

PROTOCOALE DE SINCRONIZARE MULTIMEDIA: ANALIZĂ ȘI COMPARAȚIE

Șef lucrări ing. Henri - George Coandă

Universitatea VALAHIA - Târgoviște

coanda@valahia.ro

Rezumat: Odată cu expansiunea aplicațiilor multimedia distribuite, precum *videoconferință*, *video la cerere (VoD)*, *învățământ la distanță*, *jocuri distribuite*, *filme la cerere*, *videotelefonie*, sincronizarea diferitelor tipuri de fluxuri media (dependente sau independente temporar) a devenit parte integrală a diverselor protocoale, mecanisme și servicii în sistemele de calcul și de comunicații. Scopul lucrării este de a analiza și compara arhitectura și performanțele unora din cele mai cunoscute protocoale de sincronizare și a serviciilor oferite de acestea.

Cuvinte cheie: multimedia, aplicații distribuite, protocol de sincronizare, adaptiv, QoS.

1. Introducere

Dezvoltarea vertiginoasă a performanțelor tehnicilor de calcul și, în special, progresul continuu în domeniul comunicațiilor au condus la delimitarea unui domeniu aplicativ aparte – cel al sistemelor multimedia. Trăsătura esențială a unui sistem multimedia este integrarea în aplicație a datelor, semnalelor audio și semnalelor video.

Cerințele aplicațiilor multimedia pot fi specificate prin funcționalități și parametri de calitate a serviciului, QoS. *Multicast*, *multiconexiune*, *sincronizare* sunt câteva din funcționalitățile unui flux multimedia, iar *lărgimea de bandă*, *debit*, *întârziere*, *variația jitter-ului*, *procent de erori acceptabil* reprezintă parametrii QoS cei mai importanți. Parametrii sunt utilizați de sistemele de comunicații pentru a garanta execuția normală a aplicației. Aplicațiile multimedia pot fi *aplicații în timp real*, cu necesitatea garantării lărgimii de bandă și a debitului cerute de aplicație, precum și cu impunerea sincronizării între utilizatori și între fluxurile media implicate, respectiv *fără constrângeri de timp*, mai puțin consumatoare, cu privire la bandă și debit.

Fluxurile media pot fi: *discrete* – cu valori independente în timp (text, grafice) și timpi de răspuns de ordinul secundelor, respectiv *continue* – cu valori dependente în timp (voce, video). Într-un un flux media există 3 tipuri de relații: *de conținut* (dependențele între obiectele documentului), *spațiale* (definesc spațiul utilizat pentru prezentarea unui obiect la un periferic de ieșire, la un moment dat, într-o prezentare) și *temporale* (definesc dependențele temporale între obiecte). Specificarea acestora poate fi realizată *implicit* (restituite exact cum au fost capturate) sau *explicit* (obiecte create sau capturate independent).

O aplicație multimedia cere, în general, 4 tipuri de sincronizări: *inter-media* - care face referință la relația temporală între multiple unități de prezentare ale unui mediu continuu, *intra-media* - care face referință la relația dintre obiectele ce compun fluxul media, *condițională și multipunct* care fac referință la prezentarea unui eveniment dacă se îndeplinește o condiție, respectiv, la producerea acestuia, în același timp în diferite locații.

Sincronizarea *inter-media* poate fi *în timp real* (se reproduc pe durata de restituire aceleași relații temporale, prezente în perioada de achiziție) sau *artificială* (aplicațiile pot replanifica sincronizarea, existând relații flexibile de sincronizare între diferite medii). Relațiile temporale ale unor astfel de aplicații sunt variabile.

2. Stadiul actual în domeniul sincronizării multimedia

Sistemele multimedia urmăresc 4 tipuri de configurații: *unicast*, *multicast*, *de regăsire* și *de grup*. În aceste sisteme, se identifică 4 surse de asincronism: *întârzierea jitter-ului* (variația întârzierii) care este datorată timpului de achiziție, timpului de parcurgere a rețelei, timpului de restituire, *deriva ceasurilor locale*, *diferențele între timpii de inițiere a achizițiilor*, *diferențele între timpii de inițiere a restituirilor*.

Se disting 3 strategii pentru asigurarea sincronizării multimedia: *stampile temporale*, *canale de tip feedback* ce permit transmiterea periodică a informațiilor de timp și sincronizare și *multiplexarea fluxurilor media*, ce permite transferul de date legate între ele temporal, prin același canal [7].

