

## Rețele ATM și compresia în acestea

Ing. Laur IVAN, ec. Daniel VERNIȘ  
A.S.E., București

*Odată cu apariția rețelelor de calculatoare s-a pus problema transmisiei eficiente a datelor. Acest articol își propune să evidențieze câteva aspecte ale algoritmilor de compresie în rețele de mare viteză, și anume rețele ATM. Vor fi subliniate avantaje și dezavantaje ale tipurilor de algoritmi cunoscuți.*

*Cuvinte cheie: ATM, compresie, fractali.*

### Introducere

Pe măsura trecerii timpului cerințele pentru performanțele rețelelor au crescut de la transmisia de fișiere de lungime medie/mică prin modem, la rularea de aplicații aflate pe discurile altor calculatoare (remote) și mai recent la videoconferințe și transmisii multimedie de calitate.

Astfel, în 1960 s-a pus problema upgrade-ului de la linii analogice la linii care să suporte semnal analogic/digital. În Statele Unite au fost create multiplexoare de 24 de canale de voce de 64k ( $64=8$  biți\*8000 samples/sec) și în Europa multiplexoare de 32 canale de voce, ajungându-se la transmisii de 2Mb/s. Ca o consecință firească a fost apariția de rețele ISDN (Integrated Services Digital Network) și apoi a rețelelor de tip B-ISDN (Broadband Integrated Services Data Network). Pe baza standardului B-ISDN au fost create:

- SDH - Synchronous Digital Hierarchy - în Europa
- SONET - Synchronous Optical NETwork - în USA.

Ca o soluție pentru B-ISDN a apărut ATM (Asynchronous Transfer Mode).

### Rețele ATM

În comunicarea tradițională de tip analog, un apel este stabilit prin recepționarea unei căi (de la sursă la destinație). Aceasta este o proprietate păstrată tot timpul conversației. Într-un

sistem digital cu comutare de circuite, totă rata de transfer este asignată unei convorbiri doar o perioadă de timp pentru fiecare fereastră (frame). Acest procedeu este numit multiplexare prin diviziune în timp.

Pe durata unei ferestre, emițătorul va genera un număr fix de biți. Pentru o legatură, într-un frame, aceeași porțiune din ea este alocată pentru transmisie. Deci porțiunea alocată transmisiei este identificată prin poziția sa în frame.

Când conexiunea este stabilită și se găsește o cale prin rețea de la sursă la destinație, atunci aceasta rămâne fixă pe toată durata convorbirii. Ruta poate trece prin mai multe noduri de comutărie și necesită existența unui număr mare de legături pentru a se stabili un circuit de la sursă la destinație. Comutatoarele care leagă două căi trebuie să fie capabile să execute schimbarea porțiunii între cele două ferestre. În telefonia digitală, ferestrele au o frecvență de 8000 biți pe secundă (o durată de 125 μs), și sunt totdeauna 8 biți (1 byte) per slot. Fiecare canal are o rată de transfer de 64 Kbiți/sec. Cu N intervale de timp în fiecare frame, rata de transfer a liniei este de  $N \times 64$  Kbiți/sec. În practică sunt adăugate intervale de timp suplimentare sau biți suplimentari destinați funcțiilor de control și sincronizare. Spre exemplu, sistemul cu 30 de canale are încă 2 canale pentru sincronizare și control și de

aceea rata este de  $(30+2) \cdot 64K = 2048$  Kbit/sec.

Multiplexarea în timp poate fi aplicată recursiv considerând sistemul de 30 de canale ca un singur canal, fiecare frame ocupă un interval de timp într-o altă fereastră a unui nivel superior de multiplexare. Aceasta este principiul care stă la baza SDH (Synchronous Digital Hierarchy).

Principalul avantaj al unui astfel de sistem este acela că atunci când o cerere de conexiune este cerută, există și un circuit disponibil către destinație. Nu este efectuată nici o detectie de erori, aceasta rămânând la latitudinea terminalelor.

Spre deosebire de comutarea de circuite unde pachetul de date este organizat într-o fereastră de timp, în sistemul de comutare de pachete, un pachet de date este organizat din mai mulți octeți de date utilizator. Fiecare pachet are o etichetă care îl identifică unic ca aparținând unei comunicații.

