ERROR DETECTION USING CYCLIC REDUNDANCY

*

Asis Kumar Rout 181IT108
Information Technology.
NITK Surathkal.
Karnataka,India
asisrout7@gmail.com

Ashok Bhobhiya 181IT154
Information Technology
NITK Surathkal.
Karnataka,India
ashokkumar26042000@gmail.com

Jeeukrishnan Kayshyap 181IT220
Information Technology.
NITK Surathkal
Karnataka,India
geetkayshyap@gmail.com

***Abstract*—This paper introduces a way to authenticate the data transmitted over the network using Cyclic Redundancy Check (CRC) error detection technique which work on the concept of binary division. A network must be capable of transmitting the data from one end to other end with accuracy. But transmission errors are the common fact of data communication. It is not mandatory that data received at receiver end is identical to the data transmitted by the sender. There are a number of reasons responsible for data corruption as thermal noise, impulse noise, etc. For reliable communication, it is required that the system must be enriched with error detection and error correction techniques. Therefore, a number of error control techniques have been introduced and one of them is CRC to achieve accuracy in data communication.**

I. Introduction

CRCs are based on the theory of cyclic error-correcting codes. The use of systematic cyclic codes, which encode messages by adding a fixed-length check value, for the purpose of error detection in communication networks was first proposed by W. Wesley Peterson in 1961. Cyclic codes are not only simple to implement but have the benefit of being particularly well suited for the detection of burst errors, contiguous sequences of erroneous data symbols in messages. This is important because burst errors are common transmission errors in many communication channels, including magnetic and optical storage devices. Specification of a CRC code requires definition of a so- called generator polynomial. This polynomial resembles the divisor in a polynomial long division, which takes the message as the dividend and in which the quotient is discarded and the remainder becomes the result, with the important distinction that the polynomial coefficients are calculated according to the carry-less arithmetic of a finite field. The length of the remainder is always less than the length of the generator polynomial, which therefore determines how long the result can be. In practice, all commonly used CRCs employ the finite field. This is the field of two elements, usually called 0 and 1, comfortably matching computer architecture. The simplest error- detection system, the parity bit, is in fact a trivial 1-bit CRC: it uses the generator polynomial x+1.

The Data Link layer in OSI model is responsible for error control. Error control is defined as the combination error correction and error detection. When the data is transmitted over the network, it is the responsibility of Data Link layer to check for the error between the routers and to correct them. Error is any unwanted change which reduces the usefulness of original data. There are two types of error-

- Single bit error
- Burst error

II. Goals and Objectives

- Main goal of this project is to become familiar with Raw Socket Programming.
- In this project we use CRC as error detection technique.
- Raw Socket Programming allows us to access packets sent over protocols other than TCP / UDP.
- Raw Socket Programming allows us to control the IP,TCP header and make changes in error detection mechanism.

III. Literature Survey

Error Correction and Detection using Cyclic Redundancy Check

Author-Dr. T. Logeswari

In this paper Cyclic Redundancy Check codes are implemented to detect the usefulness of various types of errors that might occur through the transmission of data stream carrying message signal through the internet. The explanation has been prepared in such a way that bit position, number of bits of data word and codeword of the generator polynomial are measured random. This helped to obtain satisfactory result with the proposed polynomial.

Cyclic Redundancy Code (CRC) Polynomial Selection for Embedded Networks

Author- Philip Koopman Tridib Chakravarty

This paper describes a polynomial selection process for embedded network applications and proposes a set of good

general-purpose polynomials. Mathematically, a CRC is treating a binary data word as polynomial and performing polynomial division by a generator polynomial (CRC polynomial). For embedded networks, the property of interest is usually the Hamming Distance (HD) which is the minimum possible number of bit inversions that must be injected into a message to create an error that is undetectable by that messages CRC-based Frame Check Sequence. A set of 35 new polynomials in addition to 13 previously published polynomials provides good performance for 3- to 16-bit CRCs for data word lengths up to 2048 bits.

