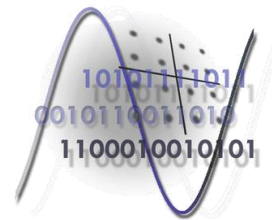


**MIXIT**

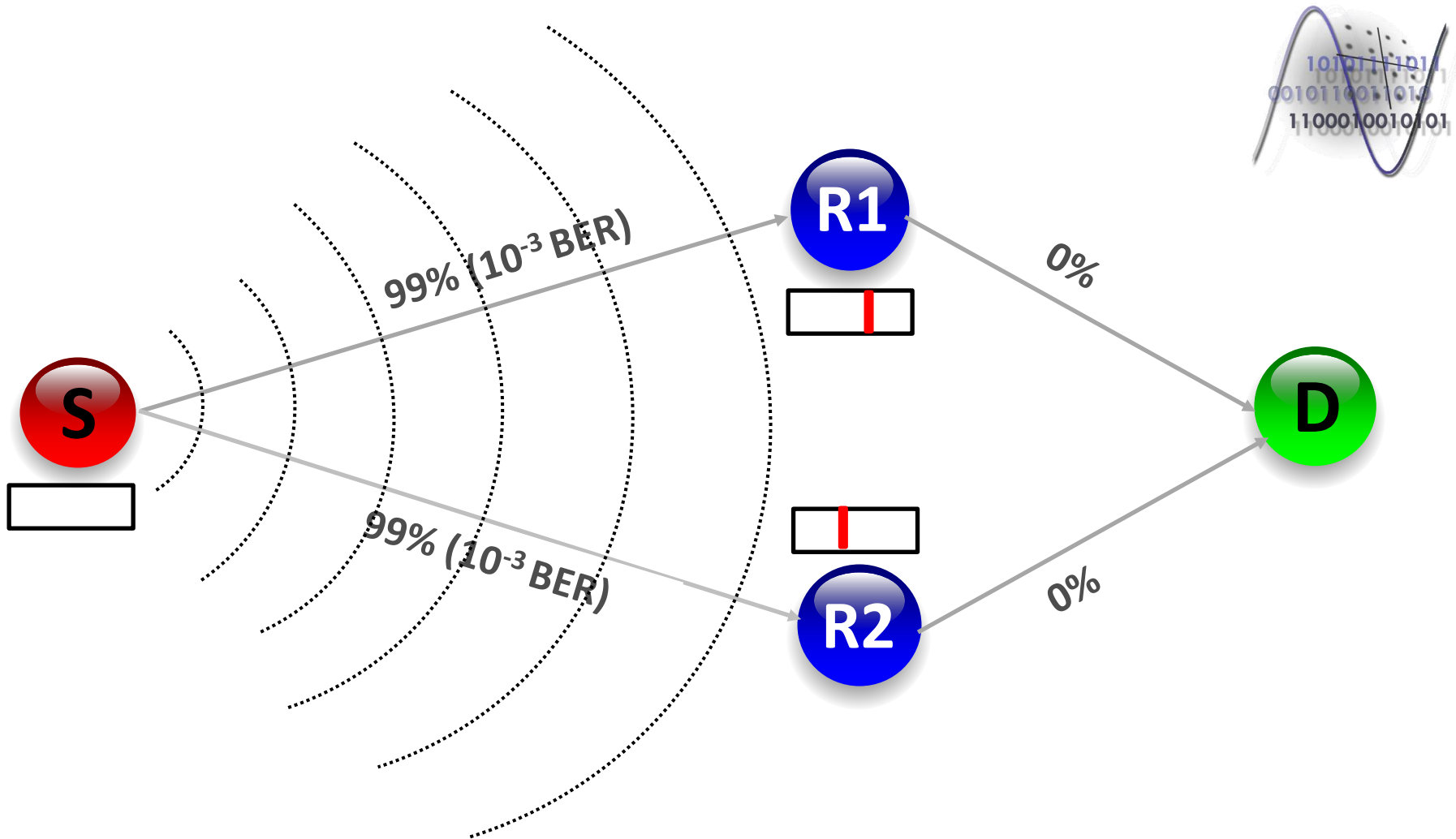
TACCFDRT Curs 7

1010111011  
1010111011  
0010110011010  
1100010010101

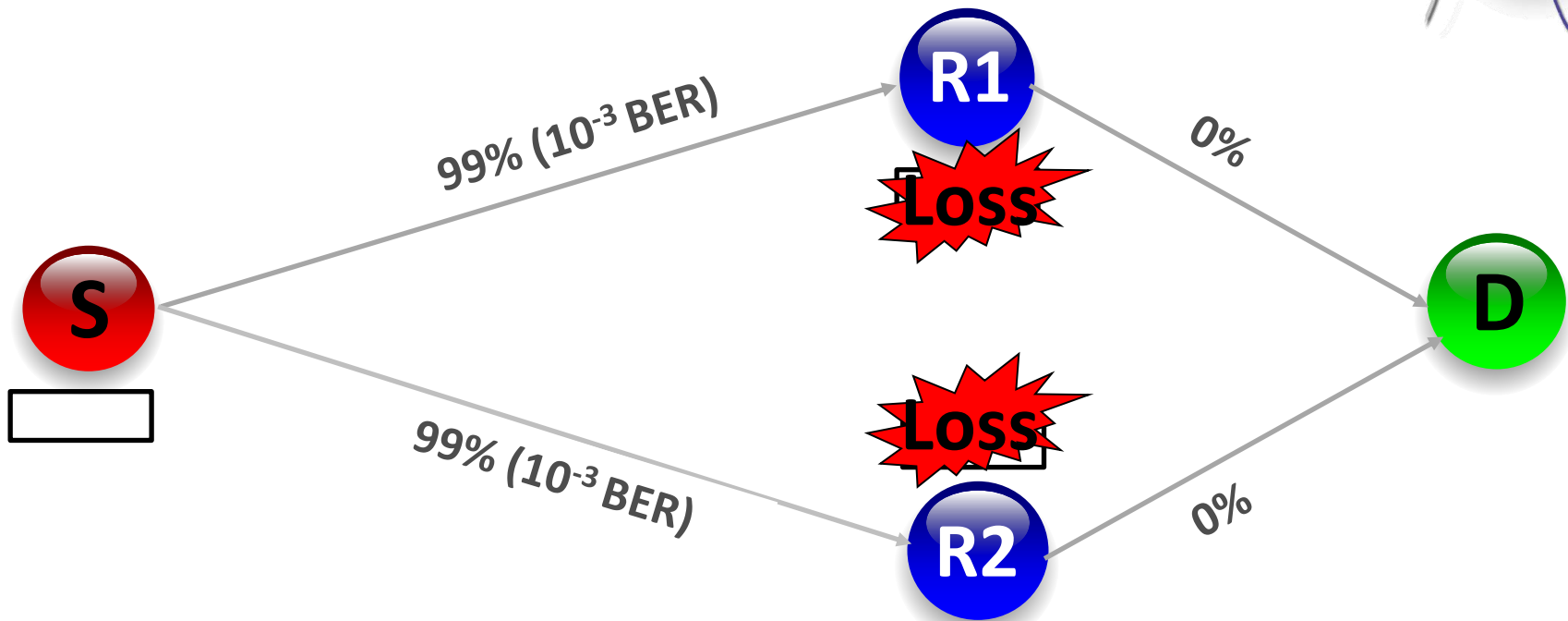
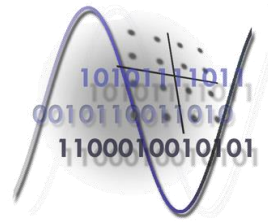
# Cuprins



