Implementarea conceptului de DF cu ajutorul codurilor cu rată finită.

Utilizarea codurilor LDPC pentru implementarea conceptului DF

TACCFDRT Curs 3

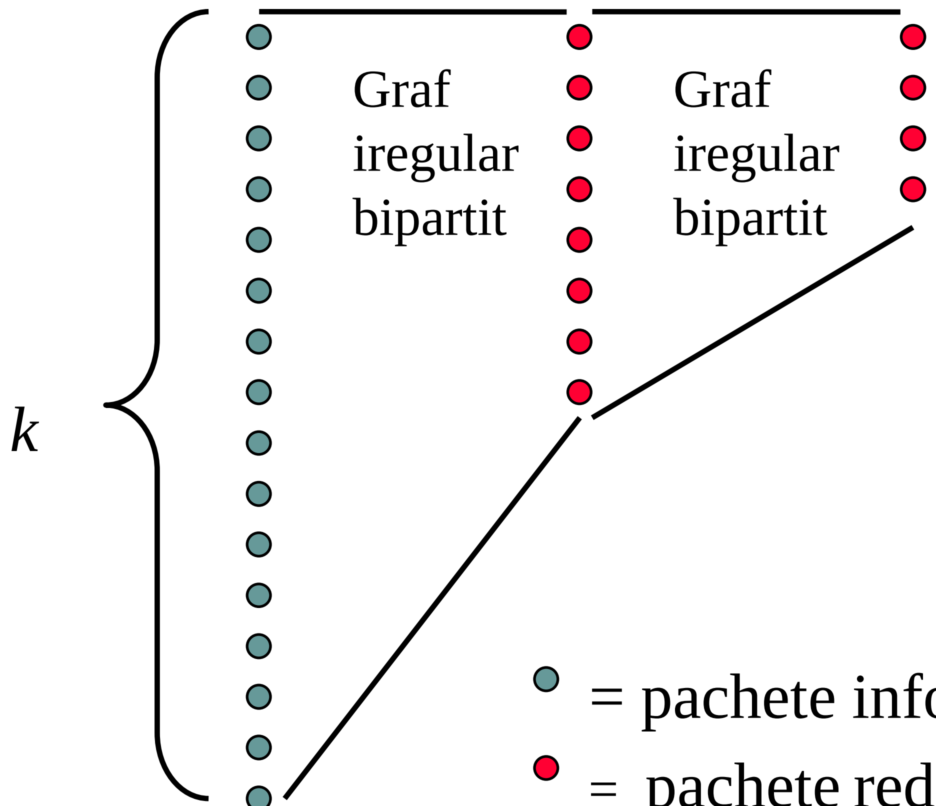
Tehnici de codare de tip Digital Fountain

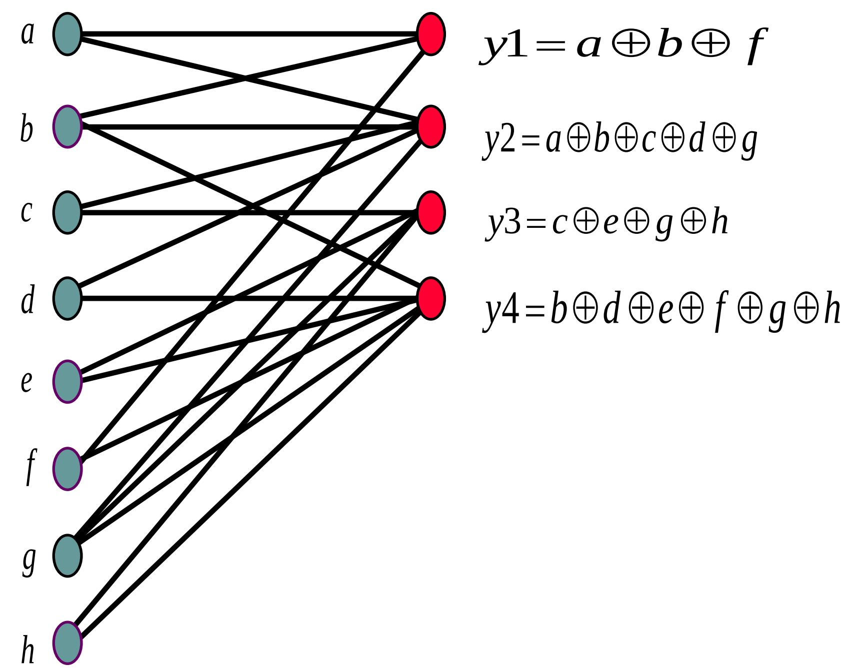
- Coduri Tornado
- Coduri Raptor
  - Coduri LDPC
- Tipuri de coduri Raptor
- Utilizarea codurilor DF

- Deoarece gradul pachetelor nu este constant
  - Timpul de codare/decodare nu este liniară
  - Probabilitatea de pierdere a pachetelor nu este uniformă
- Problema este rezolvată de codurile Raptor introduse de Amin Shokrollahi

Coduri Tornado

- Codurile Tornado sunt construite pe baza unor grafuri bipartite



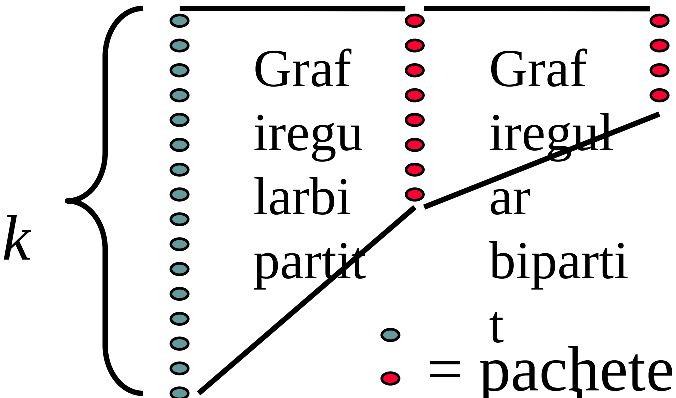


= pachete info

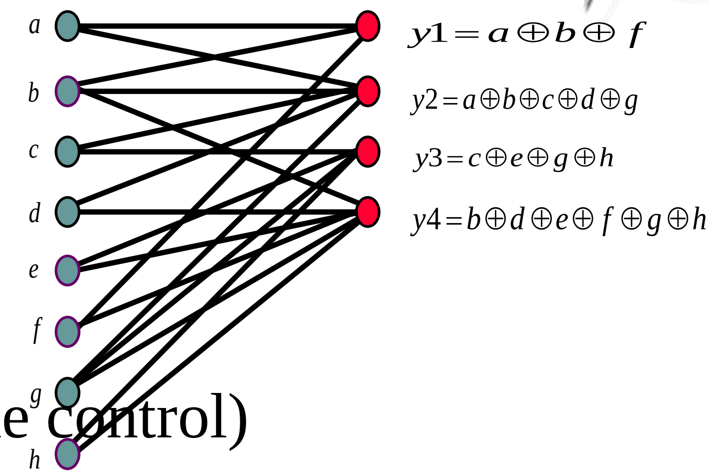
= pachete redundante (de control)

[2] John W.Byers, Michael Luby, Michael Mitzenmacher, Ashutosh Rege; “A Digital Fountain Approach to Reliable Distribution of Bulk Data” -Proceedings of the ACM SIGCOMM '98

Coduri Tornado

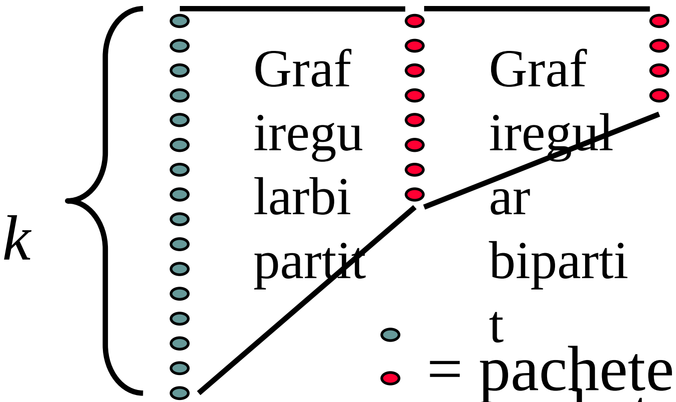


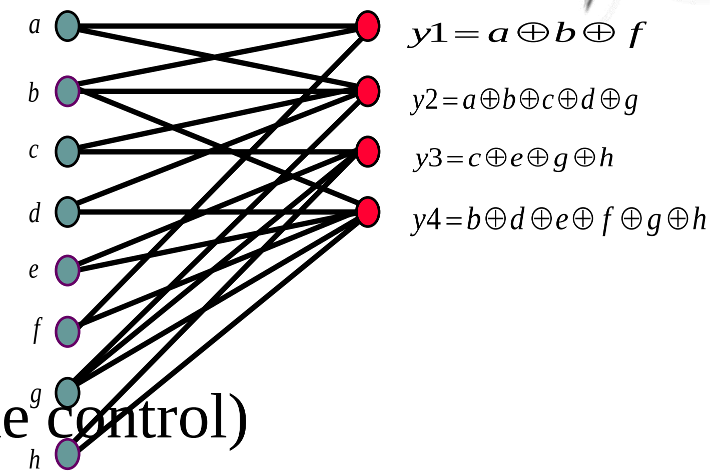
= pachete info
= pachete redundante (de control)



- în prima etapă din cele k pachete informaţionale se obţin βk pachete de control (β număr pozitiv subunitar) conform grafului B1
- în a doua etapă pornind de la cele βk pachete de control obţinute în etapa anterioară se obţin alte ββk pachete de control conform grafului B2

Coduri Tornado



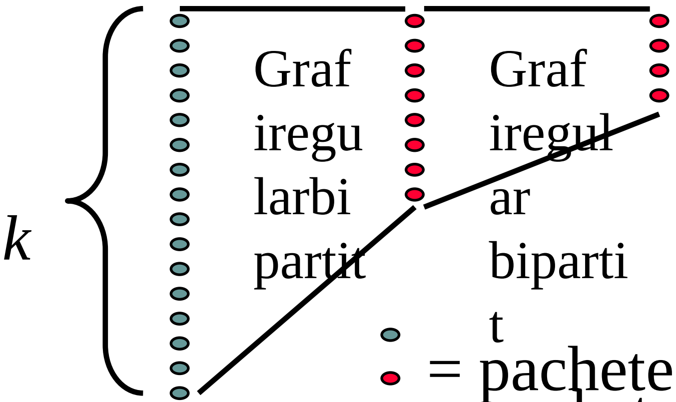


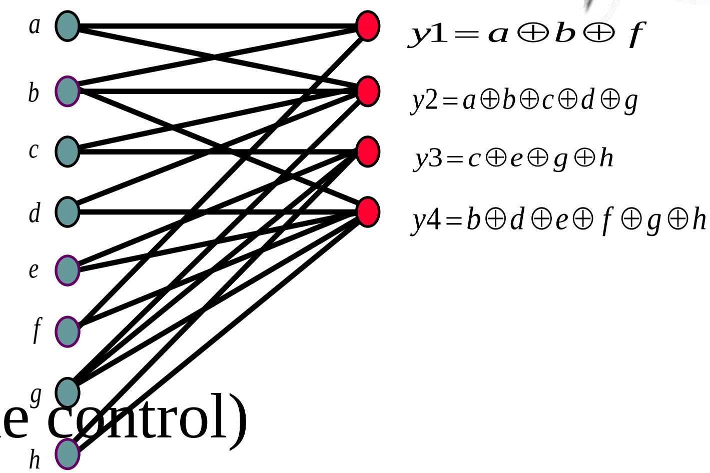
= pachete info
= pachete redundante (de control)

