of the causes of undetected error is when two or more string of bits have the same redundant bits. But the proposed model has been modelled in such a way that all string of bits must have different redundant bits as shown in table 3 using a different number of bits.

Table 3: Data bits and their redundant bits

Data	$F(x)$	$\frac{F(2)}{2}$	Redundant bit
0000	$-x^4-x^3-x^2-x$	-15	0000
0001	$-x^4-x^3-x^2+x$	-13	0010
0010	$-x^4-x^3+x^2-x$	-11	0100
0011	$-x^4-x^3+x^2+x$	-9	0110
0100	$-x^4+x^3-x^2-x$	-7	000
0101	$-x^4+x^3-x^2+x$	-5	010
0110	$-x^4+x^3+x^2-x$	-3	00
0111	$-x^4+x^3+x^2+x$	-1	0
1000	$x^4-x^3-x^2-x$	1	1
1001	$x^4-x^3-x^2+x$	3	11
1010	$x^4-x^3+x^2-x$	5	101
1011	$x^4-x^3+x^2+x$	7	111
1100	$x^4+x^3-x^2-x$	9	1001
1101	$x^4+x^3-x^2-x$	11	1011
1110	$x^4+x^3+x^2-x$	13	1101
1111	$x^4+x^3+x^2+x$	15	1111

The transmitter and receiver must agree on;

- The number of bits per frame that the network must transmit
- $\frac{F(2)}{2} - C = 0$ For calculating the value of C.

The block diagram of the proposed model is shown in Figure 2. In the figure, the error detection encoder extracts the message and uses it to form a polynomial function using the procedure explained in section 3.2. Then it applies the formula in equation (16) in order to calculate the redundant bits in decimal form. When the binary converter converts the redundant in decimal to binary, encoder appends the redundant bit to the message and transmits the codeword through a communication channel.

At the receiver, the error detection decoder receives the codeword and separates the message from the redundant bit. The decoder applies the same procedure as the encoder on the received message to gets its own redundancy and both the received redundancy and calculated redundancy are fed to a logic decision (comparator). If the redundancies are the same, then no error has occurred and the message is accepted else, an error has occurred and the message is discarded. The proposed encoder and decoder flow charts are shown in Figures 3 and 4, respectively.

