Bazele Tehnologiei Informaţiei
Curs 5

Prof. dr. Răzvan Daniel Zota
Facultatea de Cibernetică, Statistică şi Informatică Economică
ASE Bucureşti
https://zota.ase.ro/bti

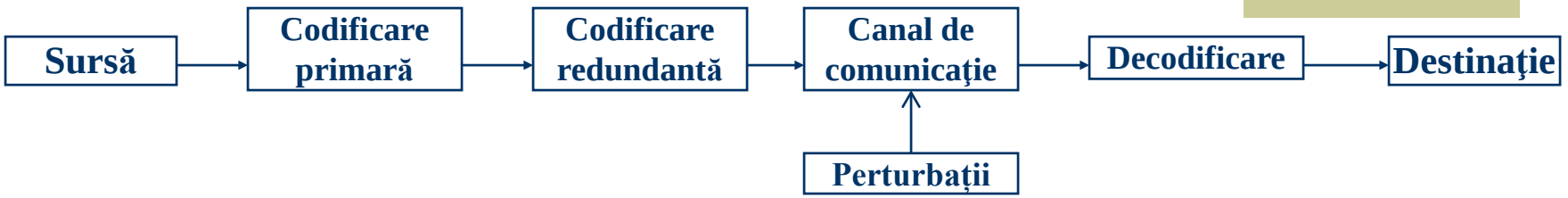
Coduri detectoare şi/sau corectoare de erori

Codurile detectoare și /sau corectoare de erori detectează și/sau corectează erori care apar inevitabil la transmiterea unui mesaj pe un canal cu zgomot.

Această detectare/corectare se realizează prin introducerea de redundanţă în mesaj (adică în loc de a transmite mesajul original, un mesaj mai lung este transmis, în speranţa că simbolurile adăugate vor ajuta la detecţia/corectarea erorilor).

Practic, orice comunicare digitală, orice stocare de date foloseşte o formă de coduri corectoare de erori. Compact discurile, hard-discurile, memoriile interne ale calculatoarelor, memoriile de tip flash, DVD-urile, etc., sunt protejate împotriva alterării accidentale a datelor folosind astfel de coduri.

Coduri detectoare şi/sau corectoare de erori



La sursă are loc codificarea; codificarea redundantă adaugă informația de control.

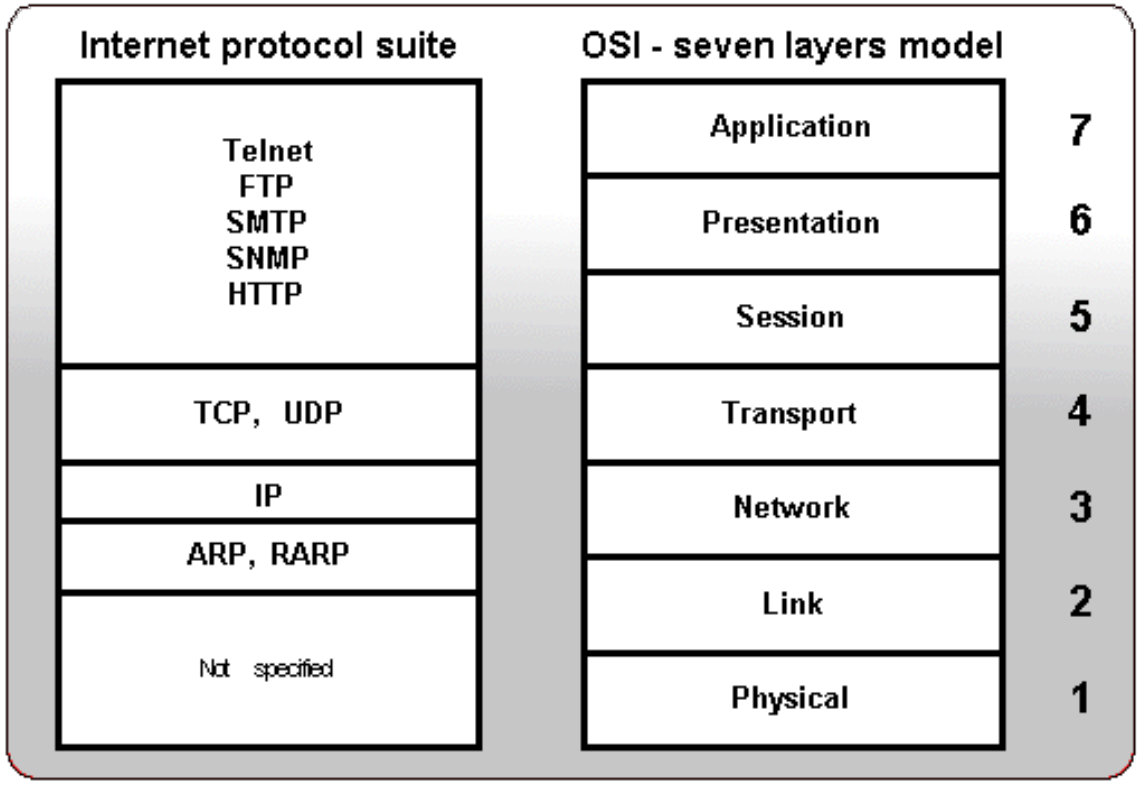
La destinație are loc decodificarea + detecția și/sau corecția erorilor.