De asemenea, protocoalele pot fi împărțite în 2 clase: cele care presupun existența ceasurilor nodurilor rețelei sincronizate și, ca atare, se pot inspira din metoda anterioară, și cele care nu fac această presupunere, caz în care este necesară o altă metodă decât cea propusă până acum.

În mod curent, sistemele permit și oferă două medii de gestiune a resurselor [3], ce pot fi privite și ca 2 clase de servicii:

- *tip best-effort* - implementate în sistemele de operare și rețelele tradiționale, prin utilizarea cozilor FIFO și a proceselor cu partajare în timp (UNIX, IP prin rețeaua Ethernet); într-un astfel de mediu, rețeaua și OS maximizează eficiența, și nu oferă garanții pentru QoS, astfel încât aplicațiile ce doresc garanții trebuie să fie adaptive sau să utilizeze protocoale adaptive;
- *cu rezervare* – utilizează controlul admisiei conexiunii, priorități statice, multiple forme de trafic și lungimi variabile pentru cozile de așteptare (RSVP/IP sau ATM); într-un astfel de mediu, rețeaua și OS utilizează algoritmi pentru rezervare de resurse și control al admisiei conexiunii pentru a garanta QoS.

Abordările cu privire la protocoalele de sincronizare, existente în literatura de specialitate, diferă în funcție de tipurile de configurații suportate. Unele dintre acestea impun prezența clienților unui grup de sincronizare în același nod [2], altele presupun existența unei arhitecturi centralizate, bazată pe un server care memorează și distribuie fluxurile de date [8], [10]. În acest ultim caz, clienții grupului trimit periodic mesaje de reacție spre server, utilizate de acesta în estimarea stării fiecărui flux individual. Dacă ceasurile nodurilor clienților nu sunt sincronizate, calitatea estimării va depinde de jitter-ul mesajelor de reacție, care se presupune a fi limitat. O abordare similară este realizată în [1], care nu cere limitarea jitter-ului, dar estimează diferențele prin intermediul *mesajelor de probă (de sondare)*. Protocoale precum cele propuse de [4] și [5] presupun sincronizarea ceasurilor și suportă configurații în care elementele grupului sunt distribuite. Totuși, niciunul dintre acestea nu permit clientului să reacționeze imediat, în situația în care starea fluxului devine critică. Nu se suportă noțiunea de flux master, excluzându-se anumite politici de sincronizare.

3. Structura protocoalelor de sincronizare

Considerăm 2 fluxuri de date, ce parcurg separat și independent nodurile unei rețele. Dacă ceasurile nodurilor sunt sincronizate, pentru a menține sincronizarea fluxurilor, o posibilă soluție constă în a ștampila, temporal, datele la părăsirea surselor și de a calcula, în paralel, timpii de parcurs ai rețelei până la destinație. Timpul maxim va fi acceptat de entitatea de protocol, destinatară ca timp de bază, și, ca atare, va întârzia datele mai rapide înainte de a le oferi aplicației client.

Ideea pare simplă, dar protocoalele trebuie să facă față unor constrângeri: *trebuie să fie funcționale pentru 4 configurații, să permită fluctuații ale timpilor de parcurs în rețea și, eventual, să facă față situațiilor extreme, în care timpii de parcurs sunt foarte mari (fluxul nu poate fi recuperat)*. În plus, în aplicațiile integrând noțiunea de grup, protocolul trebuie să permită gestiunea multiplelor grupuri de sincronizare, în același mediu de lucru.

Cele 6 protocoale propuse abordează problematica expusă, niciunul nu tratează toate aspectele, dar fiecare abordează unul sau mai multe, în funcție de tipul de aplicație pentru care a fost elaborat.

3.1 AFSP – Protocol de sincronizare adaptivă a fluxului

Protocolul AFSP („Adaptive flow synchronization protocol”) permite sincronizarea expedierii datelor dinspre și spre site-uri geografice distribuite. Acesta introduce *sincronizarea adaptivă*, în cazul modificării condițiilor rețelei, *flexibilitate în menținerea sincronizării grupurilor multiple și o arhitectură modulară*, ce permite aplicației client să efectueze calcule de sincronizare în funcție de exigențe [5].

Protocolul utilizează o *arhitectură modulară*, care autorizează aplicațiile să influențeze procesele de egalizare (de ex.: cum să reacționeze la timpii de parcurs ai rețelei foarte lungi) și *propriile sale date ștampilate temporal* pentru a depinde cât mai puțin de mecanismele de transport ale diferitelor rețele.

