ASUPRA DEBLOCĂRILOR MESAJELOR EXISTENTE ÎN REȚEA

mat. Marius Mincă

Institutul de Cercetări în Informatică

Rezumat: În cele ce urmează, vor fi definite tipurile de mişcări (mutări) ce pot exista într-o rețea, precum şi definiția generală de controlor (supraveghetor) cu ajutorul căruia se pot evita blocările în rețea. Vor fi prezentate două metode de rezolvare: una *structurată* (în care se determină buffer-ul potrivit pentru primirea unui pachet ce a fost generat sau transmis) şi alta *nestructurată* (în care toate bufferele sunt egale şi deplasarea unui mesaj în rețea depinde de numărul de salturi realizate şi de informațiile din fiecare nod). Pe parcursul acestei lucrări vor fi date exemple de controlori, specifici metodei utilizate, ce pot evita blocările din rețea. În final, sunt enumerate unele rezultate privind modificări în abordarea deblocării (prin schimbarea topologiei), tipuri practice de blocări în rețea precum şi o altă situație *(livelock)* ce poate apărea în rețea.

Cuvinte cheie: buffer, mesaj (pachet), configurație, graf, graf orientat.

1. Introducere.

Articolul de față este realizat pe baza unor rezultate apărute în cartea "Introduction to Distributed Algorithms" de GERARD TEL de la Universitatea din Cambridge, 1994.

Mesajele (pachetele) care parcurg o rețea de comunicație trebuie stocate în fiecare nod, înainte ca ele să treacă la nodurile următoare, aflate pe drumul către destinațiile lor.

Cum fiecare nod rezervă un număr finit de buffere pentru acest scop, apar foarte des situa-țiile de blocare (store-and-forward deadlocks), în care un grup de pachete nu va ajunge niciodată la destinație, deoarece toate pachetele din el sunt în aşteptare pentru ocuparea unui buffer în care se găseşte un alt pachet din grup.

Vor fi prezentate metode de evitare a acestor situații prin punerea în evidență a unor controlori ce coordonează activitatea într-o rețea.

În mod obişnuit, o rețea este modelată printr-un graf G=(V,E), iar distanța între noduri este măsurată prin numărul de muchii (salturi). Se notează cu B mulțimea de buffere aflate nodurile din rețea.

Pachetele (mesajele) transmise, sunt descrise prin următoarele tipuri de mutări:

(1) Generarea: Un nod u "creează" un nou pachet (acceptă pachetul de la un nivel mai înalt de protocol) şi îl plasează în buffer-ul gol al lui u. În acest caz, u se numeşte sursa lui p;

(2) Înaintarea: Un pachet p este transmis de la un nod u la un buffer gol al urmatorului nod v pe drumul său (drumul este determinat de algoritmul folosit). Un rezultat al acestei mutări reprezintă eliberarea buffer-ului din nodul de unde a plecat pachetul (mesajul);

(3) Prelucrarea: Când un pachet (mesaj) ajunge la destinație este scos din buffer. Se presupune că rețeaua permite întotdeauna o astfel de mişcare.

Se notează cu P mulțimea tuturor drumurilor pe care mesajele le pot parcurge; aceasta se poate determina prin rularea algoritmului, iar cu k lungimea drumului cel mai mare din P.

Observație: Nu întotdeuna este valabilă relația k=diam(G).

Se numeşte controlor, un algoritm care permite sau interzice diferite mişcări în rețea, supunându-se următoarelor cerințe:

(1). prelucrarea (consumarea) pachetelor (la des-tinație) este întotdeuna permisă;

(2). generarea unui pachet într-un nod în care toate buffer-ele sunt goale este întotdeauna permisă;

(3). un controlor utilizează doar informații locale, i.e. decizia dacă un pachet poate fi acceptat în nodul u depinde de informația cunoscută de u sau conținuta în pachet.

Observație: Condiția (2) elimină situațiile triviale de evitare a blocărilor (prin negenerarea de pachete).

Notații:

Zu= mulțimea de stări ale nodului u;

M = mulțimea de mesaje posibile.

Definiție:

Un controlor pentru o rețea G=(V,E) este o colecție de perechi **con**={GENu,FORu} , u în V

unde GENu şi FORu sunt sumulțimi ale produsului Zu*M.