Tipuri de coduri:

- *Coduri bloc* – pentru care prelucrările necesare obţinerii proprietăţilor de detecţie sau de corecţie se fac în blocuri de câte *n* simboluri.
- *Coduri convoluţionale (recurente)*: în acest caz prelucrarea simbolurilor generate de sursă se realizează în mod continuu.

Coduri detectoare şi corectoare de erori

- Acţionează de regulă la nivelul data-link (corecţia erorilor şi controlul fluxului) din modelul ISO-OSI (International Organization for Standardization - Open System Interconnection)



Distanţa de cod

- https://mathshistory.st-andrews.ac.uk/Biographies/Hamming/
- Distanţa de cod (Hamming) este o funcţie definită de:

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

Probabilitatea de detecţie şi corecţie a unui cod depinde de distanţa minimă între două cuvinte de cod. Se poate demonstra că pentru un cod ce poate detecta un număr de *e* erori existente într-una din secvenţele sale, este necesar ca:

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

Pentru detectarea unui număr de *e* erori şi corectarea de *c* erori, formula devine:

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

Exemplu de calcul al distanţei de cod

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

În cazul reprezentării binare, distanța de cod este dată de numărul de biți egali cu 1 din rezultatul obținut prin operația XOR bit cu bit dintre cele două reprezentări.

Exemple: codul Hamming

Codul Hamming

- Detectează și corectează o singură eroare.

Notăm cu ***n*** **–** numărul de simboluri ale cuvântului de cod

n = k + m, k = numărul simbolurilor de control,

m = numărul simbolurilor de informaţie

Pentru a se putea asigura detecţia şi corecţia unei erori,

$2^m \geq n + 1$

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

Codul Hamming: caracteristici

- Cifrele de control se află pe pozițiile 2^0, 2^1, 2^2, 2^3, etc.
- Pe restul poziţiilor se află cifrele de informaţie.
- Un cuvânt de cod ***v*** se va scrie:

$$\boldsymbol{v} = c_1 c_2 i_3 c_4 i_5 i_6 \ldots i_n$$

Codul Hamming (cont.)

- La **sursă** are loc **codificarea**

Caz particular: n=7, rezultă $\boldsymbol{v} = c_1c_2i_3c_4i_5i_6i_7$

c_1, c_2, c_4 se calculează după următoarele formule:

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

Codul Hamming (cont.)

- La **sursă** are loc **codificarea**

Caz particular: n=12, rezultă ***v*** = $c_1c_2i_3c_4i_5i_6i_7c_8i_9i_{10}i_{11}i_{12}$

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

Codul Hamming (cont.)

- La **destinaţie** are loc **verificarea mesajului** (corecţia)

Presupunând că am recepționat mesajul: $v' = c_1'c_2'i_3'c_4'i_5'i_6'i_7'$

Se vor calcula biții de eroare e_1, e_2, e_4 astfel:

$$e_1 = c'_1 \oplus i'_3 \oplus i'_5 \oplus i'_7$$

$$e_2 = c'_2 \oplus i'_3 \oplus i'_6 \oplus i'_7$$

$$e_4 = c'_4 \oplus i'_5 \oplus i'_6 \oplus i'_7$$

Dacă toți cei trei biți de eroare sunt zero, atunci mesajul este recepționat corect; altfel, mesajul este eronat.

Eroarea se află pe poziția dată de valoarea numărului binar $\overline{e_4 e_2 e_1}$, transformată în zecimal.

Codul Hamming – aplicații practice astăzi

- Memorii RAM (Random Access Memory) în calculatoare:

Aceasta este probabil cea mai răspândită aplicație a codurilor Hamming. Serverele, stațiile de lucru de înaltă performanță și, uneori, chiar și calculatoarele personale folosesc memorii ECC RAM (Error-Correcting Code RAM).

