Open system Interconnection (OSI)

DataLink Layer Error Control

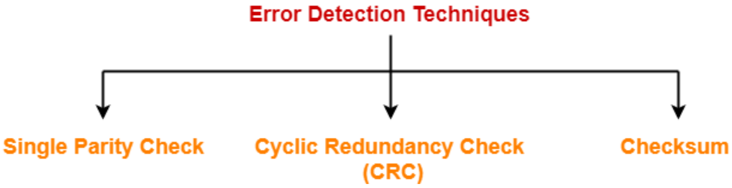
Munesh Singh

Indian Institute of Information Technology, Design and Manufacturing Kancheepuram, Chennai, Tamil Nadu 600127

September 27, 2020

Error Detection in Computer Networks

- When sender transmits data to the receiver, the data might get scrambled by noise or data might get corrupted during the transmission.
- Error detection is a technique that is used to check if any error occurred in the data during the transmission.
- Error can be single bit or multi-bit error (burst error)
- What is length of error from first bit change from MSB to LSB call the length of bit error. $1001 - - - > 0001$ (4 length of error)
- Some popular error detection methods are-



Single Parity Check-

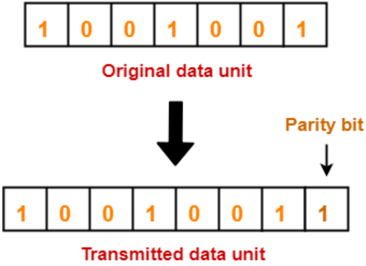
- In this technique,
  - One extra bit called as parity bit is sent along with the original data bits.
  - Parity bit helps to check if any error occurred in the data during the transmission.
- Steps Involved
  - Error detection using single parity check involves the following steps-
  - **At sender side, Step-01:**
    1. Total number of 1s in the data unit to be transmitted is counted.
    2. The total number of 1s in the data unit is made even in case of even parity.
    3. The total number of 1s in the data unit is made odd in case of odd parity.
    4. This is done by adding an extra bit called as parity bit.

Single Parity Check-

- **Step-02:**
  - The newly formed code word (Original data + parity bit) is transmitted to the receiver.
- **At receiver side, Step-03:**
  - Receiver receives the transmitted code word.
  - The total number of 1s in the received code word is counted.
  - Then, following cases are possible-
    1. If total number of 1s is even and even parity is used, then receiver assumes that no error occurred.
    2. If total number of 1s is even and odd parity is used, then receiver assumes that error occurred.
    3. If total number of 1s is odd and odd parity is used, then receiver assumes that no error occurred.
    4. If total number of 1s is odd and even parity is used, then receiver assumes that error occurred.

Parity Check Example

- Consider the data unit to be transmitted is 1001001 and even parity is used.
- **At Sender Side-**
  1. Total number of 1s in the data unit is counted.
  2. Total number of 1s in the data unit = 3.
  3. Clearly, even parity is used and total number of 1s is odd.
  4. So, parity bit = 1 is added to the data unit to make total number of 1s even.
  5. Then, the code word 10010011 is transmitted to the receiver.



Parity Check Example

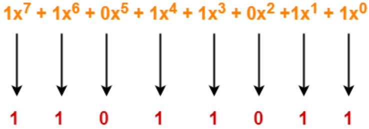
- **At Receiver Side-**
  1. After receiving the code word, total number of 1s in the code word is counted.
  2. Consider receiver receives the correct code word = 10010011.
  3. Even parity is used and total number of 1s is even.
  4. So, receiver assumes that no error occurred in the data during the transmission.
- **Advantage-**
  1. This technique is guaranteed to detect an odd number of bit errors (one, three, five and so on).
  2. If odd number of bits flip during transmission, then receiver can detect by counting the number of 1s.

Parity Check Example

- **Limitation-**
  1. This technique can not detect an even number of bit errors (two, four, six and so on).
  2. If even number of bits flip during transmission, then receiver can not catch the error.
- **Example:**
  - Consider the data unit to be transmitted is 10010001 and even parity is used.
  - Then, code word transmitted to the receiver = 100100011
  - Consider during transmission, code word modifies as 101100111. (2 bits flip)
  - On receiving the modified code word, receiver finds the number of 1s is even and even parity is used.
  - So, receiver assumes that no error occurred in the data during transmission though the data is corrupted.

Cyclic Redundancy Check

- Cyclic Redundancy Check (CRC) is an error detection method.
- It is based on binary division.
- **CRC Generator**
  - CRC generator is an algebraic polynomial represented as a bit pattern.
  - Bit pattern is obtained from the CRC generator using the following rule:
    - The power of each term gives the position of the bit and the coefficient gives the value of the bit.
  - Consider the CRC generator is $x^7 + x^6 + x^4 + x^3 + x + 1$.
  - The corresponding binary pattern is obtained as-



  - Thus, for the given CRC generator, the corresponding binary pattern is 11011011.

