

## Modelarea formală a protocoalelor

Lect.dr. Floarea NĂSTASE,  
Catedra de Informatică Economică, A.S.E., București

*Rețelele Petri constituie o tehnică de modelare formală eficientă, care și-a găsit o largă aplicabilitate în numeroase și diverse domenii. Pot fi modelate fenomenele de partajare a resurselor, protocoalele de transmisie, sincronizarea proceselor, gestiunea producției etc. Rețelele Petri au avantajul de a garanta un bun compromis între posibilitatea de reprezentare a sistemelor complexe și analiza lor. Modelarea prin rețelele Petri ține seama de aspectul structural și comportamental al rețelei de calculatoare.*

**Cuvinte cheie:** rețele Petri, modelare formală.

### Protocolul IEEE 802.3

Majoritatea rețelelor locale și metropolitane sunt în conformitate cu normele realizate de **IEEE** (Institute of Electrical and Electronics Engineers)

și **ANSI** (American National Standards Institute) (figural). Aceste norme, care au fost adoptate și de ISO, definesc, în principal, modul de utilizare a suportului de comunicație.

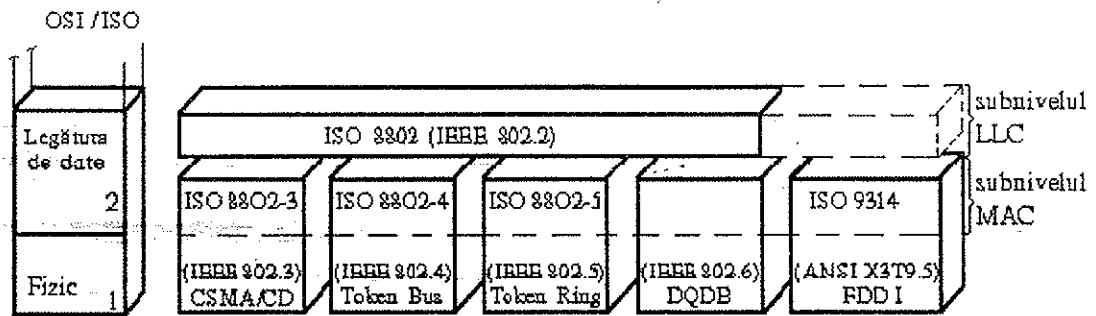


Fig.1. Norme relative la rețele locale și metropolitane

Prin aceste protocoale, nivelului 2 al modelului de referință OSI îi corespund două subnivele: subnivelul **MAC** (Medium Access Control), specific fiecărui tip de rețea locală și subnivelul **LLC** (Logical Link Control), comun tuturor tipurilor de suporturi fizice și metode de acces.

Protocolul **IEEE 802.3** se referă la rețelele locale, în care stațiile sunt conectate într-o topologie magistrală și utilizează metoda **CSMA/CD** (Carrier Sense Multiple Access/Collision Detection) ca protocol de acces. Prin această metoda se realizează :

- utilizarea algoritmului 1-persistent pentru accesul la canalul de comunicație ;

- detectarea conflictelor ;
- rezolvarea lor, pe baza adaptării timpului de reluare a controlului local. Algoritmul care stă la baza acestei norme, este constituit în principal, dintr-un algoritm de emisie și unul de recepție.

### Descrierea formală a protocolului

Protocolul poate fi descris formal cu ajutorul rețelelor Petri etichetate. Deoarece, ca orice protocol real, este relativ complex, modelul se poate realiza utilizând strategia de descompunere funcțională. O primă descompunere se face prin algoritmi de emisie și recepție. Descompunerea funcțională poate con-

tinua. Astfel, la rândul său, algoritmul de emisie realizează subfuncțiile:

- formatarea cadrului și inițializarea transmisiei (figura 2);

- transmisia cu/fără succes și achitarea (figura 3.);
- retransmisia (figura 4.).