Biții pot fi afectați de "zgomot" sau diverse evenimente externe. Memoria ECC folosește codul Hamming (sau variante ale acestuia) pentru a detecta și corecta erori de un singur bit în timp real, prevenind astfel blocajele sistemului sau coruperea datelor.

Este crucială utilizarea unui cod de detecție a erorilor în sistemele unde fiabilitatea datelor este esențială (ex: servere de baze de date, sisteme financiare, calcul științific, etc.).

Coduri liniare cu control încrucişat

- Se transmit blocuri de informaţie
  - Paritate laterală (transversală)

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

	Simboluri informaţionale	Controlul liniei
	$a_{11}\ a_{12}\ ....a_{1n}$	l_1
	.....	.
	.....	.
	$a_{m1}\ a_{m2}\ ....a_{mn}$	l_m
Control coloană	$c_1\ c_2.....c_n$	

Coduri liniare cu control încrucişat

- Paritate longitudinală

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

Coduri liniare cu control încrucişat

- Corecţia la primire

	Simboluri informaţionale	Controlul liniei
	$a'_{11}\ a'_{12}$ ……a'_{1n}	l'_1
	$a'_{21}\ a'_{22}$ ……a'_{2n}	l'_2
	……	.
	……	.
	$a'_{m1}\ a'_{m2}$ ……a'_{mn}	l'_m
Control coloană	$c'_1\ c'_2$ ……c'_n	l'_{m+1} (c'_{n+1})

Coduri polinomiale ciclice

Codurile ciclice sunt coduri bloc în care cele *n+1* simboluri ce formează o secvenţă de cod sunt considerate ca fiind coeficienţii unui polinom de grad *n* şi anume:

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

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

În cazul utilizării codurilor polinomiale ciclice, mesajului M ce se va transmite i se asociază polinomul M(x).

În continuare, printr-un algoritm de codificare, M(x) se transformă într-un polinom T(x), astfel încât T(x) să fie multiplu al polinomului G(x) - numit ***polinom de generare***.

Coduri polinomiale ciclice (cont.)

Pentru realizarea codificării se pot utiliza algoritmul de înmulţire sau algoritmul de împărţire.

Folosind algoritmul de înmulţire: $T(x)=M(x)\cdot G(x)$ (operaţiile de înmulţire şi adunare ale coeficienţilor polinoamelor se fac modulo 2) nu se obţine o separare a simbolurilor redundante de cele informaţionale, acesta fiind principalul motiv pentru care se preferă algoritmul de împărţire, deşi este mai complicat.

Coduri polinomiale ciclice (cont.)

Algoritmul de codificare prin împărţire este:

• Fie mesajul M: $(a_n,a_{n-1},.....,a_0)$, care cuprinde *n+1* cifre binare informaţionale. Acestuia i se asociază un polinom în nedeterminata x:

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

• Se alege polinomul G(x) de grad *r*, acesta fiind polinomul de generare al codului: $G(x) = b_r x^r + b_{r-1}x^{r-1} + \ldots.. + b_0$ bj $\in \{0, 1\}$,

• Înmulţind M(x) cu x^r se va obţine $M'(x)=M(x)\cdot x^r$

• Se împarte M'(x) la G(x)

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

Coduri polinomiale ciclice (cont.)

Gradul polinomului R(x) va fi mai mic, cel mult egal cu *r-1*. Coeficienţii polinomului R(x), de grad *r-1*, constituie simbolurile de control asociate mesajului informaţional.

• Se adună R(x) cu M'(x) obţinâdu-se polinomul T(x) = M'(x) ⊕R(x).

Coeficienţii polinomului T(x) constituie mesajul ce se va transmite:

T: ($a_n a_{n-1}....a_0 c_{r-1}.....c_0$) care conţine în poziţiile semnificative cele *n+1* simboluri informaţionale iar în poziţiile mai puţin semnificative cele *r* simboluri de control.

Polinomul ataşat mesajului transmis este un multiplu al polinomului de generare. Avem:

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

Coduri polinomiale ciclice (cont.)

Înlocuind $\frac{M'(x)}{G(x)}$ prin relaţia (1) se va obţine:

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

Conform relaţiei de mai sus, T(x) este divizibil prin G(x).

Această proprietate este folosită drept ***criteriu pentru detecţia erorilor***.

Coduri polinomiale ciclice (cont.)

