Data Link Layer Introduction

- Data Link layer provides a 'error free' point-to-point bit pipe for transmission of network layer PDU's.
  - Framing
  - Error Detection
  - Error Control
  - Flow Control

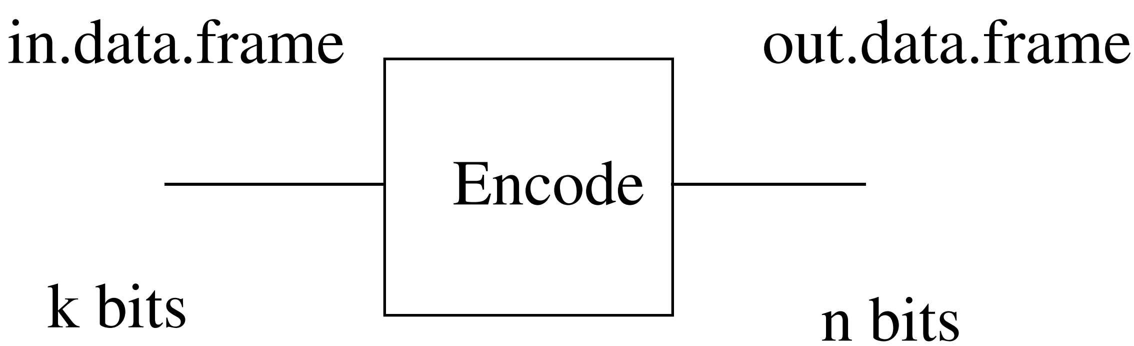
Framing

- Character Count
  - Insert PDU length as first character in the frame.
- Special Characters
  - Insert special characters at start and end of the frame (SOF, EOF characters, e.g... Data link escape (DLE) and End-of-text(EOT)).

Framing

- Flags
  - Insert special bit patterns, called ‘flags’ at start and end of the frame.
    - 01111110
- Symbol Stuffing
- Line Code Violations

Error Detection



$n>k$

$r = n - k$ =redundant bits added to detect errors

Error Detection

- Example: n=2k
  - Repeat the frame
  - k redundant bits

$$\text{Redundancy Ratio} = \frac{n}{k}$$

Error Detection

- Example: Simple Parity Check
- Even Parity
  - number of '1' per block is even
- Odd Parity
  - number of '1' per block is odd
- Redundancy ratio = k+1/k

Error Detection

- Performance criteria for error detection codes is the probability of an undetected error, P_e

Let p = probability of a bit error

and assume random errors.

then

Prob[1 bit error in a block of n bits]

$= p(1) = np(1-p)^{n-1}$

for m errors

$$p(m) = \frac{n!}{(n-m)!m!} p^m (1-p)^{n-m}$$

Error Detection

$$P_e = P(2) + P(4) + P(6) \ldots$$

Note $P(m) \approx 0$ for $M > 3$.

$$P_e = \frac{n(n-1)}{2} p^2 (1-p)^{n-2}$$

Example: Find the time between undetected errors for:

n = 1000
p = 1 in million
Rate = 100 Mb/s

$P_e = 5 \times 10^{-7}$

Time to transmit a block $= 10\mu s$

Time between undetected errors $= \dfrac{10\mu s}{5 \times 10^{-7}}$

$= 20$ seconds

Block Codes

Representation and manipulation of message blocks

k - bit message is represented by a binary vector of length k

$$\vec{d} = (d_1,\ d,\ d_3,\ \ldots.,d_k)$$

$$\text{Example,}\quad 01001100 \ \rightarrow (0,1,0,0,1,1,0,0)$$

Block Codes

n - bit code word is represented by a binary vector of length n

$\vec{c} = (c_1,\ c_2,\ c_3,\ \ldots.,c_n)$

Example,

$01001100110\ \rightarrow (0,1,0,0,1,1,0,0,1,1,0)$

Block Codes

Transformation between message and code vectors.

Let $c_1 = d_1,\ c_2 = d_2,\ c_3 = d_3, ..\ c_k = d_k$

$c_{k+1} = p_{11}d_1 \oplus p_{21}d_2 \oplus p_{31}d_3 \oplus p_{41}d_4 ... \oplus p_{k1}d_k$

$c_{k+2} = p_{12}d_1 \oplus p_{22}d_2 \oplus p_{32}d_3 \oplus p_{42}d_4 ... \oplus p_{k2}d_k$

.

.