Dacă c este o stare a lui u în care toate buffer-ele sunt goale, atunci pentru orice mesaj p avem (c,p) în GENu.

Un controlor **con** permite generarea unui pachet p în nodul u a cărui stare este c *dacă şi numai dacă* (c,p) în GENu şi permite înaintarea pachetului p de la u la v *dacă şi numai dacă* (c,p) în FORu.

Mişcările într-o rețea sub contolorul **con** sunt doar acele mişcări permise de acesta.

Un pachet se numeşte blocat, dacă nu poate ajunge la destinația sa prin nici o secvență de mişcări în rețea.

Definiție:

Dându-se o rețea G, un controlor con pentru G și o configurație l a lui G, un pachet p (existent în configurația l) este blocat, dacă nu există nici o secvență sub con aplicata în l în care p este prelucrat (consumat).

O configurație este numită blocată, dacă aceasta conține pachete blocate.

Configurația inițială a unei rețele este aceea în care nu există nici un pachet.

Definiție:

Un controlor se numește deadlock-free, dacă nu apare nici o configurație blocată sub acesta pornindu-se de la o configurație inițială.

În continuare, vor fi prezentate metode (soluții) de rezolvare a situațiilor de blocare a mesajelor într-o rețea.

2. Soluții structurate.

Merlin și Schweitzer au introdus noțiunea de buffer graf.

S-a pornit de la observația că, o blocare este datorată unei situații de așteptare în ciclu.

Se consideră o rețea G și *B* mulțimea de buffere.

Definiție:

Un buffer graf (pentru G si *B*) este un graf orientat BG de buffere din rețea, i.e. BG=(*B*,BE) astfel încât:

1) BG aciclic;

2) bc în BE, atunci b și c buffere în același nod sau sunt buffere în noduri diferite, dar conectate în G;

3) pentru orice P în *P* drum, există un drum în BG cu imaginea P.

Observație:

Dacă b0,b1,..,bs drum în BG, ui nodul în care se afla buffer-ul bi, atunci u0,u1,...,us șir de noduri a.i. {ui,ui+1} n E sau ui=ui+1. Drumul obținut în G prin omiterea de duplicate se numește imagine a dumului de buffere din BG.

Un pachet nu poate fi plasat într-un buffer arbitrar, ci unul de la care poate ajunge la destinația sa , deci un buffer potrivit, dat de următoarea definiție:

Definiție:

Fie p un pachet în nodul u cu destinația v. Un buffer b din u este potrivit pentru p, dacă există un drum în BG de la b la un buffer c din v, a cărui imagine este un drum pe care p îl poate urma în G.

Un astfel de drum va fi numit drum garantat.

Se va nota cu nb(p,b) următorul buffer pe acest drum și pentru fiecare pachet nou generat p în u există un buffer potrivit fp(b) în u.

Observații:

- nb(p,b) este întotdeauna un buffer potrivit pentru p;
- se va considera, deocamdată că nb(p,b) este situat într-un nod diferit de cel în care se găsește b;
- se pot utiliza muchii interne în BG, adică muchii intre doua buffere din acelasi nod.

Se va defini, în continuare, un controlor notat cu **bgc(BG)** și care îndeplinește funcțiile:

1. generarea unui pachet în u este permisă, *dacă și numai dacă* bufferul fp(b) este liber. Dacă pachetul este generat, atunci este plasat în acest buffer.
2. înaintarea unui pachet p de la u la w este permisă, *dacă și numai dacă* nb(p,b) (în w) este liber, iar în acest caz p va fi plasat în el.

Teorema: bgc(BG) este un controlor deadlock-free.

Demonstrație:

Dacă nodul u are toate buffer-ele goale, atunci generarea este permisă, ceea ce implică faptul că **bgc(BG)** este un controlor.

Fie b din *B* și se notează cu *b* clasa sa reprezentând lungimea celui mai lung drum din BG care se termina în b.

Observație: dacă b0,b1,...,bs drum în BG, atunci $b0<b1<...<bs$. Deci $nb(p,b)>b$.

Așa cum un controlor permite plasarea pachetelor doar în buffere potrivite așa și **bgc(BG)** permite plasarea pachetelor numai în buffere potrivite.

Se va arăta prin inducție pe clasa de buffere că nici un buffer din clasa r nu conține un pachet blocat. Fie R cea mai mare clasa buffer.