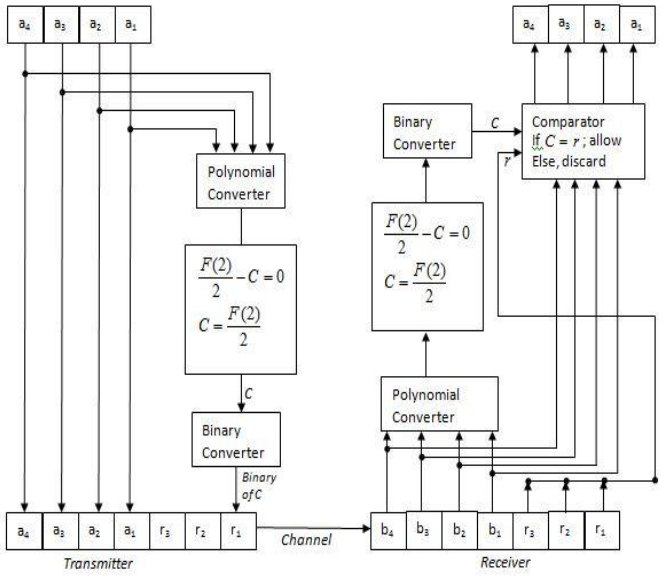


Figure 2: Proposed error detection encoder and decoder

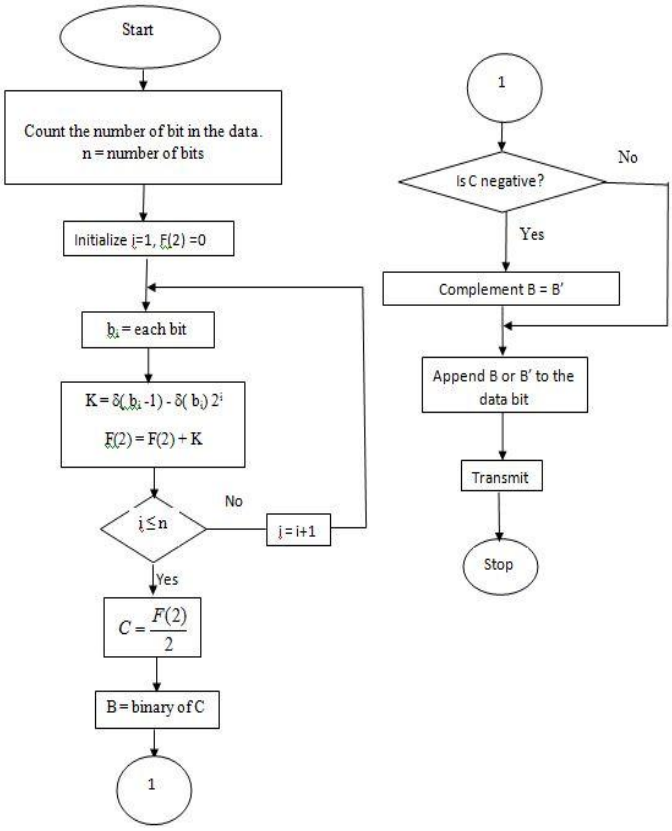
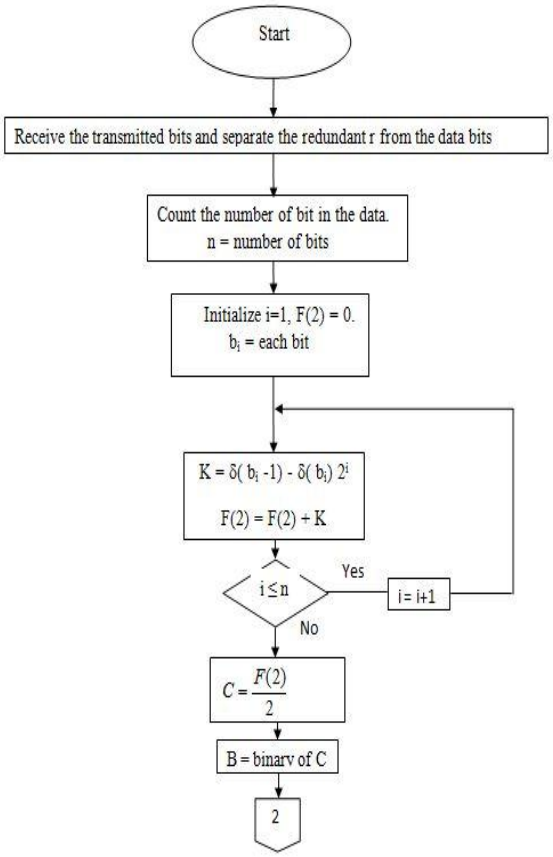


Figure 3: Proposed encoder flow chart



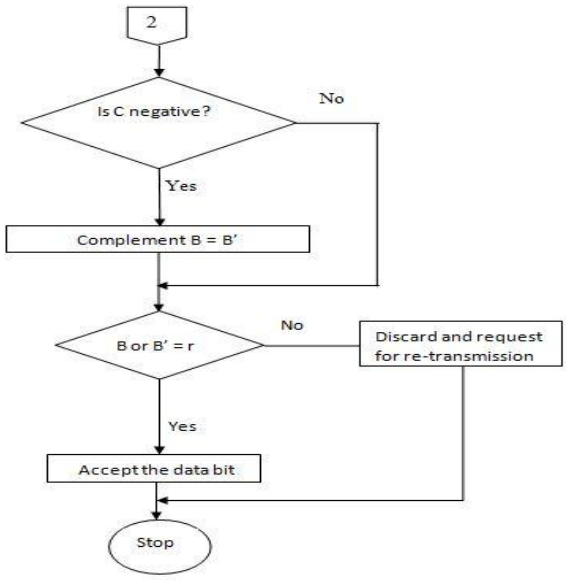


Figure 4: Proposed decoder flow chart

evaluate the performance of the designed error detection scheme. In this section, simulation results are presented to demonstrate the performance of the proposed error detection scheme in terms of a number of the erroneous frames that is undetected when 100,000 frames are sent, with each frame containing 10 bits. Its performance is compared with the performance of CRC alone, since[24] has proven CRC as the best among the existing error detection scheme so it will be a waste of time and energy to start comparing it with other existing ones that CRC outperformed.