$c_n = p_{1r}d_1 \oplus p_{2r}d_2 \oplus p_{3r}d_3 \oplus p_{4r}d_4 ... \oplus p_{kr}d_k$

Note $0 \oplus 0 = 0,\ 1 \oplus 0 = 1,\ 0 \oplus 1 = 1,\ 1 \oplus 1 = 0$

Block Codes

In matrix form $\vec{c} = \vec{d}G$

where

$$G = \begin{matrix} 1 & 0 & 0 & 0 & 0 & p_{11} & p_{12} & p_{13} & p_{14} & p_{15} & p_{1r} \\ 0 & 1 & 0 & 0 & 0 & p_{21} & p_{22} & p_{23} & p_{24} & p_{25} & p_{1r} \\ 0 & 0 & 1 & 0 & 0 & p_{31} & p_{32} & p_{33} & p_{34} & p_{35} & p_{1r} \\ 0 & 0 & 0 & 1 & 0 & p_{41} & p_{42} & p_{43} & p_{44} & p_{45} & p_{1r} \\ 0 & 0 & 0 & 0 & 1 & p_{k1} & p_{k2} & p_{k3} & p_{k4} & p_{k5} & p_{kr} \end{matrix}$$

$G = [I_k P]$ where I_k = k x k idenity matrix

G = Code Generator Matrix

Block Codes

- G Defines a *Linear Systematic (n,k) Code*

k Message Bits	r Code Bits

Code Word of n bits

Block Codes

- Example:

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix} \quad P = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

There are 8 unique code words

$$c_1 = d_1 G = [0 \quad 0 \quad 0] \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$c_2 = d_2 G = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

Block Codes

- Problem: given a received message determine if an error has occurred.

Model for the recieved message

$$\vec{r} = \vec{c} \oplus \vec{e}$$

where

$\vec{e} =$ error vector

$e_i = 1$ means that a error occured in the i^{th} bit of the code word

Block Codes

- Define a *Parity Check Matrix H*

$$H = [P^t I_r] = \begin{bmatrix} p_{11} & . & . & . & p_{k1} & 1 & 0 & 0 & 0 & 0 & 0 \\ . & . & . & . & . & 0 & 1 & 0 & 0 & 0 & 0 \\ . & . & . & . & . & 0 & 0 & 1 & 0 & 0 & 0 \\ . & . & . & . & . & 0 & 0 & 0 & 1 & 0 & 0 \\ . & . & . & . & . & 0 & 0 & 0 & 0 & 1 & 0 \\ p_{1r} & . & . & . & p_{kr} & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Block codes

For all 2^k valid code words,

it can be shown that $\vec{c}H^t = 0$

Then $\vec{r}H^t = (\vec{c} \oplus \vec{e})H^t = \vec{c}H^t \oplus \vec{e}H^t = \vec{e}H^t$

and if $\vec{e}H^t \neq 0$ then $\vec{e} \neq 0$ and an error has occured.

So if $\vec{r}H^t \neq 0$ then an error has occured.

$\vec{s}$ = Error Syndrome = $\vec{r}H^t$

Therefore, if $\vec{s} = 0$ then no error.

Block Codes

- Example:

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix} \quad P = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

$$H = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \quad H^t = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\vec{r} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$\vec{s} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \neq 0$$

An error has been detected.

Hamming Distance

A Geometric Interpretation of Error correcting Codes

- Hamming distance between two code words is defined as the number of differing bit positions.

Let

$c_i = [1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1]$ and

$c_j = [0 \quad 0 \quad 1 \quad 1 \quad 0 \quad 0]$

then the Hamming distance between

c_i and c_j is $d_{ij} = 2$.

Note $d_{ij} = \sum c_i \oplus c_j$

The minimum distance decoder selects the code word 'closest' in Hamming distance to the received message.

A minimum distance decoder can correct m errors where m is the largest integer not to exceed $1/2(d_m - 1)$, where d_m is the minumum hamming distance between any two valid code words.

Hamming Distance

Example of Hamming Distances

d	c
d1=00	c1=0000
d2=01	c2=0101
d3=10	c3=1010
d4=11	c4=1111

r	d1	d2	d3	d4
0000	0	2	2	4
0001	1	1	2	3
0010	1	3	1	3
0011	2	2	2	2
0100	1	1	2	3
0101	2	0	4	2
0110	2	2	2	2
0111	3	2	3	1
1000	1	2	1	3
1001	2	2	2	2
1010	2	4	0	2
1011	3	3	1	1
1100	2	2	2	2
1101	3	1	2	1
1110	3	3	1	1
1111	4	2	2	0

Binary Cyclic Codes

- A subclass of linear block codes are the binary cyclic codes.
- If c_k is a code word in a binary cyclic code then a lateral (or cyclic) shift of it is also a code word.