- în ultima (m-a) etapă se utilizează un cod corector de ştergeri "clasic", C, cu rata 1- β care poate să corecteze β ştergeri
- Codul C are $\beta^{m-1}k$ simboluri (pachete) la intrare, şi generează β $\beta^{m-1}k/(1-\beta)$ simboluir de control adiţionale
- Numărul simbolurilor de control generate în cele m etape este

$$\sum_{i=1}^{m-1}\beta^i k + \frac{\beta^m k}{1-\beta} = \frac{k\beta}{1-\beta}$$

Coduri Tornado





= pachete info

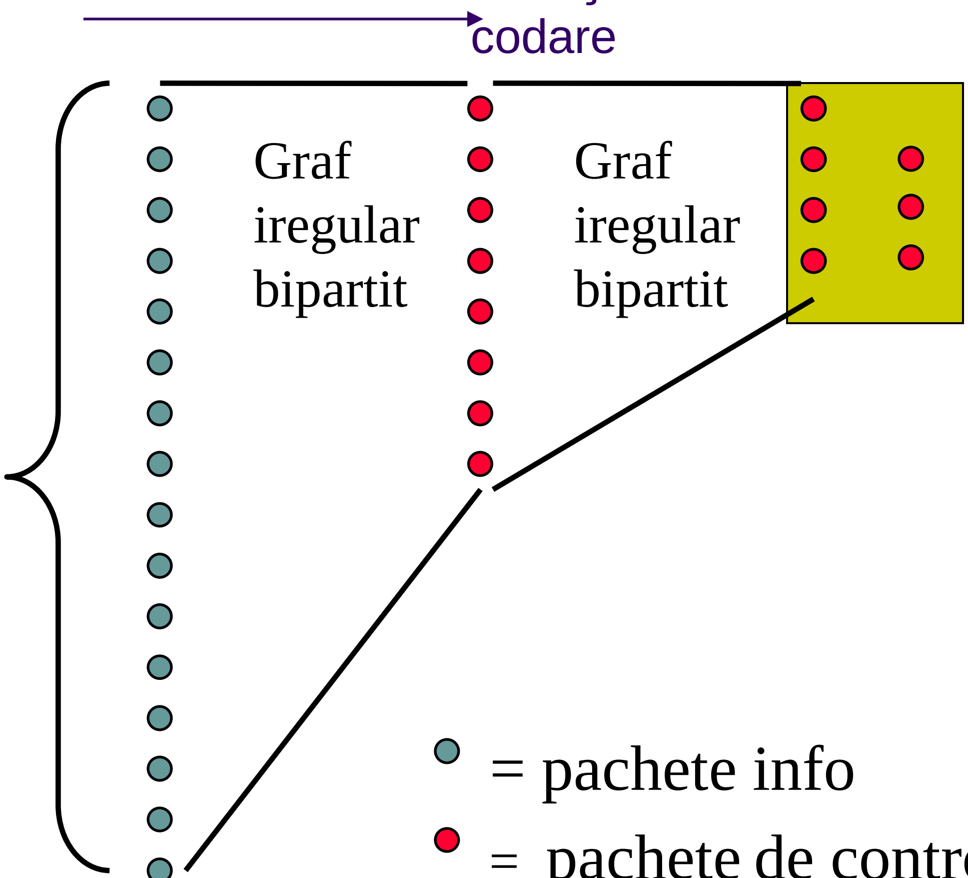
= pachete redundante (de control)

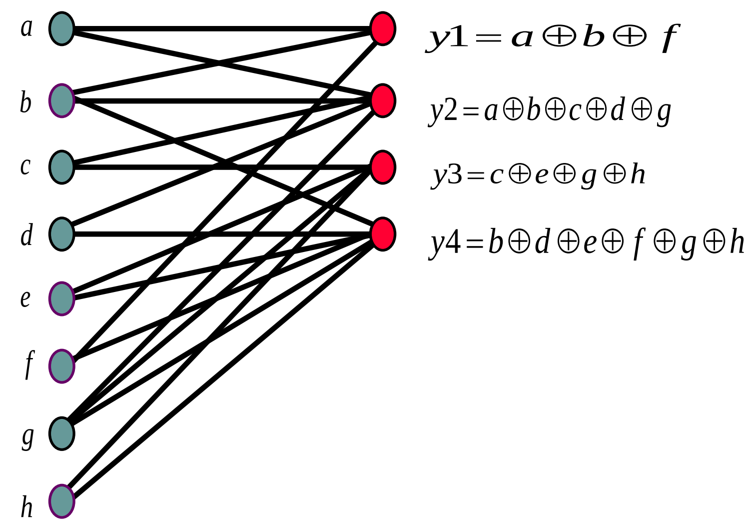
- rata globală va fi:

$$R = \frac{k}{k + \frac{k\beta}{1-\beta}} = \frac{k}{\frac{k - k\beta + k\beta}{1-\beta}} = 1-\beta$$

- Codul global $C(B_1, B_2, \ldots B_{m-1}, C)$ este un corector de ştergeri cu rata 1-β, care poate să corecteze cu probabilitate mare orice ștergere până la lungime·β

Direcţia de codare





= pachete info

= pachete de control

Direcţia de de-codare

Coduri Tornado

- Procesul de codare constă în sumarea modulo 2 a pachetelor conform grafului "generator"
- Pachetele recepţionate sunt "substituite" în ecuaţiile de control
- Prin înlocuirea variabilelor în ecuaţii de control, se pot reconstitui pachetele pierdute, utilizând adunări modulo 2
- De regulă primele k pachete sosite generează un număr redus de "rezolvări", dar când numărul pachetelor receptionate este mai mare de k, un pachet recepţionat generaează o avalanşă de "rezolvări" permiţând reconstrucţia pachetelor informaţionale

- Deoarece nodurile din graf au un număr redus de vecini (ecuaţii cu puţini termeni) operţiunea de codare şi decodare nu necesită multe operaţii
- Numărul de operaţii pentru obţinerea pachetelor redundante depinde numai de gradul nodului respectiv
- Complexitatea decodării depinde tot de gradul nodurilor şi de “poziţia” pachetelor recepţionate în graf
- Codurile Tornado sunt tot coduri cu lungime finită, deci pentru utilizarea lor ca DF pachetele codate se transmit întreţesute ciclic

- Comparaţie între timpii de codare şi decodare

Timpul de codare, pachete 1K		
Size	Reed-Solomon	Tornado
250 K	4.6 sec.	0.11 sec.
500 K	19 sec.	0.18 sec.
1 MB	93 sec.	0.29 sec.
2 MB	442 sec.	0.57 sec.
4 MB	30 min.	1.01 sec.
8 MB	2 hrs.	1.99 sec.
16 MB	8 hrs.	3.93 sec.

Timpul de decodare		
Size	Reed-Solomon	Tornado
250 K	2.06 sec.	0.18 sec.
500 K	8.4 sec.	0.24 sec.
1 MB	40.5 sec.	0.31 sec.
2 MB	199 sec.	0.44 sec.
4 MB	13 min.	0.74 sec.
8 MB	1 hr.	1.28 sec.
16 MB	4 hrs.	2.27 sec.

[1] John W.Byers, Michael Luby, Michael Mitzenmacher, Ashutosh Rege; “A Digital Fountain Approach to Reliable Distribution of Bulk Data” -Proceedings of the ACM SIGCOMM '98

