IT Basics
5

Prof. dr. Răzvan Daniel Zota
Cybernetics, Statistics and Economic Informatics Faculty
BUES
https://zota.ase.ro/itb

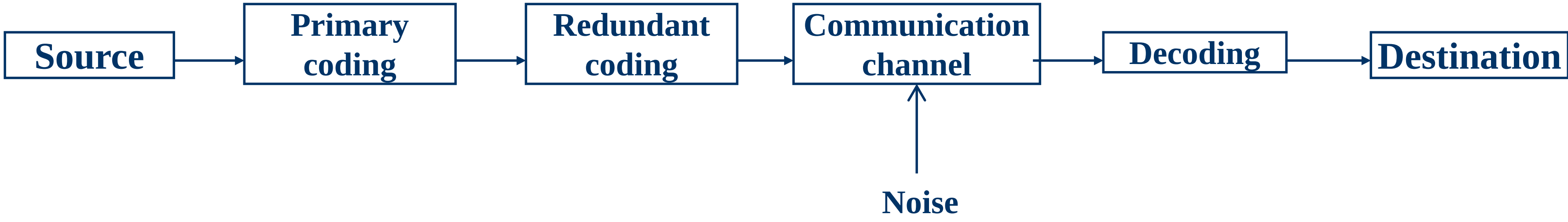
Error detection and/or correction codes

Error detection and/or correction codes detect and/or correct errors that appear in case of transmitting a message on a channel with noise.

The detection/correction is done by inserting of some redundancy in the initial message (instead of transmitting the original message, a longer message is transmitted, hoping that the added symbols will help detecting/correcting an error or errors).

Practically, any digital communication or data storage is using a kind of error detection coding. The CD-s, hard-disks, internal memories of the computers, flash memories, DVDs, etc., are protected against altering the data by using such codes.

Error detection and/or correction codes

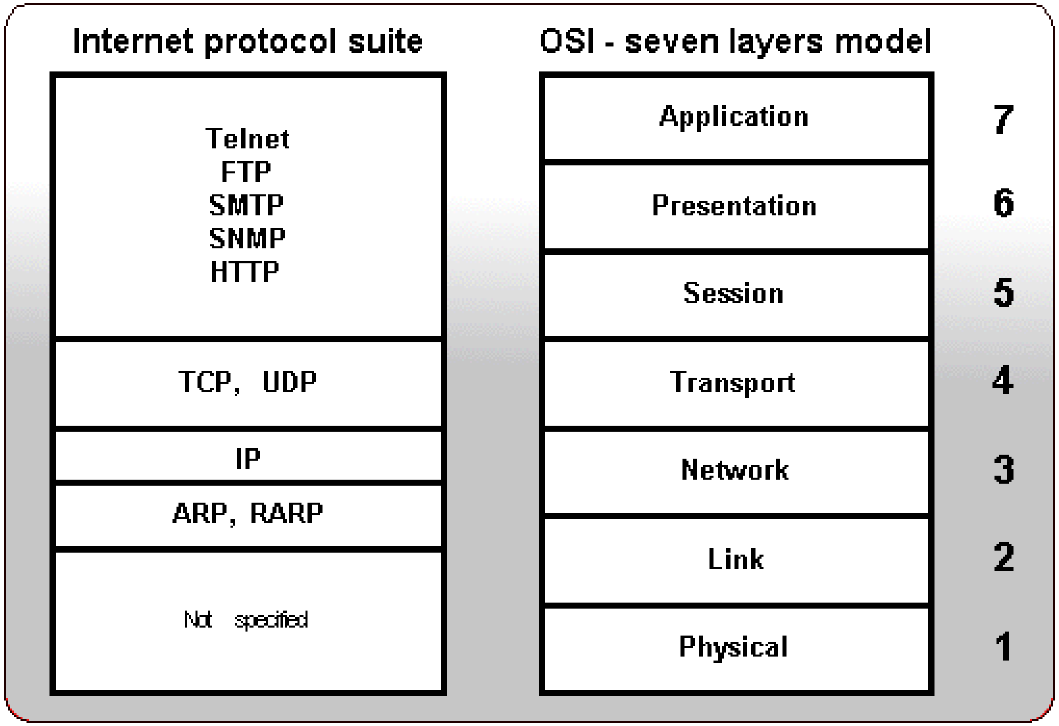


Techniques that enable reliable delivery of digital data over (unreliable) communication channels (usually subject to *noise*).

