Error Detecting Codes*

K.M.Shafi†

December 9, 2016

1 Introduction

Digital communication is part and parcel of our life. Whether wired or wireless, we all nowadays use different types of digital devices who interact with other digital devices like cell phone, telephone, TV remote control, WiFi , blue-tooth enabled devices. In any case, data gets transfered from sender to receiver and vice versa. There are various possibilities that the data in transit may get lost either partially or fully due to errors introduces by various factors.

1.1 Error Detection

To make sure the integrity of data [intactness] to the receiver of data, extra information is also transmitted with the actual data. This extra information called as error detection codes is used by the receiver to detect the errors introduced in the data during transit but do not determine the precise location of error. Error can be a single bit per word/byte or can be in multiple bits of a word/byte in which case is called burst error.

*This tutorial is originally intended for the PGDCA students batch 2015-16 of DDE, KU

†mshafi710@gmail.com

2 Parity

Single bit error is usually detectable by adding an extra bit called parity bit usually at the end of word/byte. Usually two schemes are used in this case.

1. Odd parity: when the number of 1s in a word is odd, the parity bit is 0 else 1
2. Even parity: when the number of 1s in a word is even, the parity bit is 0 else 1

Example: If the sender transmitter has to send a word **1100101000101010** to destination receiver.
Using odd parity scheme, it calculates the number of 1s which here is 7 so the parity bit will be 0.
Using even parity scheme, the number of 1s is odd, the parity bit will be 1 to make it over all number of 1s even.
The receiver at other end, will calculate the number of 1s again including parity bit. If it is odd then the last bit should be 0 in case of odd parity scheme. If it is 1, then the error has introduced in the data. Vice versa case is with the even parity scheme.

The problem with both the parity schemes is they can only detect single bit errors. For instance, if a 1 in the word gets inverted to 0 and some other 0 in the same word gets inverted to 1, then the receiver will calculate the parity which will match with the parity bit and the errors will remain undetected.

To over come this, usually checksum or cyclic redundancy check [CRC] schemes are used to detect burst errors.

3 Checksum

Checksum or hash is a value that is used to determine the integrity of data. Checksum or hash value serves as a unique identifier for the data. If the data changes [either through introduction of error or deliberate tampering along the transit] then so does the checksum value. To test data integrity, the sender of data calculates checksum value using some particular algorithm. At receivers end, the same calculation is performed on the data and compared with the sender provided checksum value . If the two values match, then the

$$\begin{array}{r} 1010110 \\ +1011010 \\ \hline 0001100 \\ \hline \end{array}$$

Figure 1: Module-2 Arithmetic

integrity of data has not comprised during the transit. The most commonly used algorithms for calculation of hash values are MD5,SHA-1,SHA-2.

4 CRC

Cyclic redundancy check is the powerful way of detecting single or multiple errors in data. It uses binary division in calculating the error detecting codes both at the sender's and receiver's end.
The data to be transmitted [k bits] is appended with bit sequence of [n-k] bits by the transmitter to make these n bits exactly divisible by some predetermined number. The receiver at the other end divides the whole stream of n bits by the same predetermined number. If it divided fully, then no error has introduced and first k bits are treated as actual data and last n-k bits are thrown.
CRC uses Modulo-2 arithmetic while calculating the [n-k] bits. Mod-2 does addition of bits without carrying further. Mod-2 addition of

This is same as Exclusive-OR operation where different bits on operation yield 1 and same bits 0.

Example:[1] Let the message [k bits] to be transmitted is 1010001101 (10 bits) and the predetermined number **P** is 110101(06 bits). As per CRC, the value **F** to be calculated will be 05 bits. Thus n=15, k=10 and n-k=5.