Dacă r=R: se consideră b buffer in nodul u.Un pachet care a ajuns în b din u, poate fi prelucrat (controlorul permite întotdeauna consumarea). Deci, nici un buffer din clasa R nu conține pachete blocate.

Dacă r<R: se presupune prin inducție că, pentru orice r', r<r'<=R, în r' nici un buffer nu conține pachete blocate.

Fie l o configurație acceptabilă cu pachetul p in bufferul b din clasa r<R în nodul u.

Dacă u este destinația lui p, atunci el poate fi consumat, deci nu este blocat. Altfel, notăm cu c=nb(p,b).

Observație: r'=*c*>r=*b*. Din ipoteza de inducție c nu conține pachete blocate.

Deci, există o configurație care permite mutarea lui p în c, ceea ce arată că p nu este un pachet blocat.

Observație: dacă drumul garantat conține muchii interne ale buffer-ului grafului, atunci controlorul trebuie să permită mișcări adiționale, prin deplasarea pachetului într-un buffer din același nod.

De obicei, un drum garantat nu permite astfel de muchii. Sunt folosite doar pentru optimizarea mișcărilor pentru creșterea eficienței la înaintare.

Exemplu: dacă pachetul p se află în buffer-ul b din nodul u, nb(p,b) în nodul w este ocupat și există b2 buffer în u potrivit pentru p, pentru care nb(p,b2) este liber, atunci p poate ajunge la nb(p,b2) via b2.

Două exemple de buffer graf:

1). Schema de destinație

Utilizează N buffere pentru fiecare nod u, iar bu[v] bufferul pentru fiecare posibilă destinație v care se va numi ținta bufferului bu[v].

Trebuie presupus că, prin rularea algoritmului toate pachetele cu destinația v înainteaza printr-un arbore orientat Tv având muchiile orientate spre v.

Buffer-ul graf: BGd=(B,BE).

bu[v1]bw[v2] in BE *dacă și numai dacă* (v1=v2) and (uw muchie in Tv1).Evident Bgd este aciclic.

Fiecare drum P din *P* cu capătul final în v este un drum în Tv și, prin construcție, există un drum în Bgd de buffere cu ținta v a cărui imagine este P. Acest drum este ales ca fiind drum garantat (pentru fiecare pachet p cu destinația v, generat în nodul u, fb(p)=bu[v] si dacă acest pachet trebuie să înainteze la w, atunci nb(p,b)=bw[v]).

Se definesc **dest=bgc(BGd)** cu fb(p) și nb(p,b) declarate mai sus.

Teorema:

Există un controlor deadlock-free pentru o rețea conectata arbitrar, care utilizează N buffere în fiecare nod și permite pachetelor să urmeze un itinerariu prin arbore.

Demonstrație: **dest** este un deadlock-free folosind același număr de buffere.

Avantajul folosirii controlorului **dest** este acela de a fi simplu de utilizat, iar dezavantajul - că folosește un număr mare de buffere în fiecare nod.

2). Schema funcție de numărul de salturi realizate

Fiecare nod u conține k+1 buffere bu[0],...,bu[k]. Se presupune că fiecare pachet conține o variabilă indicând numărul de salturi făcute de pachet de la sursa sa.

Bufferul graf: BGh=(B,BE).

bu[i]bw[j] în BE *dacă și numai dacă* (i+1=j și uw in E).

Observație: BGh este aciclic, iar indexul buffer-elor crește strict cu fiecare muchie din Bgh.

Drumul garantat este descris de fp(p)=bu[0] și nb(p,bu[i])=bw[i+1], pentru p care înaintează de la u la w.

Se definește **hsf=bgc(BGh).**

Teorema:

Există un controlor deadlock-free pentru o rețea astfel încât folosește D+1 buffere în fiecare nod (D diametrul din retea) și cere ca pachetele să fie trimise prin drumuri de lungime minimă de salturi.

Observație: Se consideră k=D.

Se poate folosi și schema în funcție de numărul de salturi pe care pachetul le mai are de făcut până la destinația sa.

Orientări pentru rețeaua G

Se va da o metodă de construcție a unui buffer graf folosind un număr mic de buffere în noduri.

În controlorul din exemplul 2) de mai sus, indexul buffer-ului în care un pachet este reținut crește cu fiecare salt.

