Previously

- HDLC – High Level Data Link Control
- PPP – Point-to-Point Protocol

Error Detection & Correction

- Error Correction
  - Hamming distance
- Error Detection
  - Parity bits
  - Block Sum Check
  - CRC

Errors

- Occurrence of errors
  - Rare on Digital Transmission
  - More frequent on analogue portions and wireless networks.
- Type of errors
  - Errors generally occur as isolated bits or alternatively as bursts.
  - Burst errors mean that frames are likely to get through but burst errors can be harder to detect and correct.
- Redundant Info is needed to
  - Just detect the error,
  - Correct the error. Used on wireless links where channel is unreliable.

Basics

- Frame consists of $n = m + r$ bits
  - m a frame consists of a number of message bits,
  - r together with a number of redundant bits
  - There are only m valid codewords.
- e.g. Parity bit: Is a single bit added to a piece of data
  - Even parity: Causes the number of 1 bits to be even,
  - Odd parity: Causes the number of 1 bits to be odd,
  - 1001010 → **1** (even) or **0** (odd)
  - There are only 2^7 = m(128) valid codewords
  - We can detect only an odd number of bits errors,
  - No way to correct them.

Forward Error Correction

- Hamming Distance is the number of bits different between two bit patterns of equal size,
  - Determine by applying an XOR
  - For example, distance is 2 and 5

10010100	00000111
10011110	11001100

- Hamming distance of a code, (i.e., the complete set of codewords) is the minimum distance between any two codes
  - E.g. Hamming distance is 3 in example
- To detect an error of d bits…
  - Hamming distance must be >= **d+1**
- To correct an error of d bits…
  - Hamming distance must be **2d+1**
  - where correction means changing the received data to the closest codeword.

00111
01100
10010
11001

Single bit correction

- To correct single bit errors…
  - Given 2^m messages, for each message there are **n+1** bit patterns dedicated to it (Corrupt each bit of the message)
  - $(n+1).2^m \leq 2^n$ Since the total number of bit patterns is $\mathbf{2^n}$
  - $(m+r+1) \leq 2^r$ With **n=m+r,** we reformulate
- Hamming codes achieves this lower limit

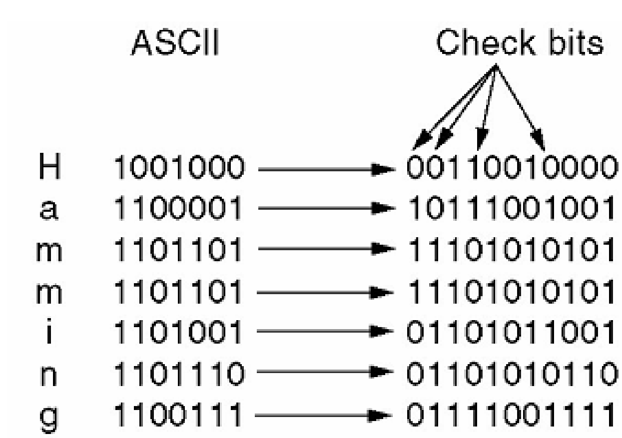
Hamming codes

- Hamming code achieves this lower limit
- Check bits: 1, 2, 4, 8, … If the position value is a power of 2 then the bit is used as a check bit
- Data bit indices: 3, 5, 6, 7, 9, 10, 11, … Otherwise is a data bit
- Check bits are parity bits computed from those data bits indices, whose binary decomposition includes the check bit:

Data bit:	1	2	3	4	5	6	7	8
Data bit index:	[3]	[5]	[6]	[7]	[9]	[10]	[11]	[12]
Binary decomposition:	1	1		1	1		1	
	2		2	2		2	2	
		4	4	4				4
					8	8	8	8

- If one check bit is incorrect, then it is the check bit which is in error,
- If multiple check bits are incorrect, the it is the bit whose index is the sum of the check bit indices that is wrong,
- To correct burst errors we change, the order of transmission and transmit columns of the matrix, rather than the rows.

