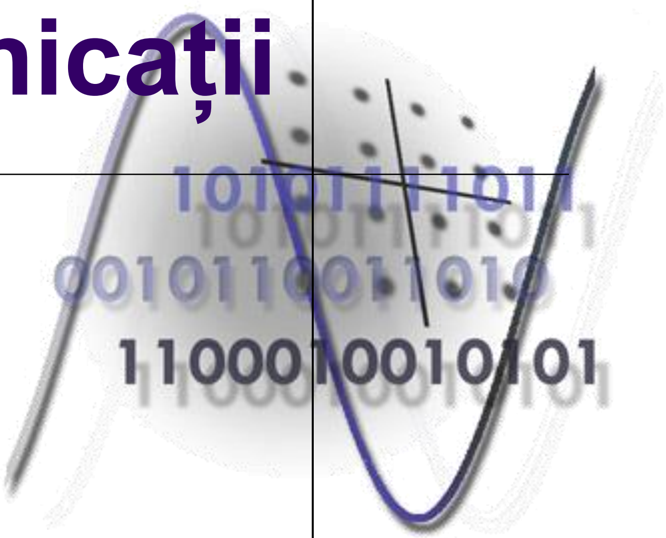
Tehnici avansate de codare și control al fluxului de date în rețelele de telecomunicații



TACCFD Curs 1

Cuprins

- Tehnici de codare de tip Digital Fountain (DF).
- Coduri de tip Tornado și de Raptor.
- Implementarea conceptului de DF cu ajutorul codurilor cu rată finită.
- Tehnici de codare de tip Network Coding.
- Generarea și optimizarea rețelei de codare Network Coding.
- Comunicații de tip swarm.
- Coduri liniare de tip Network Coding utilizate sisteme de tip swarm.
- Tehnici de codare NC utilizate în rețele wireless de tip “mesh”.
- Tehnici de codare distribuită și NC utilizate în rețele celulare cooperative.
- Tehnici de optimizare bazate pe Teoria Jocurilor
- Tehnici de optimizare bazate pe Algoritmi Genetici

Modul de examinare și atribuire a notei

- Formula de calcul a notei:
    - N=0,6Examen+0,4Miniproiect

Tehnici de codare de tip Digital Fountain

- Fundamente teoretice
- Algoritmi de codare și decodare

Tehnici de codare de tip Digital Fountain

- Conceptul DF
- DF cu coduri corectoare de ștergeri
- Coduri DF lineare, aleatoare
- Coduri LT

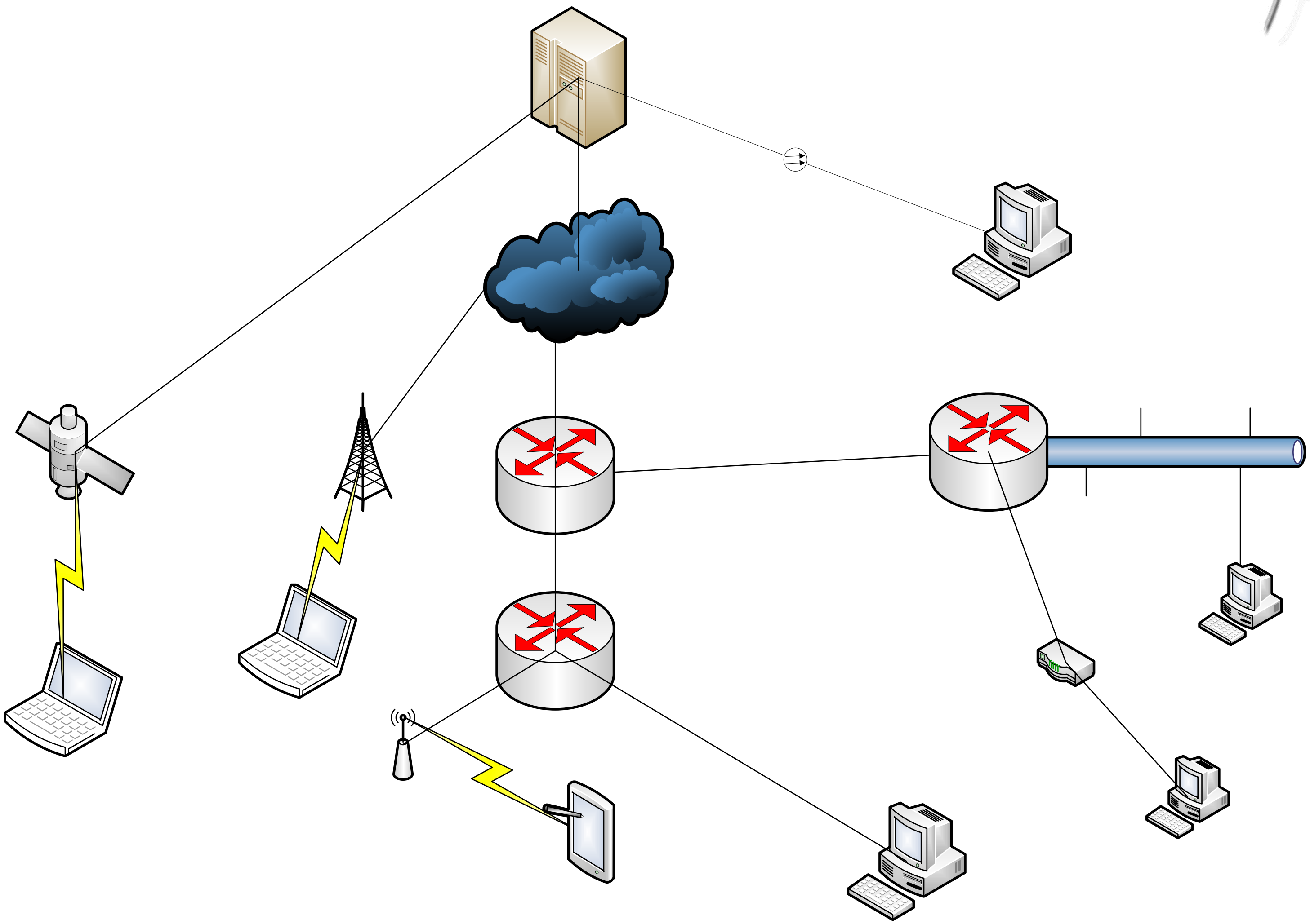
Proprietățile unui protocol ideal[1]

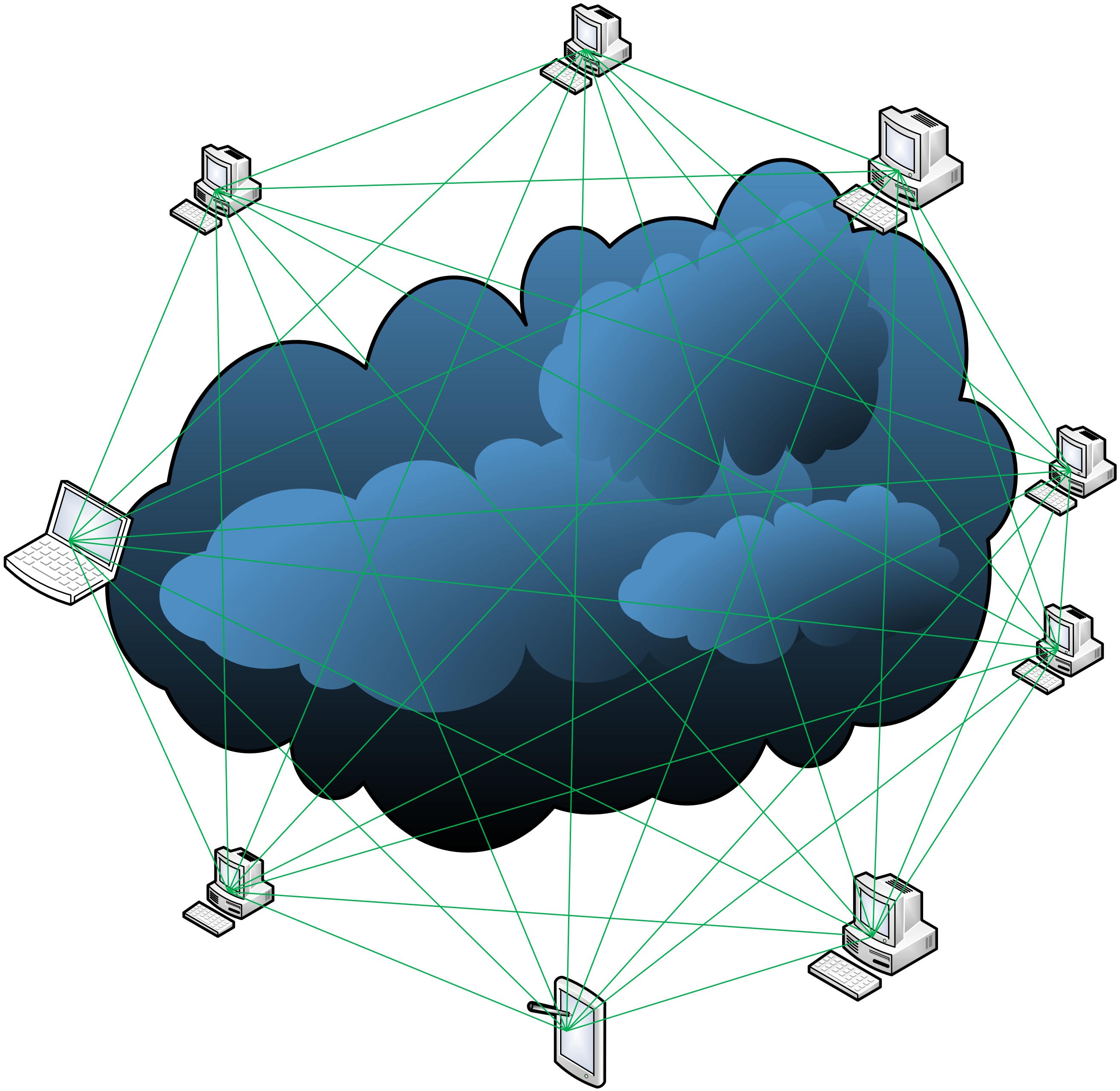
- **Fiabil** – este garantat că datele ajung în întregime
- **Eficient** - atât numărul de pachete necesare pentru reconstrucția informației cât și timpul necesar reconstrucției din pachetele recepționate trebuie să fie minim
- **Tolerant** – protocolul trebuie să suporte o populație de receptori eterogena (deferite debite, rata de pierderi etc.)
- **La cerere** – Utilizatorii pot inițializa sesiunile de comunicări in momente aleatoare, pot întrerupe respectiv relua sesiunile în orice moment de timp