If $\begin{bmatrix} c_1 & c_2 & c_3 & c_4 & c_5 \end{bmatrix}$ is a code word

then $\begin{bmatrix} c_5 & c_1 & c_2 & c_3 & c_4 \end{bmatrix}$ and

$\begin{bmatrix} c_4 & c_5 & c_1 & c_2 & c_3 \end{bmatrix}$ can also be code words.

Binary Cyclic Codes

- Advantages:
  - Ease of syndrome calculation
  - Simple and efficient coding/ decoding structure
- Cyclic codes restrict the form of the G matrix
- The cyclic nature of these codes indicates that there is a underlying pattern.

Binary Cyclic Codes

Generator Polynomial g(x)

Note r = number parity bits

$$g(x) = 1x^{r} \oplus g_{r-1}x^{r-1} \oplus g_{r-2}x^{r-2} .... \oplus g_1 x \oplus 1$$

Example :

$$g(x) = x^{16} \oplus x^{15} \oplus x^{2} \oplus 1$$

Binary Cyclic Codes

- Construction of G from g(x)
  - 1) Use g(x) to form the k^{th} row of G.
  - 2) Use k^{th} row to form (k-1) row by a shift left, or xg(x).
  - 3) If (k-1) row in not in standard form then add kth row to shifted row, i.e.., (k-1) row becomes xg(x) + g(x).
  - 4) Continue to form rows from the row below.
  - 5) If the (k-j) row not in standard form the add g(x) to the shifted row.

Binary Cyclic Codes

$Example$:

$\mathrm{g(x)} = x^3 \oplus 0x^2 \oplus x \oplus 1$

$n = 7 \quad k = 4 \quad r = 3$

$Step\ 1: k^{th}\ \ of\ G \rightarrow [0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 1 \quad 1]$

$Step\ 2: k-1\ row\ of\ G \rightarrow [0 \quad 0 \quad 1 \quad 0 \quad 1 \quad 1 \quad 0]$

$Form\ ok.$

$Step\ 3: k-2\ row\ of\ G \rightarrow [0 \quad 1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 0]$

$$\begin{array}{lll} Form\ not\ ok. & & [0 \quad 1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 0] \\ & + & [0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 1 \quad 1] \\ & & [0 \quad 1 \quad 0 \quad 0 \quad 1 \quad 1 \quad 1] \end{array}$$

$Form\ ok.$

$Step\ 4: k-3\ row\ of\ G \rightarrow [1 \quad 0 \quad 0 \quad 1 \quad 1 \quad 1 \quad 0]$

$$\begin{array}{lll} Form\ not\ ok. & & [1 \quad 0 \quad 0 \quad 1 \quad 1 \quad 1 \quad 0] \\ & + & [0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 1 \quad 1] \\ & & [1 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 1] \end{array}$$

$Form\ ok. \qquad\qquad Done$

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

Binary Cyclic Codes

Codes for this example are:

$$\vec{c}_1 = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

$$\vec{c}_2 = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix}$$

Note that $\vec{c}_1$ and $\vec{c}_2$ are cyclic shifts of each other.

Binary Cyclic Codes

- Standard generator polynomials

$$CRC-16 \rightarrow g(x) = x^{16} + x^{15} + x^{2} + 1$$

$$CRC-CCITT \rightarrow g(x) = x^{16} + x^{12} + x^{5} + 1$$

$$CRC-12 \rightarrow g(x) = x^{12} + x^{11} + x^{3} + x^{2} + x + 1$$

$$AAL\ 3/4 \rightarrow g(x) = x^{10} + x^{9} + x^{5} + x^{4} + x + 1$$

$$AAL5 \rightarrow g(x) = x^{32} + x^{26} + x^{22} + x^{16} + x^{11} + x^{10} + x^{8} + x^{7} + x^{5} + x^{4} + x^{2} + x + 1$$

Binary Cyclic Codes

- Performance of standard codes
  - Detects all single errors
  - Detects all double errors
  - Detects all odd number of errored bits
  - Detects all burst errors of 16 or fewer bits
  - Detects 99.997% of all burst of 17 or fewer bits