De aceasta dată va fi o creștere mai lentă a indexului buffer-ului, aceasta realizându-se numai pentru anumite salturi.

Definiție:

O orientare aciclică a lui G este un graf aciclic, orientat obținut prin direcționarea tuturor muchiilor lui G.

O secvență G1,G2,..,GB de orientari aciclice a lui G este o acoperire orientată aciclică de mărime B pentru mulțimea *P* de drumuri, dacă oricare ar fi P din *P* poate fi scris ca o conca-tenare de B drumuri P1,P2,...,PB, unde Pi este un drum în Gi.

Un pachet este mereu generat în nodul u în bufferul bu[1].

Un pachet în buffer-ul bu[i] care trebuie să înainteze la nodul w este plasat în buffer-ul bw[i], dacă muchia uw este direcționată spre w în Gi și în buffer-ul bw[i+1], dacă muchia uw este orientată spre u in Gi.

Teorema:

Dacă o acoperire orientată aciclică pentru *P* de mărime B există, atunci există un controlor deadlock-free folosind doar B buffere în fiecare nod.

Demonstrație:

Fie G1,G2,...,GB o acoperire şi bu[1],bu[2],...,bu[B] bufferele nodului u.

Spunem că uw în Ei, dacă muchia uw este orientată spre w în Gi şi wu în Ei, dacă muchia uw este orientată spre u în Gi.

Se consideră buffer graful BGa=(B,BE) astfel încât:

bu[i]bw[j]în BE, *dacă şi numai dacă* (uw în E) şi [(i=j si uw in Ei) sau (i+1=j si wu în Ei)]

Se ia fb(p)=bu[1] şi

nb(p,bu[i])= bw[i], dacA uw în Ei

nb(p,bu[i])=bw[i+1], dacă wu în Ei.

Atunci **acoc=bgc(BGa)** este este un controlor deadlock-free folosind B buffere în fiecare nod.

Observație: Acoperirile orientate aciclice pot fi folosite pentru a da controlori deadlock-free pentru diverse clase de rețele:

a) dacă există o rețea de tip cerc, atunci există un controlor care utilizează numai 3 buffere pe nod şi permite ca pachetele să fie transmise pe cele mai mici drumuri; pentru aceasta, este suficient să se dea o acoperire orientată aciclică de mărime 3 pentru o colecție de drumuri care include un cel mai mic drum între fiecare pereche de noduri;

b) pentru rețele de tip arbore, există controlor deadlock-free care foloseşte doar două buffere pe nod.

3. Soluții nestructurate

Toueg si Ullman au intodus o nouă clasă de controale, care nu descriu în care buffere un pachet trebuie să fie plasat, ci doar folosesc informații locale, cum ar fi numărul de salturi şi numărul de buffere dintr-un nod.

3.1. Controlor cu numărare înainte (controlor care depinde de numărul de salturi pe care pachetul - mesajul - le mai are de făcut până la destinație)

Pentru un pachet p. se consideră sp numărul de salturi pe care le mai are de făcut până la destinație,0<=sp<=k.

Este important de menținut valoarea sp în pachet, deoarece mai mulți algoritmi încarcă această informație în fiecare nod.

Pentru un nod u, se notează cu fu numărul de buffere libere 0<=fu<=B.

Definiție: Controlorul **FC** acceptă un pachet p în nodul u, dacă şi numai dacă sp<fu .

Teorema:

Dacă B>k, atunci **FC** este un deadlock-free controlor.

Demonstrație:

Dacă nodul u are toate bufferele goale, atunci fu=B şi cum B>k, iar p face cel mult k salturi atunci pachetul p este acceptat de nodul u.

Se arată că este un deadlock-free prin reducere la absurd.

Se presupune l1 o configurație blocată pentru controlor. Configurația l2 se obține prin aplica-rea asupra configurației l1 a unei secvențe maxi-male de mişcări de înaintare si de consumare.

Nici un pachet nu se mai poate deplasa în l2 şi cum l1 este o configurație blocată, rezultă că există cel puțin un pachet blocat în configurația l2. Fie p un pachet în l2 cu distanța minimă față de destinatia sa, i.e. sp este cea mai mica dintre valorile pachetelor aflate in l2.