- Tehnici de codare de tip "Xor in the Air"
  - MIXIT

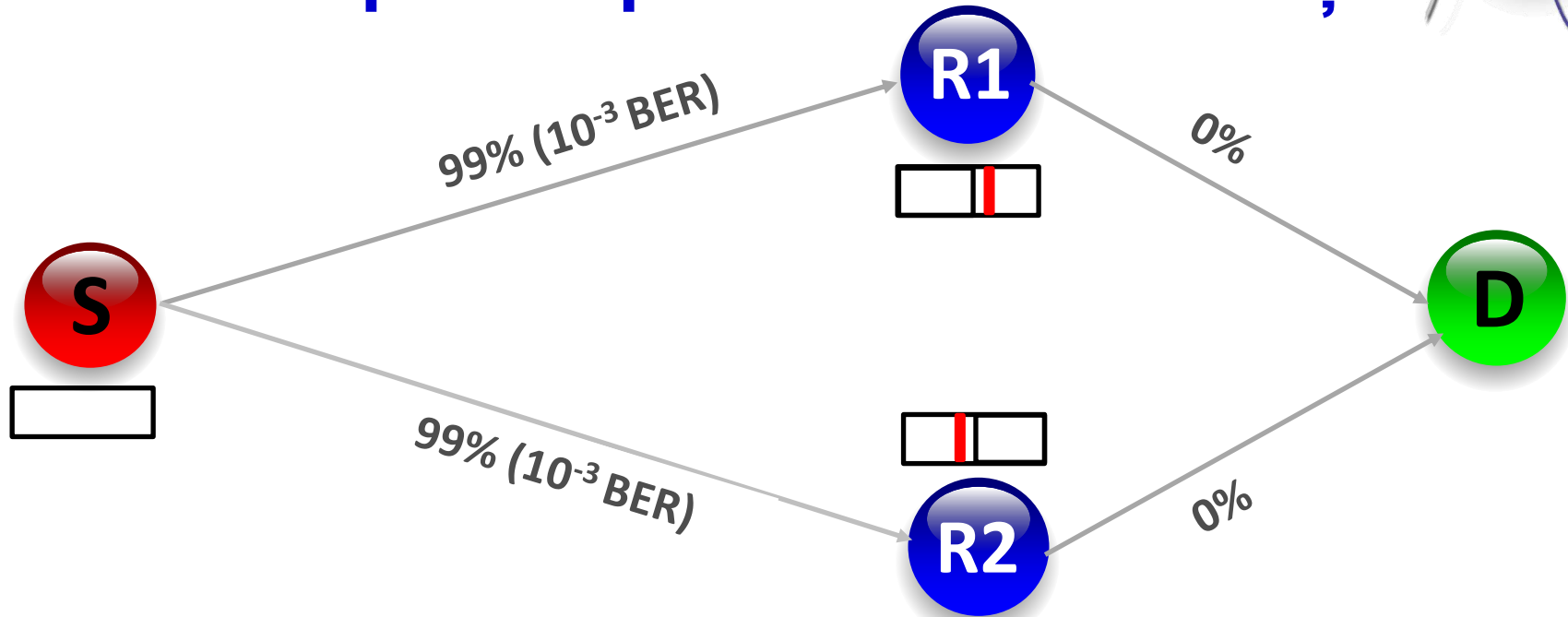
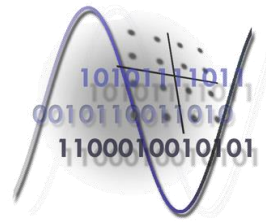


**Chiar daca un bit este eronat din 1000 → 99% probabilitate de  
eronare a pachetelor**



**Soluția „clasică” → Asigurarea fiabilității pe fiecare legătură  
→ 50 re-transmisii.....**

# Asigurarea fiabilității Rețelelor Wireless prin exploatarea diversității

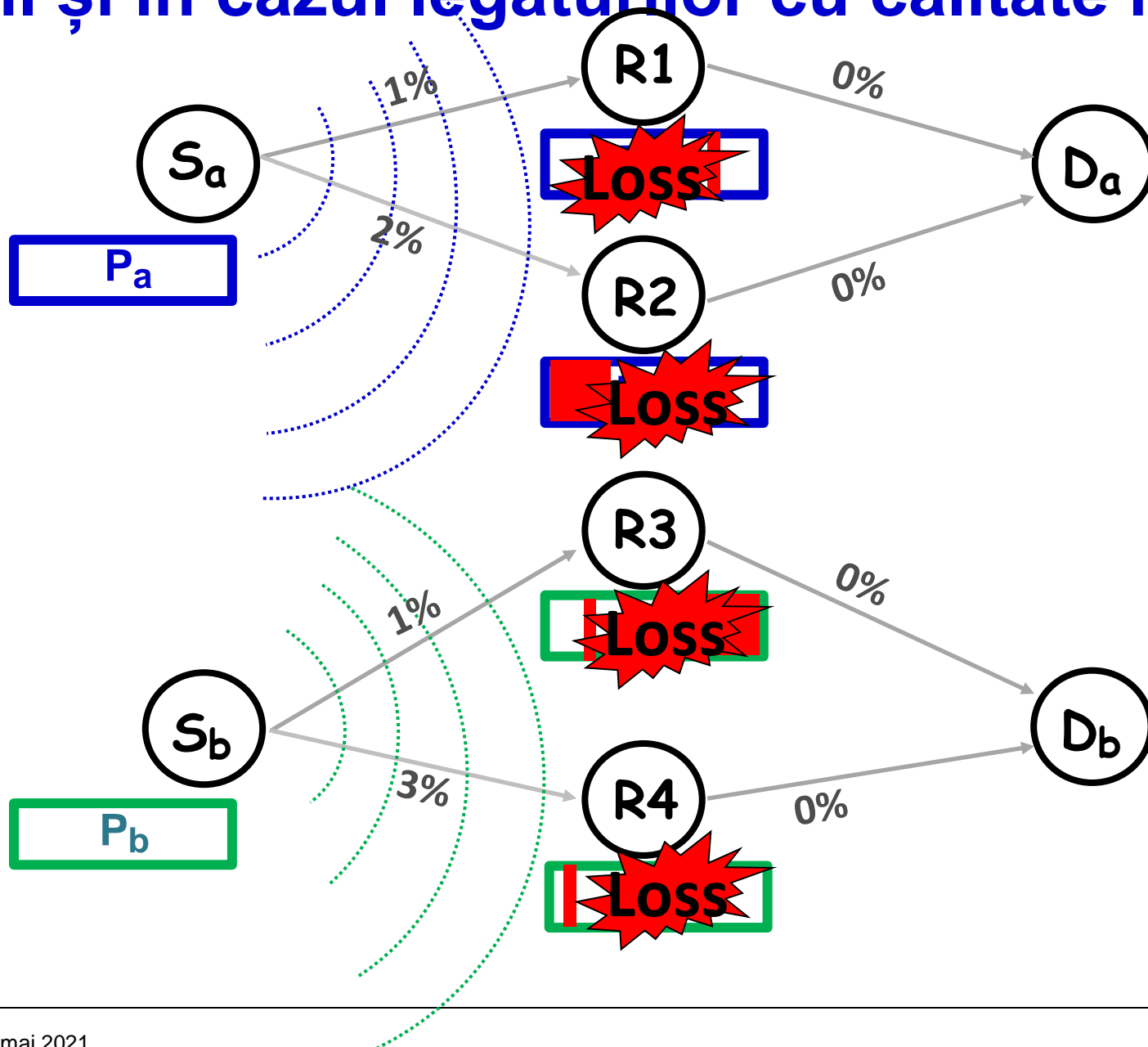
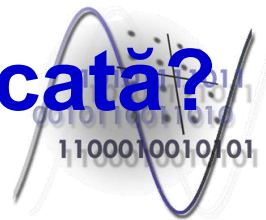