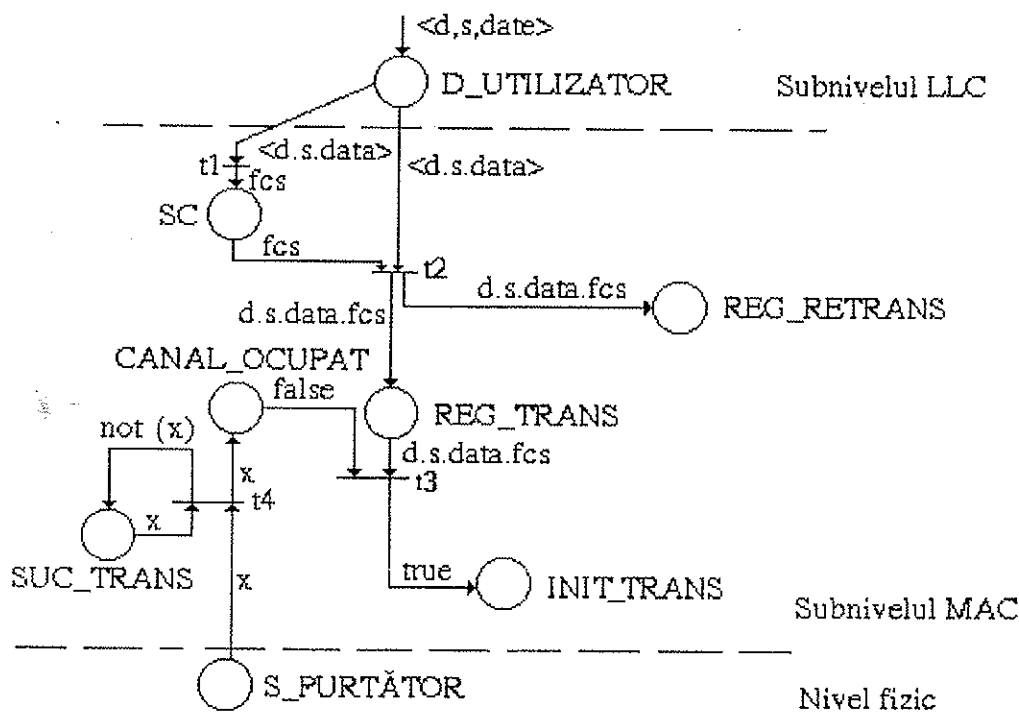


Fig.2. Realizarea cadrului și inițierea transmisiei

Modelul se referă la o stație emițătoare care asigură transmisia unui singur cadru la un moment dat. Utilizatorul poate cere transmisia unui nou cadru, după recepționarea unui mesaj de achitare a cadrului precedent.

Funcția de formatare este declanșată când locația D\_UTILIZATOR recepționează un mesaj de forma <d,s,date>. Acesta are rolul unei primitive, care cere subnivelului MAC transmisia datelor "date" de la o sursă "s" către o destinație "d". Locația D\_UTILIZATOR are rolul de interfață între MAC și LLC. Odată calculată secvența de control (SC) prin polinomul de generare, are loc asamblarea cadrului care va fi stocat atât într-un registru de transmisie REG\_TRANS cât și într-un registru de retransmisie REG\_RETRANS. Subnivelul MAC este impli-

cat în observarea canalului, pentru a se evita orice coliziune cu o transmisie în curs. Locația de semnalizare a purtătoarei S\_PURTĂTOR are rolul de interfață între nivelul fizic și subnivelul MAC. Când canalul devine liber (prezența unui token FALSE în locația S\_PURTĂTOR), subnivelul MAC lasă să treacă un interval de timp, corespunzător celui dintre cadre, apoi, consideră că transmisia are toate șansele de a se termina cu succes (stocheză un jeton FALSE în CANAL\_OCUPAT și jeton TRUE în locația SUC\_TRANS). Fiind îndeplinite condițiile de acces la canal, transmisia poate să înceapă.