Protocolul implementează o sincronizare bazată pe *modelul întârzierii datelor* (vezi figura 1), cu următorii parametri:

- δ_c – timp de achiziție (de ex.: de codare a datelor);
- δ_t – timp de tranzit, până la atingerea entității de protocol distanțe;
- δ_e – timp de egalizare, inserat de protocol pentru a egaliza timpii de parcurs în rețea ai fluxurilor de date;
- δ_d – timp de eliberare (restituire), între momentele de deblocare a datelor (t_r) de către protocol și prezența acestora în cadrul aplicației distanțe;
- Δ – reprezintă suma celor 4 timp;
- $d = \delta_c + \delta_t + \delta_d$ – întârzierea fluxului de date neegalizat.

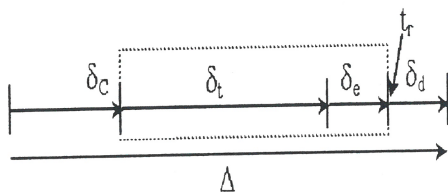


Figura 1. Modelul întârzierii datelor

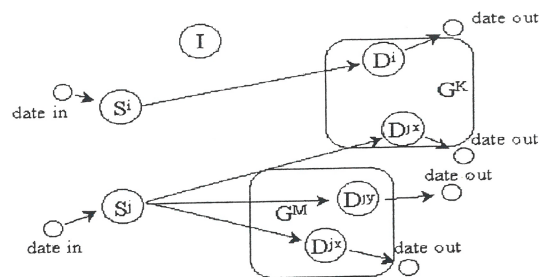


Figura 2. Modelul de protocol

Figura 2 conține modelul de protocol. Procesul inițiator I distribuie informațiile de control: un flux procesului sursă S^i și altul procesului destinar D^i . S^i ștampilează temporal datele, iar D^i implementează egalizarea utilizând buffere pentru stocarea datelor. Un flux de date este reprezentat de datagrame, celule sau șiruri de biți. Un flux multicast j este reprezentat de multiple fluxuri (j_x, j_z, j_z), ce partajează aceeași sursă. Fluxurile ale căror destinații aparțin aceluiași grup de sincronizare (G^K, G^M) vor fi sincronizate între ele de către protocol. Controlul executat de protocol utilizează 3 tipuri de mesaje: de control, de eveniment și de perioadă (figura 3).

Inițiatorul expediază mesaje de control tuturor proceselor (sursă și destinație) pentru a inițializa sau schimba parametrii sesiunii. Procesele sursă expediază mesaje eveniment proceselor destinație pentru a le comunica parametrii ce descriu fluxurile, dacă este necesar. Aplicația grupează destinațiile în grupuri de sincronizare, procesele aceluiași grup schimbând regulat informații prin intermediul mesajelor de perioadă în scopul calculării întârzierii în sincronizare Δ^K comună unui grup.

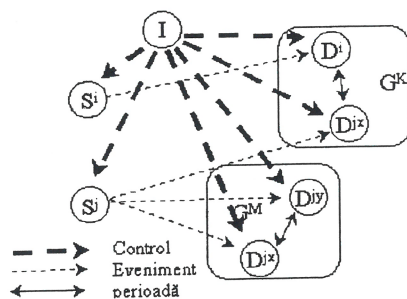


Figura 3. Mesaje de protocol

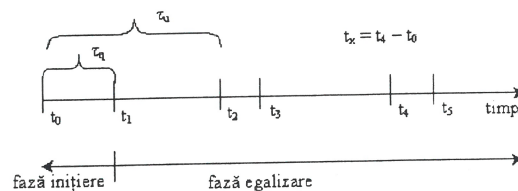


Figura 4. Parametrii de timp și faze de protocol

Inițiatorul distribuie 4 parametri de timp, care partajează intervalul de timp al procesului de sincronizare în perioade de timp de sincronizare:

- t_0 – timp de referință pentru ștampilare;
- τ_u – lungimea perioadei de actualizare;
- τ_q – lungimea intervalului de repaus la începutul perioadei în care trebuie să se efectueze schimbul mesajelor de perioadă;
- τ_x – indică momentul la care conținutul mesajului de control intră în vigoare.

Parametrii de timp și fazele de protocol sunt reprezentate în figura 4. Aplicația client activează procesele sesiunii și protocolul începe prin faza de inițializare, în care mesajele de control (eveniment și perioadă) inițializează parametrii de timp. La momentul t_1 faza de egalizare începe și se menține sincronizarea.