In this case, *noise* represents an error or undesired random disturbance of a useful information signal in a communication channel.

Error detection and/or correction codes

- They are used mostly at the data-link level (whose major functions are: error correction and flow control) of ISO-OSI model
- ISO = International Organization for Standardization
- OSI = Open System Interconnection



Code distance

- https://mathshistory.st-andrews.ac.uk/Biographies/Hamming/
- The *code distance (Hamming distance)* is a function defined by:

$$D(v_i, v_j) = \sum_{k=1}^{n} (a_{ik} \oplus a_{kj}), \quad where\ v_i = (a_{i1}, a_{i2}, ..., a_{in})\ and\ v_j = (a_{j1}, a_{j2}, ..., a_{jn})$$

The probability for the detection and correction of a code is depending of the minimum distance between two code words. It can be shown that for a code that can detect a number of *e* errors (in one of its sequences), it is necessary that:

$$D_{min} \geq e + 1$$

In order to detect *e* errors and correct *c* errors, the formula becomes:

$$D_{min} \geq e + c + 1$$

Calculus example of the *Hamming distance*

vector 1	codare	126359	01101011
vector 2	notate	226389	01001110
distanta Hamming	3	2	3

In case of binary representation, the Hamming code is given by the number of bits of 1 from the XOR result (bit by bit) between these two representations.

Hamming code

(Richard) Hamming code

- It detects and corrects ***one*** error.

If $\boldsymbol{n}$ = number of symbols of the code word

n = k + m, k = number of the control symbols,

m = number of the information symbols

To assume detection and correction of an error, it must be satisfied the condition: $2^m \geq n + 1$

$(2^m \geq m + k + 1)$

Hamming code (cont.)

- Control bits are on the positions: 2^0, 2^1, 2^2, 2^3, etc.
- On the rest of the positions there are the information bits.
- We shall write a code word ***v*** like: $c_1c_2i_3c_4i_5i_6\ldots i_n$

Hamming code (cont.)

- At the source the coding is done:

Particular case: n=7, so $\boldsymbol{v} = c_1c_2i_3c_4i_5i_6i_7$

In this case we have 4 information bits and 3 correction bits.

Redundancy rate: ¾=75%

$$\begin{cases} c_4 = i_5 + i_6 + i_7 \\ c_2 = i_3 + i_6 + i_7 \\ c_1 = i_3 + i_5 + i_7 \end{cases}$$

Hamming code (cont.)

- At the destination, the message is checked for errors (correction)

Assuming that we received the message: $\boldsymbol{v'} = c_1'c_2'i_3'c_4'i_5'i_6'i_7'$

We compute the error bits e_1, e_2, e_4 as:

$$\begin{cases} e_1 = c'_1 \oplus i'_3 \oplus i'_5 \oplus i'_7 \\ e_2 = c'_2 \oplus i'_3 \oplus i'_6 \oplus i'_7 \\ e_4 = c'_4 \oplus i'_5 \oplus i'_6 \oplus i'_7 \end{cases}$$

If all these 3 bits are zero, then the message is received correctly; otherwise, the message is wrong.

The position of the error is given by the value of the binary number $\overline{e_4e_2e_1}$, transformed into decimal.

Hamming code (cont.)

- Another particular example:

n=12, so $\boldsymbol{v} = c_1c_2i_3c_4i_5i_6i_7c_8i_9i_{10}i_{11}i_{12}$

In this case we have 8 information bits and 4 correction bits

Redundancy rate: 4/8=50%

Formulas for redundant/correction bits?

Hamming code applied today

Random Access Memory (RAM):

This is probably the most common application of Hamming codes. Servers, high-performance workstations, and sometimes even personal computers use ECC RAM (Error-Correcting Code RAM).

Bits can be affected by "noise" or various external events. ECC memory uses the Hamming code (or variants thereof) to detect and correct single-bit errors in real time, thus preventing system crashes or data corruption.

It is crucial to use an error-detecting code in systems where data reliability is essential (e.g. database servers, financial systems, scientific computing, etc.).