Se notează cu u nodul în care se afla p. Cum u nu este destinația lui p, există un vecin w al lui u la care p ar trebui să ajungă. Deoarece mişcarea nu este permisă sub **FC** rezultă că, avem sp-1>=fw, ceea ce implică fw<B.

Relația de mai sus, arată că există cel puțin un pachet în nodul w şi se va nota cu q pachetul cel mai recent acceptat de w. Fie f'w numărul de buffere libere aflate în w chiar înainte de acceptarea lui q in w.

Astfel:

f'w<=fw+1 şi cum q acceptat de w implică sq<f'w.

Rezultă,

sp<f'w<=fw+1<=sp,

contradicție cu alegerea lui p.

Deci, dacă B>k, atunci **FC** este un deadlock-free controlor.

3.2. Controlor cu numărare înapoi (controlor care depinde de numărul de salturi făcute de la sursa de către pachet)

Pentru un pachet p, se notează cu tp numărul de salturi pe care pachetul le-a făcut de la sursă.

Definiție:

Controlorul **BC** acceptă un pachet p în nodul u, *dacă și numai dacă* tp>k-fu.

Observație:

Controlorul **BC** este un "dual" pentru **FC**.

În cele ce urmează, se va da un rezultat analog celui de mai sus.

Teorema:

Dacă B>k, atunci **BC** este un deadlock-free controlor.

Demonstrație:

Dacă nodul u are toate buffer-ele goale, atunci fu=B și cum B>k, iar p face cel mult k salturi atunci pachetul p este acceptat de nodul u.

Se arată că este un deadlock-free prin reducere la absurd.

Se presupune I1 o configurație blocată pentru controlor. Configurația I2 se obține prin aplicarea asupra configurației I1 a unei secvențe maximale de mișcări de înaintare și de consumare.

Nici un pachet nu se mai poate deplasa în I2 și cum I1 este o configurație blocată, rezultă că există cel puțin un pachet blocat în configurația I2.

Fie p un pachet în I2 cu distanța maximă față de sursa sa, i.e. tp este cea mai mare dintre valorile pachetelor aflate în I2.

Se notează cu u nodul în care se află p. Cum u nu este destinația lui p, există un vecin w al lui u la care p ar trebui să ajungă. Deoarece mișcarea nu este permisă sub **FC**, rezultă că avem tp+1<=k-fw, ceea ce implică fw<B.

Relația de mai sus arată că există cel puțin un pachet în nodul w și se va nota cu q pachetul cel mai recent acceptat de w.

Fie f'w numărul de buffere libere, aflate în w chiar înainte de acceptarea lui q în w.

Astfel: f'w<=fw+1 și cum q acceptat de w implică tq>k-f'w.

Rezultă, tq>k-f'w>=k-fw-1>=tp,

contradicție cu alegerea lui p.

Deci, dacă B>k, atunci **BC** este un deadlock-free controlor.

3.3. Controlor funcție de stare-înainte și controlor funcție de stare-înapoi

Pentru fiecare nod u, se definește un *vector de stare* <j0,j1,...,jk> unde js reprezintă numărul de pachete p în u cu sp=s, sp definit anterior.

Definiție:

Controlorul **FS** acceptă un pachet p în nodul u cu vectorul de stare <j0,j1,...,js>, *dacă și numai dacă* pentru orice i, 0<=i<=sp există i<B-(j1+j2+...+jk)

Se notează cu <i0,i1,...,ik> un vector de stare, unde it este numărul de pachete în nodul u care au făcut t salturi.

Definiție:

Controlorul **BS** acceptă un pachet p în nodul u cu vectorul de stare <i0,i1,...,ik>, *dacă și numai dacă* pentru orice j, tp<=j<=k există j> (i0+i1+...+ij)-B+k, tp definit anterior.

Observații:

1) Cele două controloare sunt deadlock-free.

2)Se introduce noțiunea de "liber" în legătura cu de controlor:

Spunem că un controlor A este mai liber decât controlorul B, dacă o mutare permisă de A este permisă și de B.

Vom avea următoarele rezultate:

2.1) **FS** este mai liber decât **FC**

Se presupune că acceptarea lui p de u este permisă de către **FC**.

Atunci, sp<fu=B-(j0+j1+...+jk). Deci, pentru i<=sp, i<B-(ji+...jk), ceea ce arată că mișcarea este permisă de **FS**.