Soluția clasică → 50 tx → Throughput scăzut

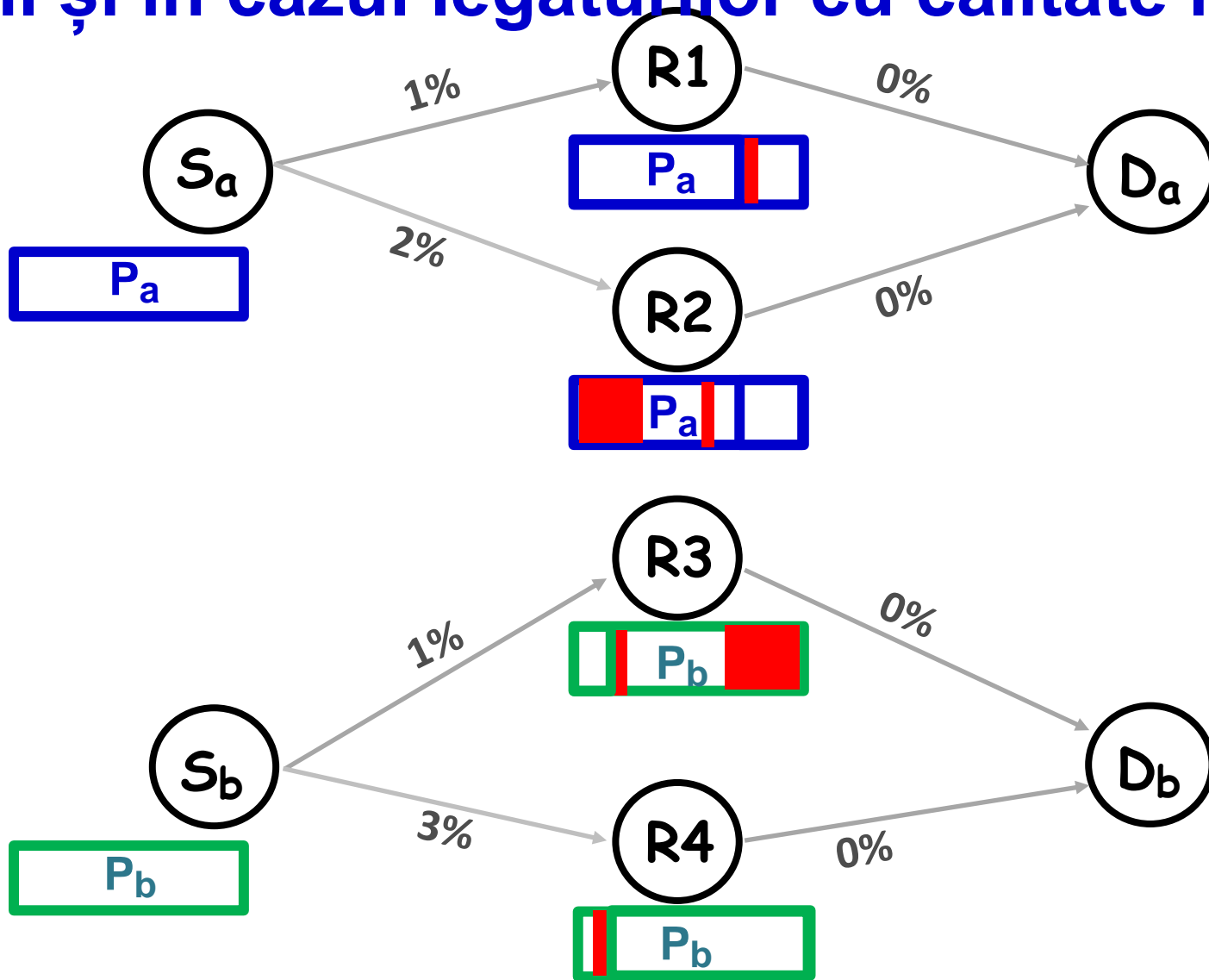
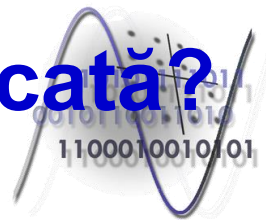
fiecare bit este recepționat corect de un nod al rețelei

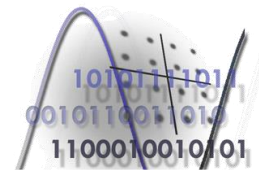
Explorând caracteristicile rețelei wireless → 3 tx → Throughput ridicat

# Util și în cazul legăturilor cu calitate ridicată?



# Util și în cazul legăturilor cu calitate ridicată?





# Soluția clasică

<b>PHY + LL</b>	<b>Păstrează pachetele corecte</b>
Network	Retransmite pachetele corecte către destinație

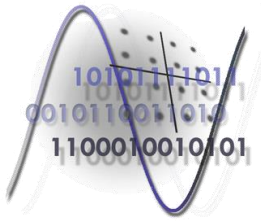
## Exploatănd noile paradigme...

<b>PHY + LL</b>	<b>Furnizează simbolurile corecte către nivelele superioare</b>
Network	Retransmite pachetele corecte către destinație

Throughput ridicat, competitivitate ridicată....

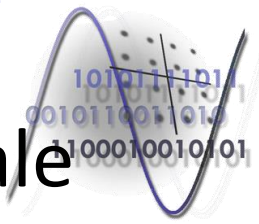


# MIXIT

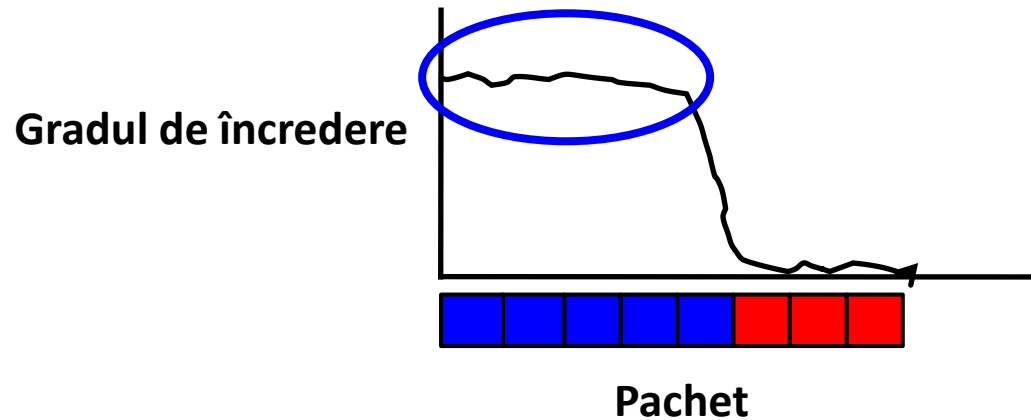


- Alt tip de colaborare dintre straturile rețelei pentru combaterea erorilor în rețele fără-fir
- Network coding la nivel de simbol care asigură transportul simbolurilor corecte către destinație
- MAC cu competitivitate ridicată
- În urma implementării si evaluării
  - Câștig 3-4x față de *shortest path routing*
  - Câștig 2-3x față de *rutare op. la nivel de pachet*

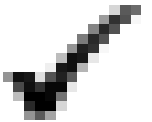
# Cum identifică router-ul simbolurile corecte?



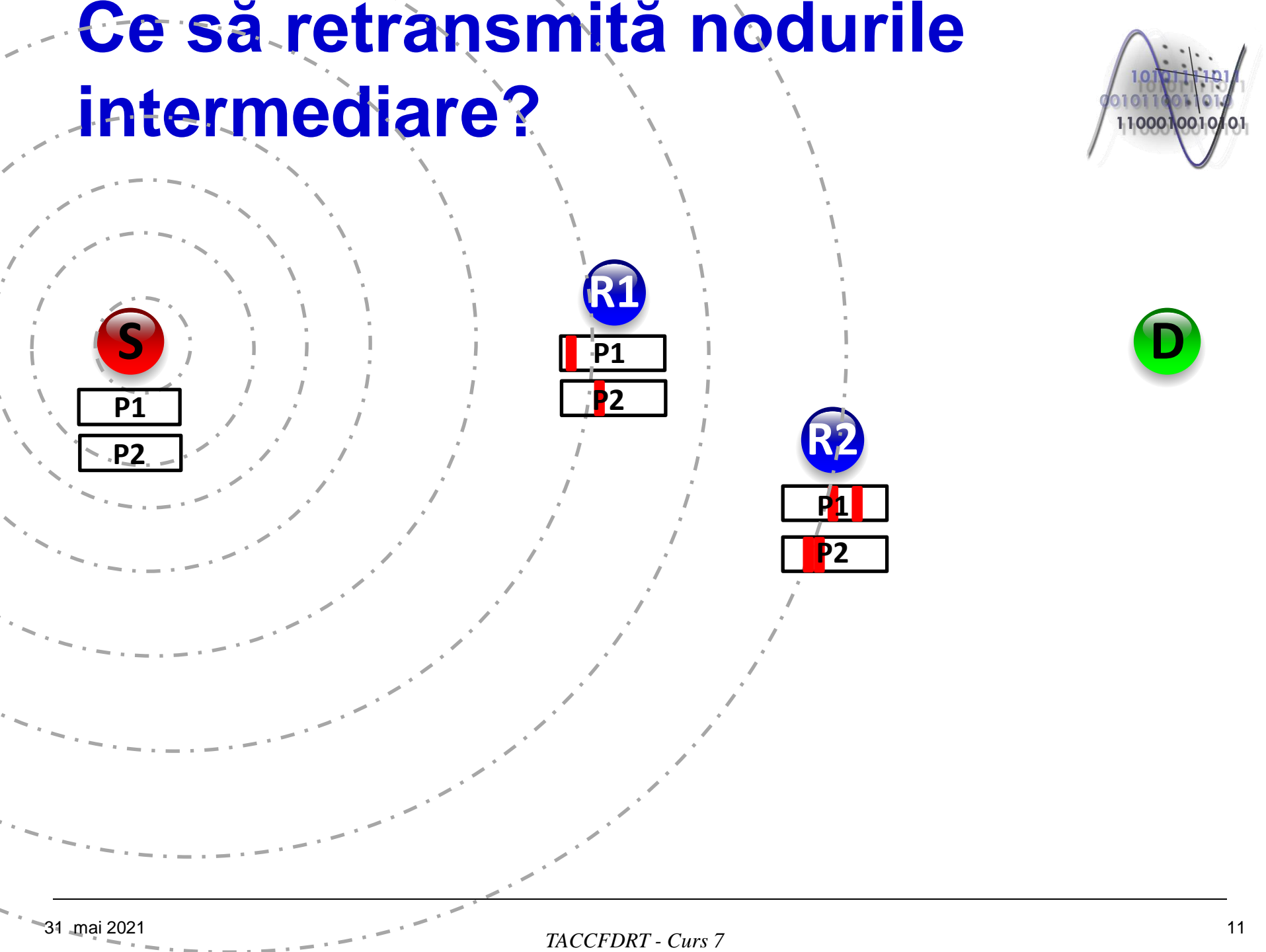
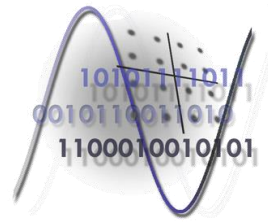
- PHY oricum estimează gradul de încredere ale simbolurilor recepționate
- PHY + LL furnizează numai simbolurile cu grad ridicat de încredere