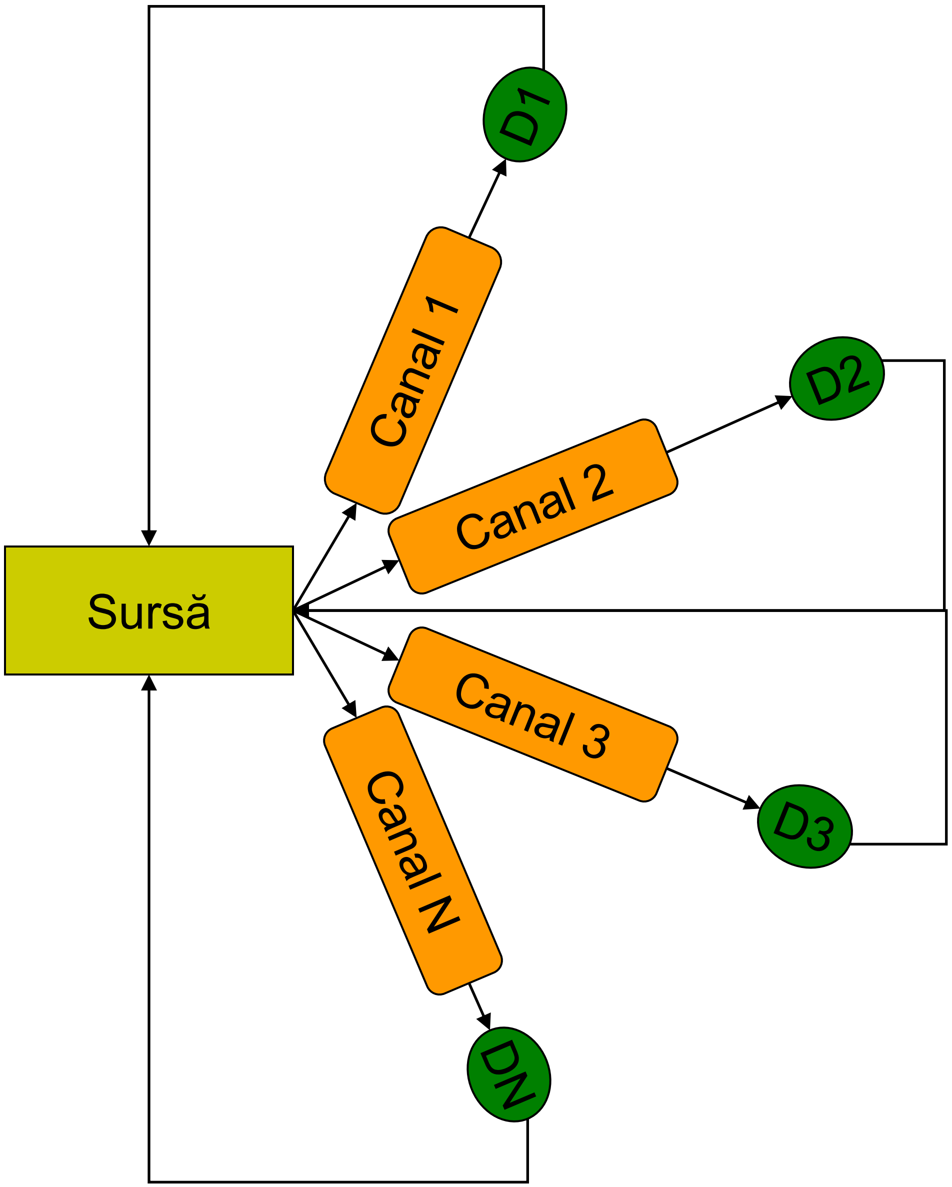
[1] John W.Byers, Michael Luby, Michael Mitzenmacher, Ashutosh Rege; “A Digital Fountain Approach to Reliable Distribution of Bulk Data” -Proceedings of the ACM SIGCOMM '98

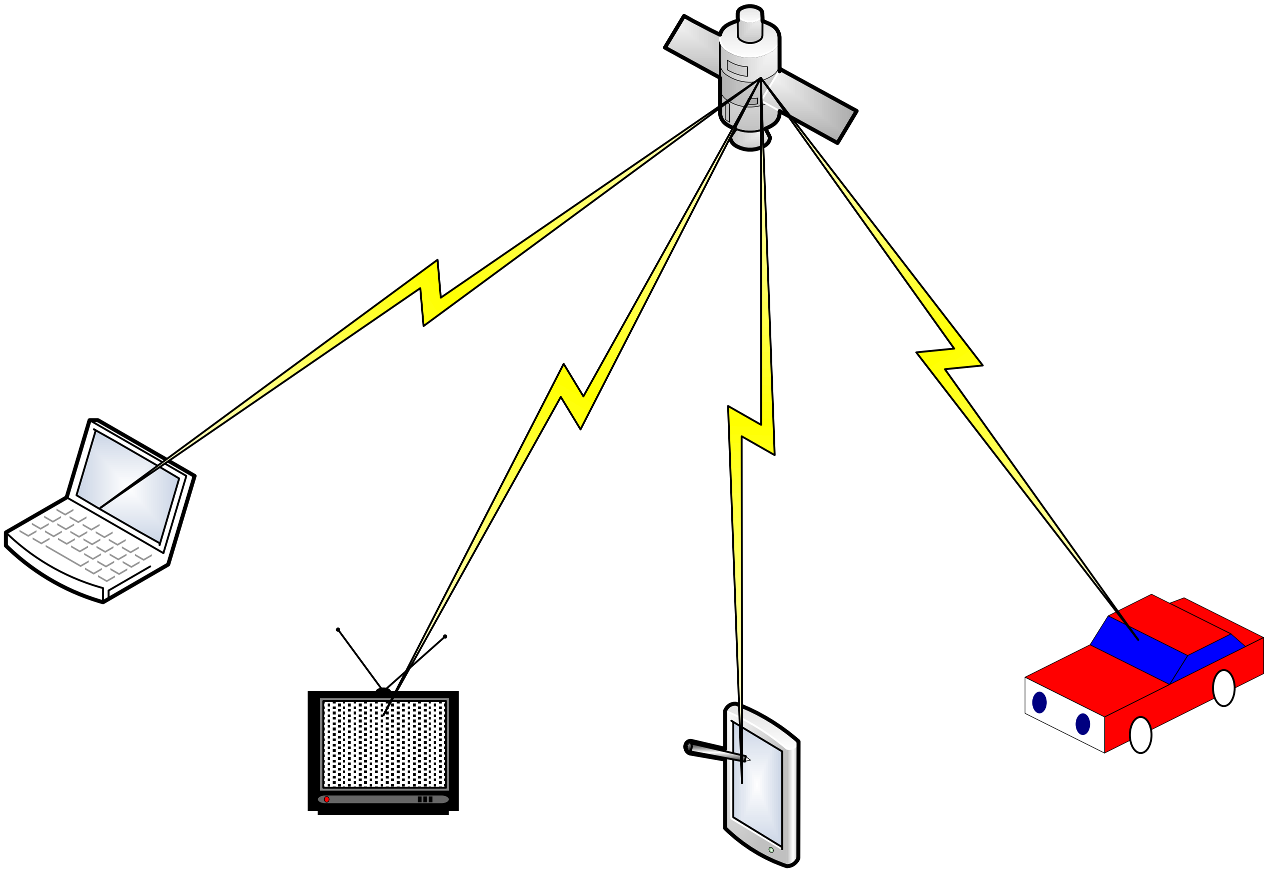
Fiabilitatea pe internet

- Pentru asigurarea fiabilități se utilizează
  - ARQ (pe canale cu Feedback)
    - Avantaje:
      - Asigură fiabilitatea transmisiei si la condiții severe de canal
    - Dezavantaje:
      - Resurse utilizate
      - Necesită legătură bidirecțională
  - Codarea canalului
    - FEC
      - Nu necesită (teoretic) feedback
      - Poate să corecteze un număr predefinit de erori (depinde de rata)