The computer simulation model was simulated in the Simulink environment to evaluate the performance of the proposed error detection scheme. It is made up of Bernoulli binary generator, proposed error detection encoder, Binary symmetrical channel and proposed error detection decoder. The Bernoulli binary generator generates ones and zeros randomly depending on the configured probability, proposed error detection encoder receives the bits and performs some calculations (explained in section 3.1 and 3.2) on the bits to get the check bit; and the check bit is appended to the original bit for error detection. The binary symmetrical channel (BSC) adds error to the bits (flips the bit based on the configured error probability, the higher the error probability the higher error is added). The proposed decoder removes the check bit from the data bit and recalculates the check bit. If the recalculated check bit is the same with the original check bit there is no error but if they are not the same it means an error has occurred.

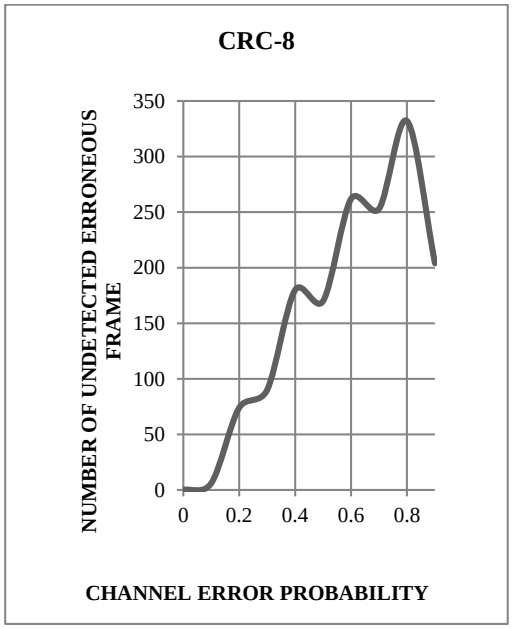


Figure 5: Performance of CRC-8.

4.1. RESULTS AND DISCUSSION

Object oriented computer simulation has been identified as one of the effective tools for providing a simpler, quicker and more cost effective of resolving this problem. MATLAB version R2013a was chosen as the simulation method to

Figure 5 shows that the CRC-8 decoder detected almost all errors between 0 and 0.1. This is because few bits are corrupt at this range of channel error probability. The system has its maximum undetected erroneous frame when the channel error probability is 0.8 and reduced as it reached 0.9.

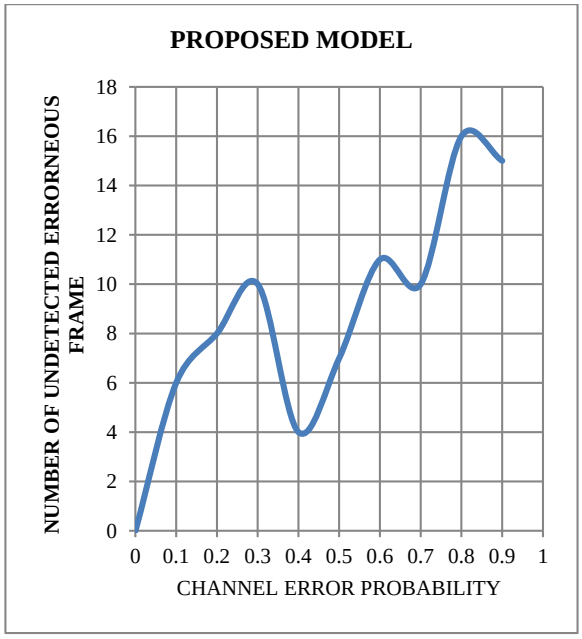


Figure 6: Performance of the proposed model

Figure 6 depicts the performance of the proposed model. At zero channel error probability, no undetected erroneous frame was recorded because the channel is ideal at zero channel probability. The undetected erroneous frame increases between 0 - 0.1, it has a high increment at 0.3 channel error probability then from 0.4 - 0.9, the proposed error detection scheme behaves like CRC-8 that has reduced undetected erroneous frame.

