Cyclic Codes for Error Detection
W. W. Peterson and D. T. Brown

by

Fuad Mohamed
Outline

- Concept of Cyclic Redundancy Checks
- Error detections
- How does it work
- Implementation
- Applications
The problem

- Error detection is especially important when computer programs are transmitted or stored, because an error of even one bit is often sufficient to make a program faulty.

- Providing this accuracy at low cost
CRC concept

- CRC is a technique for detecting data transmission errors based on generator polynomial.

- Transmitted messages are divided into predetermined lengths that are divided by a fixed divisor.

- According to the calculation, the remainder number (CRC) is appended onto and sent with the message.

- When m is received (or when stored), the remainder gets recalculated and compared to the transmitted remainder. If the numbers do not match, an error is detected.
Notations

- $k =$ Number of binary digits in the message before encoding
- $n =$ Number of binary digits in the encoded message
- $n - k =$ number of check bits
Notations cont.

\[ b \] = length of a burst of errors.

\[ G(X) \] = message polynomial

\[ P(X) \] = generator polynomial

\[ R(X) \] = remainder on dividing \( X^{n-k} G(X) \) by \( P(X) \).

\[ F(X) \] = encoded message polynomial

\[ E(X) \] = error polynomial

\[ H(X) \] = Received encoded message polynomial

\[ H(X) = F(X) + E(X) \]
Polynomial Representation of Binary Information

- Think of binary digits as coefficients of a polynomial in the dummy variable X.

- Polynomial is written low-order-to-high-order.

- A polynomial is a value expressed in a particular algebraic form:
Polynomials

Each term of the form $x^i$ is either present or absent 0 or 1

Examples:

- message: 110101
  is represented as: $1 + x + x^3 + x^5$
- $x^7 + x^6 + 1$
  $= 1 \cdot x^7 + 1 \cdot x^6 + 0 \cdot x^5 + 0 \cdot x^4 + 0 \cdot x^3 +$
  $0 \cdot x^2 + 0 \cdot x^1 + 1 \cdot x^0$
Polynomial arithmetic mod 2

- Polynomial arithmetic mod 2 differs slightly from normal computer arithmetic.
- They are treated according to the laws of ordinary algebra with an exception: addition is to be done modulo two.
- A modulo polynomial has no carry operation.
- Allows an efficient implementation of a form of division.
- Fast, and easy to implement using shift registers.
polynomials cont’d

- $1 \cdot x^a + 1 \cdot x^a = 0 \cdot x^a$
- XOR
- $1 \cdot x^a + 0 \cdot x^a = 1 \cdot x^a$
- $0 \cdot x^a + 1 \cdot x^a = 1 \cdot x^a$
- $0 \cdot x^a + 0 \cdot x^a = 0 \cdot x^a$
- $-1 \cdot x^a = 1 \cdot x^a$
Addition

“Add”: XOR each element individually with no carry:

\[ x^4 + x^3 + x^2 + x + 1 \]

\[ + x^4 + x^3 + x^2 + x \]

\[ \underline{x^3 + x^2 + 1} \]
Multiplication

“Multiply”: multiplying by $x^i$ is like shifting to the left.

\[
\begin{array}{c}
x^2 + x + 1 \\
\times \quad x + 1 \\
x^2 + x + 1 \\
x^3 + x^2 + x \\
x^3 + 1
\end{array}
\]
Algebraic Description of Cyclic Codes

- A cyclic code is defined in terms of a generator polynomial $P(X)$ of degree $n-k$.
- If $P(X)$ has $X$ as a factor then every code polynomial has $X$ as a factor and therefore zero-order coefficient equal to zero.
- Only codes for which $P(X)$ is not divisible by $X$ are considered.
Encoded Message Polynomial F(X)

- Compute \( X^{n-k} G(X) \)
- \( R(X) = X^{n-k} G(X) / P(X) \)
- Add the remainder to the \( X^{n-k} G(X) \)
  - \( F(X) = X^{n-k} G(X) + R(X) \)
  - \( X^{n-k} G(X) = Q(X) P(X) + R(X) \)
  - Also \( F(X) = Q(X) P(X) \)
- \( Q(x) \) is the Quotient
Example