3.2 MMA – Algoritm de sincronizare prin multiplexarea fluxurilor

Algoritmul MMA („Media mixing algorithm”) a fost creat pentru a suporta aplicații de teleconferință [11]. Utilizează o ierarhie în combinarea datelor și un unificator denumit *rădăcină*, ce expediază datele spre toți participanții. Nu utilizează ceasul global al rețelei și acceptă prezența întârzierilor în timpii de comunicație. Nu există alte mesaje în afara datelor fluxurilor media, între participanți și sursa de date. În implementare, s-a preferat o arhitectură ierarhizată, în loc de una centralizată sau distribuită, în scopul oferirii de suport pentru un mare număr de participanți (peste 400).

Algoritmul de mixare va fi discutat pentru o conferință în care sunt m surse $S_1, S_2, \dots, S_m$ și fiecare generează pachete la viteză constantă, cu perioada p pe care le transmite spre mixer (M). Întârzierea în comunicație (timp transmisie, memorare noduri intermediare etc.) va fi mărginită de $\Delta_{\min}$ și $\Delta_{\max}$.

Dacă τ_s este momentul sosirii la mixer a pachetului numerotat n_s generat de sursa S_s , atunci cel mai rapid și cel mai târziu moment de generare a pachetului vor fi: $g_s^e(n_s) = \tau_s - \Delta_{\max}$ respectiv $g_s^l(n_s) = \tau_s - \Delta_{\min}$.

Măsurat de ceasul mixer-ului, momentul de timp de generare $g_s(n_s)$ se va regăsi în intervalul $[g_s^e(n_s), g_s^l(n_s)]$, care poartă denumirea de *interval de generare* al pachetului n_s .

Mecanismul de mixare constă în memorarea pachetelor media, recepționate de la cele m surse, până când cel puțin un pachet este recepționat de la fiecare din ele (creare set de fuziune). Prin regula de mixare, diferența între pachete poate fi cel mult p . Totuși, o astfel de schemă „memorează și mixează” are și dezavantaje. Datorită efectului jitter-ului, 2 pachete generate la distanță de $p + 2(\Delta_{\max} - \Delta_{\min})$ pot ajunge la mixer în fereastra de dimensiune $p + (\Delta_{\max} - \Delta_{\min})$ ceea ce va conduce la violarea regulii de mixare (caz WAN). Pe de altă parte, pierderile de pachete pot fi, de asemenea, problematice.

3.3 AFP – Protocol de sincronizare adaptiv prin reacție

Protocolul AFP („Adaptive feedback protocol”) rezolvă problema asincronismelor pentru serviciile multimedia la cerere (configurație multicast). Ceasurile nodurilor rețelei nu sunt sincronizate. Protocolul realizează o sincronizare inter-media, folosind un server ce utilizează mesaje de reacție (*feedback*), transmise de periferice în scopul estimării momentelor de restituire a datelor și pentru detectarea problemelor de desincronizare [8]. Server-ul corectează desincronizarea prin mărirea sau diminuarea vitezei de restituire a perifericelor (*mediaphones*) care pot fi, astfel, accelerate sau decelerate. Se propun 3 reguli de resincronizare: *conservatoare* (server-ul garantează o restituire asincronă), *agresivă* (există o mică șansă ca restituirea să fie asincronă) și *probabilistică* (regula intermediară între cele două anterioare; utilizată atât timp cât distribuția statistică a întârzierilor în rețea și distribuția frecvenței de restituire sunt cunoscute).

Se consideră că rețeaua care interconectează server-ul cu clienții este presupusă a avea o întârziere limitată de $\Delta_{\min}$ și $\Delta_{\max}$ pentru fiecare unitate de date, transmisă pe canalul de feedback. Întârzierea în transmisie a rețelei poate fi compensată prin rezervarea resurselor necesare inițierii restituirilor – *memorii de rezervare la nivelul nodurilor rețelei, a server-ului și a clienților*. Mesajele de feedback sunt replici ale unităților de date, dar fără date. Neexistând un ceas global, decalajele între restituiri pentru clienți vor fi limitate la valoarea $\pm p$ pe perioada de restituire θ .

Decalajele între frecvențele de restituire introduc desincronizări între clienții, unii fiind mai rapizi/lenți decât alții. Protocolul autorizează clienții să emită mesaje doar când au recepționat un număr suficient de unități de date. În absența mesajelor de reacție, asincronismul maxim este calculat prin:

$$\left[\frac{(\Delta_{\max} - \Delta_{\min})}{\theta \cdot (1 - \rho)} \right] + \left[\frac{2 \cdot \theta \cdot \rho \cdot \mu}{\theta \cdot (1 - \rho)} \right]$$

unde primul termen este întârzierea jitter-ului, al doilea este contribuția decalajelor dintre frecvențele de restituire, iar μ este unitatea de date curentă.

