Teoria da Informação - 060046 - UNISINOS

Aula 3 – 9/mar/2007

CRC – Cyclic Redundancy Check

A mensagem a ser enviada é tratada como uma seqüência de bits que representam os coeficientes de um polinômio, tendo a seguinte forma:

$$P_n(x) = \sum_{i=0}^{n} a_i x^i = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n,$$

A mensagem, "shiftada" a esquerda em n bits, é dividida pelo polinômio G, obtendo-se desta forma o resto R (o *checksum*), que é anexado a mensagem a ser enviada. Então,
no envio: $(D * X^n) / G \rightarrow R$ (Q é descartado) e ***na recepção***, é verificado se $(D * Xn + R) / G = = 0$

$(X^n*D)+R = (Q*G)+0$ onde **D**=data (a mensagem), **X^n**=0's inseridos a direita, **R**=resto **Q**=quociente, **G**=polinômio gerador

Exemplos:

1

mensagem= 101101001|000001011|011110010|001101110|111101101| 101001110|011101001|111010111

$$\begin{aligned} P(x) &= 1 \times x^{63} + 0 \times x^{62} + 1 \times x^{61} + 1 \times x^{60} + \cdots + 1 \times x^2 + 1 \times x^1 + 1 \times x^0 \\ &= x^{63} + x^{61} + x^{60} + \cdots + x^2 + x + 1. \end{aligned}$$

$$\mathrm{CRC}_{32}(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x^1 + 1.$$

CRC32=1 0000 0100 1100 0001 0001 1101 1011 0111

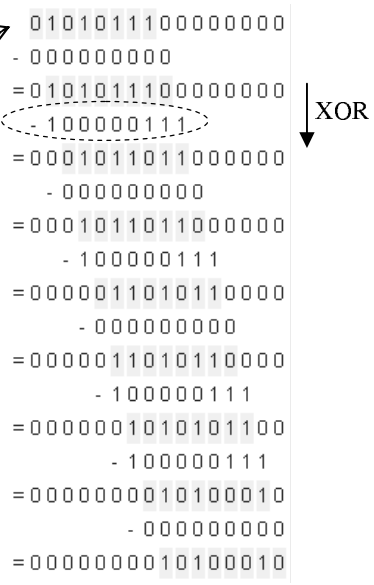
2

CRC-8 → $x^8 + x^2 + x + 1$ = 1 0000 0111 = 263 $_{dec}$ = 107 $_{hex}$

D=87 $_{dec}$ = 57 $_{hex}$ = 0101 0111

Adicionando bits zero a direita de D (D * Xn), temos

R = $x^7 + x^5 + x$ = 1010 0010 = A2 $_{hex}$ = 162 $_{dec}$



Alguns padrões para polinômios empregados em CRC:

CRC-5-USB	$x^5 + x^2 + 1$ (USB)
CRC-8-ATM	$x^8 + x^2 + x + 1$ (ATM)
CRC-8-CCITT	$x^8 + x^7 + x^3 + x^2 + 1$
CRC-12	$x^{12} + x^{11} + x^3 + x^2 + x + 1$ (telecomunicação)
CRC-16-CCITT	$x^{16} + x^{12} + x^5 + 1$ (XMODEM,X.25, V.41, Bluetooth, PPP, IrDA)
CRC-32-MPEG2	$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$ (e IEEE 802.3)
CRC-64-ISO	$x^{64} + x^4 + x^3 + x + 1$ (ISO 3309)

CRC-128 IEEE-ITU (substituído por MD5 & SHA-1)

Codificação da Fonte

Códigos de comprimento fixo

Fonte onde L seqüências geradas a partir do alfabeto A (de n símbolos), mapeados usando alfabeto-código B de m símbolos, então:

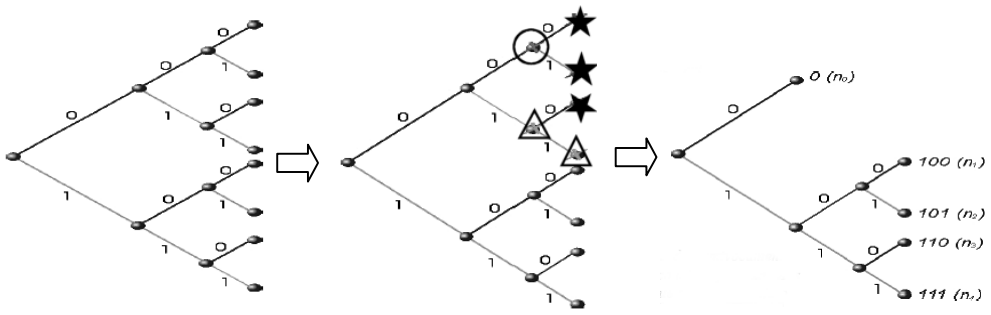
$$m^{\Lambda} \geq n^{L} \Rightarrow \frac{\Lambda}{L} \geq \log_m n$$ onde Λ denota o comprimento das *codewords*

Códigos de comprimento variável

Fonte onde alfabeto A={a1,a2,...,an}, alfabeto-código B= {b1,b2,...,bn} e λ =comprimento do *codeword* da i-ésima letra de A, então o comprimento médio das *codewords* será:

$$\overline{\lambda} = \sum_{i=1}^{n} p_i \lambda_i$$

Criando um código de condição Prefixa (univocamente decodificável): alfabeto A=5 letras (n) e B=2 (m)
Então, deve-se eliminar m^r-n nós, sendo **r**=comprimento do código tal que $m^r \geq n$



Teorema de codificação da fonte:

$$L \geq \frac{H(X)}{\log_2 D}$$

Codificação Huffman

Alfabeto $A=\{a_1, a_2, a_3, a_4, a_5\}$ $P=\{0,3;\ 0,25;\ 0,25;\ 0,1;\ 0,1\}$ H(A)=2,18bits