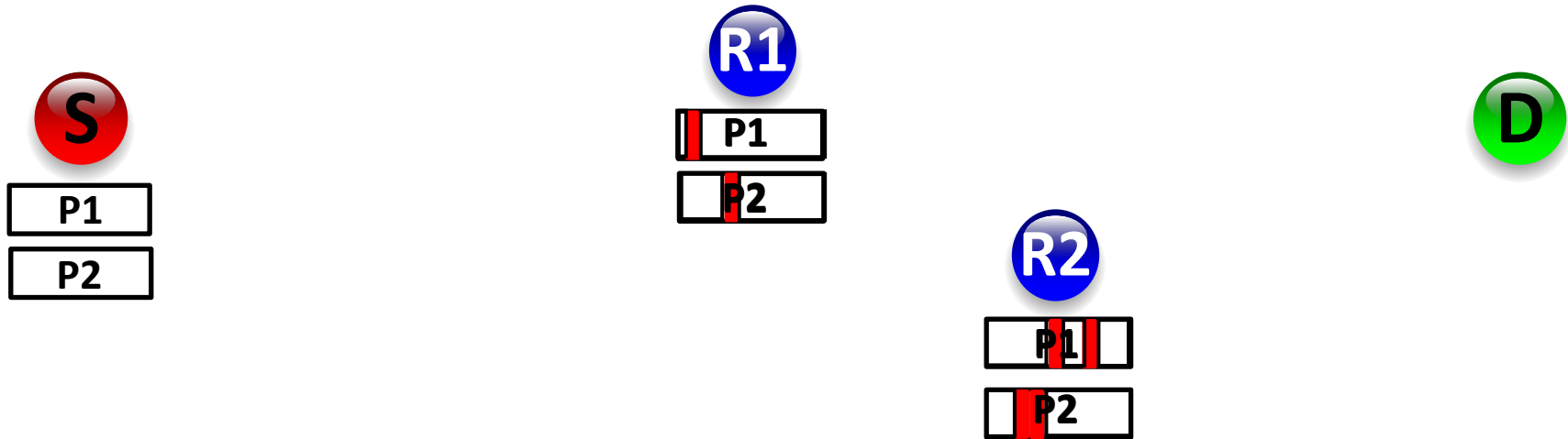
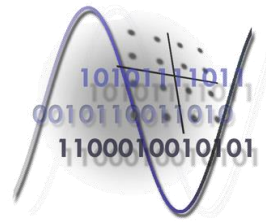
<b>PHY + LL</b>	<b>Furnizează simbolurile corecte către nivele superioare</b>
Network	Retransmite simbolurile corecte către destinație



# Ce să retransmită nodurile intermediare?



# Ce să retransmită nodurile intermediare?

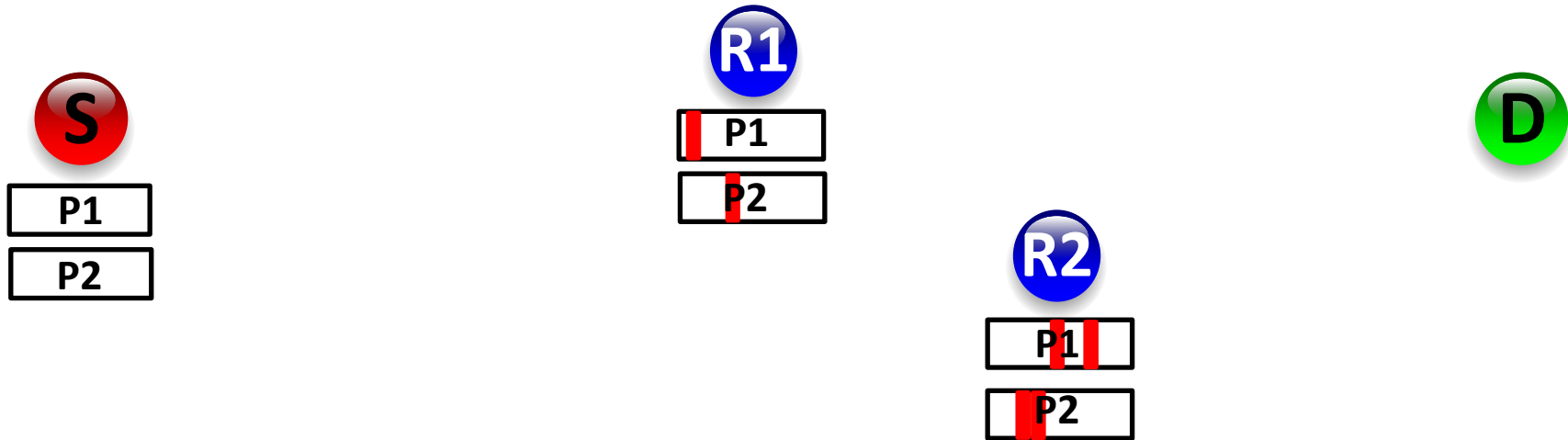
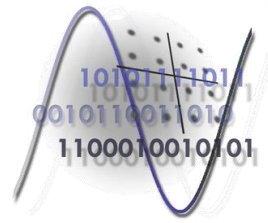


Suprapuneri între simboluri corecte

Soluții posibile

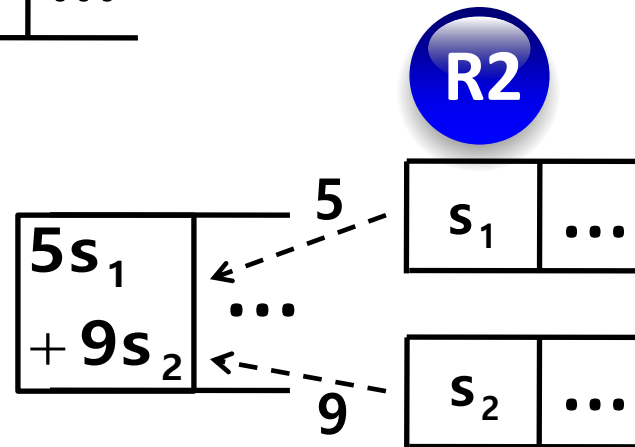
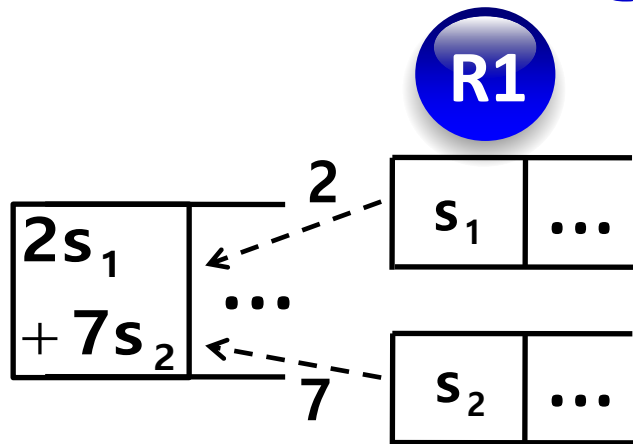
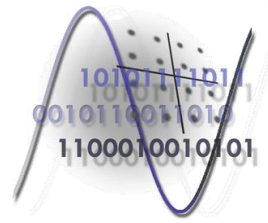
- 1) Să retransmită tot ce este corect → Ineficient
- 2) Retransmisie coordonată → Ne-scalabil

# MIXIT elimină duplicatele utilizând Network Coding la nivel de simbol



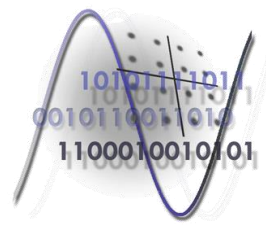
Retransmite combinații aleatoare ale simbolurilor corect recepționate