Un mecanism simplu de resincronizare constă în a autoriza server-ul să accelereze anumiți clienți și să-i încetinească pe alții sau să oprească continuitatea restituirii, în cele mai defavorabile cazuri. Soluția unui flux master poate evita aceste asincronisme. Cât timp fluxul master este restituit la frecvența sa normală, fluxurile slave pot fi accelerate sau încetinite, rămânând sincrone cu fluxul master. Alegerea fluxului master depinde de aplicație.

3.4 Protocolul de sincronizare al grupului Yen – Akyildiz

Protocolul elimină cele 4 surse de asincronism, suportând configurații unicast, multicast, de regăsire și de grup, și include servicii de difuziune în timp real, pentru a asigura sincronizarea la nivelul aplicație. În același timp, nivelul aplicație trebuie să informeze protocolul despre obiectele multimedia, ce includ flux sincron, și despre perioadele de achiziție și de restituire.

Asincronismul dintre emițătorul i și receptorul j poate fi rezolvat prin utilizarea unui buffer de dimensiune $B_j = \lceil J_{ij}/\mu_j \rceil$, unde J_{ij} reprezintă întârzierea jitter-ului între i și j , egală cu $\Delta_{ij} - \delta_{ij}$; Δ_{ij} fiind timpul de parcurs maxim al rețelei între i și j ; δ_{ij} și fiind timpul de parcurs minim între i și j ; μ_j care reprezintă timpul de restituire a datelor fluxului media, la nivelul clientului j . Pentru corecția decalajului ceasurilor locale, s-a ales o schemă de sincronizare cu ajustare dinamică, aceasta folosind un timp de referință numit *timp global virtual (VGT)*. Valoarea acestuia este egală cu valoarea ceasului global al stației master, aleasă printr-un algoritm de alegere

distribuit. Periodic, stația distribuie informațiile temporale (valoarea VGT) tuturor utilizatorilor grupului. Ceasul local al fiecărui utilizator poate fi sincronizat cu cel al stației master, prin aplicarea periodică a protocolului de ajustare dinamică de perioadă pentru toate cuplurile client-master. Perioada de ajustare μ_j depinde de nevoia de sincronizare a aplicației și de nivelul de decalaj al ceasurilor. În ce privește sincronizarea timpilor de inițiere a achiziției, în faza de negociere, fiecare utilizator măsoară timpii de propagare dintre el și ceilalți membri activi ai grupului. O dată stația master aleasă, clientul i poate utiliza θ_{chj} dintre el și master pentru a calcula VGT. După ce master-ul începe să distribuie valoarea VGT, fiecare utilizator îi va înștiința pe ceilalți membri ai grupului asupra timpului inițial de achiziție I_i în termenii VGT. După recepția tuturor I_i , fiecare utilizator va alege cea mai mare valoare ca timp global inițial de achiziție.

3.5 ASSP – Protocol de sincronizare de flux adaptiv

Protocolul ASSP (“Adaptive stream synchronization protocol”) a fost creat pentru a oferi adaptare rapidă în sincronizare la modificările condițiilor rețelei [9]. Este flexibil, suportând o varietate de reguli (politici) de sincronizare, permițându-se comutarea de la una la alta, dinamic pe durata unei sesiuni. Protocolul presupune sincronizarea ceasurilor nodurilor rețelei, suportând doar 2 tipuri de configurații: *unicast* și *multicast*. Setul de fluxuri necesar va fi prezentat într-o manieră sincronă și se numește *grup de sincronizare* (*grup sync*). Pentru fiecare grup, există un *controller* (*server*) și mai mulți *agenți de sincronizat* (*agenți sync*). Controller-ul este entitatea software, care menține informații de stare, și realizează controlul operațiilor ce privesc grupul sync. El comunică cu agenții sync – unități software ce controlează fluxuri individuale. Pentru fiecare flux, există un agent sursă și un agent sync care împreună, pornesc/opresc prezentarea sau/și modifică rata de restituire. Agenții sync ai unui grup de sincronizare pot comunica în scopul adaptării momentelor de restituire.