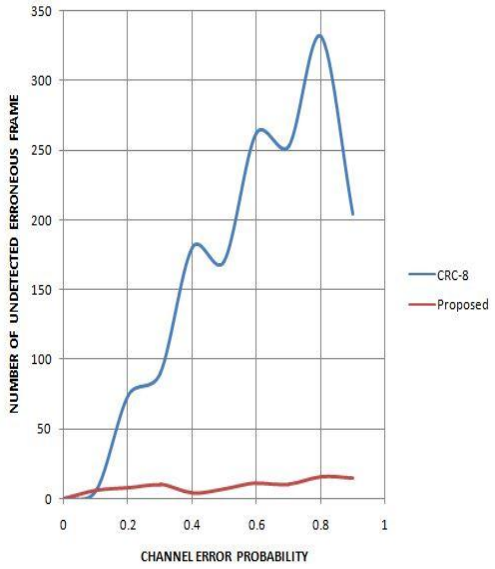


Figure 7: The comparison between the performance of CRC-8 and the performance of the proposed model.

The comparison between the performance of CRC-8 and the proposed model is shown in Figure 7. It can be seen that the proposed model outperformed CRC-8 as the number of undetected error is by far less than the number of undetected error in CRC-8 when two of them are subjected to the same condition. The result also shows that the new error detection scheme reduces the number of the undetected erroneous frame by 93.6%.

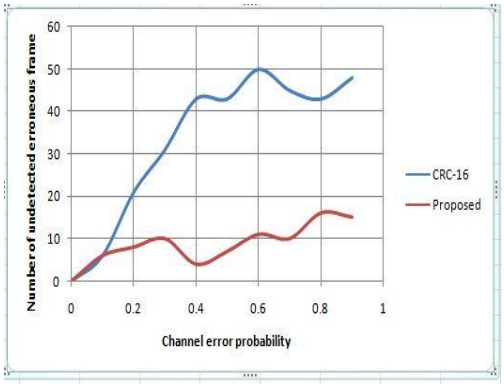


Figure 8: The comparison between the performance of CRC-16 and the performance of the proposed model.

From the figure 8, it can be shown that the proposed model out performed CRC-16 but not as much as CRC-8

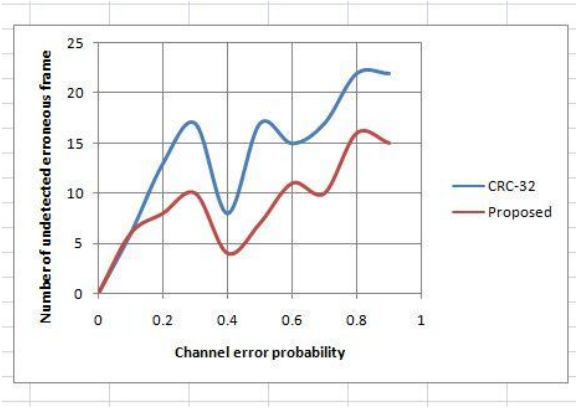


Figure 9: Comparison between the performance of CRC-32 and the performance of the proposed model.

From figure 9, it can be shown that the proposed model slightly out performed CRC-32

4.2 BANDWIDTH COMPARISON

The proposed model consumes more bandwidth than CRC-8 because it uses more number of redundant bits to perform error detection.

5.1. CONCLUSION

Reliable error detection scheme is very important in communication networks for proper delivery of frames from

the transmitting system to the receiving system. The higher the number of check bits the higher its ability to detect an error but the higher the bandwidth consumption. The proposed error detection scheme was developed to detect all types of error that occur during transmission and reception. This is an added advantage. During implementation and simulation, it was observed that when the code word passes through a binary symmetrical channel, both the data bit and the check bit get corrupted.(It rarely occurs). When the data bit and check bit are corrupt the error detection ability of the proposed model drops but if only the data bit is corrupt, the system operates at its optimal error detection capability. In conclusion, from the result of the proposed model, it can be shown that the proposed model with variable length redundant bit outperforms CRC-8 with constant length redundant bit as a number of undetected error is by far less than the number of undetected error in CRC-8 when two of them are subjected to the same condition.