# MIXIT elimină duplicatele utilizând Network Coding la nivel de simbol



Routererele creează combinații aleatoare ale simbolurilor corect recepționate

# MIXIT elimină duplicatele utilizând Network Coding la nivel de simbol



R1

$2s_1$ $+ 7s_2$	...
--------------------	-----

D

R2

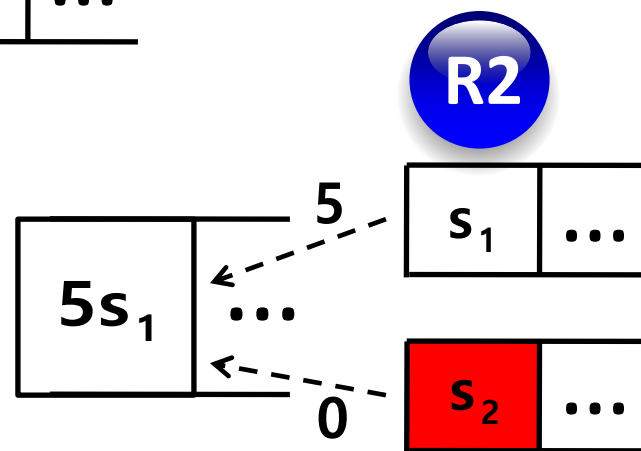
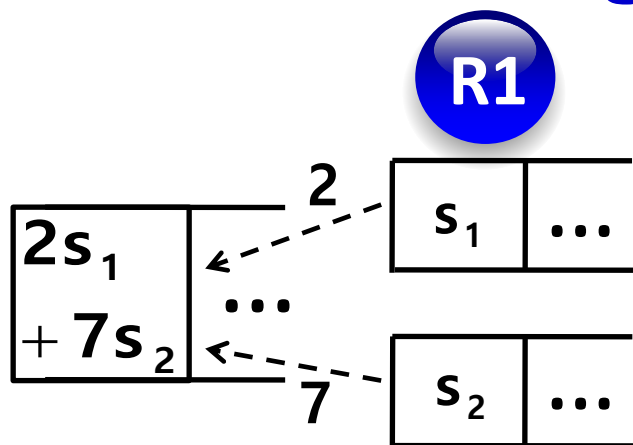
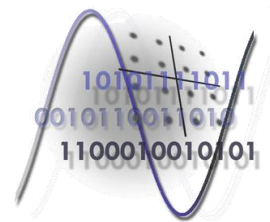
$5s_1$ $+ 9s_2$	...
--------------------	-----

rezolvă 2  
ecuații

↓  
 $s_1, s_2$

Aleatorizarea elimină transmisii multiple fără coordonare

# MIXIT elimină duplicatele utilizând Network Coding la nivel de simbol



Routererele creează combinații aleatoare ale simbolurilor corect recepționate



# MIXIT elimină pachete duplicate prin Symbol Level Network Coding



R1

$2s_1$ $+ 7s_2$	...
--------------------	-----

D

R2

$5s_1$	...
--------	-----

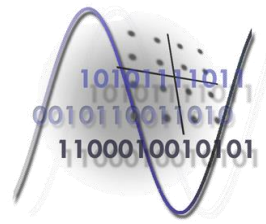
Rezolvă 2  
Ecuatii



Network Coding la nivel de simbol asigură:

- Eliminarea transmisiilor multiple → Eficient
- Nu necesită coordonare → Scalabil

# Destinația trebuie să cunoască ce combinație a recepționat



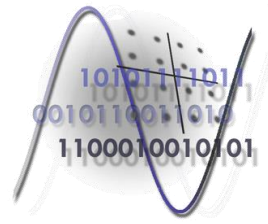
**$5s_1 + 9s_2$**  (daca ambele simboluri sunt corecte)

**$5s_1 + 0s_2$**  (daca numai  $s_1$  este corect)

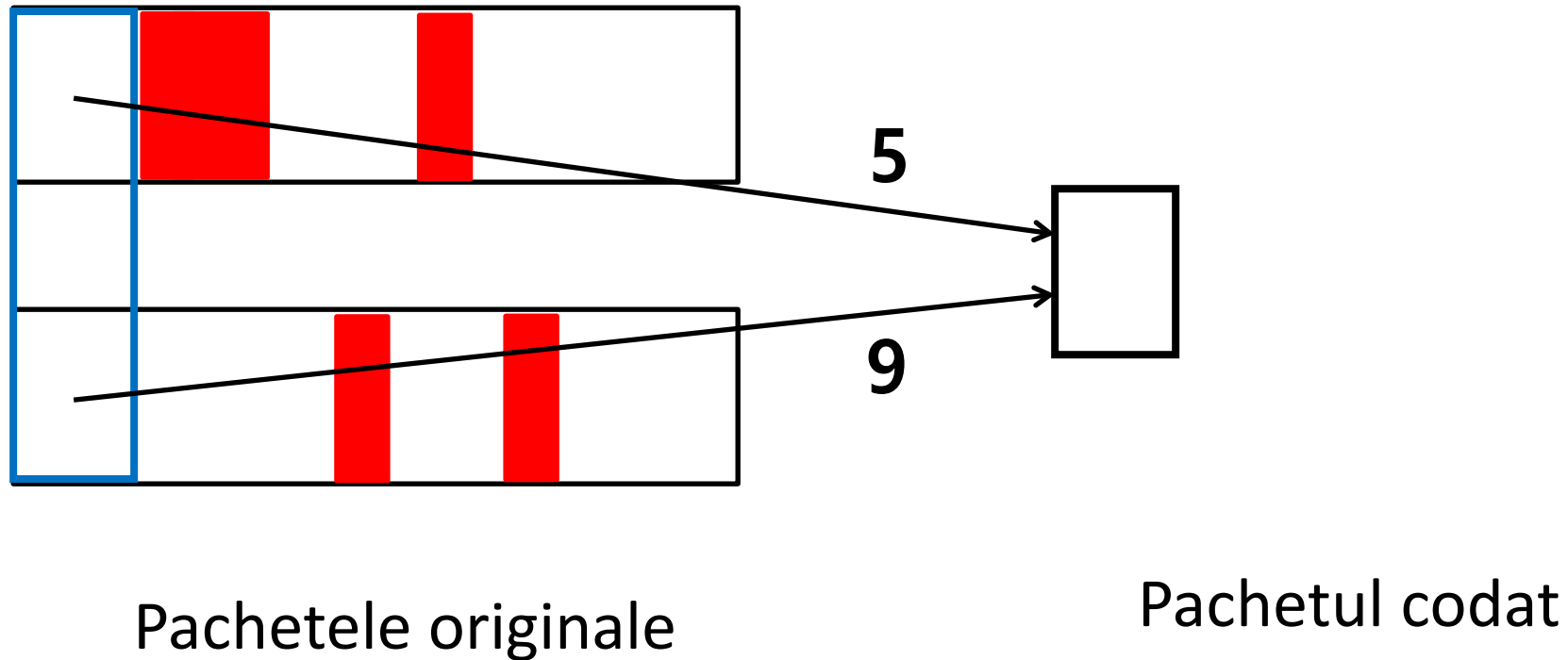
**$0s_1 + 9s_2$**  (daca numai  $s_2$  este corect)

**Nimic** (Dacă nici unul nu este corect)

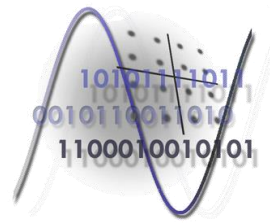
# Destinația trebuie să cunoască ce combinație a recepționat