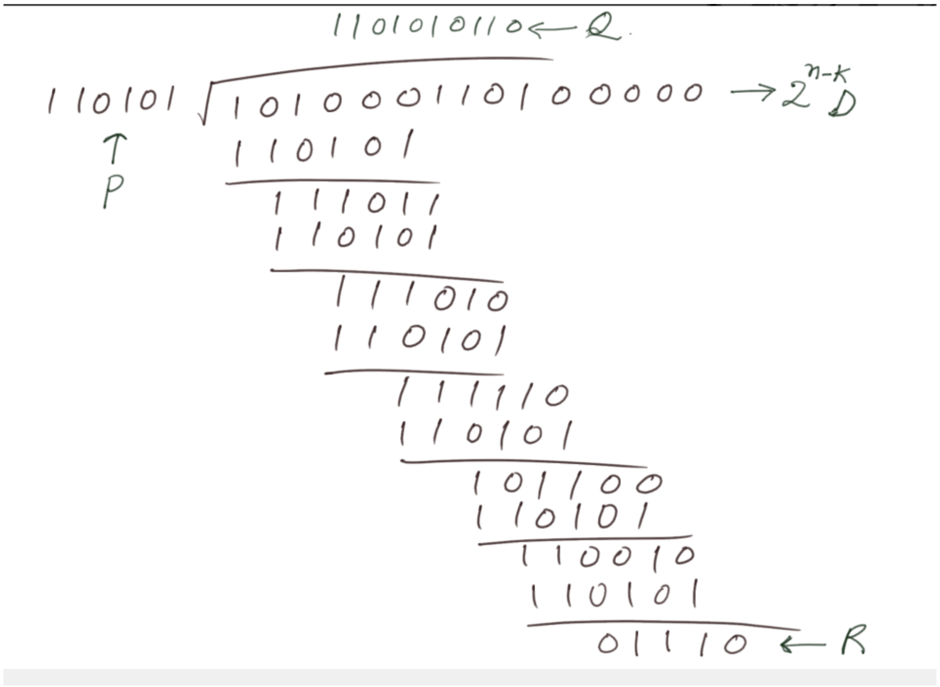
1:

[1] Data and Computer Communications, William Stallings,Pearson Education, 7th Edition,2004

The message 1010001101 is multiplied by 2^{n-k} i.e., 2^5 giving the result of 15 bits as 101000110100000.

2:

The result is continuously divided by P as shown below



```
                 1101010110 ← Q.
                                          n-k
110101 √ 101000110100000  →  2   D
   ↑      110101
   P      ------
           111011
           110101
           ------
             111010
             110101
             ------
               111110
               110101
               ------
                 101100
                 110101
                 ------
                  110010
                  110101
                  ------
                   01110 ← R
```

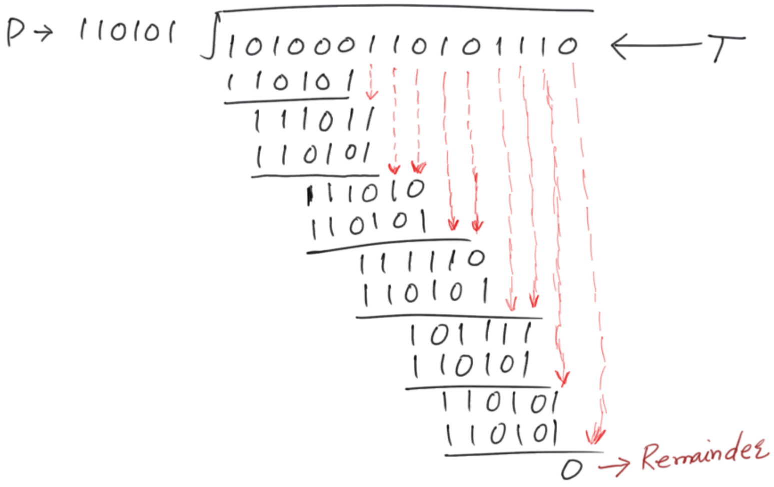
3: The remainder is added to 2_5D to give T= 101000110101110 which is transmitted.

4:

If there are no errors, the receiver receives intact. The received frame is divided by P:

5:

Because there is no remainder, it is assumed that there have been no errors.


P → 110101
101000110101110 ← T
110101
111011
110101
111010
110101
111110
110101
101111
110101
110101
110101
0 → Remainder