Linear codes with cross control

- In this case there are transmitting blocks of information
  - Transversal parity (for lines)

$$\underset{(i=\overline{1,m})}{l_i} = \begin{cases} \bigoplus_{k=1}^{n} a_{ik} \text{ realizeaza paritatea para} \\ \bigoplus_{k=1}^{n} a_{ik} \oplus 1 \text{ realizeaza paritatea impara} \end{cases}$$

	Information bits	Line control
	a_{11} a_{12} ....a_{1n} ………… ………… a_{m1} a_{m2} ....a_{mn}	l_1 . . l_m
Column control	c_1 c_2.......c_n	

Linear codes with cross control (cont.)

- Longitudinal parity (for columns)

$$\underset{(j=\overline{1,n})}{c_j} = \begin{cases} \bigoplus_{k=1}^{m} a_{kj} & \text{realizeaza paritatea para} \\ \bigoplus_{k=1}^{m} a_{kj} \oplus 1 & \text{realizeaza paritatea impara} \end{cases}$$

Linear codes with cross control (cont.)

- Correction

	Information bits	Line control
	a'_{11} a'_{12} …………..a'_{1n}	l'_1
	a'_{21} a'_{22} …………..a'_{2n}	l'_2
	………………………	.
	………………………	.
	a'_{m1} a'_{m2} …………..a'_{mn}	l'_m
Column control	c'_1 c'_2 ………………c'_n	l'_{m+1} (c'_{n+1})

Polynomial cyclic codes

The cyclic codes are block codes where the *n+1* symbols which are making a code sequence are considered as being the coefficients of a *n* degree polynomial, like:

$M(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots\ldots + a_0$

where $a_i \in \{0, 1\}$, $i = 1..n$.

When using the cyclic polynomial codes, a polynomial *M(x)* is associated to message *M*.

In the following, using a coding algorithm, *M(x)* is transforming in another polynomial *T(x)*, such as *T(x)* will be a multiple of polynomial *G(x)* – denoted ***generator polynomial***.

Polynomial cyclic codes (cont.)

For the coding part it can be used a *multiplying* or a *dividing* algorithm.

Using the multiplying algorithm: $T(x)=M(x)\cdot G(x)$ (the multiplying and addition operations are made *modulo 2*) there is no separation between the redundant and informational bits – this being the main reason for which the *dividing algorithm is preferred*.

Polynomial cyclic codes (cont.)

The coding dividing algorithm is the following:

• Let the message M: $(a_n, a_{n-1}, \ldots, a_0)$, with $n+1$ binary information digits (bits). We associate it a polynomial in indeterminate x:

$M(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots\ldots + a_0$ ($a_i \in \{0, 1\}$);

• We choose the polynomial G(x) of r degree, being the *generator polynomial* of the code: $G(x) = b_r x^r + b_{r-1} x^{r-1} + \ldots\ldots + b_0$ $b_j \in \{0, 1\}$,

• Multiplying M(x) by x^r we will have $M'(x) = M(x) \cdot x^r$

• We divide M'(x) to G(x)

$$\frac{M'(x)}{G(x)} = C(x) \oplus \frac{R(x)}{G(x)} \quad (1)$$

Polynomial cyclic codes (cont.)

The degree of the polynomial R(x) will be less or equal with *r-1*. The coefficients of polynomial R(x), of degree *r-1*, will represent the control bits associated to the information bits.

• We add R(x) with M'(x) obtaining the polynomial T(x) = M'(x) ⊕ R(x).

The coefficients of the polynomial T(x) will represent the message to be transmitted:

T: $(a_n a_{n-1} \ldots a_0 c_{r-1} \ldots c_0)$ which contains on the most significant positions the *n+1* information bits and in the least significand positions the *r* control symbols.

The attached polynomial to the transmitted message is a multiple of the generator polynomial. We have the following:

$$\frac{T(x)}{G(x)} = \frac{M'(x) \oplus R(x)}{G(x)} = \frac{M'(x)}{G(x)} \oplus \frac{R(x)}{G(x)}$$

Polynomial cyclic codes (cont.)