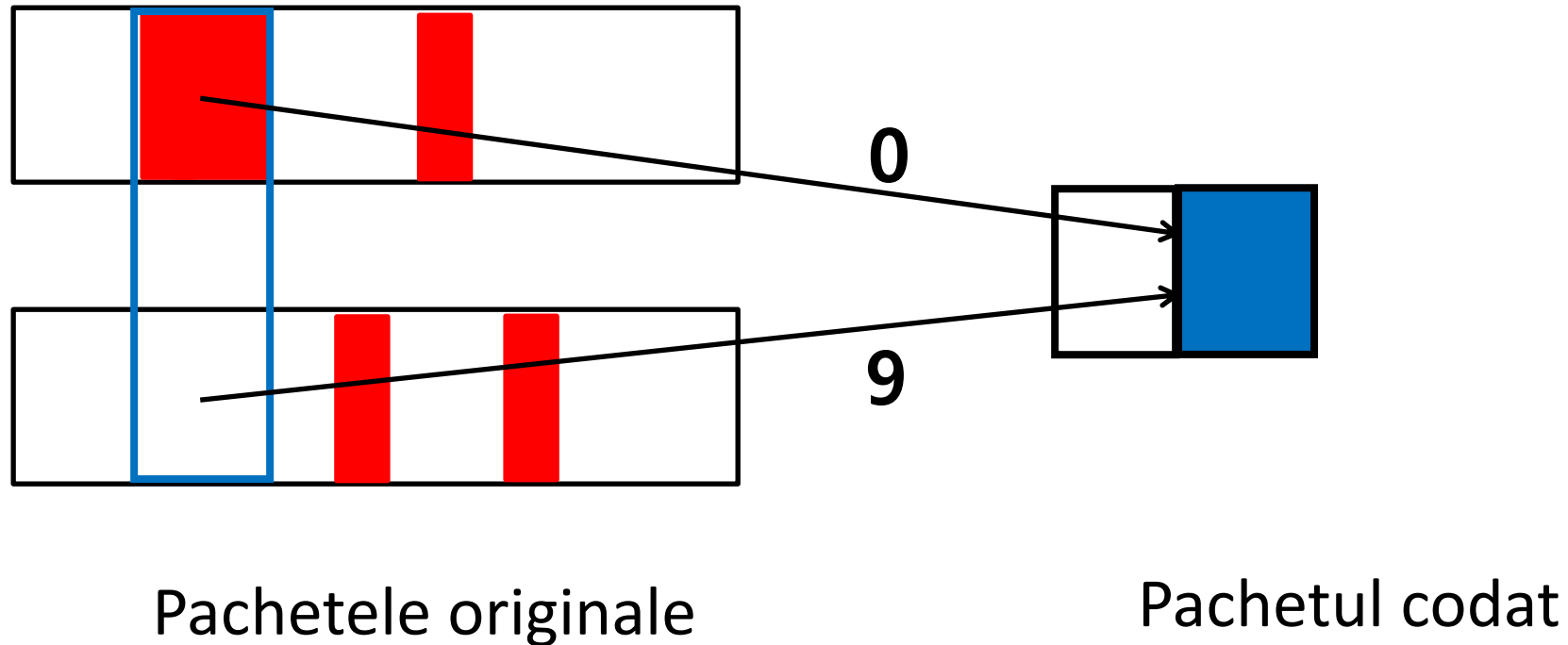
Utilizează *run length encoding*



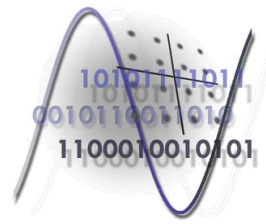
# Destinația trebuie să cunoască ce combinație a recepționat



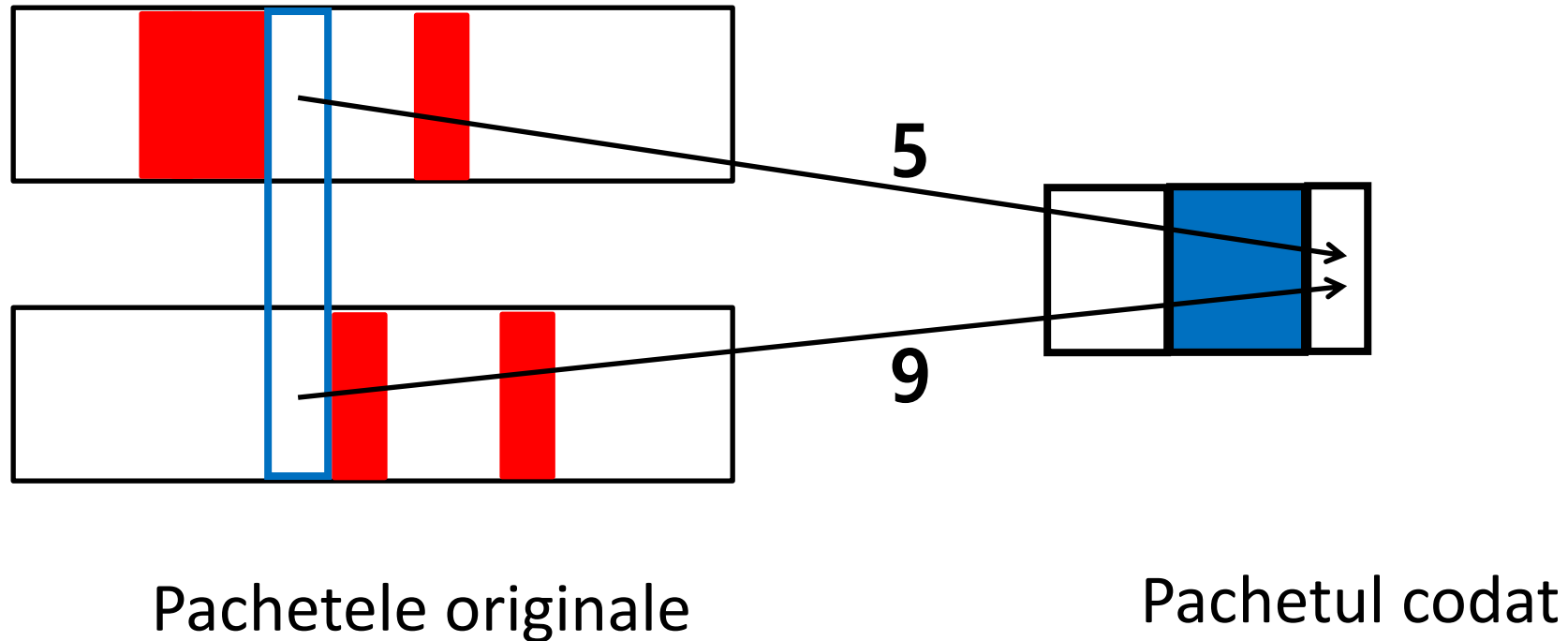
Utilizează *run length encoding*



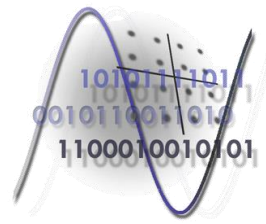
# Destinația trebuie să cunoască ce combinație a recepționat



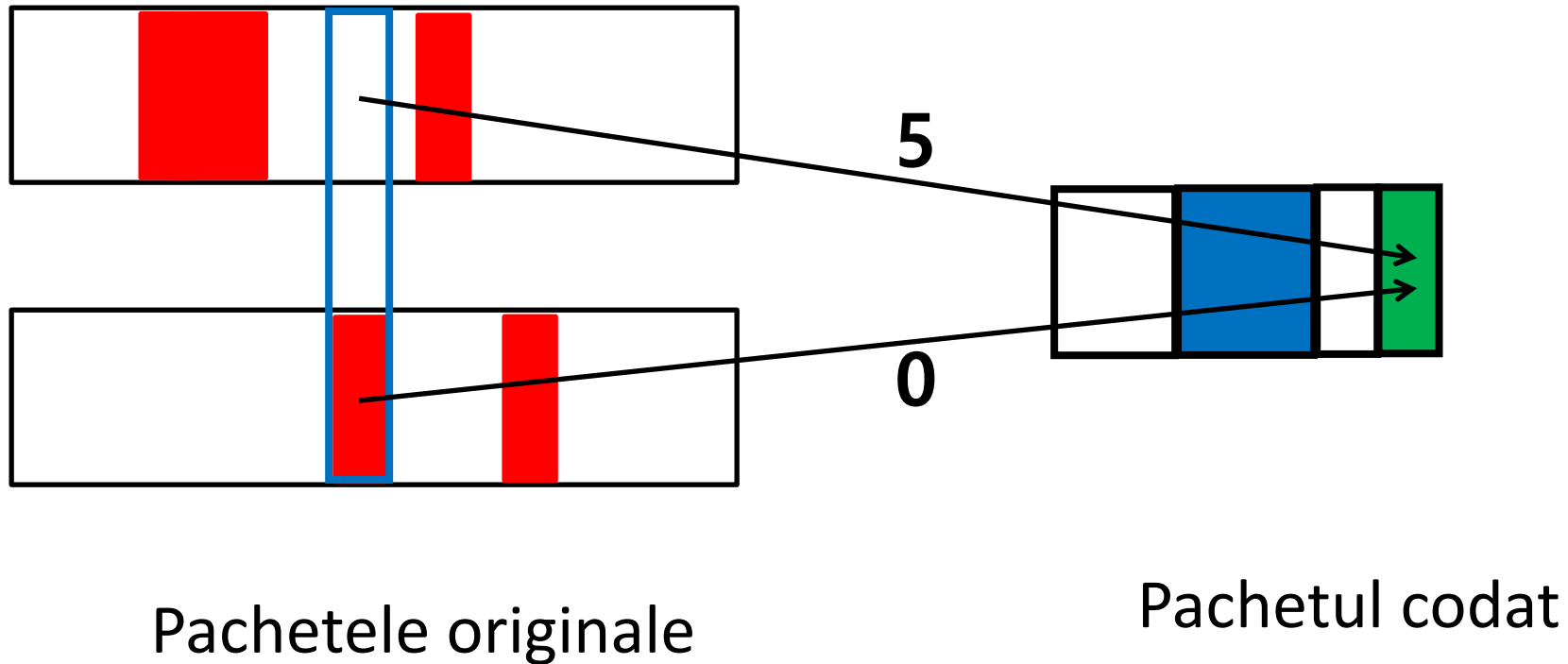
Utilizează *run length encoding*



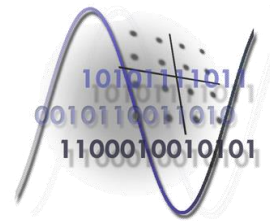
# Destinația trebuie să cunoască ce combinație a recepționat