În urma execuției tranzițiilor se realizează:

t1 : calcularea secvenței de control a erorilor pentru cadrul care se va transmite. Aceasta se poate scrie:

$$(D\_UTILIZATOR, \langle d, s, date \rangle) \xrightarrow{t1} (D\_UTILIZATOR, \langle d, s, date \rangle) \cap (SC, fcs)$$

t2 : asamblarea cadrului, ce constă din concatenarea biților corespunzători

adreselor, datelor și secvenței de control.

$$(D\_UTILIZATOR, \langle d, s, date \rangle) \cap (SC, fcs) \xrightarrow{t2} (REG\_TRANS, d.s.data.fcs) \cap (REG\_TRANS, d.s.data.fcs)$$

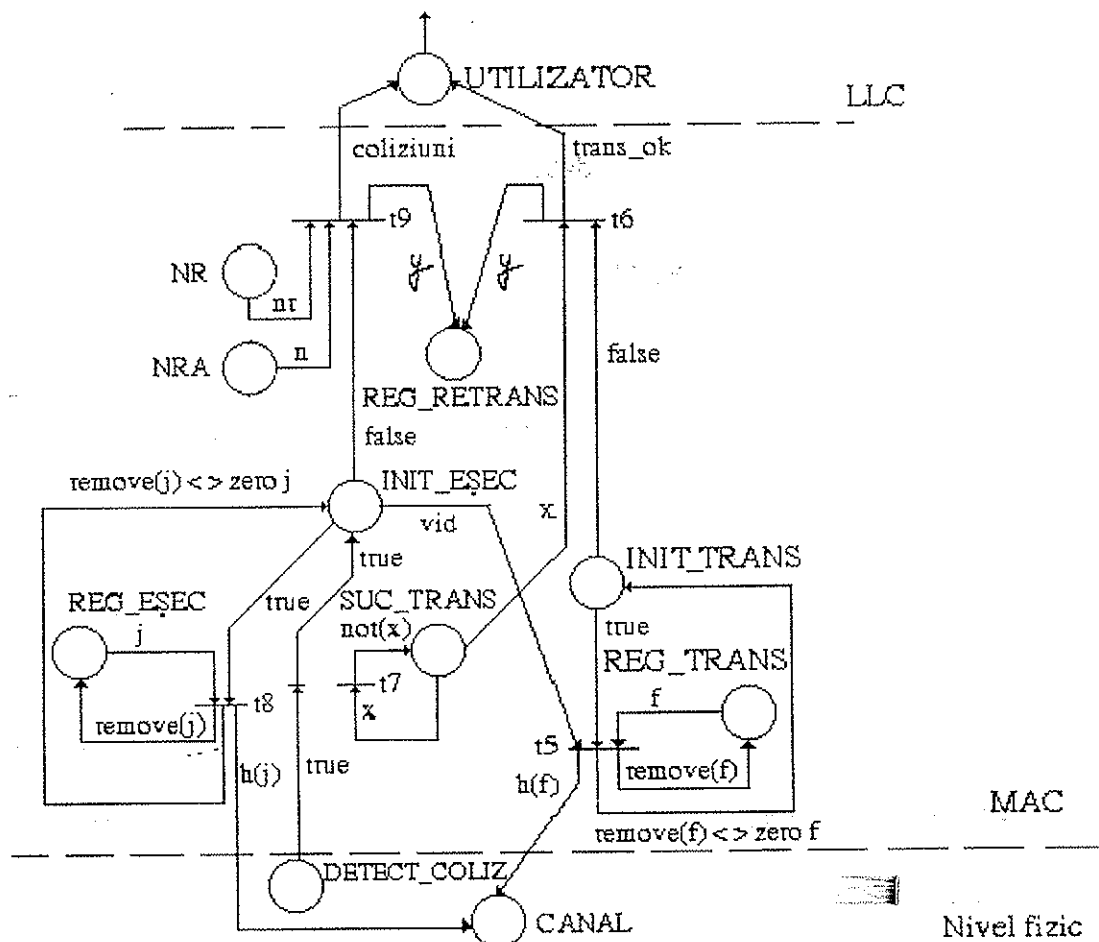


Fig.3. Transmisia cu /fără succes și achitarea

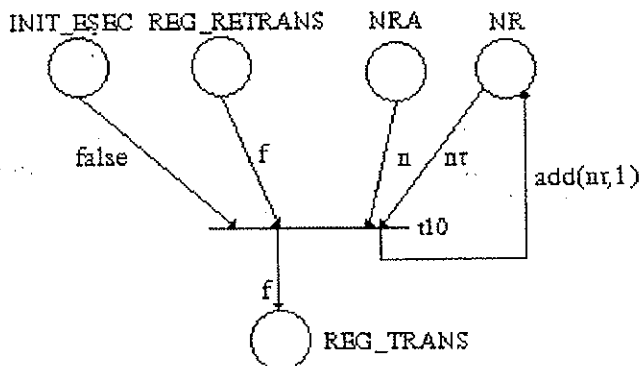


Fig.4. Retransmisia

t3 : accesul la canal, când sunt îndeplinite condițiile.

$$(CANAL\_OCUPAT, false) \cap (REG\_TRANS, d.s.data.fcs) \xrightarrow{t3} (INIT\_TRANS, true) \cap (REG\_TRANS, d.s.data.fcs)$$

*t4*: introduce o întârziere corespunzătoare duratei dintre cadre, în cazul

$$(S\_PURTĂTOR,x) \cap (SUC\_TRANS,x) \xrightarrow{t4} (CANAL\_OCUPAT,x) \cap (SUC\_TRANS,not(x)) \text{ IF } (x=false)$$

Formatarea cadrului se obține prin execuția secvenței de tranziții (*t1*, *t2*) iar inițializarea transmisiei prin secvența de tranziții (*t1*, *t2*, *t4*, *t3*).

Odată inițializată transmisia unui cadru, se va realiza emisia integrală, atâta timp cât nu este întreruptă din cauza detectării unei coliziuni. Într-o astfel de situație, pe canal se va transmite un cadru de eșec (în acest caz transmisia poate fi întreruptă prin poziționarea unui jeton în locația