Protocolul exploatează sincronizarea ceasurilor nodurilor rețelei. Agenții sursă ai grupului sync pornesc emisia datelor, simultan, la momentul t_0 . O unitate de date u va fi expediată la $t_0 + TS(u)$, unde $TS(u)$ reprezintă ștampila temporală asociată unității u . Fiecare agent sync pornește restituirea la momentul $t_0 + \Delta$, unde Δ trebuie să fie destul de mare pentru a putea permite primelor date expediate să ajungă în buffer-ul agentului sync. Fiecare unitate u va fi restituită la $t_0 + \Delta + TS(u)$. Întârzierea capăt-la-capăt maximă în rețea determină valoarea lui $\Delta = \max\{d_{ij}\}$, unde d_i reprezintă întârzierea fluxului i . Unitățile de date, ce sosesc înainte de $t_0 + \Delta$, vor fi memorate, asigurând egalizarea diferitelor întârzieri.

Adaptarea în ASP este coordonată de un *algoritm master/slave dinamic*. Există un *flux master*, responsabil de adaptarea lui Δ , care ia decizii în funcție de monitorizarea locală la agentul sync. O dată luată decizia de modificare a lui Δ , aceasta se propagă spre toți agenții sync și spre toate fluxurile slave ai/ale grupului de sincronizare. Algoritmul este dinamic în sensul în care ori de câte ori un *slave* devine critic, poate deveni imediat master. Din nefericire, într-o asemenea situație, pot exista simultan mai multe fluxuri master. ASP este capabil să rezolve situația fără pierderea sincronizării, asigurând, în final, după perioada critică, prezența unui singur master.

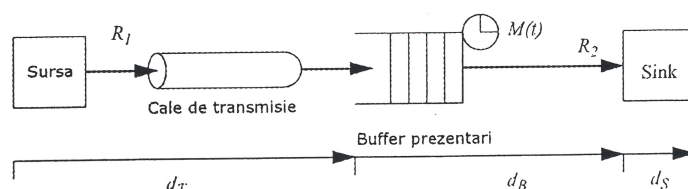


Figura 5. Flux de date și întârzieri în ASP

Modelul transmisiei fluxului și al memorării în buffer este prezentat în figura 5. ASP modifică Δ prin creșterea sau descreșterea ratei de servire R_2 pentru un anumit interval de timp. În timpul operării normale, $R_2 = R_1$. Figura evidențiază întârzierea în transfer d_T (include timpi de achiziție, comunicație, procesare, transfer în buffer), întârzierea în buffer d_B precum și întârzierea în prezentare (restituire) d_S . Starea temporală a unității de date este specificată prin *timpul media* $M(t) = TS - d_S$.

3.6. LVoD - Protocol de sincronizare adaptiv pentru video la cerere

Protocolul LVoD („Lip synchronization for adaptive VOD system”) este creat pentru aplicațiile multimedia de tip *video la cerere* (*Video On Demande*). Este adaptiv și reglează problema comportamentului nedeterminist al sistemelor client ale unei rețele în scopul realizării unei sincronizări *intra-media* cu minim de decalaj între datele dependente [6].

Particularitatea protocolului este de asigurare a sincronizării *intra-media*, modificând informațiile în interiorul fluxului de date, în funcție de resursele sistemului client.

Fiecare flux de date continuu include informații de sincronizare spațiale și temporale. În cazul în care doar informațiile temporale prezintă interes, se consideră o frecvență de achiziție diferită pentru fiecare media (audio și video). În timpul restituirii, sistemul de exploatare trebuie să furnizeze resursele necesare pentru a suporta o anumită frecvență – *frecvență de sistem*. Frecvența sistemului este determinată în faza de conexiune, utilizând un algoritm ce are la bază o *probă (sondă)*. Sonda – o *porțiune de semnal video* – este utilizată pentru a determina frecvența sistemului, posibilă în partea clientului. Pentru aceasta, server-ul expediază datele cât mai repede posibil, iar un sistem de observare în partea clientului determină frecvența posibilă pentru restituirea datelor. Algoritmul se termină fie când degradarea devine semnificativă (frecvența scade sub o anumită limită), fie când timpul de execuție al algoritmului este depășit. Frecvența de restituire, măsurată de client în timpul fazei de sondare, este utilizată ca frecvență inițială de către sistem atât ca frecvență de transmisie în partea server-ului, cât și ca frecvență de restituire în partea clientului. Tabelul 1 sintetizează operațiile executate în timpul *fazei de conexiune*, respectiv, al *fazei de transmisie*.