- Let \( n=15, \ k=10 \)
- \( n-k=5 \), uses the generator poly. \( P(x) = 1 + x^2 + x^4 + x^5 \) to encode the msg. 1010010001 corresponding to poly.

\[
G(x) = 1 + x^2 + x^5 + x^9
\]

- So we divide \( x^5 G(x) \) by \( p(x) \) to find the reminder. By long division we can see:

\[
x^5 + x^7 + x^{10} + x^{14} = (1 + x^2 + x^4 + x^5)(1 + x + x^2 + x^3 + x^7 + x^8 + x^9) + (1 + x)
\]

- Code Poly. Formed by adding the reminder \( 1 + x \) to \( x^5 \ G(x) \)
Principles of Error Detection and Error Correction

- An encoded message containing errors can be represented by
  \[ H(X) = F(X) + E(X) \]

- \( H(X) = \) Received encoded message polynomial
  - \( F(X) = \) encoded message polynomial
  - \( E(X) = \) error polynomial
Principles of Error Detection and Error Correction Cont’d

- To detect error, divide the received, possible erroneous message $H(X)$ by $P(X)$ and test the remainder.
- If the remainder is nonzero an error has been detected.
- If the remainder is zero, either no error or an undetectable error has occurred.
Theorem 1: A cyclic code generated by a polynomial $P(X)$ with more than one term detects all single errors.

Proof:

- A single error in the $i$'th position of an encoded message corresponds to an error polynomial $X^i$.
- For detection of single errors, it is necessary that $P(X)$ does not divide $X^i$.
- Obviously no polynomial with more than one term divides $X^i$. 
Theorem 2: Every polynomial divisible by $1 + X$ has an even number of terms.

Proof:

Also if $P(X)$ contains a factor $1 + X$ any odd numbers of errors will be detected.
Double and Triple Error Detecting Codes (Hamming Codes)

**Theorem 3:** A code generated by the polynomial \( P(X) \) detects all single and double errors if the length \( n \) of the code is no greater than the exponent \( e \) to which \( P(X) \) belongs.

- Detecting double errors requires that \( P(X) \) does not divisible by \( X^i + X^j \) for any \( i, j < n \).
Double and Triple Error Detecting Codes Cont’d

- **Theorem 4**: A code generated by \( P(X) = (1 + X) P_1(X) \) detects all single, double, and triple errors if the length \( n \) of the code is not greater than the exponent \( e \) to which \( P_1(X) \) belongs.
  - Single and triple errors are detected by presence of factor \( 1 + X \) as proved in Theorem 2.
  - Double errors are detected because \( P_1(X) \) belongs to the exponent \( e \geq n \) as proved in Theorem 3

Q.E.D
Detection of a Burst-Error

- A burst error of length \( b \) will be defined as any pattern of errors for which the number of symbols between first and last errors including these errors, is \( b \).
Detection of a Burst-Error Contd...

Theorem 5: Any cyclic code generated by a polynomial of degree n-k detects any burst - error of length n-k or less.

- Any burst polynomial can be factored as $E(X) = X^i E_1(X)$
- $E_1(X)$ is of degree b-1
- Burst can be detected if $P(X)$ does not divide $E(X)$
- Since $P(X)$ is assumed not to have $X$ as a factor, it could divide $E(X)$ only if it could divide $E_1(X)$.
- But $b \leq n - k$
Theorem 5 Contd…

- Therefore \( P(X) \) is of higher degree than \( E_1(X) \) which implies that \( P(X) \) could not divide \( E_1(X) \)

- Q.E.D
Detection of a Burst-Error Cont’d

- Theorem 6: The fraction of bursts of length $b > n-k$ that are undetected is
  
  $2^{-(n-k)}$ if $b > n - k + 1$
  
  $2^{-(n-k-1)}$ if $b = n - k + 1$
Theorem 7: The cyclic code generated by $P(X) = (1 + X) P_1(X)$ detects any combination of two burst-errors of length two or less if the length of the code, $n$, is not greater than $e$, the exponent to which $P_1(X)$ belongs.