Hamming code example



Block Sum Check

- Using a single parity bit the probability of an error not being detected is 50% in burst mode.
- To increase this we can use a Block Sum Check
  - Compute parity for each row AND for column of data,
  - Probability of an error not being detected is

    $2^{-n}*2^{-k}$

    Where **n** is the length of the row. However, a number of bit errors still causes problems.

P_R	B_6	B_5	B_4	B_3	B_2	B_1	B_0	
0	0	0	0	0	0	1	0	= STX
1	0	1	0	1	0	0	0	
0	1	1	0	0	0	1	0	
0	0	1	0	0	0	0	0	Frame contents
1	0	1	0	1	1	0	1	
0	1	0	0	0	0	0	0	
1	1	0	0	0	1	1	1	
1	0	0	0	0	0	1	1	= ETX
1	1	0	0	0	0	0	1	= BCC

Cyclic Redundancy Check

- Treats the bits in the frame as coefficients of a polynomial,
  - e.g. 110001 $= 1*x^5+1*x^4+0*x^3+0*x^2+0*x^1+1*x^0$
- Determine a Checksum which is
  - Data / Generator Polynomial
  - Checksum is usually 16/32 bits long,
  - Generator Polynomial is 1 bit longer,
  - Checksum is referred to as the FCS (Frame Check Sequence) or the CRC (Cyclic Redundancy Check)
  - Checksum appended to the end of the data frame.

Cyclic Redundancy Check

- Generator Polynomials
  - High & low bits must be 1
  - $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^{8} + x^{7} + x^{5} + x^{4} + x^{2} + x^{1} + 1$
  - $x^{16} + x^{15} + x^{2} + 1$
- Computing the checksum
  - Append zero bits $\rightarrow x^rM(x)$ where r is the order of the generator polynomial, (i.e., 16 for CRC-CCITT)
  - $x^rM(x) / G(x)$ This division is done modulo 2
  - Subtract the remainder from $x^rM(x)$
- To detect errors
  - Compute the CRC remainder on the received data stream, including the transmitted checksum,
  - If zero then checksum matches data, otherwise we have an error.

Example

```
                1 1 0 0 0 0 1 0 1 0
10011 | 1 1 0 1 0 1 1 0 1 1 0 0 0 0
        1 0 0 1 1
          1 0 0 1 1
          1 0 0 1 1
            0 0 0 0 1
            0 0 0 0 0
              0 0 0 1 0
              0 0 0 0 0
                0 0 1 0 1
                0 0 0 0 0
                  0 1 0 1 1
                  0 0 0 0 0
                    1 0 1 1 0
                    1 0 0 1 1
                      0 1 0 1 0
                      0 0 0 0 0
                        1 0 1 0 0
                        1 0 0 1 1
                          0 1 1 1 0
                          0 0 0 0 0
                            1 1 1 0
```

- $M(x) = 1101011011$
- $G(x) = 10011$
- $x^rM(x) = 11010110110000$
- Remainder = 1110
- Transmit 11010110111110

Data Link Layer, Topic 5.1.2, Error Detection and Correction

Introduction
Correction
Detection

CRC Performance

- E(x): We consider an error modeled as E(x)
- $E(x) = x^i$ is a single bit error. This is not divisible by G(x) as long as G(x) has 2 terms or more.
- $E(x) = x^i + x^j = x^j (x^{i-j} + 1)$ represents 2 single bit errors. Detectable as long G(x) is not divisible by X^k+1. Well known low order polynomials give this protection.
  - E.g. $x^{15}+x^{14}+1$
- To catch all Odd errors we make x+1 a factor of G(x).
- Burst errors
  - Catch all errors of length less than or equal to the number of check bits except for a burst error length equal to r+1, the only error which would get through is the generator itself.
  - Probability (Long errors are unnoticed) = $(1/2)^r$

13

Data Link Layer, Topic 5.1.2, Error Detection and Correction

Introduction
Correction
Detection

Exercise

- Given a message 100110111010 and a generator polynomial $(x^4 + x^3 + x^1 + 1)$ compute the CRC.
- Also, what errors can this CRC detect?
  - It can detect all single bit errors – it has more than 1 term,
  - It can detect all odd number error – it has (x+1) as a factor,
  - It can detect all burst errors of 4 bits or less,
  - However, it cannot detect all double bit errors.

14