Tabelul 1. Faze ale protocolului client / server

	Server		Client
Faza de negociere		←	deschidere conexiune TCP pentru informații de reacție
	Expediere listă fișiere	→	recepție listă fișiere
	Recepție nume fișier	←	expediere nume fișier
	Expediere $n_g, n_l, n_p, n_B, a_{freq}, r_u$	→	Recepție $n_g, n_l, n_p, n_B, a_{freq}, r_u$
		←	deschidere conexiune UDP pentru video și audio
Faza de adaptare și renegociere	Expediere prim grup (determina QoS)	→	
			$t_w = (n_g/r_u - n_g/r_s)$
	Recepție t_w	←	Dacă $t_w \geq 0$ atunci expediază t_w
	Formează APDU fără suprimări	→	
	Așteaptă $(t_w/n_g - t_{send}^l)$		Dacă $t_w < 0$ atunci
			Dacă $((diff = n_g/r_s - t_p \cdot n_l - t_p \cdot n_p) \geq 0)$ pune $\{ n_B = diff/t_B \}$
	Recepție formă	←	Determină și expediază forma de afișare (tablou) }
	Construcție APDU cu suprimări	→	Altfel, dacă $((diff = n_g/r_s - t_p \cdot n_l) \geq 0)$ $\{ n_p = diff/t_p \}$
			Suprimă toate cadrele B
	Recepție formă	←	determină și expediază o formă de afișare a n_p în grup }
	Construcție APDU cu suprimări	→	Altfel, $\{ \text{elimină toate imaginile B, P în grupurile } n_g/n_l \}$
			Afișează fiecare al $(t_p \cdot r_s/n_g)$ -lea cadru I
	Recepție formă	←	Expediază cererea pentru cadre I }
	Construcție APDU	→	Se gestionează n_g/n_l grupuri

4. Evaluarea performanțelor, comparații și concluzii

Tabelul 2 oferă o sinteză a funcționalităților și a parametrilor de calitate, ai protocoalelor analizate.

- analiza protocolului "Adaptive flow synchronization protocol" (3.1) permite verificarea avantajelor oferite de integrarea conceptelor: întârziere în sincronizare adaptivă, sincronizare de grup multiplă și modularitate pentru funcțiile orientate pe aplicație; se demonstrează faptul că plasarea utilizatorilor în grupuri de sincronizare permite calculul timpului de sincronizare comun grupului, ceea ce, în anumite cazuri, induce corecții ale jitter-ului și ale timpilor de achiziție și restituire; din nefericire, protocolul presupune existența sincronizării ceasurilor nodurilor rețelei, cerință care limitează aplicabilitatea metodei;
- „Media mixing algorithm” (3.2) permite corecția de jitter și deriva ceasului local, dar procedura de inițializare a sincronizării unei conferințe multimediale nu este clar exprimată; problemele legate de timpii de achiziție și de restituire nu sunt rezolvate, iar unificatorii implicații introduc o întârziere suplimentară în parcurgerea rețelei ce poate fi critică pentru WAN;

- "Adaptive feedback protocol" (3.3) rezolvă problema desincronizării în configurația multicast, fără să fie necesară sincronizarea globală a ceasurilor rețelei, dar introduce necesitatea unei conexiuni suplimentare pentru fiecare cuplu emițător-receptor în vederea transportului informațiilor de tip feedback; protocolul este performant în LAN, dar nu este adaptat pentru WAN deoarece garanția sincronizării este limitată de timpul maxim de parcurgere al rețelei;
- protocolul de sincronizare al grupului Yen – Akyildiz (subcapitolul 3.4) este primul care corectează toate asincronismele, nu necesită conexiuni suplimentare de tip feedback, răspunde rapid și precis celei mai mici desincronizări apărute, este performant atât în LAN, cât și în WAN;
- "Adaptive stream synchronization protocol" (subcapitolul 3.5) răspunde perfect obiectivului: asigură sincronizarea inter-media între evenimente distribuite; din păcate doar 3 din cele 4 tipuri de asincronisme sunt abordate, iar ceasurile rețelei se presupun sincronizate;
- "Lip synchronization for adaptive VOD system" (subcapitolul 3.6) privilegiază fluxul audio în defavoarea fluxului video, ultimul fiind degradat, abordarea multiplexării nefiind o soluție care să aducă avantaje pentru cazul rețelei ATM; protocolul este performant deoarece 85% din decalajele audio-video sunt în gama -80÷80ms imperceptibile pentru om, restul situându-se în gama -160÷160ms încă acceptabilă.