Proof

There are four types of error patterns

- $E(X) = X^i + X^j$
- $E(X) = (X^i + X^{i+1}) + X^j$
- $E(X) = X^j (X^j + X^{j+1})$
- $E(X) = (X^i + X^{i+1}) + (X^j + X^{j+1})$
Other Cyclic Codes

- There are several important cyclic codes which have not been discussed in this paper.
  - BCH codes by Chaudhuri, are a very important type of cyclic codes.
  - Reed-Solomon codes are a special type of BCH codes that are commonly used in compact disc players.
Implementation

- Briefly, to encode a message, $G(X)$, $n-k$ zeros are annexed (multiplication of $X^{n-1}G(X)$ is performed) and then $X^{n-1}G(X)$ is divided by the polynomial $P(X)$ of degree $n-k$. The remainder is then subtracted from $X^{n-1}G(X)$. (It replaces the $n-k$ zeroes).

- This encoded message is divisible by $P(X)$ for checking out errors.
It can be seen that modulo 2 arithmetic has simplified the division considerably.

Here we do not require the quotient, so the division to find the remainder can be described as follows.

1) Align the coefficient of the highest degree terms of the divisor and dividend and subtract (same as addition)

2) Align the coefficient of the highest degree terms of the divisor and difference and subtract again

3) Repeat the process until the difference has the lower degree than the divisor
The hardware to implement this algorithm is a shift register and a collection of modulo two adders. The number of shift register positions is equal to the degree of the divisor, $P(X)$, and the dividend is shifted through high order first and left to right.
As the first one (the coefficient of the high order term of the dividend) shifts off the end we subtract the divisor by the following procedure:

1. In the subtraction the high-order terms of the divisor and dividend always cancel. As the higher order term of the dividend is shifted off the end of the register, this part of the subtraction is done automatically.

2. Modulo two adders are placed so that when a one shifts off the end of the register, the divisor is subtracted from the contents of the register. The register then contains a difference that is shifted until another one comes off the end and then the process is repeated. This continues until the entire dividend is shifted into the register.
Fig. 1—A shift register for dividing by $1 + X^2 + X^4 + X^5$. 
Example: Polynomial division

\[ \begin{array}{c|ccccc}
1 & 1 & 1 & 0 & 1 & 0 \\
1 & 0 & 1 \\
\hline
0 & 1 & 0 & 0 & 1 & 0 \\
1 & 0 & 1 \\
\hline
0 & 1 & 1 & 0 \\
1 & 0 & 1 \\
\hline
1 & 1
\end{array} \]
Fig. 2—One method of encoding on detecting errors. (In this example, $P(X) = 1 + X^2 + X^4 + X^6$.)
CRC encoding

Message sent:

1 0 1 1 1 0 0 1 1 0 1 0
CRC decoding

1 0 1 1 0 0 1 1 0 1 0

0 0 0 0 0 0

0 0 0 0 0
To minimize the hardware it is desirable to use the same register for both encoding and error detection.

If circuit of fig: 3 is used for error detection, the remainder on dividing $X^{n-k}H(X)$ by $P(X)$ instead of remainder on dividing $H(X)$ by $P(X)$

This makes no difference, because if $H(X)$ is not evenly divisible by $P(X)$ than obviously $X^{n-k}H(X)$ will not be divisible either.
Error Correction: It is a much more difficult task than error detection.

It can be shown that each different correctable error pattern must give a different remainder after division by \( P(X) \).

Therefore error correction can be done.
Applications

- Following is a list of the most used CRC polynomials
  - CRC-12: $X^{12}+X^{11}+X^3+X^2+X+1$
  - CRC-16: $X^{16}+X^{15}+X^2+1$
  - CRC-CCITT: $X^{16}+X^{12}+X^5+1$
  - CRC-32:
    $$X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^8+X^7+X^5+X^4+X^2+X+1$$
Applications cont’d

- IBM 8-inch floppy disk (CRC-CCITT)
- now used in almost all disk controller devices
- XMODEM (or Christensen) file transmission protocol uses the CRC-CCITT
- 16 bits: [X25 standard]
- 32 bits Ethernet CRC-32
- Standardized for different uses in TCP/IP for **error** checking
- Burst errors in networks, disks
Conclusion

- Cyclic codes for error detection provides high efficiency and the ease of implementation.

- It provides standardization like CRC-8 and CRC-32
Questions