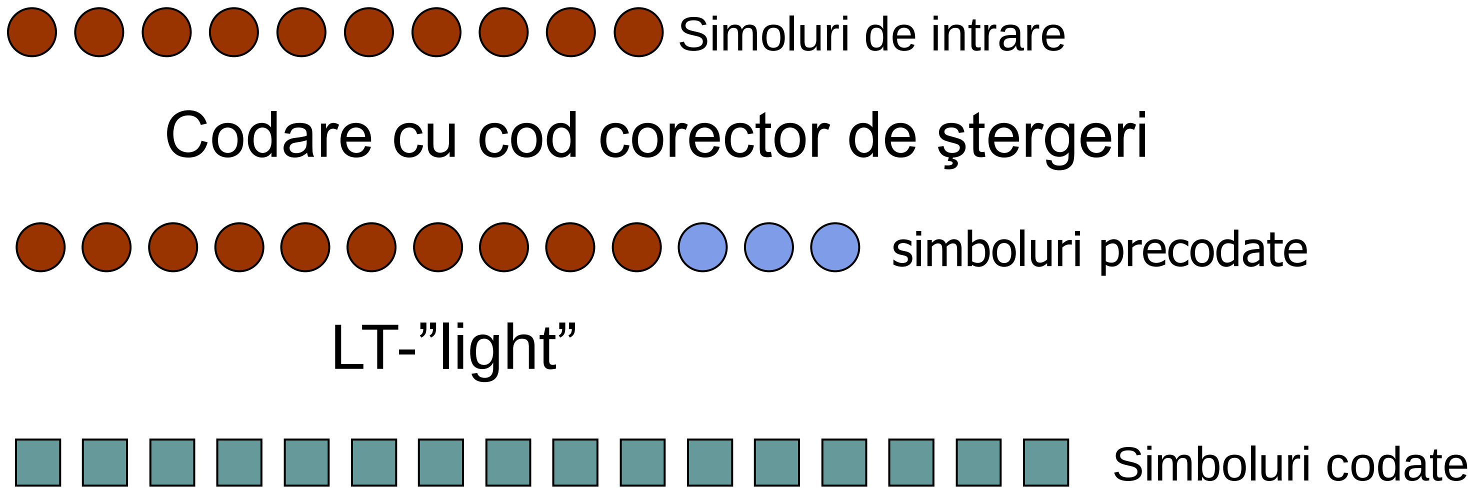
Tornado vs LT

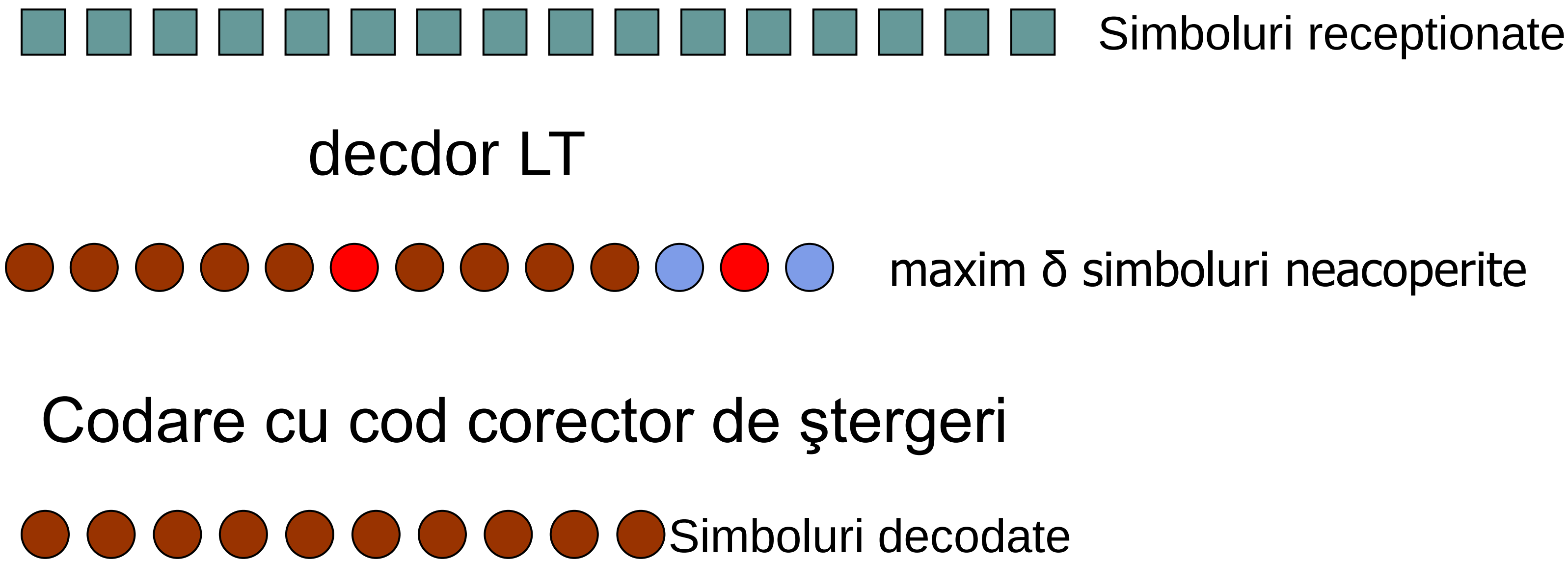
- Atât codurile RS cât şi codurile Tornado sunt coduri sistematice, în schimb codurile LT sunt nesistematice
- memoria necesară pentru codarea decodarea codurilor tornado este mult mai mare decât în cazul codurilor LT
- codurile LT sunt coduri *rateless* iar codurile tornado au rata finită
- Codurile Tornado sunt generate pe baza grafurilor cu grad maxim constant, în schimb codurile LT au grafuri cu densitate logaritmică

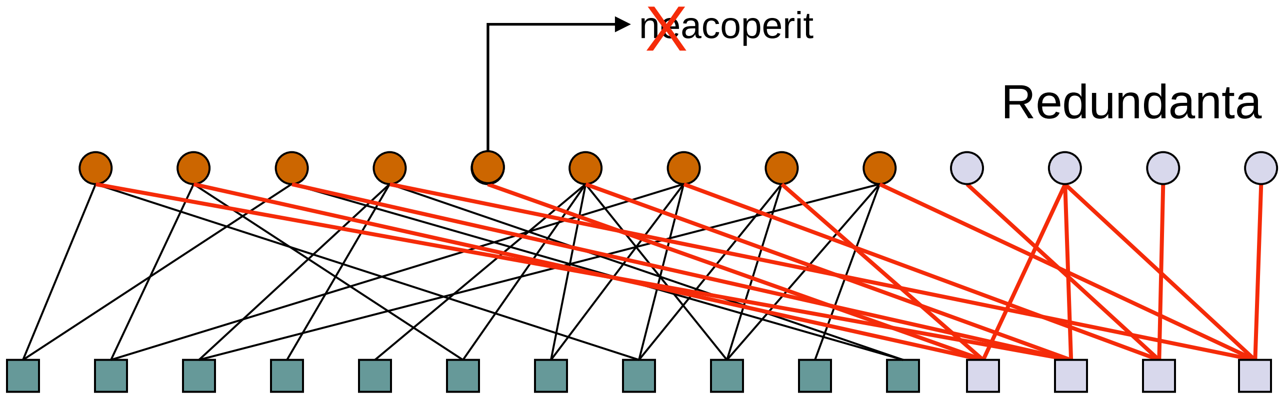
Coduri Raptor(RAPide TORnado)

- Primul cod DF cu timp de codare şi decodare liniară
- au fost inventate in 2000/2001 publicate in 2004
- Se acceptă ca o fracţiune maxim δ din simbolurile de intrare a unui cod LT să nu fie acoperite la terminarea procesului de decodare LT
- Aceste simboluri “pierdute” se pot recupera utilizând un cod corector de ştergeri classic
  - Se cunoaşte numărul maxim de ştergeri posibile
- Implică o precodare cu un cod corector de ştergeri classic

[5] A. Shokrollahi,„Raptor Codes”,http://www.cs.huji.ac.il/labs/danss/p2p/resources/raptor.pdf


Simoluri de intrare
Codare cu cod corector de ştergeri
simboluri precodate
LT-”light”
Simboluri codate

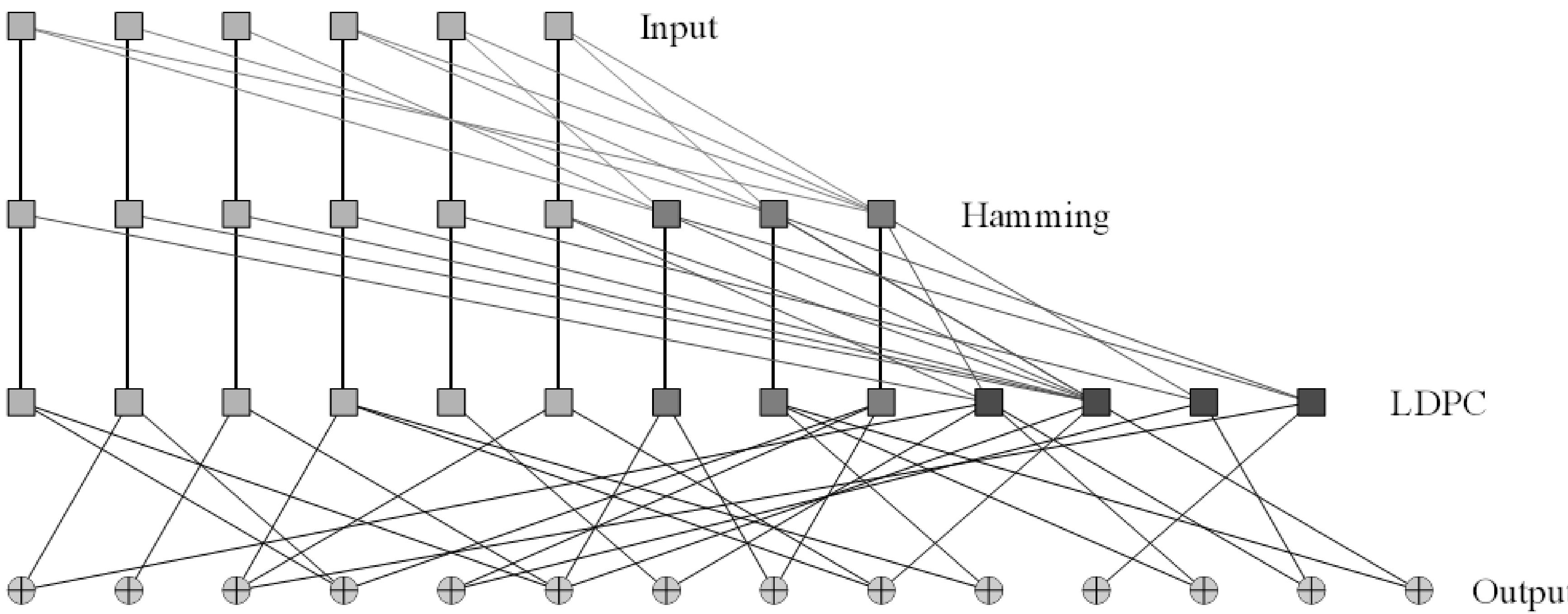

Simboluri receptionate
decdor LT
maxim δ simboluri neacoperite
Codare cu cod corector de ştergeri
Simboluri decodate



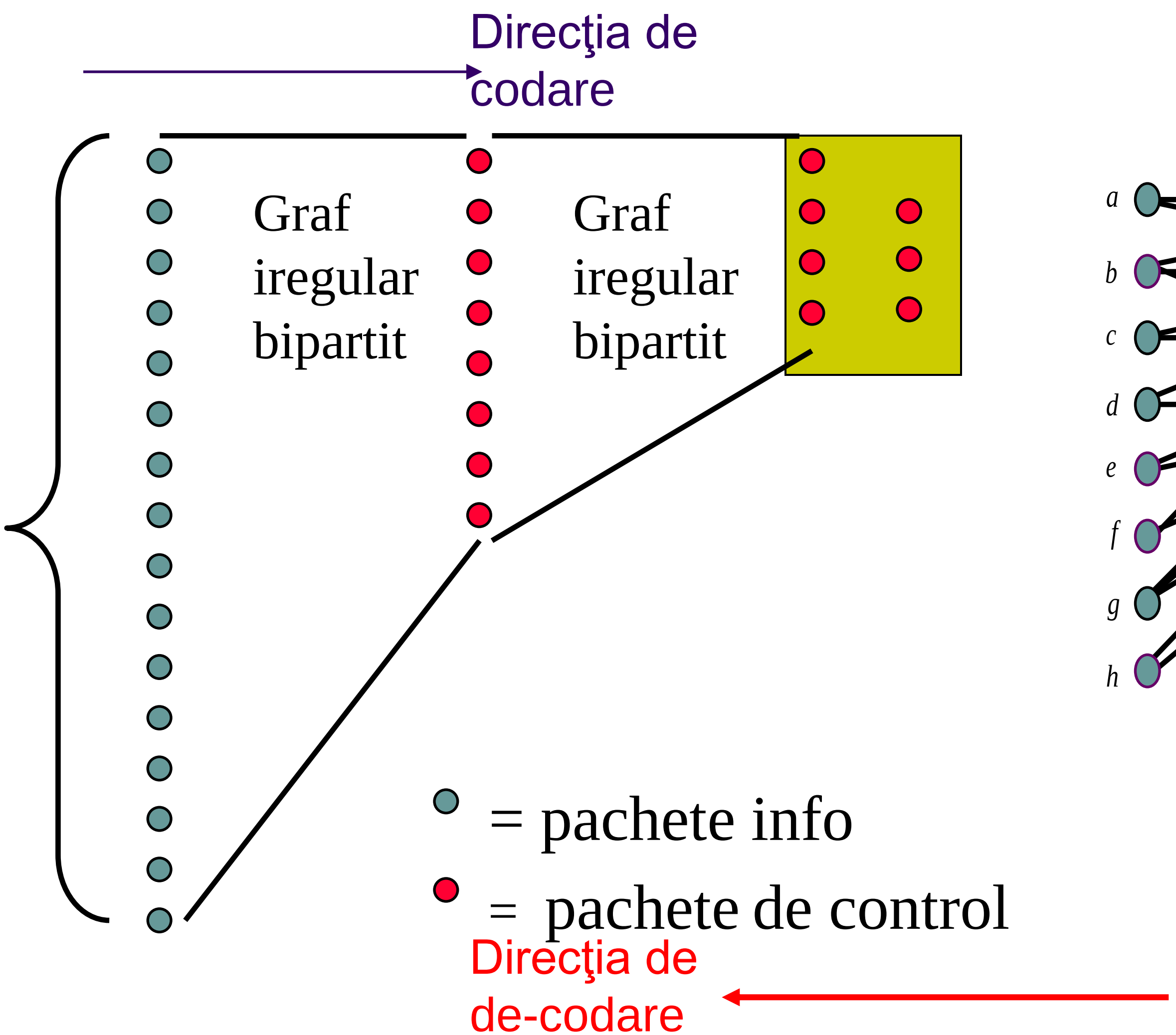
- Dacă precodorul este ales în mod corespunzător, atunci se poate utiliza un cod LT cu gradul mediu constant, care asigură timp de codare liniară
- Un cod Raptor (k,C, Ω(d))este definit de numărul de pachete informaţionale k, codul corector de ştergeri C şi distribuţia gradurilor al codului LT Ω(d)