Sursă
Canal 1
Canal 2
Canal 3
Canal N
D1
D2
D3
DN



- Numărul cererilor de retransmisie se poate reduce utilizând coduri corectoare de ștergeri (RS)
  - La *k* pachete de informație se adună *n-k* pachete de control
  - Codul RS poate corecta pana la *n-k* ștergeri din cuvântul de cod format din n pachete
- Deoarece fiecare legătură are propria probabilitate de pierdere a pachetelor rata codului utilizat se determină pe baza legăturii cu probabilitatea cea mai mare de pierderi
  - Pe legăturile bune se transmit prea multe pachete redundante

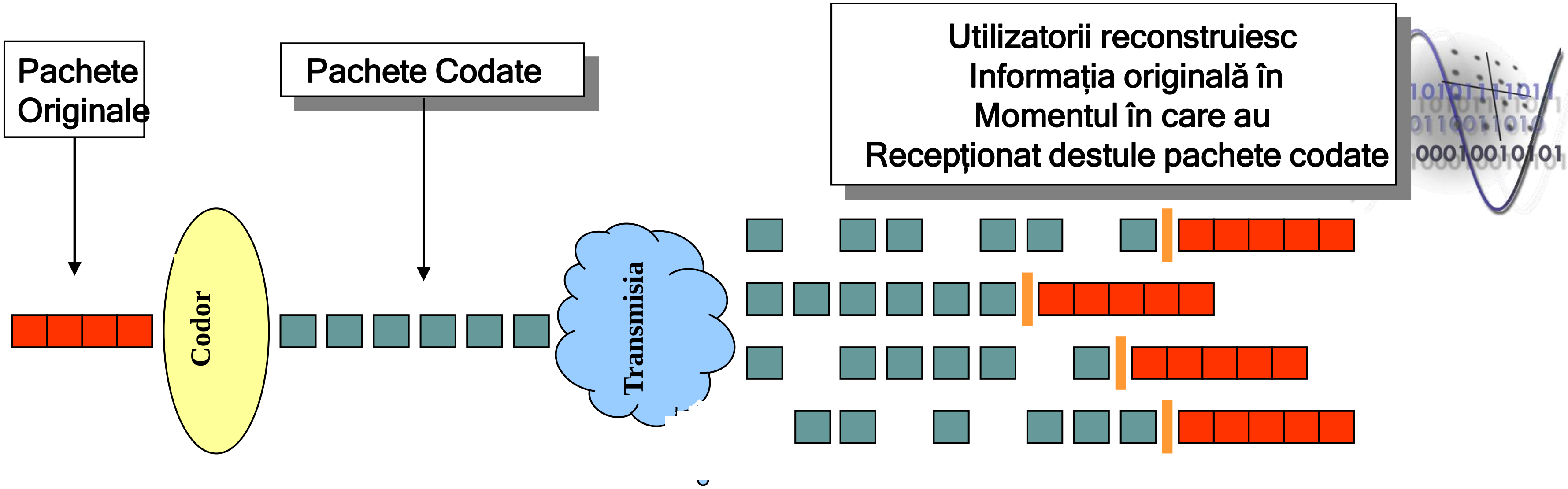
- Avantajele codului RS
  - Poate sa corecteze cele mai multe ștergeri la o lungime și o rată dată

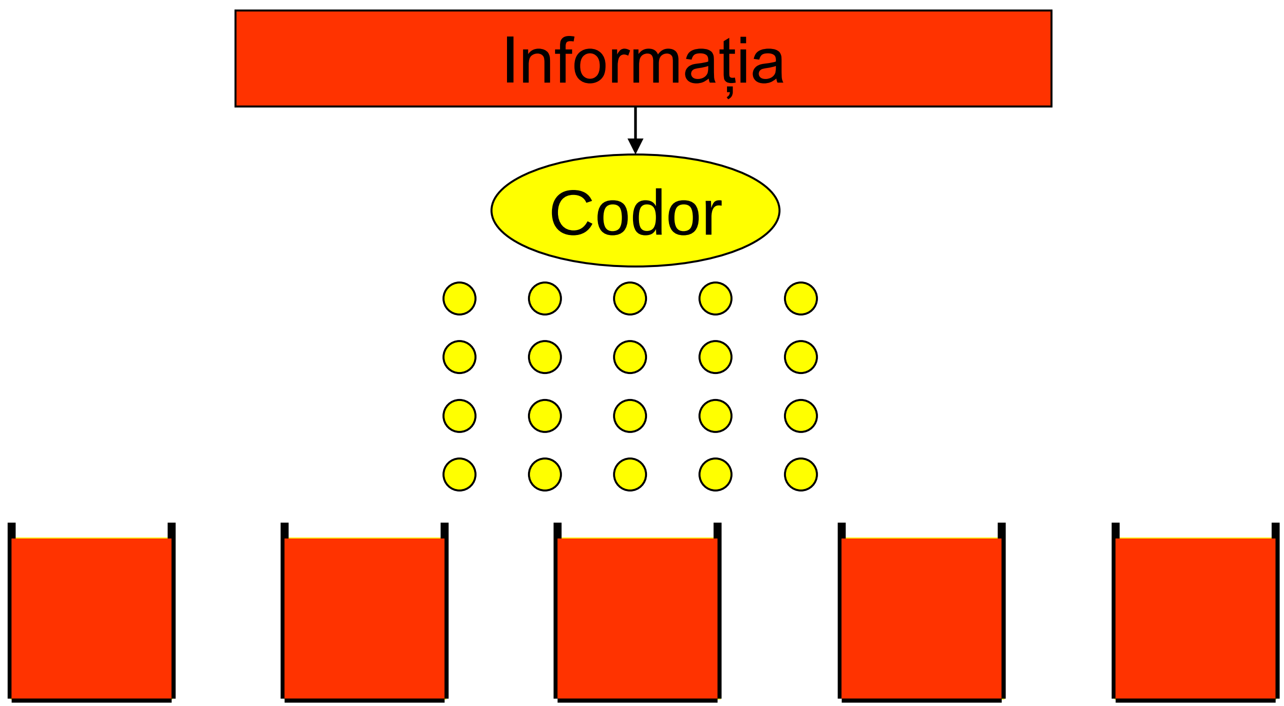
- Dezavantajele codului RS
  - Lungimea pachetelor foarte mică
  - Algoritmi de codare și decodare complicate
  - Imposibilitatea adaptării ratei

- Restricții (dezavantajele) utilizării codurilor corectoare de ștergeri clasice
  - Receptorul trebuie să cunoască exact poziția pachetului pierdut
  - Pachetele care ajung trebuie să ajungă în ordinea transmiterii, sau trebuie să fie numerotate
  - Lungimea pachetelor trebuie să fie foarte mică
  - Este complicat adaptarea capacității de corecție la caracteristicile fiecărui legături
- Probleme la implementarea protocoalelor eficiente și *la cere*

Conceptul Digital Fountain

- Posibilitatea generării dintr-un număr *k* finit de pachete informaționale un număr infinit de pachete codate
- Receptorul din oricare set de k+ε pachete codate informaționale poate reconstrui cele k pachete informaționale