Replacing $\frac{M'(x)}{G(x)}$ by relation (1) we will get the following:

$$\frac{T(x)}{G(x)} = C(x) \oplus \underbrace{\frac{R(x)}{G(x)} \oplus \frac{R(x)}{G(x)}}_{=0} = C(x)$$

As to this relation, T(x) is a multiple of G(x).

This characteristic is used as an ***error detection criteria***.

Polynomial cyclic codes (cont.)

Having the received message T', we associate to it the polynomial T'(x).
We may write that T'(x)=T(x) ⊕ E(x), where E(x) is the ***error polynomial***.

By applying the error detection criteria, we get:

$$\frac{T'(x)}{G(x)} = \frac{T(x) \oplus E(x)}{G(X)} = \frac{T(x)}{G(x)} \oplus \frac{E(x)}{G(x)} = C(x) \oplus \frac{E(x)}{G(x)}$$

Polynomial cyclic codes (cont.)

It can be noticed that if E(x) is a multiple of G(x), the received message is validated, even if it contains errors.

If E(x) it is not a multiple of G(x) then the error is detected.

By this method are detected all the error packets with a length less than the degree of G(x)+1.

It is called an *error packet* a sequence of symbols, correct or not, where the first and the last symbol are wrong.

Coding example

Problem 1. The binary message M: 1110101 is transmitted after encoding using the generating polynomial $G(x) = x^3 + x + 1$. What is the binary representation of the transmitted message?

Solution: The binary message M: 1110101 is associated with the polynomial M(x):

- $M(x) = x^6 + x^5 + x^4 + x^2 + 1$;
- $M'(x) = M(x) \cdot x^3$, since the degree of G(x) is 3;
- $M'(x) = x^9 + x^8 + x^7 + x^5 + x^3$;

We divide M'(x) by G(x):

$$
\begin{array}{l|l}
x^9 + x^8 + x^7 + x^5 + x^3 & x^3 + x + 1 \\
x^9 + x^7 + x^6 & x^6 + x^5 + 1
\end{array}
$$

$x^8 + x^6 + x^5 + x^3$

$x^8 + x^6 + x^5$

----

x^3

$x^3 + x + 1$

$x + 1 = R(x)$

(Addition and subtraction in modulo 2 are equivalent.)

$R(x) = x + 1$

We obtain the polynomial $T(x) = M'(x) \oplus R(x)$:

$T(x) = x^9 + x^8 + x^7 + x^5 + x^3 + x + 1$

The coefficients of this polynomial represent the message that will be transmitted: **1110101011**

Error checking example

Problem 2. Knowing that the received message T': 1010101011 was transmitted after encoding using the generating polynomial $G(x) = x^3 + x + 1$, verify its correctness.

Solution:
The received message T' is associated with the polynomial:
$T'(x) = x^9 + x^7 + x^5 + x^3 + x + 1$.
By applying the error detection criterion, we get:

Error checking example

$$
\begin{array}{l|l}
x^9 + x^7 + x^5 + x^3 + x + 1 & x^3 + x + 1 \\
x^9 + x^7 + x^6 & x^6 + x^3 + x^2 + x + 1
\end{array}
$$

$x^6 + x^5 + x^3 + x + 1$

$x^6 + x^4 + x^3$

$x^5 + x^4 + x + 1$

$x^5 + x^3 + x^2$

$x^4 + x^3 + x^2 + x + 1$

$x^4 + x^2 + x$

$x^3 + 1$

$x^3 + x + 1$

$x = R(x)$

Thus, the received message is incorrect because $E(x) \neq 0$.

Key points – CRC codes

Cyclic polynomial codes

Why are they called "cyclic"?

Because the operations of division with remainder are similar to polynomial arithmetic modulo a given polynomial.

Advantages: CRCs are extremely effective at detecting single errors, burst errors (several adjacent bits corrupted), and most multiple errors. They are relatively simple to implement in hardware.

Disadvantages: They are not error-correcting codes (they cannot repair errors themselves, they only detect them). There is a small probability that an error will result in a remainder of zero, in which case the error is not detected.

Key points – CRC codes

Where they are used in practice:

- Computer networks: Ethernet, Wi-Fi, Frame Relay.
- Storage: Hard drives, SSDs (for data integrity), SD cards.
- Archives: ZIP, RAR files (to verify archive integrity).
- Communication protocols: A number of communication protocols use CRC to ensure reliability.