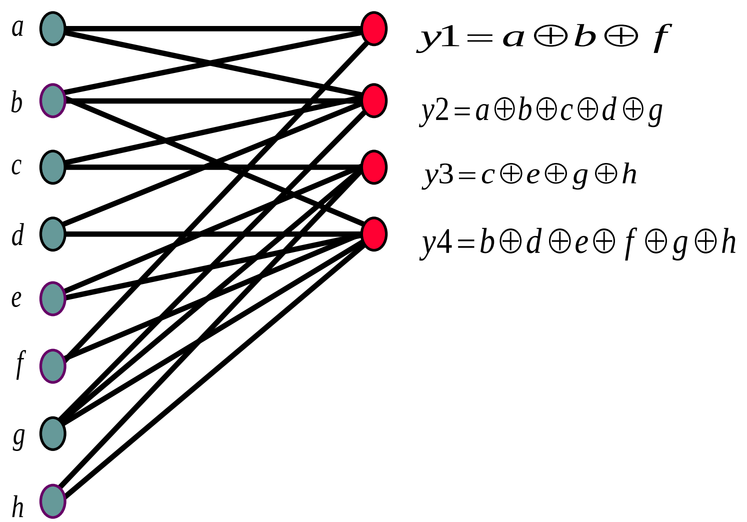
Coduri Raptor

- Parametrii principali de performanţă ai codului Raptor sunt definite după cum urmează:
  - Spaţiul de memorie: Codurile Raptor necesită spaţiu de stocare pentru simbolurile intermediare, Consumul de spaţiu al codurilor Raptor este k/R, unde R este rata pre-codului.
  - Overhead: Overheadul este o funcţie a algoritmului de decodare folosit, şi este definit ca numărul de simboluri de ieşire pe care trebuie să aibă decodorul pentru a recupera cu probabilitate mare simbolurile de intrare. Un overhead de $1+\varepsilon$ înseamnă că trebuie recepţionate $k(1+\varepsilon)$ simboluri de ieşire pentru a asigura o decodare cu success cu o probabilitate mare.
  - Costul: Costul procesului de codare şi de decodare.

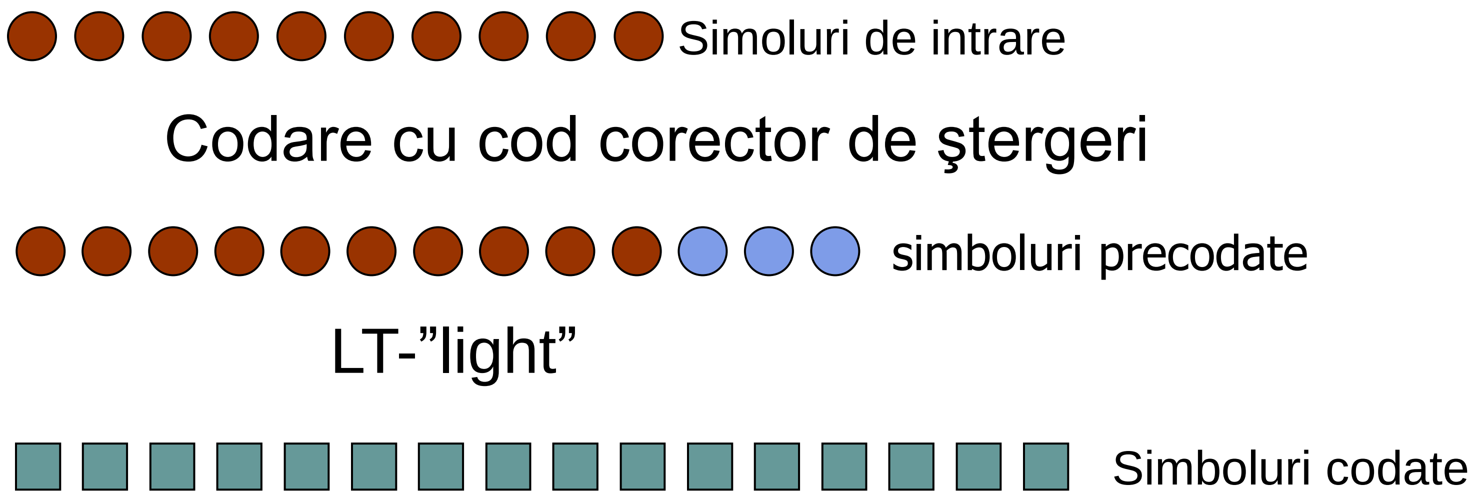

Input
Hamming
LDPC
Output

Coduri Tornado

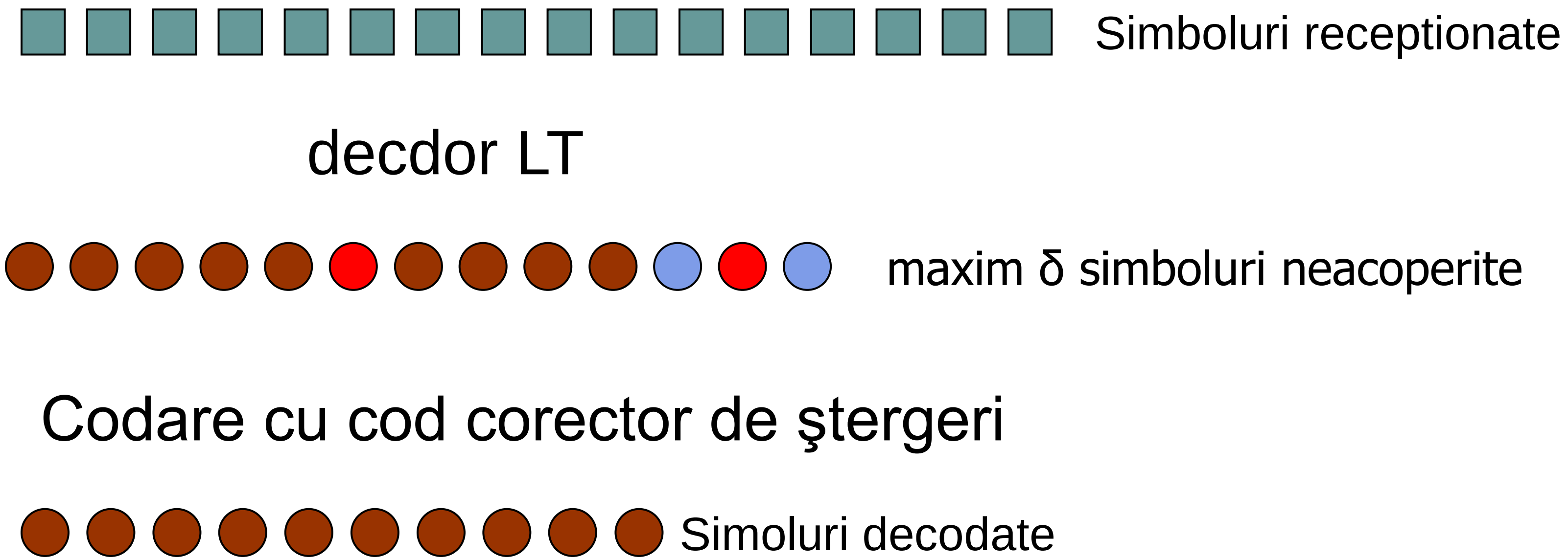




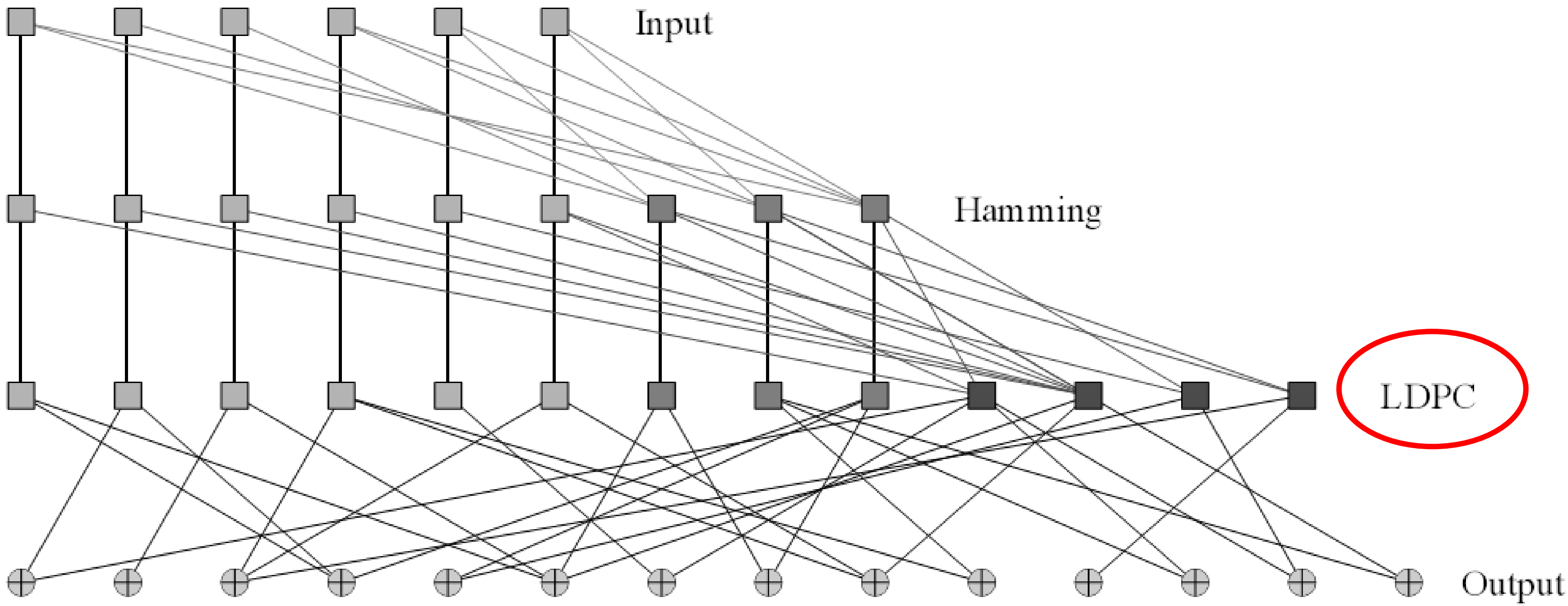
Coduri Raptor(RAPide TORnado)