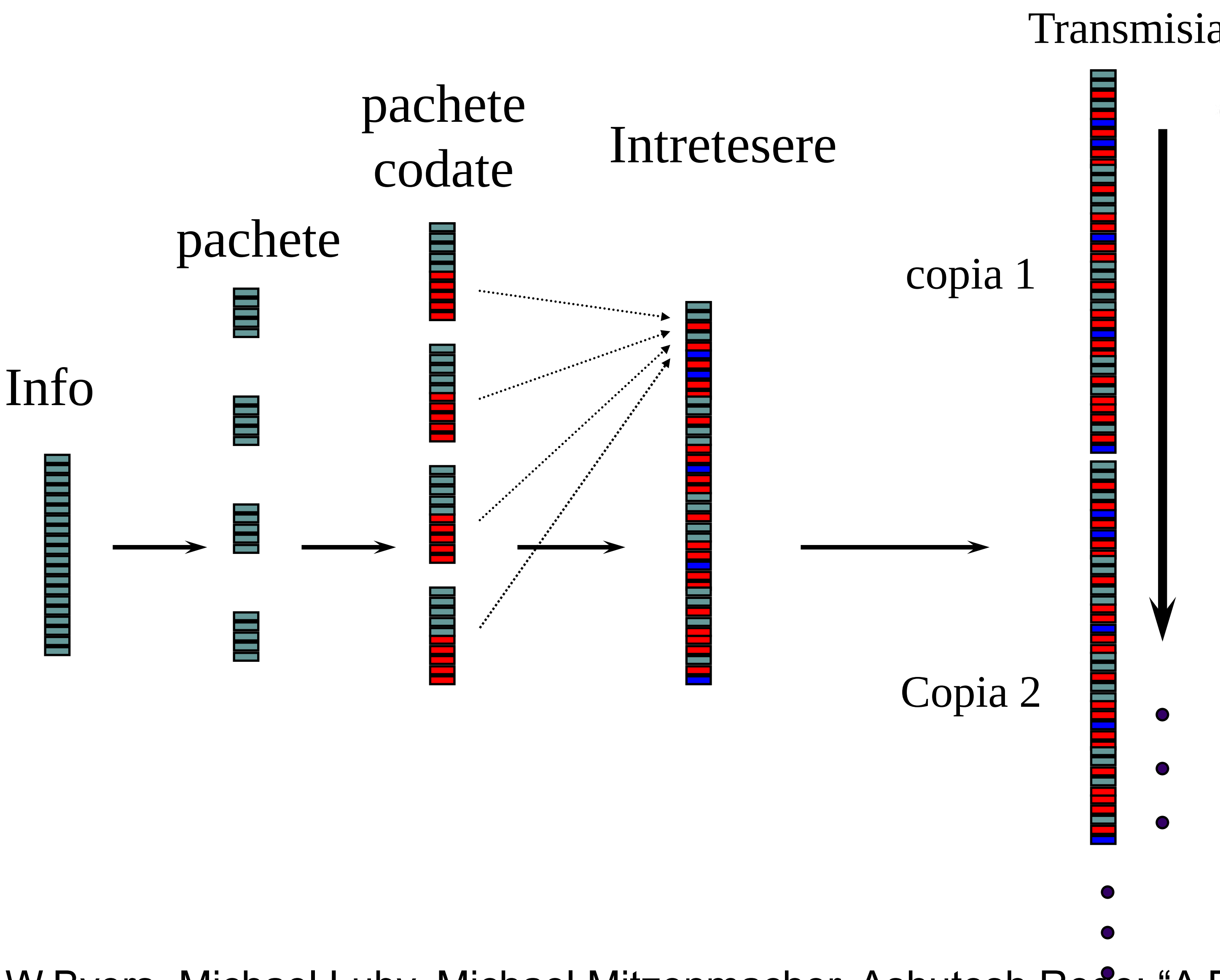

Pachete Originale
Pachete Codate
Utilizatorii reconstruiesc
Informația originală în
Momentul în care au
Recepționat destule pachete codate
Codor
Transmisia


Informația
Codor

- Nu necesită feedback
- Timpul de recepție pentru fiecare utilizator depinde de calitatea legăturii
- Sursa nu trebuie sa gestioneze separat fiecare conexiune
- Rata se adaptează automat la caracteristicile legăturii

DF cu coduri corectoare de ștergeri

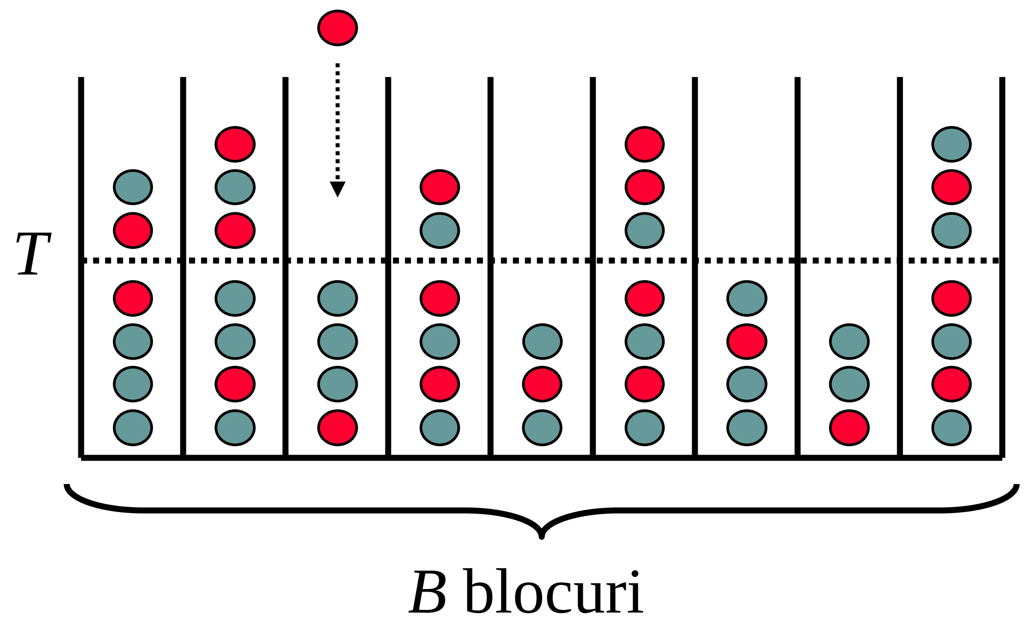
- Utilizând coduri RS
  - Informația originală se împarte în k pachete (simboluri)
  - Se alege un cod cu rata suficient de mică și se generează cele n simboluri codate
  - Sursa transmite repetat simbolurile codate
  - Când un utilizator a recepționat cel puțin k simboluri independente poate să reconstruiască informația originală
  - Fiecare utilizator poate să înceapă recepția pachețelelor în momente aleatoare
  - Fiind-că pachetele codate se transmit repetitiv, utilizatorii nu trebuie să recepționeze cele k pachete independente pe durata unei ciclu
    - Posibilitatea de a recepționa de mai multe ori același pachet



[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

Dezavantajele întrețeserii ciclice

- Așteaptă după ultimele blocuri



[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

Limitările utilizării codurilor RS

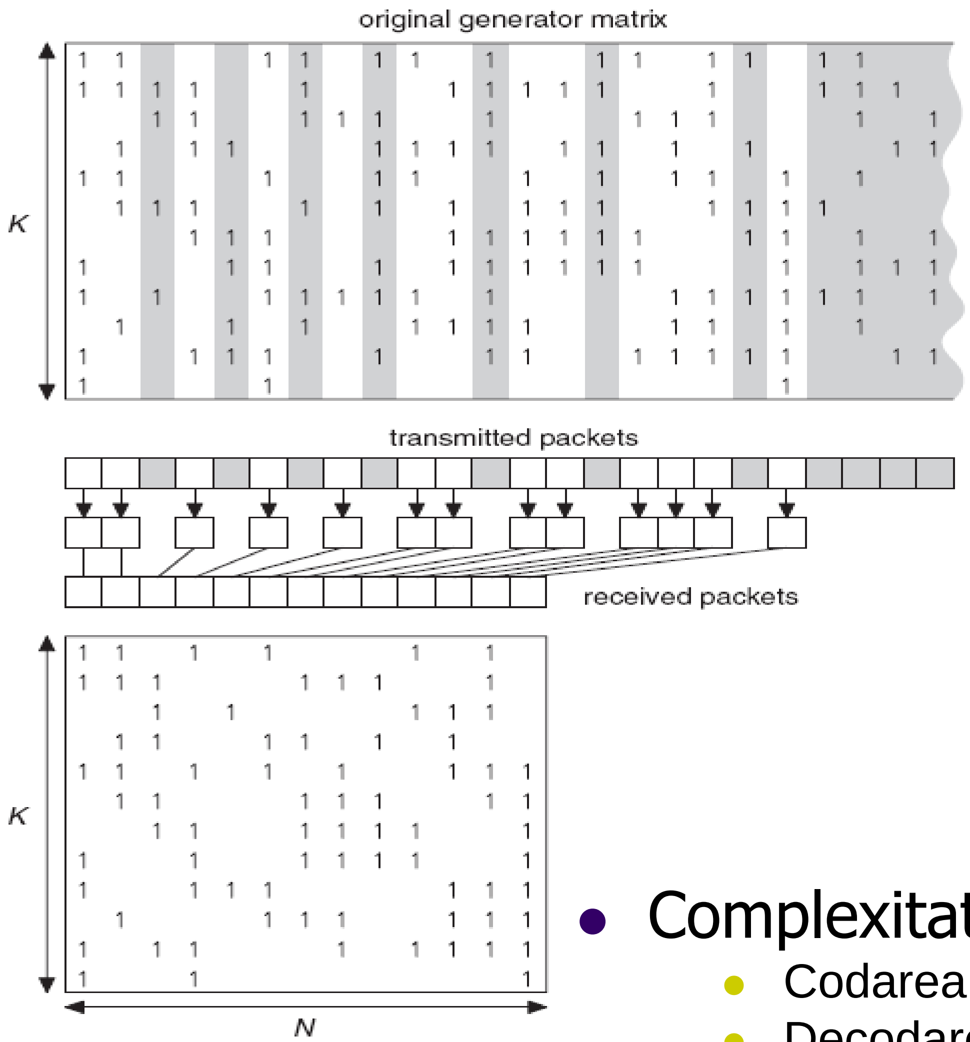
- Datorită complexității de implementare a codorului și a decodorului dimensiunea pachetelor în practică nu poate să fie mai mare de 16biţi (operații în GF(2^x))
- Datorită întrețeserii ciclice poate să crească numărul pachetelor redundante
- Complexitatea decodării depind de n, și k (cod "*puternic*" timp de decodare mare)