Fie mesajul recepţionat T', acestuia i se asociază polinomul T'(x). Putem scrie că T'(x)=T(x) ⊕E(x), unde E(x) este polinomul erorilor.

Aplicând criteriul de detecţie a erorilor, obţinem:

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

Coduri polinomiale ciclice (cont.)

Se observă că dacă E(x) este multiplu al lui G(x), mesajul recepţionat este validat, deşi conţine erori.

Dacă E(x) nu este multiplu al lui G(x) atunci eroarea este sesizată.

Prin această metodă sunt determinate toate pachetele de erori de lungime mai mică decît gradul lui G(x)+1.

Se numeşte ***pachet de erori*** o succesiune de simboluri, corecte sau eronate, în care primul şi ultimul simbol sunt eronate.

Exemplu de codificare

Exemplul 6. Mesajul binar *M: 1110101* se transmite după codificarea prin polinomul de generare $G(x) = x^3 + x + 1$. Care este reprezentarea binară a mesajului transmis?

Rezolvarea.

Mesajului binar M : 1110101 se asociază polinomul M(x):

- $M(x) = x^6 + x^5 + x^4 + x^2 + 1$;
- Se calculează $M'(x) = M(x) \cdot x^3$; deoarece gradul lui G(x) este 3;
- $M'(x) = x^9 + x^8 + x^7 + x^5 + x^3$
- Se împarte M'(x) la G(x):

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

Exemplu de codificare (cont.)

$x^9 + x^8 + x^7 + x^5 + x^3 \quad | x^3 + x + 1$

$\quad | \text{---------}$

$x^9 + x^7 + x^6 \quad | x^6 + x^5 + 1$

$/ \quad x^8 + x^6 + x^5 + x^3$

$\quad x^8 + x^6 + x^5$

$\quad$ -------------------------

$\quad / \quad / \quad / \quad x^3$

$\qquad x^3 + x + 1$

$\qquad$ -----------

$\qquad / \quad x + 1$

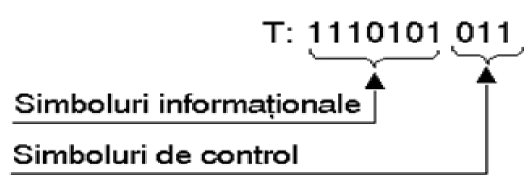
(Adunarea şi scăderea în modulo 2 sunt echivalente)

$R(x) = x + 1$

Se obţine polinomul $T(x) = M'(x) \oplus R(x)$

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

Coeficienţii acestui polinom reprezintă mesajul ce se va transmite:



Exemplu de verificare

Exemplul 7. Ştiind că mesajul recepţionat *T' : 1010101011* a fost transmis după codificarea prin polinomul de generare $G(x) = x^3 + x + 1$, să se verifice corectitudinea lui.

Rezolvarea. Mesajului recepţionat T' se asociază polinomul:
$T'(x) = x^9 + x^7 + x^5 + x^3 + x + 1$.

Prin aplicarea criteriului de detecţie a erorilor se obţine:

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

$x^9 + x^7 + x^5 + x^3 + x + 1$ | $x^3 + x + 1$

$x^9 + x^7 + x^6$ | $x^6 + x^3 + x^2 + x + 1$

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

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

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

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

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

$x^4 + x^2 + x$

$x^3 + 1$

$x^3 + x + 1$

x

Aşadar, mesajul recepţionat este eronat deoarece $E(x) \neq 0$.

Rezumat - 1

Coduri polinomiale ciclice

- De ce se numesc "ciclice"? Deoarece operațiile de împărțire cu rest sunt similare cu aritmetica polinomială modulo un anumit polinom.
- **Avantaje:** CRC-urile sunt extrem de eficiente în detectarea erorilor izolate, a erorilor în "burst" (mai mulți biți adiacenți alterați) și a majorității erorilor multiple. Sunt relativ simplu de implementat hardware.
- **Dezavantaje:** Nu sunt coduri de corecție (nu pot repara singure erorile, doar le detectează). Există o mică probabilitate ca o eroare să ducă la un rest de zero, caz în care eroarea nu este detectată.

Rezumat - 2

Unde se folosesc în realitate:

- Rețele de calculatoare: Ethernet, Wi-Fi, Frame Relay.
- Stocare: Hard disk-uri, SSD-uri (pentru integritatea datelor), carduri SD.
- Arhive: Fișiere ZIP, RAR (pentru a verifica integritatea arhivelor).
- Protocoale de comunicație: o serie de protocoale de comunicație folosesc CRC pentru a asigura fiabilitatea.