Coduri Raptor(RAPide TORnado)



Coduri Raptor(RAPide TORnado)



Codurile LDPC – Low Density Parity Check Codes

- Codurile LDPC, introduse în 1963 de Robert G. Gallager, reprezintă o familie de coduri bloc liniare
- se obţin pornind de la o matrice „rară” de control a parităţii H, care conţine un număr mic de elemente nenule şi un număr mare de elemente de 0 („sparse matrix”)
- Codurile LDPC pot fi privite ca şi coduri bloc liniare, descrise de matricea de control a parităţii H de dimensiune $M \times N$ care satisface ecuaţia: $Hx^T = 0$ pentru toate cuvintele de cod x

Codurile LDPC

- codurile LDPC au următoarele proprietăţii specifice:
  - dimensiunile M şi N ale matricii H sunt mult mai mari faţă de dimensiunile unei matrici de control specifică unui cod Hamming;
  - matricile H sunt definite, prin construcţie, într-o formă nesistematică şi conţin un număr de elemente de 1 mult mai mic decât numărul de elemente de 0;
  - matricea de control a parităţii ce defineşte un cod LDPC A(j, k) are exact j elemente de 1 pe fiecare coloană şi exact k elemente de 1 pe fiecare linie.

- **Tipuri de coduri LDPC**
  - construite aleator (random)
  - construite pe baza unor structuri regulate („array.based”)
  - pe baza unor construcţii combinatoriale
  - pe baza unor construcţii geometrice
  - construite pe baza grafurilor
  - cuasi-ciclice

Codarea codurilor LDPC

- Considerând cuvântul de cod $v=[c_0,\ldots,c_{M-1}, i_0,\ldots,i_{(N-M)-1}]$, biţii de control c_m se pot determina, în funcţie de biţii informaţionali i_l, prin rezolvarea sistemului de M ecuaţii liniare:

$$Hv^T=0$$

- Această abordare are două dezavantaje majore
  - pentru valori mari al parametrului j, M ia valori mari necesitând un mare volum de calcule, ceea ce conduce la creşterea timpului şi/sau resurselor hardware necesare procesării.
  - are nevoie de toţi biţii informaţionali i_l, $l = 0,\ldots, (N-M)-1$, în acelaşi timp; această cerinţă induce o întârziere suplimentară de un cuvânt de cod în sistemul de transmisie

Codarea codurilor LDPC

- Matricea H, (M x N), este împărţită în două matrici D şi E, reţinând primele M coloane în matricea D şi restul de (N-M) coloane în matricea E.
- Cele două matrici au dimensiunile:

D: (M x M)

E: ((N-M) x M)

- Matricea D este pătrată şi are determinant nenul, deci ecuaţia codării poate fi rescrisă sub forma

$$[H]\cdot[v]^t=[0]\Leftrightarrow[D]\cdot\begin{bmatrix}c_0\\ \vdots\\ c_{M-1}\end{bmatrix}+[E]\cdot\begin{bmatrix}i_0\\ \vdots\\ i_{(N-M)-1}\end{bmatrix}=[0]\quad\Rightarrow\quad\begin{bmatrix}c_0\\ \vdots\\ c_{M-1}\end{bmatrix}=[F]\cdot\begin{bmatrix}i_0\\ \vdots\\ i_{(N-M)-1}\end{bmatrix}$$

Codarea codurilor LDPC

- Prin dezvoltarea matricii de codare F, (M x (N-M)), dacă notăm cu $[f_l]$ coloanele sale (fiecare din ele un vector de M biţi) ecuaţia de codare poate fi exprimată sub forma:

$$[f_0]\cdot i_0 + \ldots [f_{(N-M)-1}]\cdot i_{(N-M)-1} = [C]$$

- Matricea F, calculată off-line, este stocată cloană cu coloană în memorie; fiecare bit de informaţie se înmulţeşte cu coloana cu acelaşi index, iar vectorii-produs rezultaţi sunt acumulaţi, astfel obţinându-se vectorul biţilor de control [C]

Decodarea codurilor LDPC

- Algoritmul de decodare MP este un algoritm iterativ care utilizează ca date de intrare probabilităţile *a posteriori* ale biţilor cuvântului de cod, furnizate de funcţie de demapare soft.
- La fiecare iteraţie, algoritmul modifică valorile acestor probabilităţi, în funcţie de probabilităţile *a posteriori* ale celorlaţi biţi care intră in aceleaşi ecuaţii de control cu bitul dat. Aceste actualizări sunt făcute iterativ pentru fiecare bit

Decodarea codurilor LDPC

- Matricea de control a unui cod LDPC (5,3,4)

$$H = \left[\begin{array}{cccccccccccccccccccc}
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
\hline
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
\hline
1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1
\end{array}\right]$$

Decodarea codurilor LDPC

- sistemul de ecuaţii de control a parităţii, construit pe baza relatiei:

$$\left[\begin{array}{cccccccccccccccccccc}
1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&1&1&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1\\
\hline
1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&0&0&0&0\\
0&1&0&0&0&1&0&0&0&1&0&0&0&0&0&0&1&0&0&0\\
0&0&1&0&0&0&1&0&0&0&0&0&0&1&0&0&0&1&0&0\\
0&0&0&1&0&0&0&0&0&0&1&0&0&0&1&0&0&0&1&0\\
0&0&0&0&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1\\
\hline
1&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&1&0&0\\
0&1&0&0&0&0&1&0&0&0&1&0&0&0&0&1&0&0&0&0\\
0&0&1&0&0&0&0&1&0&0&0&0&1&0&0&0&0&0&1&0\\
0&0&0&1&0&0&0&0&1&0&0&0&0&1&0&0&1&0&0&0\\
0&0&0&0&1&0&0&0&0&1&0&0&0&0&1&0&0&0&0&1
\end{array}\right]\cdot\begin{bmatrix}c_1\\c_2\\c_3\\c_4\\c_5\\c_6\\c_7\\c_8\\c_9\\c_{10}\\m_1\\m_2\\m_3\\m_4\\m_5\end{bmatrix}=\left[0\right]$$

Decodarea codurilor LDPC

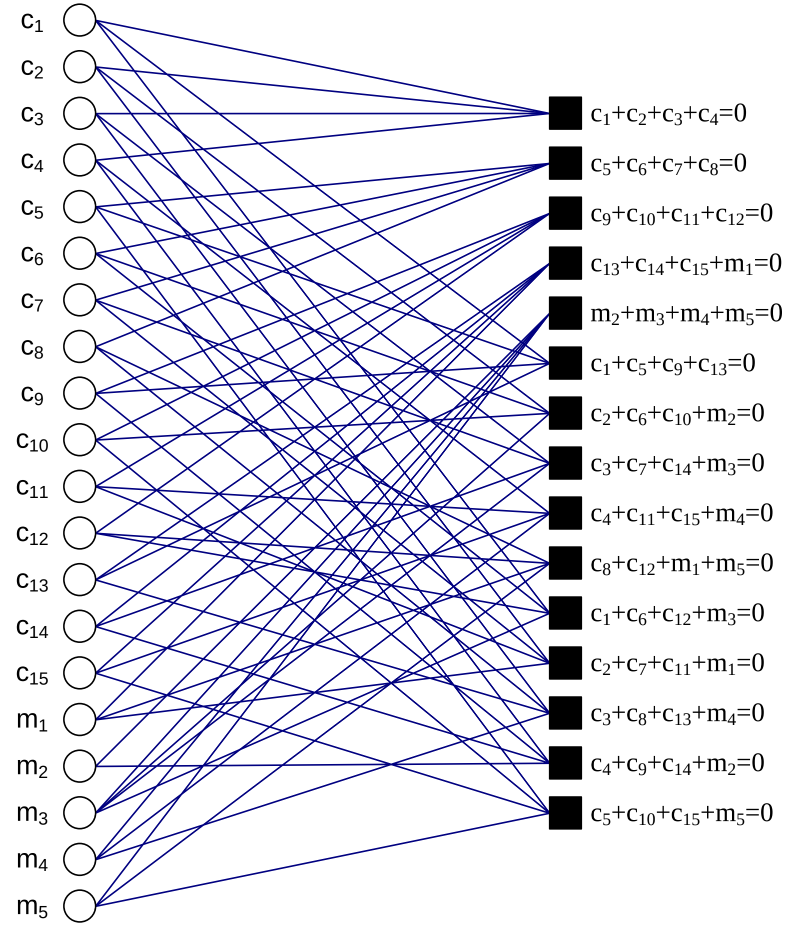
- Pentru a obţine biţii c_i asociaţi cuvântului binar m=$\{m_1, m_2,\ldots,m_K\}$ trebuie rezolvat sistemul următor:

$$\begin{cases}
c_1+c_2+c_3+c_4=0\\
c_4+c_6+c_7+c_8=0\\
c_9+c_{10}+c_{11}+c_{12}=0\\
c_{13}+c_{14}+c_{15}+m_1=0\\
m_2+m_3+m_4+m_5=0\\
c_1+c_5+c_9+c_{13}=0\\
c_2+c_6+c_{10}+m_2=0\\
c_3+c_7+c_{14}+m_3=0\\
c_4+c_{11}+c_{15}+m_4=0\\
c_8+c_{12}+m_1+m_5=0\\
c_1+c_6+c_{12}+m_3=0\\
c_2+c_7+c_{11}+m_1=0\\
c_3+c_8+c_{13}+m_4=0\\
c_4+c_9+c_{14}+m_2=0\\
c_5+c_{10}+c_{15}+m_5=0
\end{cases}$$

Decodarea codurilor LDPC

- Graful bipartit asociat codului LDPC (5,3,4) este:

$$
H = \left[\begin{array}{cccccccccccccccccccc}
1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&1&1&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1\\
\hline
1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&0&0&0&0\\
0&1&0&0&0&1&0&0&0&1&0&0&0&0&0&0&1&0&0&0\\
0&0&1&0&0&0&1&0&0&0&0&0&0&1&0&0&0&1&0&0\\
0&0&0&1&0&0&0&0&0&0&1&0&0&0&1&0&0&0&1&0\\
0&0&0&0&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1\\
\hline
1&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&1&0&0\\
0&1&0&0&0&0&1&0&0&0&1&0&0&0&0&1&0&0&0&0\\
0&0&1&0&0&0&0&1&0&0&0&0&1&0&0&0&0&0&1&0\\
0&0&0&1&0&0&0&0&1&0&0&0&0&1&0&0&1&0&0&0\\
0&0&0&0&1&0&0&0&0&1&0&0&0&0&1&0&0&0&0&1
\end{array}\right]
$$