Properties Of CRC Generator-

- The algebraic polynomial chosen as a CRC generator should have at least the following properties-
- **Rule 1**
  - It should not be divisible by x.
  - This condition guarantees that all the burst errors of length equal to the length of polynomial are detected.
- **Rule 2**
  - It should be divisible by x+1.
  - This condition guarantees that all the burst errors affecting an odd number of bits are detected.

Properties Of CRC Generator-

- **Important Notes-**
  - If the CRC generator is chosen according to the above rules, then-
    - CRC can detect all single-bit errors
    - CRC can detect all double-bit errors provided the divisor contains at least three logic 1s.
    - CRC can detect any odd number of errors provided the divisor is a factor of $x+1$.
    - CRC can detect all burst error of length less than the degree of the polynomial.
    - CRC can detect most of the larger burst errors with a high probability.

Steps Involved-

- **Step-01: Calculation Of CRC At Sender Side-**
  - A string of n 0s is appended to the data unit to be transmitted.
  - Here, n is one less than the number of bits in CRC generator.
  - Binary division is performed of the resultant string with the CRC generator.
  - After division, the remainder so obtained is called as CRC.
  - It may be noted that CRC also consists of n bits.
- **Step-02: Appending CRC To Data Unit-**
  - At sender side,
  - The CRC is obtained after the binary division.
  - The string of n 0s appended to the data unit earlier is replaced by the CRC remainder.

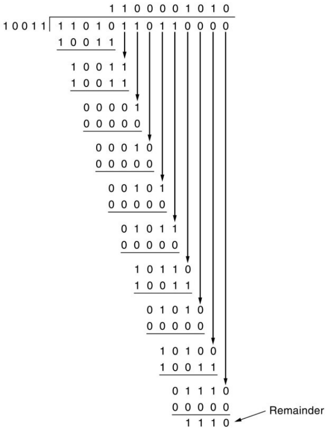
- **Step-03: Transmission To Receiver-**
  - The newly formed code word (Original data + CRC) is transmitted to the receiver.
- **Step-04: Checking at Receiver Side-**
  - The transmitted code word is received.
  - The received code word is divided with the same CRC generator.
  - On division, the remainder so obtained is checked.
- **The following two cases are possible-**
  - Case-01: Remainder $= 0$
    - Receiver assumes that no error occurred in the data during the transmission.
    - Receiver accepts the data.
  - Case-02: Remainder $\neq 0$
    - Receiver assumes that some error occurred in the data during the transmission.
    - Receiver rejects the data and asks the sender for retransmission.

PRACTICE PROBLEMS BASED ON CYCLIC REDUNDANCY CHECK (CRC)-

Q1 A bit stream 1101011011 is transmitted using the standard CRC method. The generator polynomial is x4+x+1. What is the actual bit string transmitted?

- **Solution:**
  - The generator polynomial G(x) = x4 + x + 1 is encoded as 10011.
  - Clearly, the generator polynomial consists of 5 bits.
  - So, a string of 4 zeroes is appended to the bit stream to be transmitted.
  - The resulting bit stream is 11010110110000.

- From here, CRC = 1110. Now,
- The code word to be transmitted is obtained by replacing the last 4 zeroes of 11010110110000 with the CRC.
- Thus, the code word transmitted to the receiver = 11010110111110.



Problem 2

Q2 A bit stream 10011101 is transmitted using the standard CRC method. The generator polynomial is x^3+1.

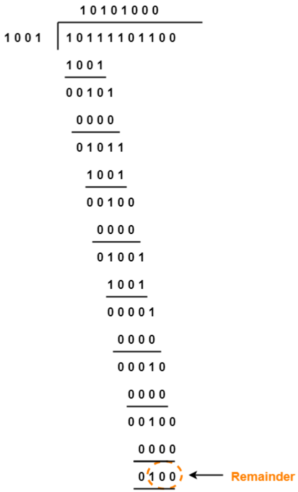
1. What is the actual bit string transmitted?
2. Suppose the third bit from the left is inverted during transmission. How will receiver detect this error?

- **Solution-**
  - The generator polynomial $G(x) = x^3 + 1$ is encoded as 1001.
  - Clearly, the generator polynomial consists of 4 bits.
  - So, a string of 3 zeroes is appended to the bit stream to be transmitted.
  - The resulting bit stream is 10011101000.

```
          10001100
     ______________
1001 | 10011101000
       1001
       ----
       00001
        0000
        ----
        00011
         0000
         ----
         00110
          0000
          ----
          01101
           1001
           ----
           01000
            1001
            ----
            00010
             0000
             ----
             00100
              0000
              ----
              0100  <---- CRC
```

- The code word to be transmitted is obtained by replacing the last 3 zeroes of 10011101000 with the CRC.
- Thus, the code word transmitted to the receiver = 10011101100.

- According to the question,
  - Third bit from the left gets inverted during transmission.
  - So, the bit stream received by the receiver = 10111101100.
- Receiver receives the bit stream = 10111101100.
- Receiver performs the binary division with the same generator polynomial as-



- The remainder obtained on division is a non-zero value.
- This indicates to the receiver that an error occurred in the data during the transmission.
- Therefore, receiver rejects the data and asks the sender for retransmission.

Thank You