Ruta de ieșire poate fi predeterminată la stabilirea comunicației sau poate fi variată funcție de condițiile de trafic (de exemplu alegerea rutei cel mai puțin încărcate). Prima metodă asigură ajungerea pachetelor în aceeași ordine ca la emisie, pe când a doua nu garantează acest lucru, deci destinația trebuie să fie capabilă să ordoneze pachetele care sosesc. Astfel se reduce rata de transfer efectivă pe linie.

O rețea cu comutare de pachete poate accepta în continuare pachete dacă este suprasolicitată, într-un regim neblockant, spre deosebire de sistemul cu comutare de circuite care respinge pachetele atunci când este suprasolicitat. Efectul acestei operații este o întârziere foarte mare din punctul de vedere al utilizatorului, întârziere inacceptabilă în cazul unor transmisii video.

ATM menține sistemul de transmisie din comutarea de circuite cu multiplexare în timp (fără ca poziția într-un

frame să aibă o semnificație deosebită), dar mărește lungimea unității de date la 53 de octeți. De asemenea, ATM menține conceptul de machetă dar îl restrâng la 53 de octeți, și în plus, necesită o transmisie sincronă a pachetelor.

Pachetul de 53 de octeți este numit celulă. O celulă are următoarele părți semnificative:

- header - 5 octeți
- date - 48 octeți

Lungimea pachetului de date de 48 de octeți a rezultat ca un compromis între 64 și 32 de octeți.

Header-ul conține o etichetă care identifică în mod unic pachetul ca aparținând unei conexiuni. Deci ATM utilizează multiplexarea pe baza etichetei și nu pe baza poziției. În ATM porțiunea din frame alocată transmisiei (time slot) se numește *cell slot*. O legătură de tip ATM se comportă ca o bandă rulantă cu compartimente de lungime fixă pe care merg celulele. Dacă există o celulă de transmis, ea trebuie să aștepte ca celula dinainte să se termine de transmis. O celulă nu are voie să ocupe două compartimente. Dacă nu există nici o celulă de transmis, atunci compartimentul aferent rămâne liber.

## Compresia pentru rețele ATM

Dat fiind faptul că rețelele ATM sunt destinate aplicațiilor cu volum relativ mare de date și cu tempi critici de răspuns, există posibilitatea compresiei, pentru a reduce din banda ocupată. Pentru aceasta, avem câteva variante :

1. Compresia "On-line"
2. Compresia "End-to-end"

Compresia de tip "On line" presupune ca switch-ul să comprezeze datele pe care le transmite mai departe pe linie. Acest lucru necesită un algoritm foarte rapid de compresie (de tip LZW spre exemplu), deoarece există o limită superioară pentru timpul de transmisie a pachetelor. Pentru aceasta avem posibilitatea de a crea un buffer în care să se depoziteze datele până la finalizarea compresiei. În acest fel, se poate obține o compresie mai bună, dar și o creștere a consumului de memorie.

bilitatea unui algoritm dinamic, care se bazează pe pachetele transmise înainte. Dezavantajul este că, la pierderea unui pachet, nu se mai poate reconstitui ce era înainte. O altă alternativă ar fi aceea a algoritmilor statici. Pachetul având o dimensiune foarte mică, nu poate fi compresat corespunzător, deci algoritmul trebuie să funcționeze pentru mai multe pachete. Aceasta implică o creștere a buffer-ului de aşteptare a pachetelor și o creștere a complexității echipamentului electronic. De asemenea, pentru această variantă apare și problema decompresiei la celalalt capăt (la receptor).

De aceea este de preferat varianta compresiei de tip "End-to-end", în care sursa efectuează compresia și destinația efectuează decompresia. Pentru sunet și pentru imagine video (în mișcare) apare problema resincronizării la pierdere de pachet, deoarece în majoritatea cazurilor se poate permite pierderea pachetelor fără a se mai cere din nou transmiterea lor (ar genera o întârziere foarte mare). Pentru aplicații de tip multimedia se preferă o compresie de tip MPEG (pentru imagine cât și pentru sunet), iar la pachetele transmise sunt adăugate niște date de control (de sincronizare) cel puțin pentru imagine. Dată fiind viteza foarte mare de transmisie, datele de control se pot reduce la o informație de genul "este un cadru nou"/"nu este un cadru nou" dintr-un sir de secvențe în mișcare.