Decodarea codurilor LDPC

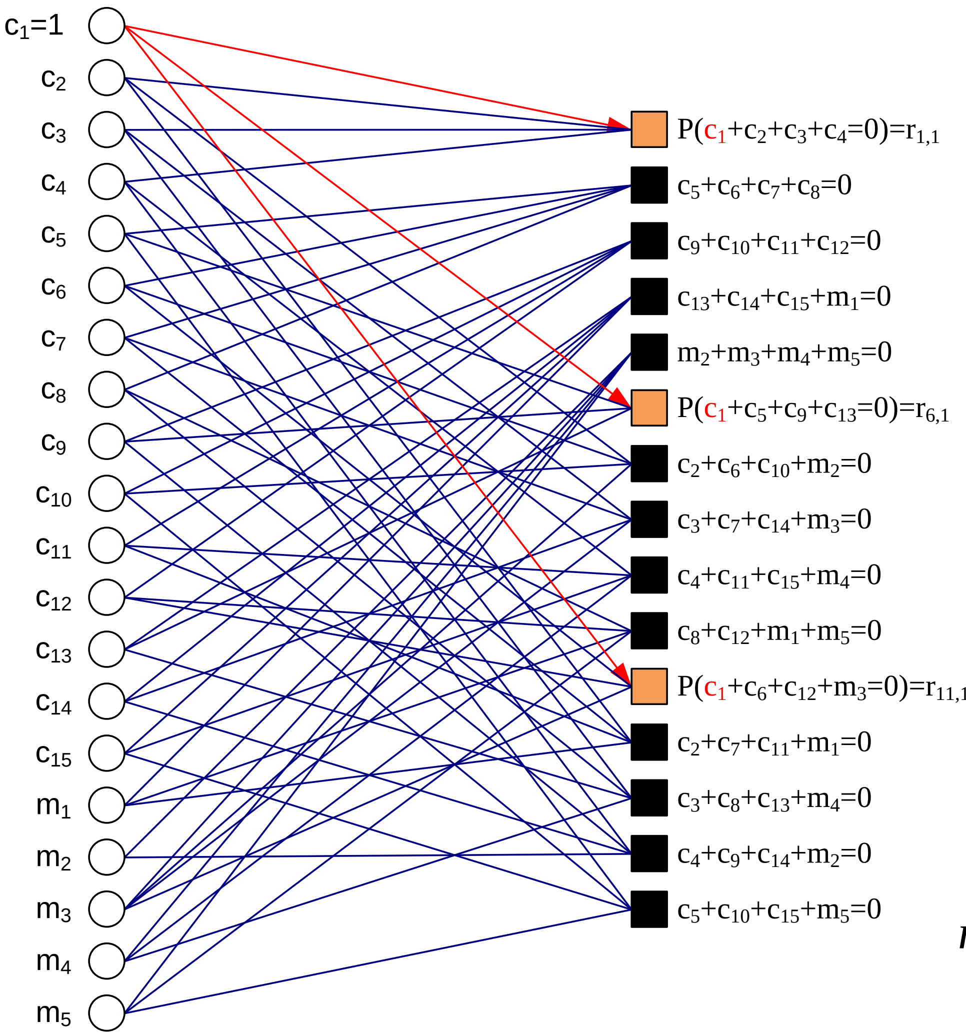
- **Pasul 0: iniţializare**

nodurile de bit sunt iniţializate cu probabilităţile *aposteriorii* iniţiale determinate de blocul de demapare pe baza semnalului recepţionat, P_j^0 (probabilitatea ca bitul j din cuvântul recepţionat să fie 0) şi P_j^1(probabilitatea ca bitul j din cuvântul recepţionat să fie 1)

Decodarea codurilor LDPC

- **Pasul 1 actualizarea nodurilor de control**
- se determină probabilităţiile $r_{m,n}^{x}$, care reprezintă probabilitatea ca ecuaţia de control m să fie satisfăcută dacă bitul n are valoarea x

Decodarea codurilor LDPC



Probabilitatea ca prima ecuaţie să fie satisfăcută, dacă c_1=1 este:

$$P(c_1+c_2+c_3+c_4=0\,|\,c_1=1)$$

$$=P(c_2+c_3+c_4=1)=$$

$$=P_2^1\cdot P_3^0\cdot P_4^0+P_2^0\cdot P_3^1\cdot P_4^0+$$

$$+P_2^0\cdot P_3^0\cdot P_4^1+P_2^1\cdot P_3^1\cdot P_4^1$$

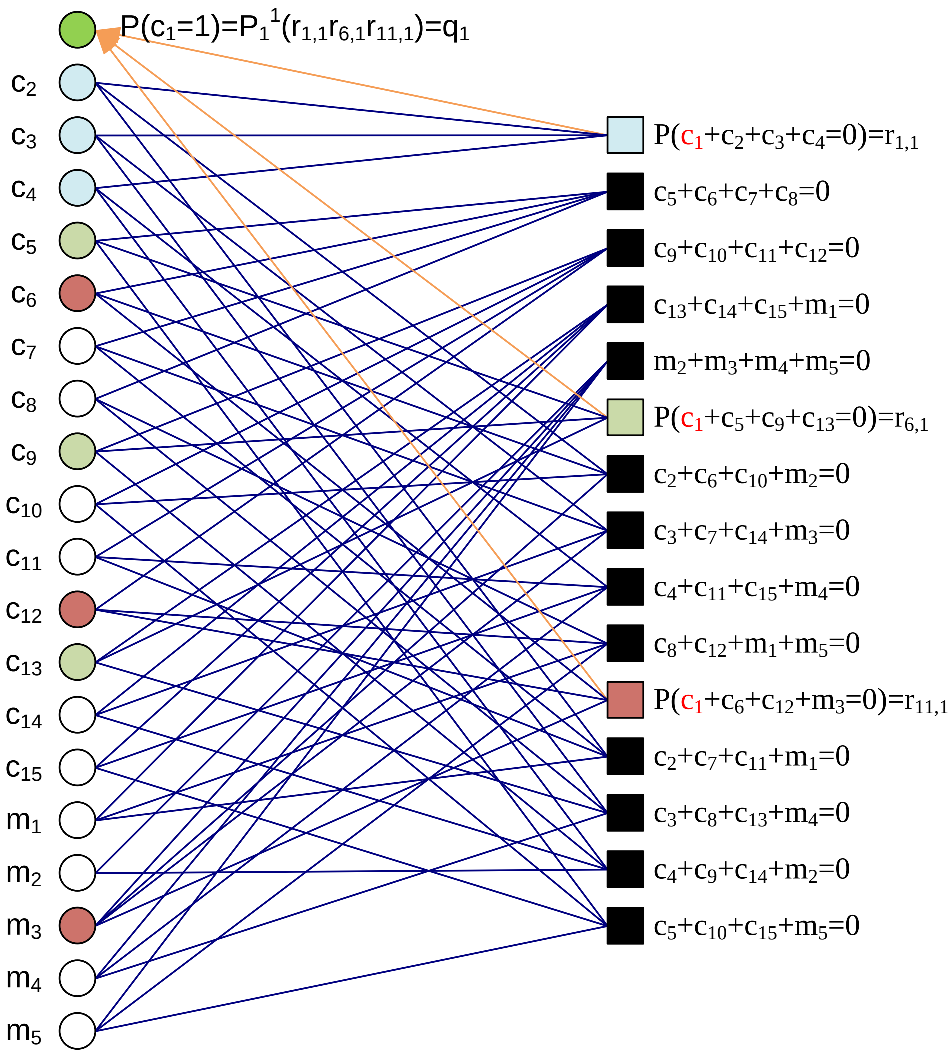
Se poate arăta că probabilitatea ca din k biţi independenţi un număr impar de biţi să fi nenul este:

$$r_{m,n}^1=\frac{1-\prod\limits_{\substack{l=1\\l\neq n}}^{k_m}\left(1-2q_{l,m}^1\right)}{2}=\frac{1}{2}\left(1-\prod\limits_{\substack{l=1\\l\neq n}}^{k_m}\left(q_{l,m}^0-q_{l,m}^1\right)\right)$$

Decodarea codurilor LDPC

- **Pasul 2 Actualizarea nodurilor de bit**
- La sfârşitul fiecărei iteraţii se determină probabilităţile a posteriori pentru fiecare bit, adică probabilitatea P ca bitul transmis pe poziţia n să fie „1” condiţionat de setul de simboluri y recepţionate şi de evenimentul S ca cele j_n ecuaţii de control, în care intră bitul n, să fie satisfăcute.
- Fiindcă ecuaţiile de control sunt independente, probabilitatea ca toate cele j_n ecuaţii de control în care intră bitul n să fie satisfăcute este produsul probabilităţilor individuale ca fiecare din aceste ecuaţii să fie satisfăcută

Decodarea codurilor LDPC



$$\mathrm{P}\left(x_n=1 \mid S,\{y\}\right)=\alpha_n \cdot P_n^1 \cdot \prod_{i=1}^{j_n}\left(r_{i,n}^1\right)$$

- În ecuaţia de mai sus constanta α se alege astfel încât

$$\mathrm{P}\left(x_n=1 \mid S,\{y\}\right)+\mathrm{P}\left(x_n=0 \mid S,\{y\}\right)=1$$