IV. Methodology

The methodology proposed for CRC error detection technique is in which one is to generate the CRC bits at sender side and another is for error detection in data unit transmitted from one end to other end at receiver side. The CRC bits computed using CRC generator methodology is appended to the data unit to be transmitted to other end. The received message is checked using CRC checker methodology and if it is correct then the CRC is discarded and data is accepted otherwise data is discarded. Initially when there is only information then 0s are appended to the data bits which act as CRC bits to compute the CRC. The computed result is replaced by 0 bits append to the data bits. This is the actual data unit which is also known as the message and transmitted over the network to the other end. At the receiver side, same generating function is used and result is computed. If it is all 0s then data received is correct otherwise it is corrupted and discarded

Raw sockets provide a way to bypass the whole network stack traversal of a packet and deliver it directly to an application.It allows an application to directly access lower level protocols, which means a raw socket receives un-extracted packets.

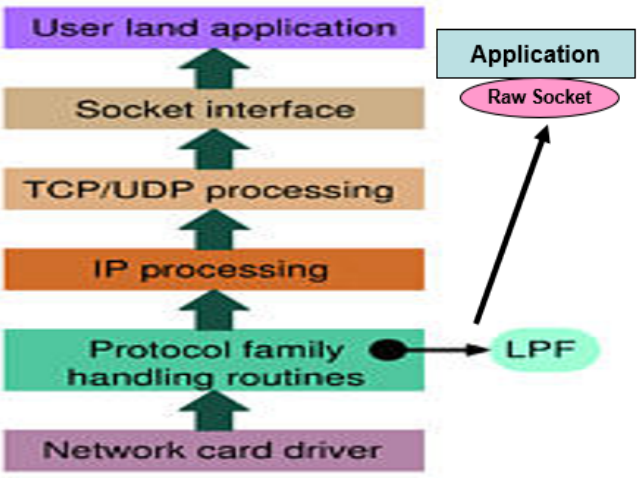

Fig. 1. Raw Socket

V. Design

The design is as below

- CRC generator
- CRC checker

CRC Generator
Input the number of data bits and CRC bits. Enter generating function bits, its value and data bits value.
Perform Adder operation on the variable entered in step-1. Store data bits, generating function bits and data unit (data bits + CRC bits) to be operated in different arrays. Repeat next step until each bit of data unit is processed. Check data unit to be operated-
if (1st bit==0)
XOR the data unit with 0 and store
the result in same array of data unit.
else
XOR the data unit with generating
function and store the result in same
array of data unit.
Display CRC bits and data unit to be transmitted over the network to receiver end after appending CRC bits to data bits.

B. CRC checker:
Input the number of data unit bits (number of data bits + number of CRC bits) received from the sender.
Enter generating function bits, its value and data unit bits value. Store generating function bits and data unit (data bits + CRC bits) received from sender in different arrays. Repeat next step until each bit of data unit is processed. Check data unit received from sender
If (1st bit==0)
XOR the data unit with 0 and store
the result in same array of data unit.
else
XOR the data unit with generating
function and store the result in same
array of data unit.