INIT\_EȘEC). Eticheta *h(f)*, indică faptul că un cadru *f* este emis în ordine normală. Locația CANAL are rolul de interfață între nivelul fizic și MAC. Eticheta *remove(f)* permite emisia serială a cifrelor cadrului, cifra care a fost emisă este eliminată. Când ultimul bit a fost emis cu succes, utilizatorul este informat prin poziționarea mesajului "*trans\_ok*" în locația UTILIZATOR. Copia cadrului transmis este apoi retrasă din registrul de retransmisie. Totodată în locația SUC\_TRANS este prezent un jeton "true" și un jeton "false" în INIT\_TRANS, ca rezultat al lui *remove(f) <> zero f* (*zero f* - cadru vid).

Când nivelul fizic detectează o coliziune de transmisie pe canal, subnivelul

oricărei retransmisii există un algoritm pentru evaluarea acestui timp.

MAC este informat prin prezența unui jeton "true" în DETECT\_COLIZ, ce are rolul de interfață. Aceasta determină poziționarea unui jeton "false" în SUC\_TRANS și inițiază transmisia unui cadru, pentru situația de eșec (poziționează "true" în INIT\_EȘEC). Cadrul de eșec *j* este apoi transmis (se poziționează *h(j)* în locația CANAL și *remove(j)* în REG\_EȘEC). După transmiterea ultimului bit al cadrului de eșec se poziționează jeton "false" în INIT\_EȘEC, utilizatorul fiind informat că transmisia nu a putut avea loc ca urmare a unui exces de coliziuni pe canalul de transmisie și poziționând un jeton "eroare coliziune" în locația UTILIZATOR. Aceasta se întâmplă după ce numărul de retransmisii (NRA) "n" au fost epuizate. Locația NR controlează numărul de retransmisii.