Recommendations

Since significant improvement has been made in the performance of error detection scheme by using variable check bit length, further research can work on the following areas.

- This model was implemented by making the transmitter and receiver to agree on the number of data bits to be transmitted per frame, researchers should work to remove this impairment since some communication networks allow the transmission of the variable frame length.
- The effect of increased number of bits on the performance of the proposed model should be investigated.

REFERENCES

[1] D. C. Lynch and M. T. Rose, *Internet System Handbook*. Addison-Wesley, 1993.

[2] K. Washburn and J. Evans, *Running a successful Network*. 1993.

[3] P. H. Corrigan, *LAN disaster preventation and recovery*. Prentice Hall, 1994.

[4] A. Behrouz, *Data Communication and Network*. 2014.

[5] D. Yoshihisa, I. Kazuhiko, M. Yukiva, and Y. Daisuke, "Double and Triple Error Detecting Capability of Internet Checksum and Estimation of Probability of Undetectable Error," *IEEE Trans. Communications*, 1996.

[6] C. Patridge and J. Hughes, "Performance of Checksums and CRCs over real data," in *Proc. of SIG-COMM*, 1995, pp. 67–76.

[7] J. Kay and J. Pasquale, "Profiling and reducing processing overheads in TCP/IP," *IEEE Trans. Networking*, vol. 4, no. 6, pp. 817–828, 1996.

[8] G. G. Fin, S. H. And, and R. Van Meter, "The impact of a zero-scan Internet checksumming mechanism," *Computer Communication Review*, vol. 26, no. 5, pp. 27–39, 1996.

[9] R. Braden and D. Borman, *Computing the internet checksum*. 1988.

[10] J. G. Fletcher, "An arithmetic checksum for serial transmission," *IEEE Trans. Communications*, vol. 30, No. 1, pp.247 – 252, 1982.

[11] F. Changxing, Z. P, and W. . C. Ke, *Communication Principle*. publishing house of electronics industry. Beijing, 2005.

[12] G. Alain, *Channel Coding in Communication Networks-From Theory to Turbocodes*. Antony Rowe Ltd, Chippenham, Wiltshire. London, 2006.

[13] M. Jianming and C. Chong, "The CRC checkout algorithm used in data acquisition system," *Journal of Fuzhou University (Natural Science)*, vol. 31, no. 3, 2003.

[14] U. Fanghui and A. Fang, *Implement and Design of CRC Based on FPGA*. 2008.

[15] F. Behrouz, *Introduction to data communications and networking*. The McGraw-Hill Companies,Inc, 1998.

[16] K. A. Witzke and C. Leung, "A comparison of some error detecting CRC code standards," *IEEE Trans. Communications*, vol. 33, p. pp 996-998, 1985.

[17] T. Fujiwara, T. Kasami, and S. Lin, "No TitleError detecting capabilities of the Shortened Hamming Codes Adopted for error detection in IEEE standards 802.3," *IEEE Trans. Communications*, vol. 37, no. 9, pp. 986–989, 1989.

[18] J. Postel, "Internet Protocol," in *RFC791*, 1981.

[19] J. Postel, "Transmission control protocol," in *RFC793*, 1981.

[20] A. Rijsinghani, *Computation of the Internet Checksum via Incremental Update*. 1994.

[21] B. Kumar, "Design and implementation of Reed-Solomon using extended inversionless massey-berukamp algorithm," *International journal of emerging research in management and technology*, vol. 3, p. 17, 2014.

[22] J. C. Amazigo *et al.*, *Introductory University Mathematics 1, algebra, trigonometry and complex number*. 2005.

[23] S. Sanjay, *Signals and systems*. S.K. Kataria and sons, 2012.

[24] J. Stone and C. Partridge, "When the CRC and TCP checksum disagree," in *ACM SIGCOMM*, 2000.