Coduri DF lineare, aleatoare

- Informația este împărțită în K pachete $s_1, s_2, \dots s_k$
- La fiecare perioadă de timp (marcat cu *n*), codorul generează k biți aleatorii $\{G_{kn}\}$
- Pachetul codat în momentul E_n se obține adunând modulo 2 pachetele pentru care elementele corespunzătoare din G_{kn} este 1, adică:

$$E_n = \sum_{k=1}^{K} s_k G_{kn}$$

- La fiecare pachet codat se adaugă vectorul G_{kn}



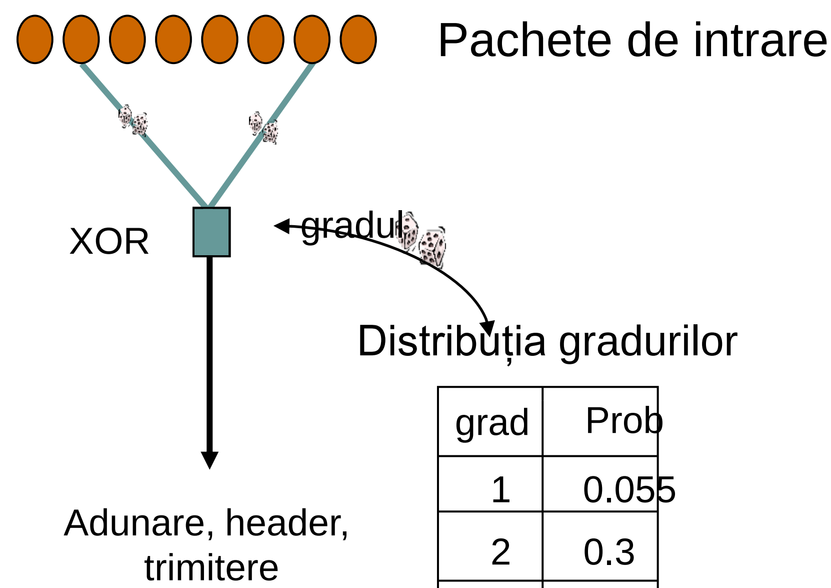
- Receptorul stochează pachetele capturate și construiește matricea de decodare
- Dacă s-a recepționat N=K pachete probabilitatea ca matricea de decodare să fie inversabilă este 0.289
- Dacă N>K informația poate fi decodată dacă și numai dacă există în matricea de decodare o sub-matrice cu dimensiuni KxK inversabilă
- Complexitatea implementării:
  - Codarea $O(k^2)$
  - Decodarea $O(k^3)$
- D.J.C. MacKay, "Fountain Codes", IEE Proceedings – Commun. Vol. 152, No 6, Dec 2005.

Codurile Luby-Transform (LT)

- Codurile LT reprezintă prima realizare practică a codurilor rateless; nu trebuie definită o rată fixă înainte de codare
- Se poate construi un flux infinit de pachete codate din pachetele informaționale

Principiul de codarea LT [3]

- Pentru pachetul codat care urmează să fie generat se alege aleator un grad *d*, conform unei distribuții predefinite
- Se aleg aleator *d* pachete din mulțimea pachetelor informaționale
- Pachetul codat se obține adunând modulo doi pachete selecționate la pasul anterior

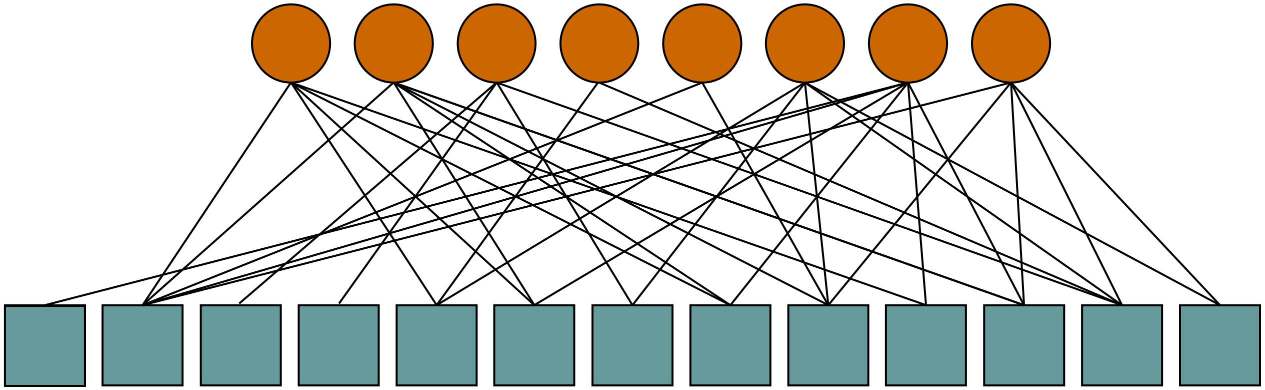


grad	Prob
1	0.055
2	0.3
3	0.1
4	0.08
⋮	⋮
100000	0.0004

[3] Michael Luby; “LT Codes” -*Digital Fountain, Inc.*, Fremont, 2002

[4] Amin Shokrollahi; “Fountain Codes” EPFL and Digital Fountain, Inc.

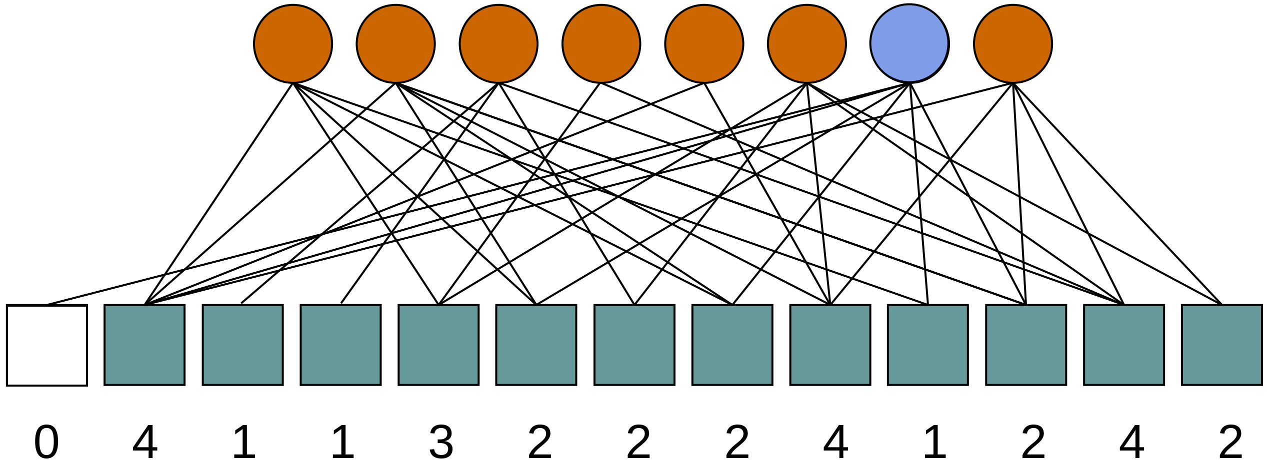
Graful asociat codului



Principiul de decodare

- Regula de decodare
  - Dacă există minim un simbol codat care are un singur vecin, atunci valoarea vecinului respectiv este o copie a simbolului codat.
  - Valoarea simbolului de intrare recuperat va fi adunat modulo 2 cu toate simbolurile codate rămase care au ca și vecin acel simbol de intrare
  - gradul simbolurilor codate, la care a fost adunat simbolul informațional recuperat este redus cu unu, și acest simbol de intrare este eliminat din lista vecinilor