VI. IMPLEMENTATION

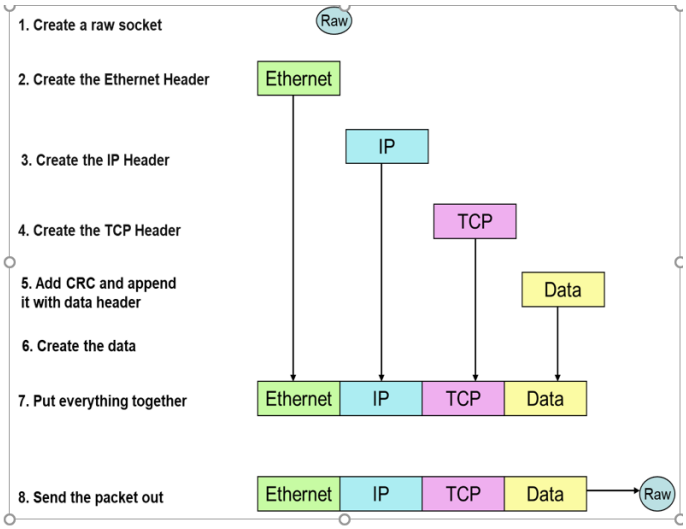

Fig. 2. Send Packet in Raw Socket

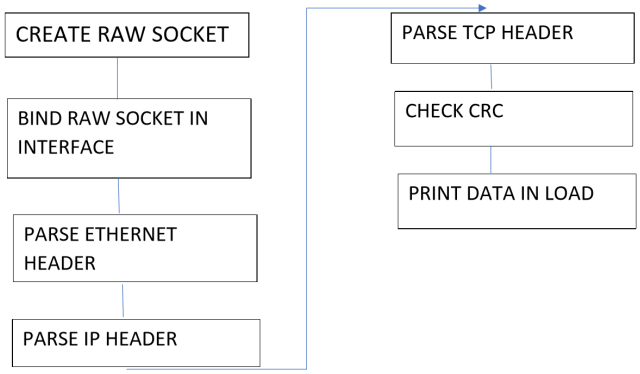

Fig. 3. Packet Sniffer

VII. RESULTS AND ANALYSIS

- The idea is mainly based on raw socket programming.
- A raw socket is used to receive all packets added in the layers of TCP/IP model. This means packets received at the Ethernet layer will directly pass to the raw socket.
- Raw sockets programming allows headers of lower level protocols to be constructed by the application.
- The RAW protocol is one of the common computer languages that documents are translated into before being sent to a networked printer. The printer interprets the protocol and prints the document.

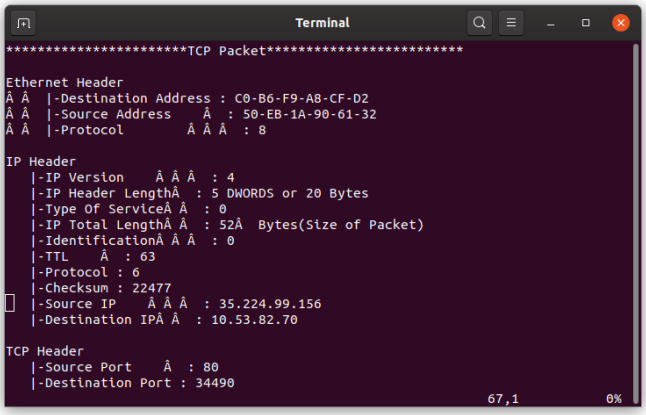

Fig. 4. TCP Header

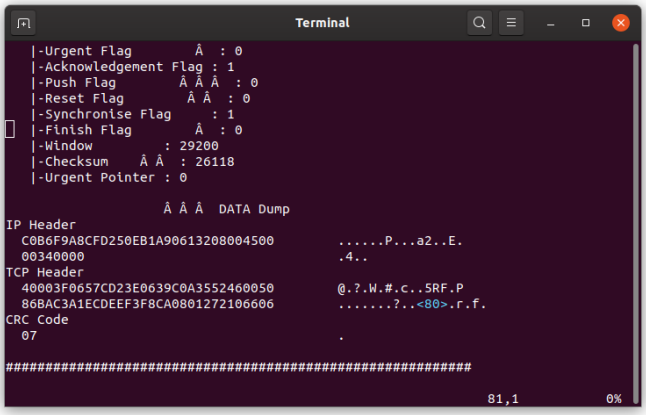

Fig. 5. CRC Code

VIII. CONCLUSION

This paper presented the error detection technique CRC based on the binary division using basic XOR bitwise operation and algorithm to implement it. It works on the concept of redundancy that is to append some extra bits in data unit known as redundant bits to detect the error and implemented at the Data Link layer of OSI model. There are a number of techniques used for error detection at data link layer among which CRC provides desirable efficiency. It provides good performance in terms of accuracy and security compared to other techniques. The future challenge is to make the code applicable to real world entity for example, audio, image etc.

IX. ACKNOWLEDGEMENT

We would like to thank our teachers in our department ,particularly Dr.Geetha V and Ms.Thanmayee for helping us shape the idea of our project.

X. REFERENCES

- Dr. T. Logeswari,"Error Correction and Detection using Cyclic Redundancy Check"; http://www.ijircce.com/;Vol. 5, Issue 1, January 2017.

- Philip Koopman,"32-Bit Cyclic Redundancy Codes for Internet Applications", The International Conference on Dependable Systems and Networks (DSN) 2002.
- BEHROUZ A. FOROUZAN,"DATA COMMUNICATION AND NETWORKING"; 5TH EDITION.