Funcția de retransmisie este activată, numai dacă un cadru de eșec a fost transmis și numărul retransmisiilor autorizate nu a fost epuizat. Retransmisia constă în transferul conținutului registrului de retransmisie în registrul de transmisie. Tranzițiile sunt :

*t5* : transmisia cadrului

$$(INIT\_TRANS,true) \cap (REG\_TRANS,f) \xrightarrow{t5} (INIT\_TRANS,remove(f) <> zero f) \cap (CANAL,h(f)) \cap (REG\_TRANS,remove(f)) \text{ IF } M(INIT\_EȘEC)=\emptyset$$

*t6* : achitarea în situația unei transmisii cu succes :

$$(INIT\_TRANS,false) \cap (SUC\_TRANS,x) \xrightarrow{t6} (REG\_RETRANS,y) \cap (UTILIZATOR,trans\_ok) \text{ IF } (x=true)$$

*t7* : testarea apariției coliziunii :

$$(DETECT\_COLIZ,true) \cap (SUC\_TRANS,x) \xrightarrow{t7} (INIT\_EȘEC,true) \cap (SUC\_TRANS,not(x)) \text{ IF } (x=true)$$

*t8* : transmisia cadrului în situația de eșec :

$$\begin{aligned} & (\text{INIT\_EȘEC, true}) \cap (\text{REG\_EȘEC, j}) \xrightarrow{t8} (\text{CANAL, h(j)}) \cap \\ & (\text{REG\_EȘEC, remove(j)}) \cap (\text{INIT\_EȘEC, remove(j)} \neq \text{zero } j) \end{aligned}$$

$t9$  : achitarea în situația unei transmisii nereușite.

$$\begin{aligned} & (\text{INIT\_EȘEC, false}) \cap (\text{NRA, n}) \cap (\text{NR, nr}) \xrightarrow{t9} (\text{UTILIZATOR, coliziuni}) \\ & \cap (\text{REG\_RETRANS, y}) \text{ IF } (\text{nr} > \text{n}) \end{aligned}$$

$t10$  : retransmisia unui cadru.

$$\begin{aligned} & (\text{INIT\_EȘEC, false}) \cap (\text{REG\_RETRANS, f}) \cap (\text{NRA, n}) \cap (\text{NR, nr}) \xrightarrow{t10} \\ & \rightarrow (\text{REG\_TRANS, f}) \cap (\text{NR, add(nr, 1)}) \text{ IF } (\text{nr} \leq \text{n}) \end{aligned}$$

În mod asemănător poate fi descrisă funcția de recepție.

## Concluzii

Alegerea tehnicii rețelelor Petri are o serie de avantaje:

- se poate ține seama, în faza de evaluare, atât de **comportamentul dinamic** al sistemului, cât și de resursele sale;
- permite evaluarea, nu numai a configurației nominale disponibilă, dar și **configurației în situație de pană**;
- oferă o **analiză formalizată**, favorizând descrierea completă a comportamentului sistemului;
- permite o **evoluție** a structurii sistemului, fie prin extinderea acestuia, fie în scopul îmbunătățirii performanțelor tehnice sau de utilizare.

## Bibliografie

- ANGOSTO, P. - *L'ingénierie des protocoles*, Inter Editions, Paris, 1993
- BOUNEMRA, K - *Normalisation des réseaux: la couche application et la gestion des réseaux*, Eyrolles, Paris, 1990
- BRAMS, G.W. - *Réseaux de Petri: Théorie et Analyse*, Masson, Paris, 1983
- DIAZ, M., ș.a. - *Un modèle formel pour la spécification de la synchronisation multimedia en environnement distribué*, Actes CFIP'93, Montréal
- FDIDA, S., PUJOLLE, G - *Modeles de Systemes et de Réseaux*, Eyrolles, Paris, 1989