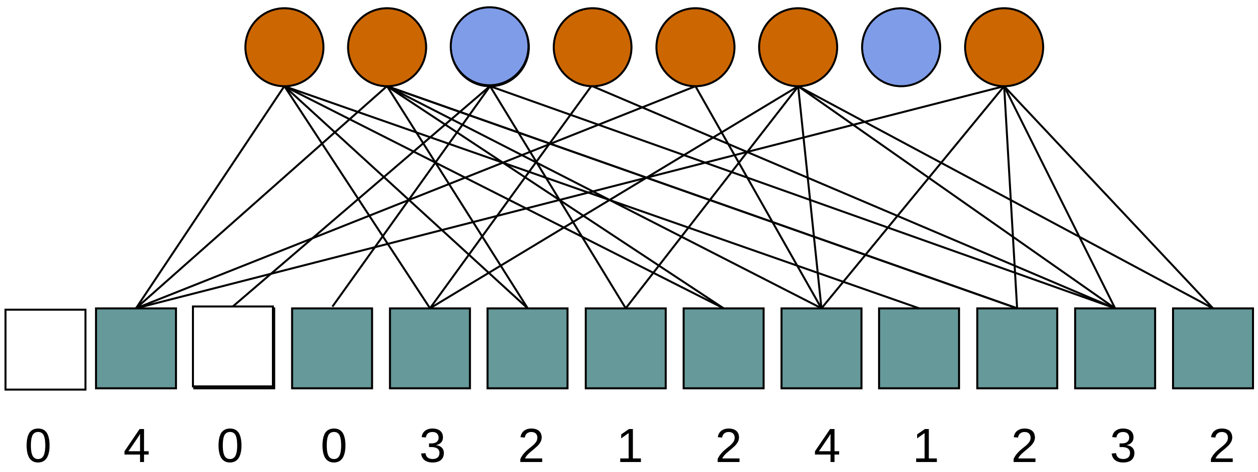
Decodare LT



[3] Michael Luby; “LT Codes” -*Digital Fountain, Inc.*, Fremont, 2002

[4] Amin Shokrollahi; “Fountain Codes” EPFL and Digital Fountain, Inc.

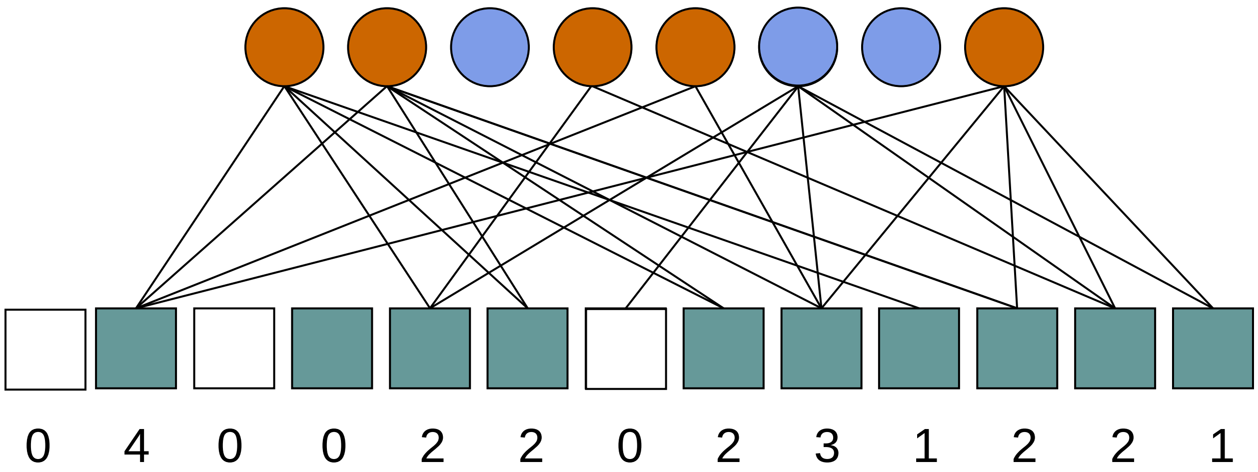
Decodare LT



[3] Michael Luby; “LT Codes” -*Digital Fountain, Inc.*, Fremont, 2002

[4] Amin Shokrollahi; “Fountain Codes” EPFL and Digital Fountain, Inc.

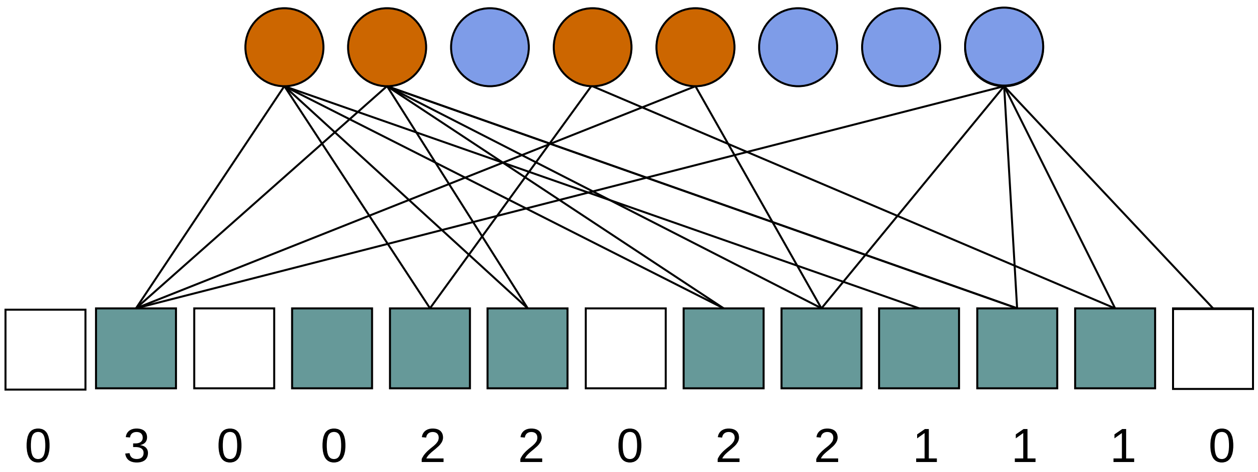
Decodare LT



[3] Michael Luby; “LT Codes” -*Digital Fountain, Inc.*, Fremont, 2002

[4] Amin Shokrollahi; “Fountain Codes” EPFL and Digital Fountain, Inc.

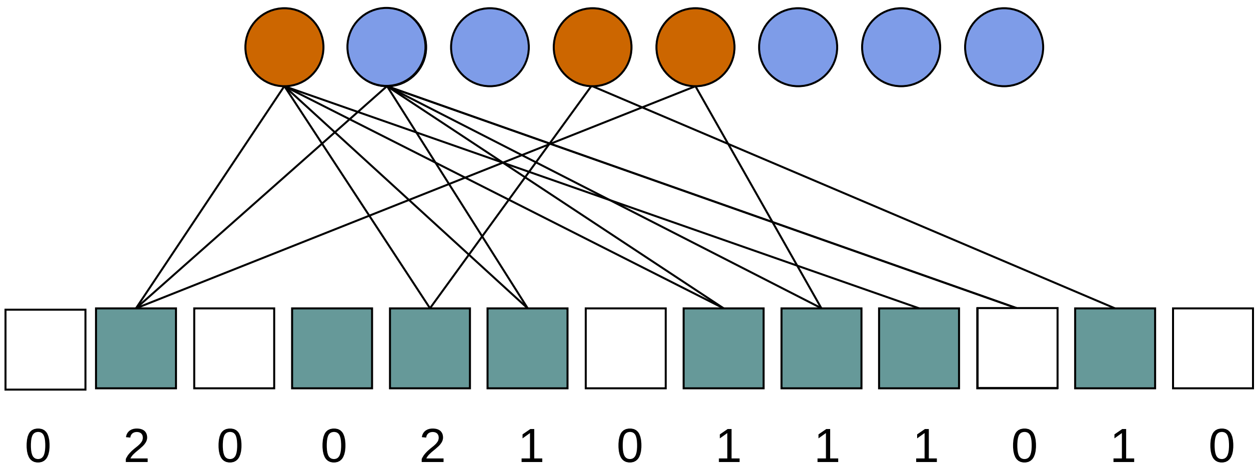
Decodare LT



[3] Michael Luby; “LT Codes” -*Digital Fountain, Inc.*, Fremont, 2002

[4] Amin Shokrollahi; “Fountain Codes” EPFL and Digital Fountain, Inc.