O variantă hibridă este aceea ca aplicația să genereze datele necompresate, iar un driver sau chiar placa de rețea să execute compresia în timp real. O soluție, în afara de compresia MPEG, este compresia cu fractali simplificată.

Pentru o imagine dintr-un sir (film) se execută următoarele operații:

1. se face diferență cu imaginea precedentă;
2. se compresează imaginea cu fractali în felul următor: se alege un caroaj *do dimensiuni convenabile* pentru ima-

gine. Dacă dreptunghiul din caroajul respectiv are toată culoarea 0 sau o pondere *suficient de mare* 0, atunci se compresează prin transmiterea / reținerea coordonatelor unui colț convenabil. Dacă nu, atunci se alege o culoare de medie (sau două culori medii pentru niște zone din dreptunghi) și eventual un parametru care va defini o funcție de decompresie (gradații de culoare după diferite axe) și se vor transmite. Dacă se folosește și un algoritm genetic într-un singur pas, crește overheadul, dar se pot mări dimensiunile celulelor caroajului până la 1.5-2 față de cel inițial.

## Concluzii

Se poate demonstra că în rețelele ATM compresia conservativă de informații (algoritmi fără pierderi) nu își are rostul pentru aplicațiile multimedia. Aceasta deoarece ar trebui impuse niște condiții foarte dure (nici un pachet să nu se piardă) pentru că la pierderea unui pachet informația nu mai poate fi refacută (cu excepția algoritmilor matematici care necesită un volum mare de calcule).

De asemenea algoritmii de compresie neconservativi pot fi folosiți pentru imagini multimedia sau pentru transmisie de fidelitate medie/înaltă. Pentru imagini de foarte mare fidelitate, dacă se pierde un pachet, în imaginea originală acesta înseamnă maxim o linie de imagine, iar în imaginea compresată, acesta poate însemna toata imaginea. Problema ce apare la compresia imaginilor constă în variabilitatea lungimilor datelor rezultate, deci transmisia la nivel de imagine compresată este asincronă, temporizarea fiind un atribut al receptorului.

## Bibliografie

- [HGP96] Harry G. Perros, "Call Admission Control Schemes: A Re-

- view”, IEEE Communications Magazine Vol. 34 No. 11, November 1996
- [AXN96] Ambalavanar Arulambalam, Xiaoqiang Chen, Nirwan Ansari, “Allocating Fair Rates for Available Bit Rate Service in ATM Networks”, IEEE Communications Magazine Vol. 34 No. 11, November 1996.
- [GLL95] Guang-Liang Li, “Admission Control for Transmitting Connection-less Traffic Across ATM Networks”, IEEE Fourteenth Annual International Phoenix Conference on Computers and Communications, Scottsdale, Arizona USA, March 28-31, 1995.
- [YAS95] Y. Chang, A van der Horst, S. Wakid, “An ATM Protocol for Local Access and Control of Internal/External Traffic”, IEEE Fourteenth Annual International Phoenix Conference on Computers and Communications, Scottsdale, Arizona USA, March 28-31, 1995

- [KYS95] Kai-Yeung Siu, Hong-Yi Tzeng, “Adaptive Proportional Rate Control for ABR Service in ATM Networks”, IEEE Fourteenth Annual International Phoenix Conference on Computers and Communications, Scottsdale, Arizona USA, March 28-31, 1995.
- [RTB95] Robert Engel, Toni Bieri, Beat Keller, “Signalling in ATM Networks: Experiences With an Object-Oriented Solution”, IEEE Fourteenth Annual International Phoenix Conference on Computers and Communications, Scottsdale, Arizona USA, March 28-31, 1995.
- [JMP96] J. M. Pitts, J. A. Schormans, “Introduction to ATM design and Performance”, John Wiley & Sons, Chichester, New York, Brisbane, Toronto, Singapore, 1996
- [MPr96] Martin de Prycker, “Asynchronous Transfer Mode Solution for Broadband ISDN”, Ellis Horwood, New York 1993.