2.2) **FC** este mai liber decât **BC**.

Se presupune că acceptarea lui p de u este permisă de către **BC**.

Atunci, tp>k-fu, de unde rezultă fu>k-tp și cum în mod evident k-tp>=sp se poate spune că mișcarea permisă de către **BC** este, de asemenea, permisă și de **FC**.

2.3) **BS** mai liber decât **BC**

Se presupune că acceptarea lui p de u este permisă de către **BC**.

Atunci tp>k-fu.

Fie j astfel încât k=>j=>tp. Avem j=>tp>k-fu>=k-[-(i0+i1+...+ij)+B], ceea ce arată că, mișcarea este permisă și de **BS**.

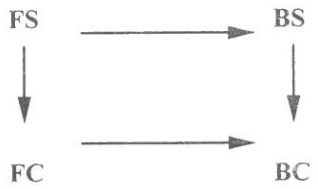
2.4) **FS** mai liber decât **BS**

Se presupune că acceptarea lui p de u este permisă de către **BS.**

Atunci, fiecarui i din 0,..,sp ii corespunde un j din tp,...,k şi pentru care se ştie că:

j> (i0+i1+...+ij)-B+k, adică k-j<B-(i0+i1+...+ij), de unde, notând pe i cu k-j, se obţine relaţia care asigură acceptarea pachetului de către **FS**.

Practic, diagrama între cele patru controloare relativ la noţiunea de "liber" arată astfel:



4. Concluzii

Se vor prezenta, în continuare, unele dintre aspectele noi, ce pot apărea în situaţii practice, privitoare la blocarea mesajelor în reţea.

1. Schimbări topologice

Există posibilitatea de schimbare a topologiei în timpul deplasărilor pachetelor prin reţea, de la sursa la destinaţie.

Actualizarea tabelelor de itinerariu se va face după fiecare schimbare şi mesajul va fi trimis folosind aceste noi date.

Ca urmare a acestor noi schimbări, există posibilitatea de a parcurge un drum care altfel nu ar fi existat posibilitatea de a-l urma.

2. Situaţiile livelock

Acestea sunt situaţiile în care, deşi, teoretic, un mesaj ar putea ajunge la destinaţie, practic, acest lucru nu se întâmplă.

Definiţie:

Dându-se o reţea G, un controlor con, o configuraţie l, un pachet p este livelock, dacă există o infinitate de secvenţe de mişcări, aplicabile în l, în care p nu este consumat.

O astfel de configuraţie care permite asemenea pachete se numeşte configuraţie livelock.

Un controlor este livelock-free, dacă nu apar astfel de configuraţii sub mişcările permise controlor.

3. Tipuri de blocări

Până acum au fost utilizate doar blocări de aşteptare şi înaintere.

Merlin şi Schweitzer au considerat 4 tipuri de blocări:

a) *blocări datorate capacităţii de creare de către un pachet al unui alt mesaj*:

Problemele apar la permiterea înaintării şi generării unui pachet. Evitarea acestora se poate face prin două cópii ale buffer-ului graf .

b) *blocări datorate păstrărilor de cópii*:

Apar când sursa păstrează o copie a unui pachet până când primeşte o înştiinţare de la destinaţie.

c) *blocări apărute când reţeaua conţine noduri cu rezerve limitate*:

exp: afişarea, mai întâi, a unor caractere înainte să poată accepta următoarele caractere pentru afişare;

d) *blocări datorate divizării mesajelor mari*:
Blocarea celorlalte mesaje până când toate diviziunile unui mesaj mare ajung la destinaţie.

Bibliografie

1. **TEL, G.**: Introduction to Distributed Algorithms, Cambridge University Press, 1994.
2. **MERLIN, P.P., SCHWEITZER,P.**: Deadlock Avoidance in Store-and-Forward Networks I: Store-and-Forward Deadlock. În: IEEE Trans.Commun, 1980
3. **MERLIN, P.P., SCHWEITZER,P.:** Deadlock Avoidance in Store-and-Forward Networks II: Other Deadlock Types. În: IEEE Trans. Commun, 1980
4. **TOUEG, S., ULLMAN, J.D.**: Deadlock-free Packet Switching Networks. În: SIAM Journal Comput, 10, 1981.