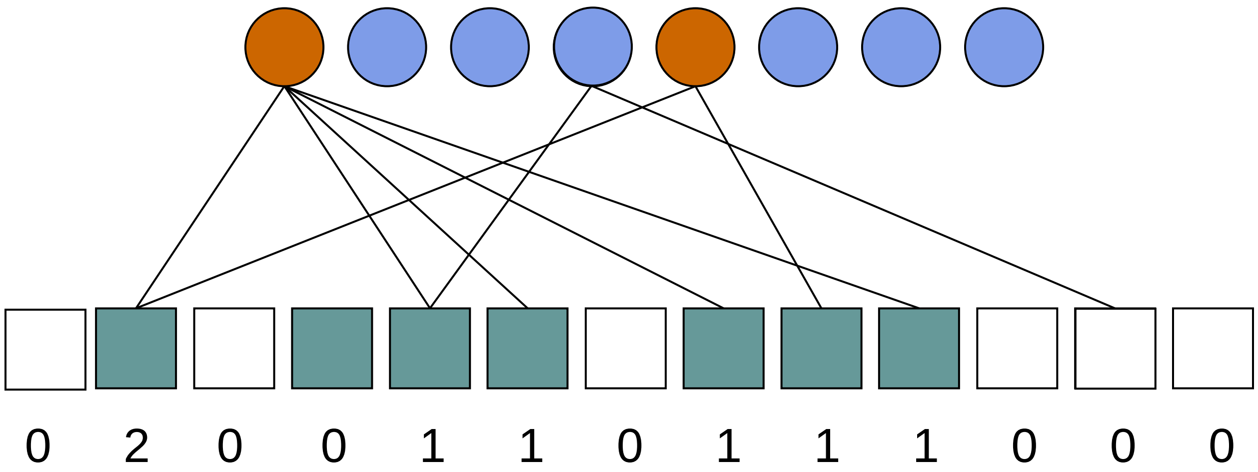
Decodare LT



[3] Michael Luby; “LT Codes” -*Digital Fountain, Inc.*, Fremont, 2002

[4] Amin Shokrollahi; “Fountain Codes” EPFL and Digital Fountain, Inc.

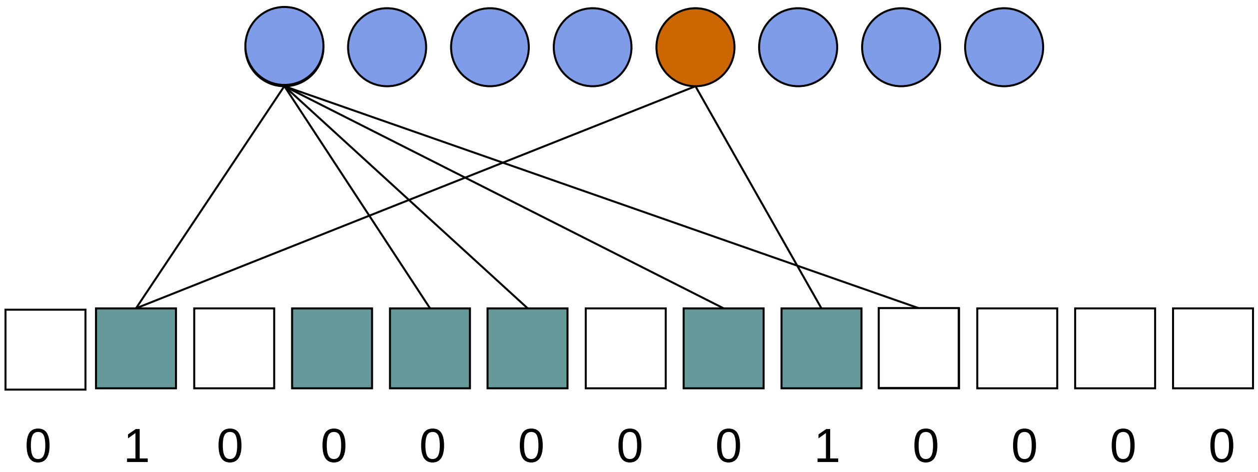
Decodare LT



[3] Michael Luby; “LT Codes” -*Digital Fountain, Inc.*, Fremont, 2002

[4] Amin Shokrollahi; “Fountain Codes” EPFL and Digital Fountain, Inc.

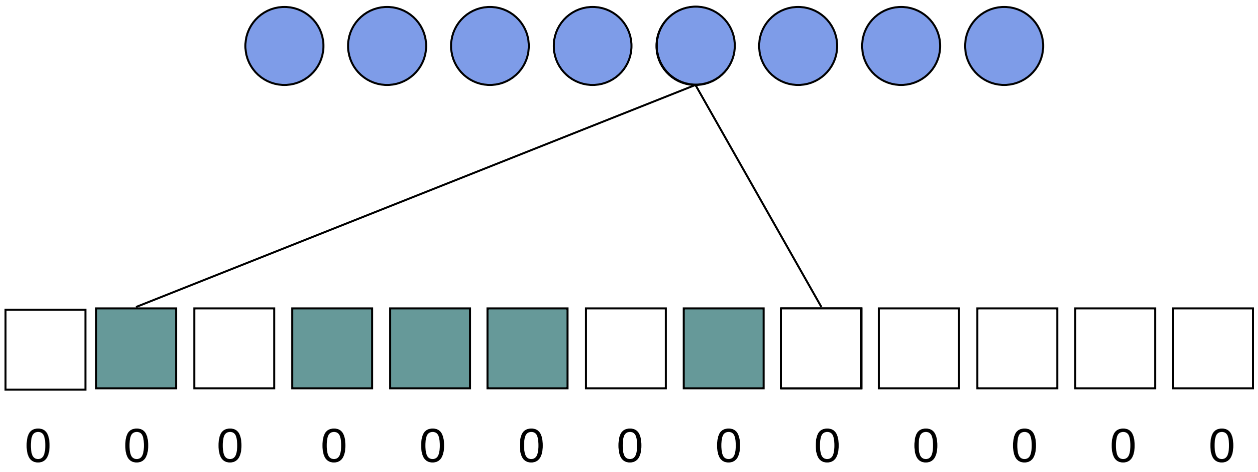
Decodare LT



[3] Michael Luby; “LT Codes” -*Digital Fountain, Inc.*, Fremont, 2002

[4] Amin Shokrollahi; “Fountain Codes” EPFL and Digital Fountain, Inc.

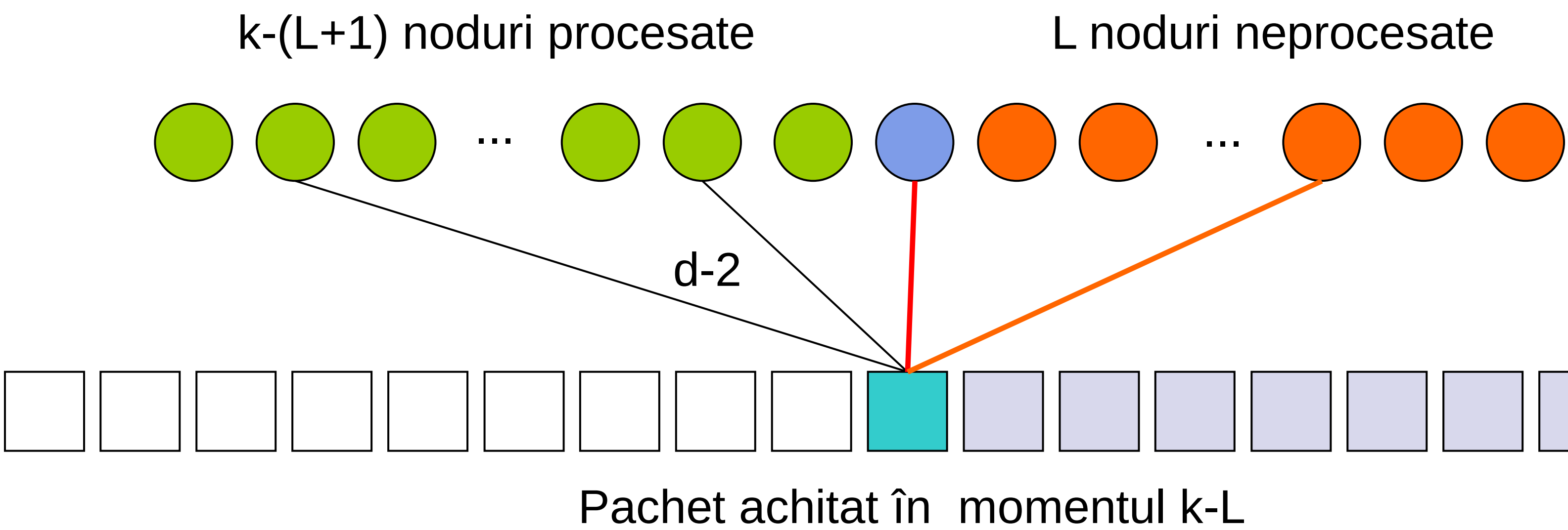
Decodare LT



[3] Michael Luby; “LT Codes” -*Digital Fountain, Inc.*, Fremont, 2002

[4] Amin Shokrollahi; “Fountain Codes” EPFL and Digital Fountain, Inc.