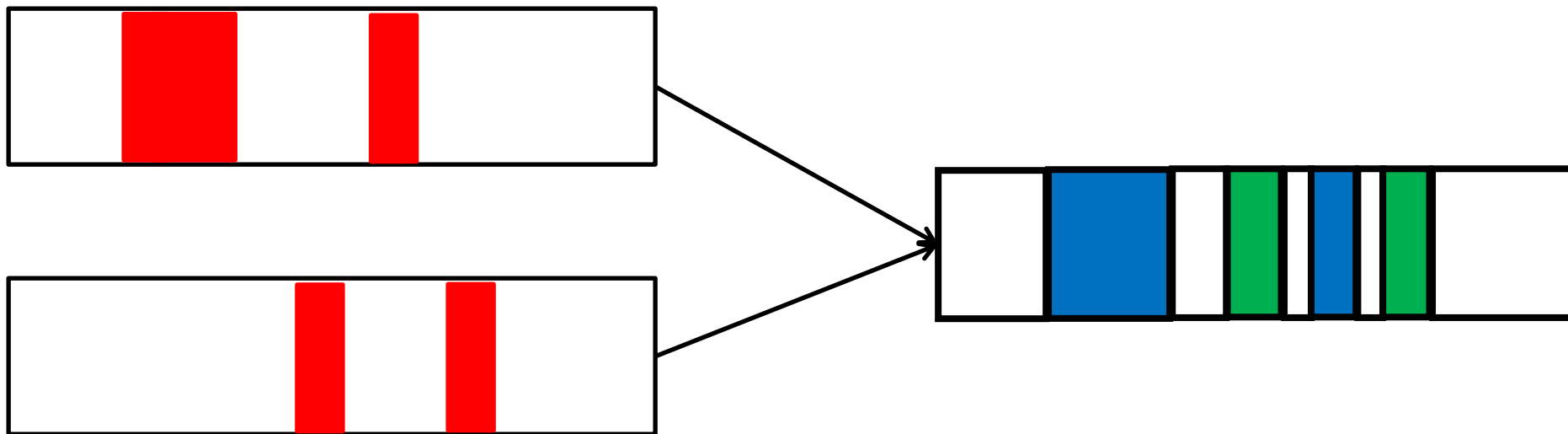
Utilizează *run length encoding*



# Destinația trebuie să cunoască ce combinație a recepționat



Utilizează *run length encoding*

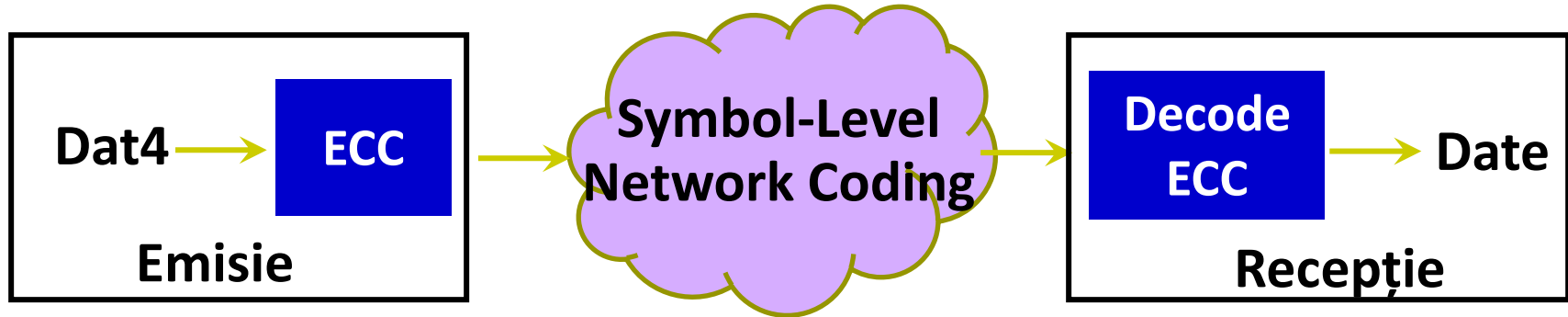


Run length encoding descrie eficient combinațiile  
posibile



**Routerele pot să retransmită biții eronați chiar și în cazul unui grad ridicat de încredere**

**MIXIT are capabilitate de corecție E2E!**



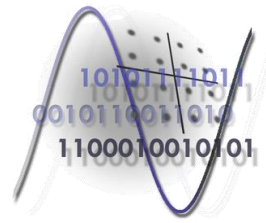
Capabilitate (ECC Error Correcting Code )

1. Routerele sunt transparente pentru ECC
2. Capabilitatea Optimă de corecție
3. Rateless

PHY + LL	Straturile superioare receptioneaza numai simbolurile corecte
Network	Retransmite numai simbolurile corecte

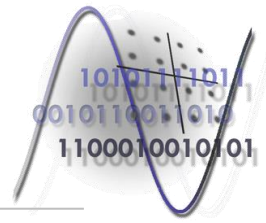




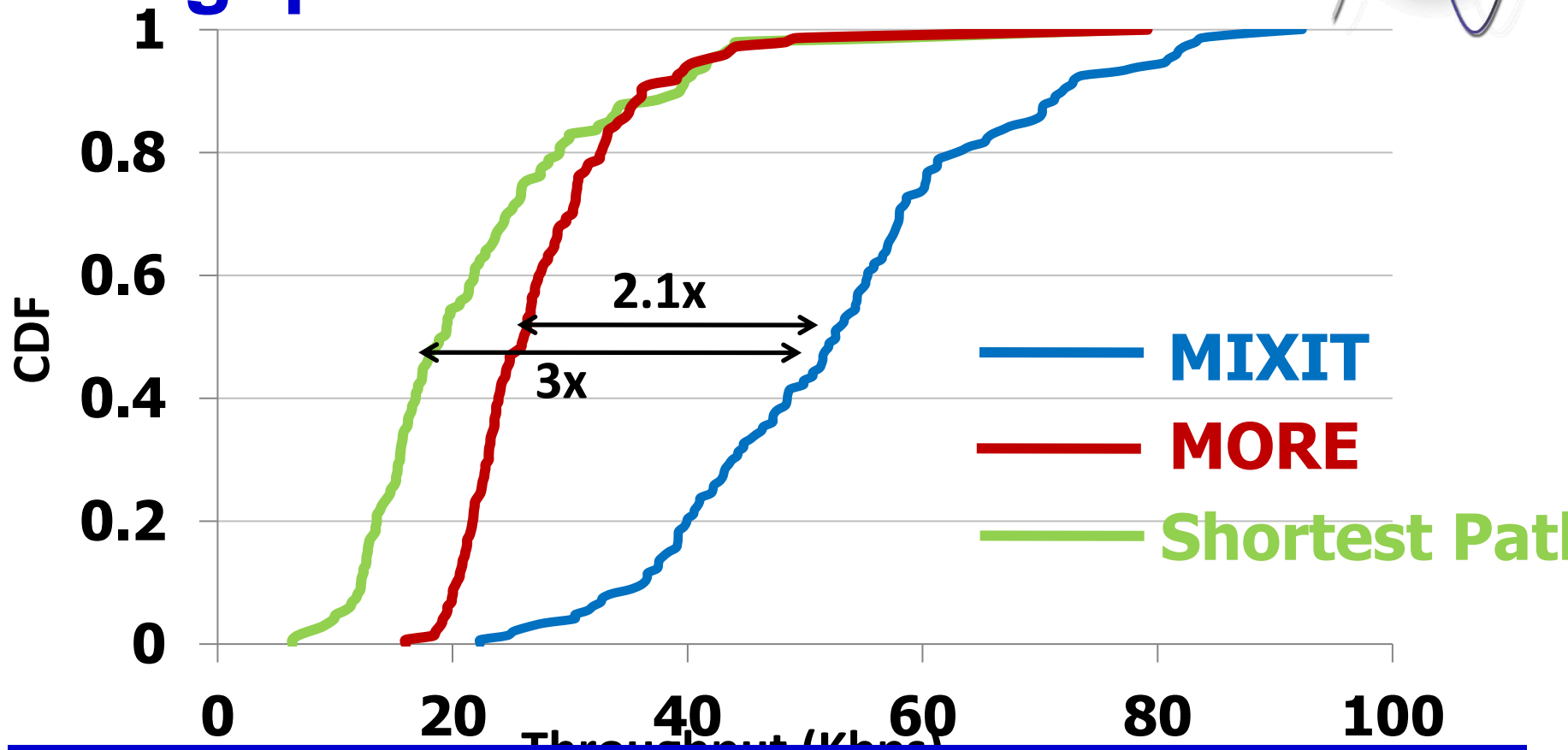


# Evaluarea

- Implementate în GNURadio pe SDR și USRP
- Zigbee (IEEE 802.15.4) link layer
- 25 noduri
- Scenarii de referință:
  1. Shortest path routing
  2. MORE: rutare oportunistă la nivel de pachete-MORE