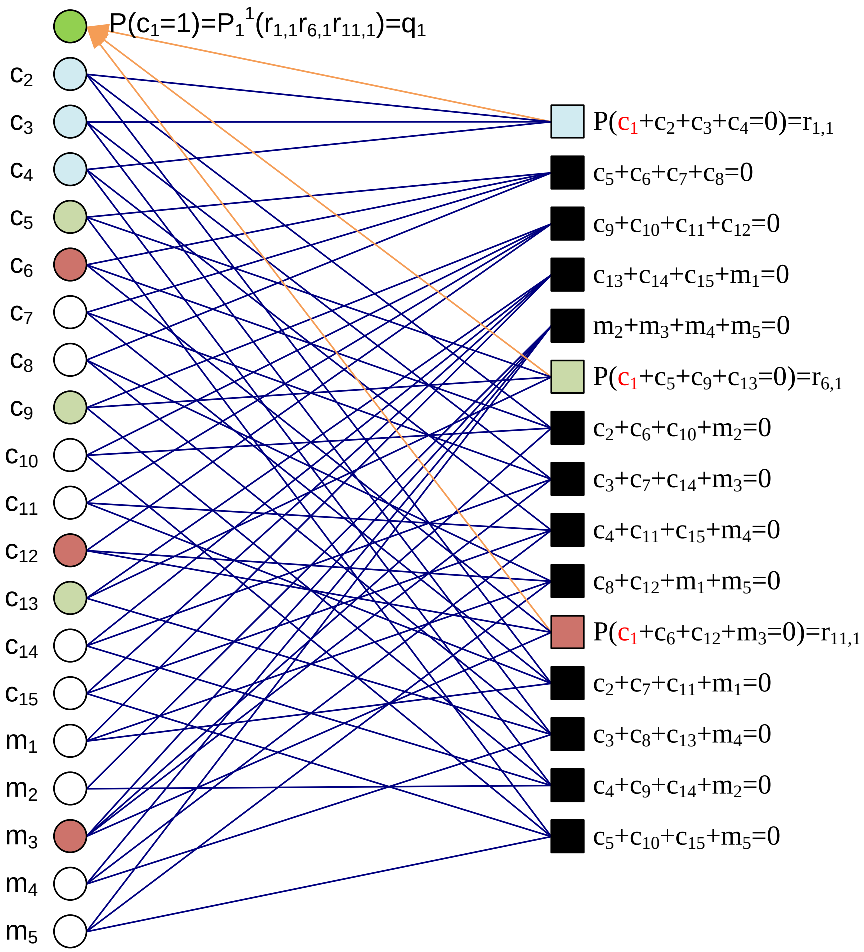
Decodarea codurilor LDPC

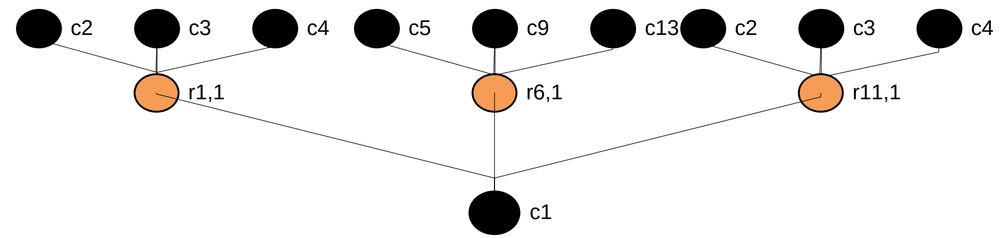
- **Pasul 3 Decizia hard a biţilor decodaţi şi calcularea sindromului**
  - **Dacă cuvântul binar x în care bitul n are valoarea 1, cu respectarea condiţiei $P(x_n=1|S,\{y\})>0.5$, satisface ecuaţia $Hx^T = 0$, algoritmul de decodare este încheiat.**
  - **Dacă sindromul nu este nul, adică Hx^T nu este zero, se efectuează o nouă iteraţie.**
  - **Acest algoritm iterativ continuă până când se găseşte un cuvânt de cod valid, sau până când graful de decodare va deveni un arbore cu l nivele, cazul în care se consideră că decodarea a eşuat**

$$\mathrm{P}\left(x_n = 1 \mid S, \{y\}\right) \geq 0.5$$

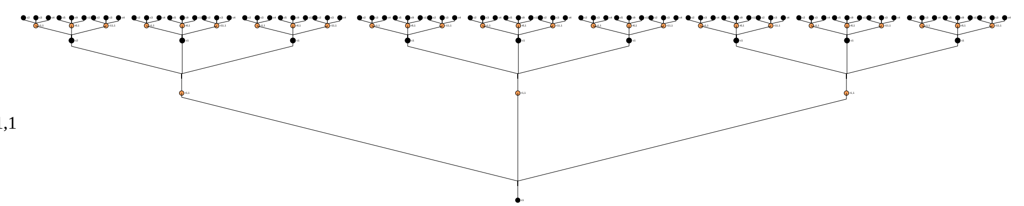
Decodarea codurilor LDPC

- Graful vecinilor





Graful vecinilor nodului de bit c_1 după prima iteraţie



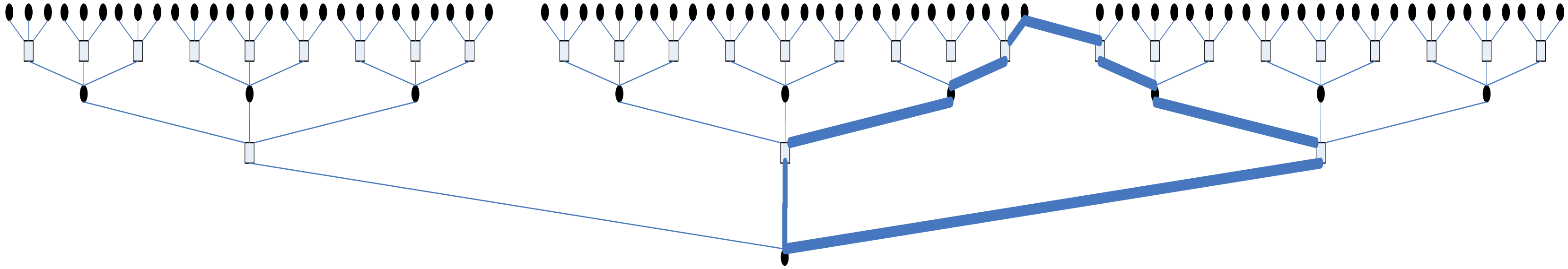
Graful vecinilor nodului de bit c_1 după adoua iteraţie

Decodarea codurilor LDPC

- ca toţi vecinii de gradul L unui nodului să fie independenţi (să apară o singură dată în graful vecinilor) după iteraţia L, lungime cuvântului de cod ar trebui să fie :

$$M \geq \left[(k-1) \cdot (j) \right]^{L}$$

  - subgraful vecinilor nodului de bit *n* rezultat în urma decodării iterative, după un număr de *I* iteraţii, nu va fi un arbore fără bucle



Decodarea codurilor LDPC

- Trebuie reţinut că acest algoritm nu încearcă să gasească cel mai apropiat cuvânt de cod, faţă de secvenţa recepţionată, ci încearcă să corecteze fiecare bit al secvenţei recepţionate, folosind informaţiile oferite de ceilalţi biţi cu care bitul dat intră în ecuaţiile de control. Sindromul este folosit ca o verificare (CRC).
- Datorită acestui fapt, numărul de biţi eronaţi după o decodare nereuşită, este mai mic decât numărul biţilor eronaţi înaintea decodării (spre deosebire de codurile bloc ciclice sau de cele convoluţionale).
- Numărul maxim de iteraţii este un parametru care se stabileşte în funcţie de durata permisă pentru decodare de aplicaţia în care este utilizat codul

Diferite tipuri de coduri Raptor

- Coduri LT
  - Codurile LT pot fi considerate coduri Raptor de forma $(k, F_2^k, \Omega(d))$, unde F_2^k este practic un cod de lungime k, care asociază simbolurile de intrare la simbolurile de ieşire
  - Inexistenţa puterii de corecţie a pre-codului impune utilizarea unei distribuţii $\Omega(d)$ sofisticate, cu complexitatea logaritmică a codării şi decodării (vezi cursul anterior)

Diferite tipuri de coduri Raptor

- PCO (Pre-Code-Only)
  - Aceste coduri reprezintă extremitatea cealaltă a codurilor *Raptor*, adică au un precodor sofisticat şi o distribuţie de ieşire foarte simplă de forma $\Omega(1)=1$
  - Algoritmul de decodare recepţionează un număr de m simboluri, din care se decodează n simboluri intermediare independente (este posibilă recepţionarea de două sau de mai multe ori ale aceluiaşi simbol), şi pre-decodorul pe baza acestor n simboluri intermediare determină cele k simboluri informaţionale

Diferite tipuri de coduri Raptor

- Coduri optimale (Asymptotically Optimal Raptor Codes)
  - Costuri de codare şi decodare constante, overheadul necesar decodării tinde spre unu
  - Este format dintr-un precodor LDPC şi o distribuţie de graduri de forma:

$$\Omega(d) = \frac{1}{\mu+1}\left(\mu d + \frac{d^2}{1\cdot 2} + \cdots + \frac{d^D}{(D-1)\cdot D} + \frac{d^{D+1}}{D}\right)$$

  - Unde $\mu \,\square\, \varepsilon/2$ şi $D \,\square\, 2/\varepsilon$ astfel se poate obţine un cod LDPC cu decodor MP liniar şi un overhead ε

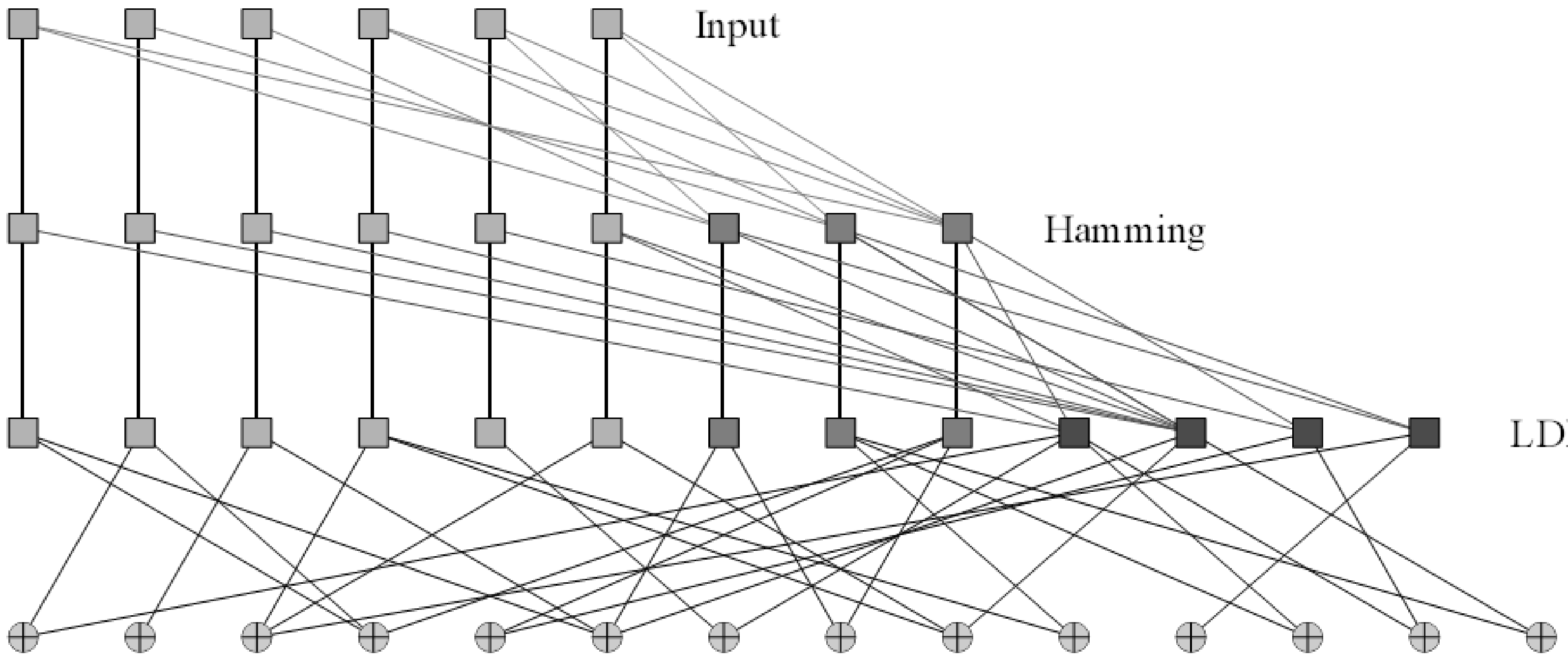
Diferite tipuri de coduri Raptor

- Cod eficient şi implementabil (practic)
  - Este format din două precodoare:
    - Precodor 1 cod Hamming
    - Precodor 2 cod LDPC
  - LT cu distribuţia gradurilor:

$$\Omega(d) = 0.008d + 0.494d^2 + 0.166d^3 + 0.073d^4 + 0.083d^5 + 0.056d^8 + 0.037d^9 + 0.056d^{19} + 0.025d^{65} + 0.003d^{66}$$