- Pt studierea distribuțiilor se reformulează procesul de decodare:
  - La început toate simbolurile informaționale sunt descoperite
  - În prima etapă toate simbolurile codate care sunt formate dintr-un singur simbol informațional, “acoperă” singurul lor vecin. Mulțimea formată din simbolurile acoperite, care încă nu au fost procesate, se numește “riplu”
  - În pașii următor este luat câte un simbol informațional din riplu, este adunat la simbolurile codate la care este vecin și se reduce gradul acestor simboluri.
  - Dacă un simbol codat astfel va avea grad 1, acest simbol va acoperi vecinul său, iar acest simbol codat astfel va fi “***achitat***” Dacă acest vecin acoperit nu a fost acoperit mai devreme, de un alt simbol codat, atunci dimensiunea riplu-lui crește.
  - Procesul se termină când riplul se golește
    - Procesul de decodare este cu succes dacă la golirea riplului nu mai sunt simboluri de intrare neacoperite.



- Probabilitatea *q(d,L)* ca un pachet cu gradul inițial d să fie achitat când mai sunt L pachete informaționale neprocesate este:

$$q(1,k)=1$$

$$\textit{pentru} \quad d=2,\ldots,k \; si \; L=k-d+1,\ldots,1$$

$$q(d,L)=\frac{\binom{k-(L+1)}{d-2}}{\binom{k-1}{d}}L=\frac{d(d-1)\cdot L\prod_{j=0}^{d-3}k-(L+1)-j}{\prod_{j=0}^{d-1}k-j}$$

$$\textit{pentru} \quad \textit{celelalte valori } d \textit{ si } L$$

$$q(d,L)=0;$$

- Probabilitatea r(d,L) este probabilitatea ca un simbol codat să aibă gradul d, și să fie achitat când mai sunt L simboluri de intrare neprocesate
    - r(d,L)=p(d)q(d,L)
- Probabilitatea r(L) ca un pachet să fie achitat când mai sunt L pachete informaționale neprocesate este:

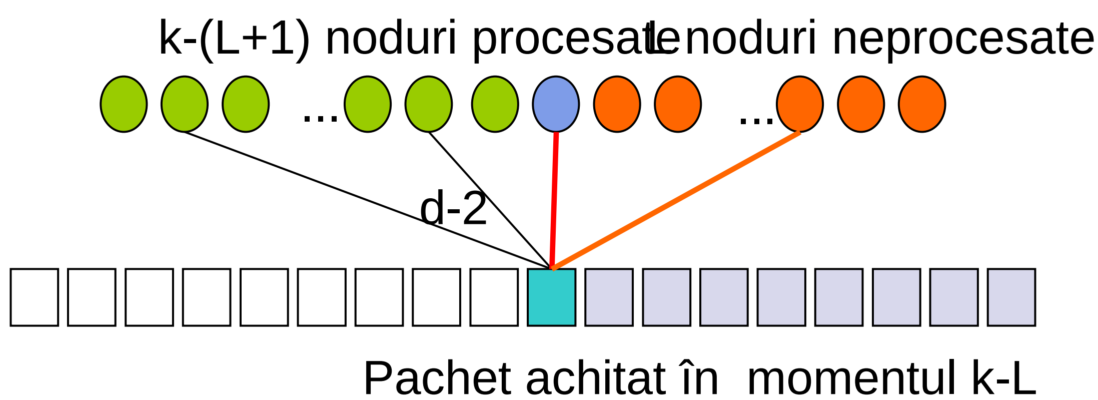
$$r(L)=\sum_{d} r(d,L)=\sum_{d=2}^{k} p(d)\frac{d(d-1)\cdot L\prod_{j=0}^{d-3} k-(L+1)-j}{\prod_{j=0}^{d-1} k-j}$$

Distribuția *Soliton* ideală

- Cerințele impuse unei distribuții de graduri sunt:
  - Un număr mediu de simboluri codate cât mai mic posibil pentru a asigura succesul procesului LT.
  - Gradul mediu al simbolurilor cât mai mic posibil. Gradul mediu definește numărul operațiilor de simbol necesare pentru generarea unui simbol codat, iar k*(grad mediu) este numărul operațiilor necesare pentru recuperarea completă a datelor.

Distribuția *Soliton* ideală

- O proprietate elementară a unei distribuții “ideale” este ca la procesul de decodare, la procesarea unui simbol informațional la riplu să fie adăugat un simbol acoperit.
- Asta asigură că dimensiunea riplului să nu fie niciodată prea mică sau prea mare.
- r(L) –este probabilitatea ca la riplu să fie adunat un singur simbol la procesarea simbolului k-(L+1)



Distribuția *Soliton* ideală

- Ținând cont că:

$$r(L)=\sum_{d} r(d,L)=\sum_{d=2}^{k} p(d)\frac{d(d-1)\cdot L\prod_{j=0}^{d-3} k-(L+1)-j}{\prod_{j=0}^{d-1} k-j}$$

și

$$\sum_{d=2}^{k}\frac{L\prod_{j=0}^{d-3} k-(L+1)-j}{\prod_{j=0}^{d-1} k-j}=1 \quad pt\ orice\ L>1$$

Rezultă distribuția Soliton ideală:

$$p(d)=\begin{cases}\frac{1}{k}; & pt\ d=1\\ \frac{1}{d(d-1)}; & pt\ d=2,\ldots,k\end{cases}$$

Distribuția *Soliton* ideală

- Sunt necesare exact k simboluri codate pentru reconstruirea celor k simboluri de intrare
- Dimensiunea riplului este 1 pe toată durata decodări
- PERFORMAŢE FOARTE SLABE în practică
  - Deoarece riplul este foarte scurt, riplul poate să se golească înaintea decodării tuturor mesajelor informaționale

Distribuția Soliton Robust

- Cu cât dimensiunea riplului este mai mare cu atât probabilitatea ca riplul să se golească înaintea recuperării tuturor simbolurilor informaționale este mai mică
- Pentru a minimiza numărul total de simbolul codate utilizate la recuperarea simbolurilor informaționale dimensiunea riplului trebuie să fie cât mai mic posibil

Distribuția Soliton Robust

- Pentru un compromis se acceptă că din K pachete recepționate, decodorul LT nu reușește să determine informațiile originale cu probabilitatea δ
- pentru asigurarea ca probabilitatea de eroare să fie maxim δ, dimensiunea riplului trebuie să fie:

$$\ln\left(\frac{k}{\delta}\right)\sqrt{k}$$

- Iar numărul pachetelor codate necesare pentru decodare este:

$$K = k + O\left(\ln^2\left(\frac{k}{\delta}\right)\sqrt{k}\right)$$

Distribuția Soliton Robust

- Se alege lungimea riplului dorit ca fiind:

$$R = c\ln\left(\frac{k}{\delta}\right)\sqrt{k} \quad unde\ c > 0$$

- Se definește τ(d) ca fiind

$$\tau(d) = \begin{cases} \dfrac{R}{dk} & pt\ d = 1,\ldots,\dfrac{k}{R}-1 \\ R\dfrac{\ln\left(\dfrac{R}{\delta}\right)}{k} & pt\ d = \dfrac{k}{R} \\ 0 & pt\ d = \dfrac{k}{R}+1,\ldots,k \end{cases}$$

Distribuția Soliton Robust

- Se adună distribuția ideală p(d) la τ(d), normalizând această sumă cu β se obține distribuția robustă μ(d)

$$\beta = \sum_{d=1}^{k} p(d) + \tau(d)$$

$$\mu(d) = \frac{p(d) + \tau(d)}{\beta}$$

  - Valoarea medie a gradurilor este c*ln*(k)
  - Ovearheadul necesar este proporțional cu K

- 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

Probleme legate de DF

- Implementarea oricărei aplicații, se poate face numai cu acordul DF inc.