CprE 489
Lecture 10

Fall Semester 1997

Bit Stuffing Revisited

- Normally, all bits between the flags, 01111110, are bit stuffed.
- If some 0 becomes 1 then potentially a flag can be created, causing the frame to be in error.

  110111110011

  110111111011
- If a 1 becomes a 0 no flag is created, but one could become destroyed.

Error Correction and Detection

- Error correction: By sending extra bits errors can be detected and corrected.
- Error detection: By sending extra bits errors are simply detected and the offending frame is thrown away. A resend of the frame will eventually lead to correct data being received.
- Error detection are limited. There is a chance that incorrect data is received but not corrected.

  Example: Parity bits detect one bit errors but not two bit errors.

  0111101 clear bits

  01111011 transmitted bits

  parity bit = sum mod 2 of the clear bits

Codes

- We have error-correcting and error-detecting codes. If x has m bits, then c(x) is the code of x which has n = m + r bits.
- There are 2^m messages and 2^n potential codes. c is a one-to-one function from the set of m length messages to n length codes.
- A code that detects d bit errors must have the property that if x and y are different, then c(x) and c(y) differ by at least d + 1 bits. That is, have "Hamming distance" at least d+1 between them.

  Parity bit: c(x) = x parity(x)

  Suppose x not equal to y.

  case 1. x and y differ by at least 2 bits. Then c(x) and c(y) do also.

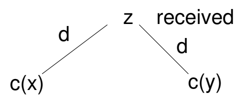
  case 2. x and y differ by one bit. The leading m bits differ by one bit and the parity bits differ. Thus, c(x) and c(y) differ in two bits.

  example: x = 0110 c(x) = 01100

  y = 0100 c(y) = 01001

Error Correcting Codes

- An error correcting code that corrects d bits must have the property that if x and y are different then c(x) and c(y) must have at least 2d + 1 bits different.



Example: c(00) = 000000

c(01) = 000111

c(10) = 111000

c(11) = 111111

This code will correct any one bit errors but not two bit errors.

Types of Errors

- Random single bit errors - usually caused by thermal noise
- Burst errors - caused by impulse noise

  A burst looks like

  The length of the burst depends on the transmission medium and other factors.

- It is important to detect burst errors.

  The parity bit and its relatives are poor at detecting burst errors.

  The parity bit is particularly bad: The two bad bits do not change the parity and the random bits mean that the parity bit will catch at most 50% of burst errors.

Cyclic Redundancy Code (CRC)

- Uses ideas from mod 2 polynomials. A little explanation is in order.
- A mod 2 polynomial only has coefficients 0 and 1. Addition, substraction, multiplication and division are all done modulo 2.

$x^4 + x^3 + 1$	11001	addition or subtraction
$x^4 + x^2 + x + 1$	10111	
$x^3 + x^2 + x$	01110	Binary repesentation of polynomials

Mod 2 Division

$$x^3 + x + 1 \overline{\big)\, x^7 + x^5 + x^3 + x + 1}$$

The degree of the remainder is less than the degree of the divisor.

Mod 2 Division Algorithm

```
1 0 1 1 | 1 0 1 0 1 0 1 1
```


General Set-up

- There is a generator polynomial G(x) of degree r.
- Transmitter
  1. Let M(x) be the polynomial corresponding to the message m.
  2. Divide x^r M(x) by G(x) to obtain a remainder R(x)
  3. Transmit the code corresponding to T(x) = x^r M(x) - R(x).

     Note that the degree of R(x) is less than r and that G(x) divides T(x).
- Receiver
  1. Suppose that A(x) = T(x) + E(x) arrives.
  2. Divide A(x) by G(x), if the remaider is not 0 then reject the frame.

     If the remainder is 0 then let m' be the message corresponding to A(x)/x^r

Properties of CRC

- If x+1 divides G(x) then CRC will detect frames with an odd number of errors.

  Suppose CRC accepts a frame. Then G(x) divides T(x) + E(x). Since G(x) divides T(x) then it must E(x). Thus x+1 must divide E(x).

  E(x) = (x+1) Q(x).

  E(1) = (1+1) Q(1) = 0.

  Thus, E(x) has an even number of terms.
- If G(x) has degree r and has the x^0 term then CRC will detect all burst of length $\leq$ r.

  A burst error of length $k \leq r$ is represented by a polynomial of the form $E(x)=x^i(x^{k-1}+...+1)$. If G(x) divides T(x) + E(x) then it divides E(x). Since G(x) is of the form H(x) +1 then G(x) must divide $x^{k-1}+...+1$. This is impossible because k-1 < r.

Common Generators

- CRC-12 1100000001111 $x^{12}+x^{11}+x^3+x^2+x+1$
- CRC-16 11000000000000101 $x^{16}+x^{15}+x^2+1$
- CRC-CCITT 10001000000100001 $x^{16}+x^{12}+x^5+1$

  CRC-CCITT which is used in X.25 detects all single and double bit errors, all errors on an odd number of bits, and all bursts of length 16 or less. In addition, 99.997% of burst of length 17 or greater.

  Calculating the CRC is relatively easy in hardware and software.