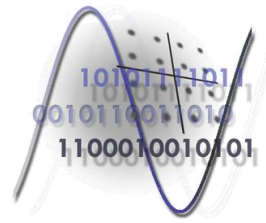


# Throughput



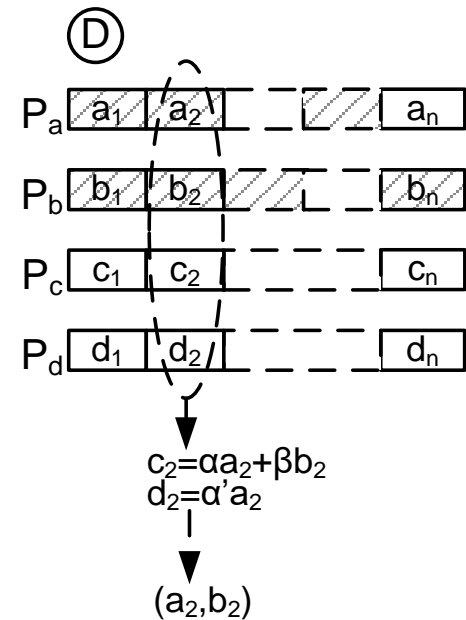
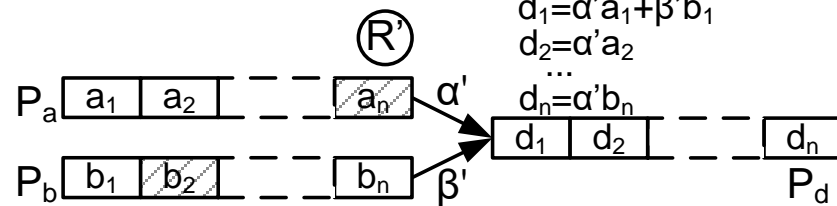
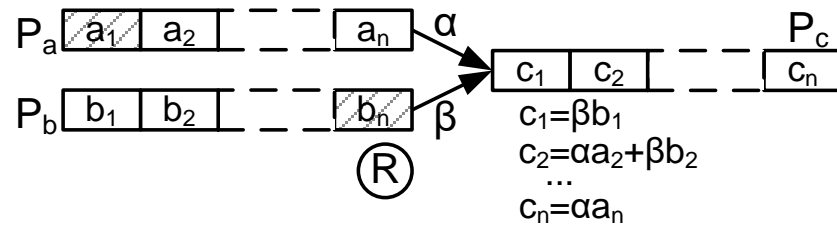
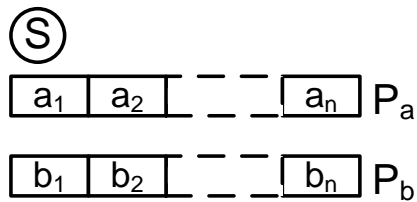
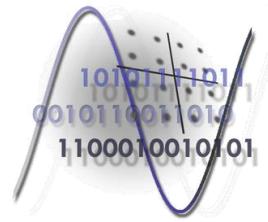
**Throughput: 3x - SPR, 2x - MORE**

# MIXIT

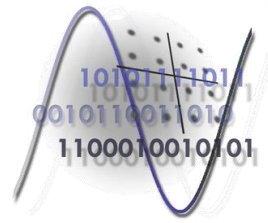


- Componenta de bază a arhitecturii MIXIT este un cod de tip NC la nivel de simbol care funcționează ca și un cod corector de erori de tip rateless.
- se încearcă simplificarea soluțiilor pentru două probleme majore:
  - Identificarea simbolurilor recepționate de către fiecare nod pentru a se preveni transmisii duplicate – poate fi o sarcină de coordonare foarte complexă a nodurilor.
  - Prin utilizarea codurilor NC la nivel de simbol, routerele transmit combinații liniare ale simbolurilor recepționate, reducându-se astfel probabilitatea de a trimite informații duplicate și se elimină necesitatea coordinării nodurilor
- Chiar dacă routerele trimit numai simboluri care au fost decodate cu grad ridicat de încredere există probabilitatea de a se trimite simboluri greșite. Codurile NC la nivel de simbol acționează și ca coduri corectoare de erori rateless, asigurându-se o redundanță adaptivă pentru corecția simbolurilor eronate

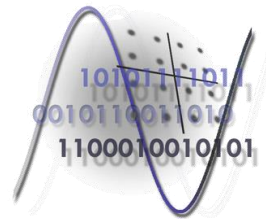
# MIXIT



# MIXIT

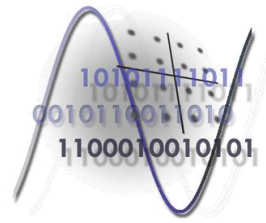


- *Componentele arhitecturii. Caracteristici*
  - **Sursa:** Nodul sursă transmite fișierele în grupuri de K pachete. Sursa creează combinații lineare aleatoare ale celor K pachete din grup și transmite pachetele codate
  - La fiecare pachet codat trebuie atașat un antet separat care arată, simbolurile care au fost combinate pentru formarea pachetului codat, respectiv cum s-a realizat procesul de codare
  - Antetul conține de asemenea și lista de relee/routere (numiți și forwarderi)



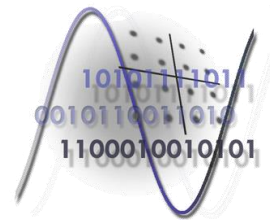
## ● Releele:

- Când un nod recepționează un pachet verifică dacă el este în lista de rele pentru pachetul respectiv (forwarders list). Dacă această condiție este îndeplinită nodul verifică dacă pachetul recepționat este inovativ, adică conține informație nouă
- relele creează combinații liniare ale simbolurilor corect recepționate, aceste combinații vor fi transmise mai departe. Astfel pachetele transmise vor conține numai simbolurile corecte
- În antetul noului pachet codat trebuie să fie incluși coeficienții de codare globali



- **Destinația:** Trebuie să recupereze simbolurile originale din simbolurile codate recepționate utilizând algoritmi de decodare standard.
  - Odată ce simbolurile originale sunt recuperate destinația le reasamblează în pachetele originale și trimite un semnal ACK către sursă pentru a permite generarea următorului grup de pachete.
  - Semnalele ACK sunt transmise utilizând o rutare clasică bazată pe selecția celei mai bune căi.
    - Se poate asocia o prioritate mărită pentru pachetele ACK și se pot proteja aceste pachete cu coduri FEC.

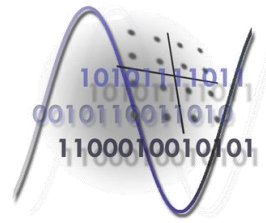
# În ce situații ajută MIXIT?



- Simulări/studii efectuate arată că se pot obține câștiguri de throughput între 1.2 și 8.
  - Din studii rezultă că se obțin câștiguri reduse de throughput în situația în care majoritatea legăturilor wireless sunt bimodale, adică fie sunt foarte bune, cu probabilitate redusă de eroare pe simbol, fie sunt foarte proaste, având o probabilitate ridicată de eroare pe simbol.
- Utilizând tehnica MIXIT se pot obține câștiguri de throughput de aproximativ de 4 ori relativ la alte metode de rutare oportunistă, dacă nu se consideră antetul mai mare al pachetelor MIXIT.
- În realitate acest câștig trebuie considerat o limită superioară.



# Cerințe cerute de aplicarea practică a NC în rețele wireless multihop



- *Capacitatea de a face față la un trafic variabil și un mediu dinamic*
- *Operații de Broadcast cu evitarea coliziunilor*
- *Algoritmi de codare și de decodare de complexitate redusă*
- *Se poate utiliza în mod transparent cu TCP*
  - recuperarea pachetelor pierdute
  - reordonarea pachetelor