  - Codul rezultat are probabilitatea de eroare de bloc de maxim 10^{-14} pentr $k \geq 2^{16} = 65536$

Cod Raptor practic

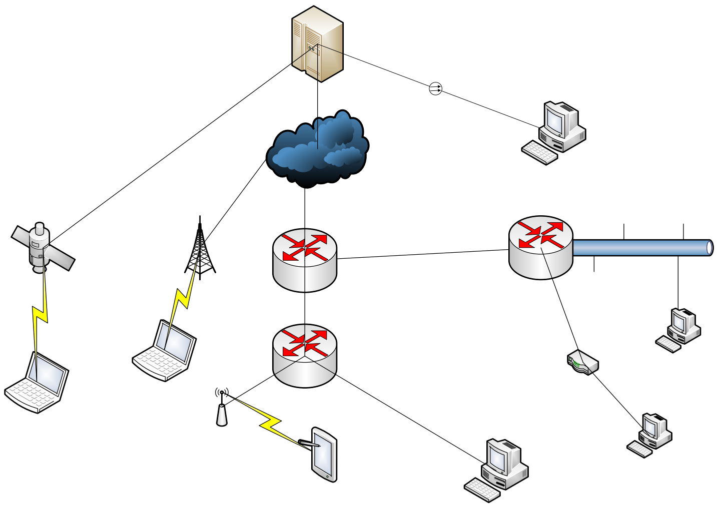


Utilizarea Codurilor DF

- Multicast
- Descărcare în paralel
- One-to-Many TCP
- Video streaming
- Transmisii în Overlay Networks
- Stocare distribuită
- Rutare dispersivă

Multicast

- DF a fost conceput pentru a asigura un multicast fiabil
- Sursa transmite într-un moment dat acelaşi informaţie către toţi utilizatorii
- Nu este nevoie de feedback
- Fiecare utilizator poate să întrerupe şi să reîncepe achiziţionarea datelor fără să informeze sursa despre acest lucru

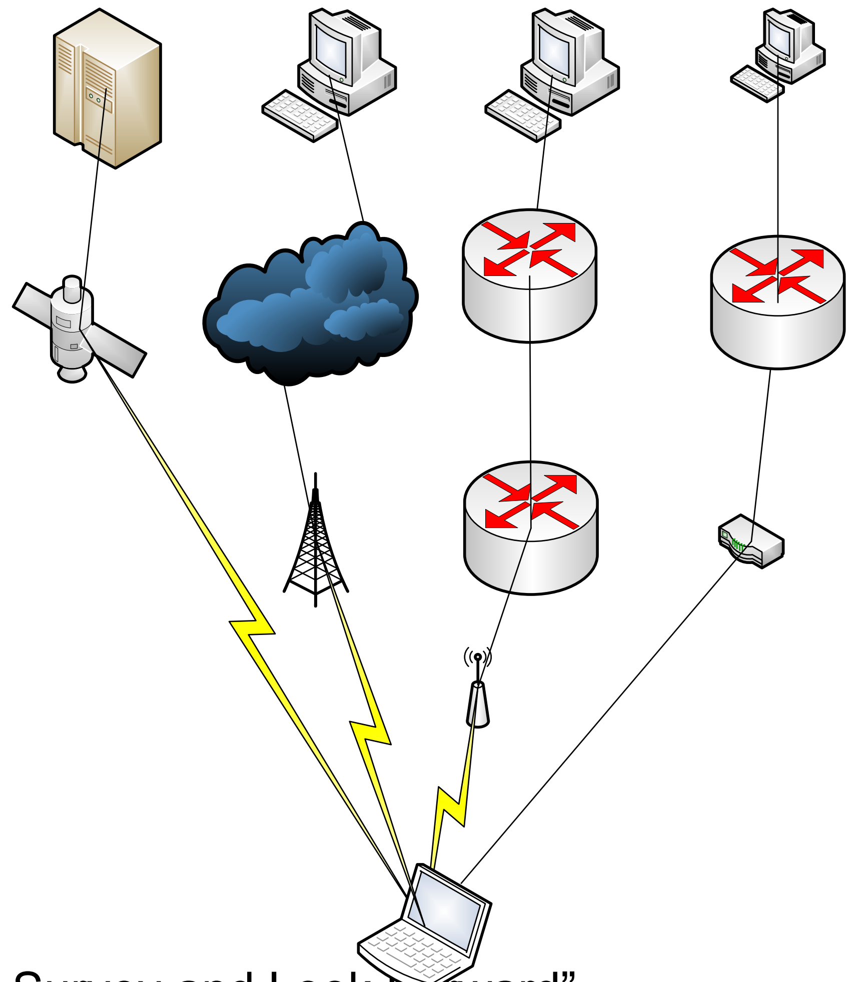


Transmisii punct la punct

- TCP are "probleme" în cazul transmisiilor la distanţe mari
  - Timpul de reacţie a sistemului este mare
  - Fereastra glisantă relativ mare, rezultă memorie mare
  - Retransmisiile introduc alte complicaţii
- În cazul utilizării unei cod DF destinaţia trebuie să transmită numai o cerere de transmisie şi o notificare despre decodarea reuşită la sfârşitul transmisiei
  - Sursa nu are nici o informaţie despre canal
    - poate să "inunde" cu pachete primele hopuri din reţea

Descărcare în paralel

- DF poate simplifica unui utilizator descărcarea unui bloc de informaţii, în paralel de la mai multe surse
- Fiecare sursă transmite secvenţa codată
- Destinaţia nu trebuie să gestioneze informaţiile recepţionate pe diferite legături
- Trebuie să recepţioneze numărul minim de pachete necesare decodării, indiferent pe ce legătură ajunge asta



[6] Michael Mitzenmacher, “Digital Fountains: A Survey and Look Forward”, http://www.eecs.harvard.edu/~michaelm/postscripts/itw2004.pdf

One-to-Many TCP

- Multicastul peste TCP înseamnă că sursa, pentru fiecare legătură trebuie să ţină evidenţa pachetelor transmise, pachetelor confirmate respectiv neconfirmate.
- Această evindenţă ocupă o cantitate de memorie pentru fiecare legătură, rezultă limitarea numărului de conexiuni deservite
- Cu DF serverul nu trebuie să gestioneze independent fiecare conexiune
- Într-un moment dat serverul trimite acelaşi pachet la toţi utilizatori indiferent dacă a primit ACK sau NACK

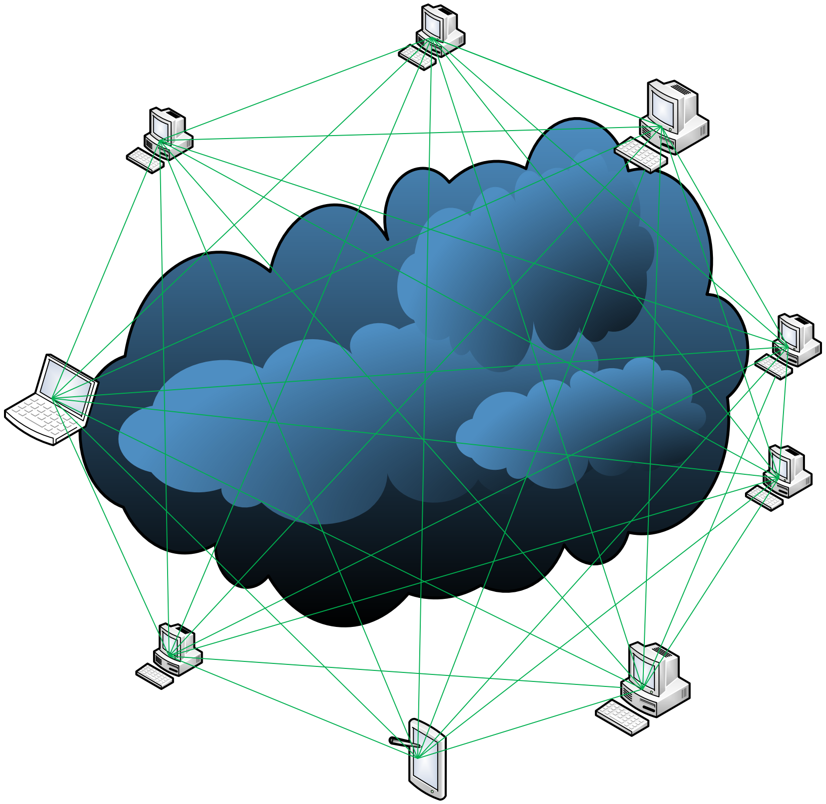
[6] Michael Mitzenmacher, “Digital Fountains: A Survey and Look Forward”, http://www.eecs.harvard.edu/~michaelm/postscripts/itw2004.pdf

Video streaming

- Aplicaţie în timp real
- Dacă seconsideră că întregul fişier reprezintă cele k simboluri de intrare, decodare poate să inceapă după recepţionarea a cel puţin k pachete codate – nu este utilizabil pt video streaming
- Fişierul se împarte în segmente, fiecare segment este codat separat
- În timp ce primul segment este redat, se recepţionează al doilea segment....
- Probleme la proiectarea sistemului:
  - Dimensiunea segmentelor
  - Numărul de pahete codate necesare (rata de transmisie)
  - Combinare cu multicastul classic (pt on demand)

Overlay Networks

- Fiecare utilizator transmite pachetele codate în loc de pachete informaţionale
- Un utilizator când a reuşit să decodeze informaţia poate să genereze alt set de pachete codate
- Probleme ex:
  - Dacă sursa iniţială dispare din reţea înainte ca să ajungă în reţea un număr suficient de pachete codate pt reconstruirea informaţiei originale



Stocare distribuită

- Fişierul este împărţit în *n* blocuri de dimensiune fixă, aceste blocuri sunt repetate şi distribuite prin sistem.
- Fiecare bloc trebuie repetat de *m* ori pentru a tolera căderea a *m-1* servere
- În cazul utilizării DF calculează *n+m* pachete codate, şi acestea se stochează.
- Clientul trebuie să descarce numai orice $n \cdot (1+\varepsilon)$, $\varepsilon \to 0$, blocuri, şi din acestea se pot calcula cele *n* blocuri ale fişierului.
  - Se simplifică găsirea şi descărcarea unui anumit bloc.
  - Accesul lent şi pierderile de pachete sunt eliminate
  - Se obţine un spaţiu de stocare stabil pentru arhivare.

Rutare dispersivă

- Spre deosebire de procedurile de rutare convenţionale, care rutează un mesaj de-a lungul unei anumite căi între sursă şi destinaţie, mecanismele de rutare dispersive (multicale) subdivizează mesajul şi îl împrăştie pe câteva căi din reţea