		Flow Synchr. (3.1)	Media mixing (3.2)	Adaptive Feedback (3.3)	Wei-Zen (3.4)	Adaptive Stream (3.5)	Adaptive Lip (3.6)
Sincronizări oportune	Intra-media	Da	Da	Da	Da	Da	Da
	Inter-media	Da	Da	Da	Da	Da	Da
	Condicionali	Nu	Nu	Nu	Nu	Nu	Nu
	Multiplex	Nu	Nu	Nu	Da	Nu	Nu
Configurații oportune	Unicast	Da	Da	Da	Nu	Nu	Da
	Multicast	Da	Da	Da	Nu	Nu	Nu
	De rețea	Da	Da	Nu	Nu	Nu	Nu
	De grup	Da	Da	Nu	Da	Da	Nu
Sincronizări intra-media	Lip sync (lip-roll)	Da	Da	Da	Da	Da	Da
	Spoken sync (artificial)	Nu	Nu	Nu	Da	Nu	Nu
	Jumpile temporale	Da	Nu	Nu	Da	Da	Da
Strategii intra-media	Canale suplimentare	Nu	Nu	Da	Da	Nu	Da
	Multiplexare	Nu	Nu	Nu	Nu	Nu	Da (MPEG)
Surse de asincronism corectate	Interruperea pierzului	Da	Da	Da	Da	Da	Da
	Deriva ceasurilor locale	Nu	Da	Da	Da	Nu	Nu
	Diferența timpilor de schimbare	Da	Nu	Nu	Da	Da	Nu
	Diferența timpilor de rețea	Da	Nu	(pentru Master/Slave)	Da	Da	Nu
	Ceas global	Da	Nu	Nu	Nu	Da	Nu
	Ceas local sincronizat	Nu	Nu (pasiv)	Nu	Da	Nu	Nu
Canal de rețea (feedback)		Nu	Nu	Da	Nu	Nu	Da
Regimul de grup		Da	Nu	Nu	Da	Nu	Nu
Tine cont de diferite reguli de sincronizare (flexibilitate unitară de aplicare, cerințe utilizare etc.)		Da	Nu	Da	Nu	Da	Da
Domani de utilizare		Aplicații multimedia distribuite	Conținut multimedia (tele-orchestra)	VOD	Tele-orchestra Multimedia la cerere	Aplicații multimedia distribuite	VOD (MPEG-1)
Observații		(1)	(2)	(3)	(4)	(5)	(6)

Tabel 2

- (1) ceasurilor ; flexibilitate în menținerea în sincronism a grupurilor multiple;
- (2) presupune că fiecare sursă expediază propriile pachete la o frecvență constantă ;
- (3) 3 tipuri de resincronizări – agresivă, probabilistică, conservatoare ; sincronizare inter-media cu feedback ;
- (4) ceas global virtual (VGT) ;
- (5) presupune sincronizarea ceasurilor ;
- (6) sincronizarea fluxurilor se face la nivelul server-ului

5. Concluzii

Ca o concluzie generală, se poate afirma că algoritmi sunt adaptivi, putând fi modificați ca urmare a schimbărilor condițiilor de mediu. Sunt capabili de sincronizare pentru grupuri independente și au o anumită flexibilitate relativ la aplicația sau cerințele utilizatorului. În fine, mare parte dintre aceștia prezintă interes doar pentru LAN, proprietățile fiind diminuate în cazul WAN. Pe baza rezultatelor sintetice din tabelul 2 se poate opta, în funcție de natura și complexitatea aplicației, pentru soluția cea mai convenabilă.

Bibliografie

1. **AGARWAL, N., S.SANG:** Synchronization of Distributed Multimedia Data in an Application-Specific Manner. The 2nd International Conference on Multimedia, San Francisco, USA, 1994, pp. 141-148.
2. **ANDERSON, D.P., G. HOMSY:** Synchronization Policies and Mechanisms in a Continuous Media I/O Server. În: Report no. UCB/CSD 91/617, Berkeley, 1991.
3. **CHEN, H.S.A., L. QIAO, K. NAHRSTEDT:** Adaptive versus Reservation-Based Synchronization Protocols-Analysis and Comparison. În: Multimedia Tools and Applications, 2001, pp. 219-257.
4. **CAMPELL, A., G. COULSON, F. GARCIA, D. HUTCHISON:** A Continuous Media Transport and Orchestration Service. În: Proc. of SIGCOMM'92, Conference on Communications Architectures and Protocols, 1992, pp. 99-110.
5. **ESCOBAR, J., D. DEUTSCH, C. PARTRIDGE:** Flow Synchronization Protocol. În: IEEE Global Communication Conferences, 1992, pp. 1381-1387.
6. **NAHRSTEDT, K., L. QIAO:** Lip Synchronization within an Adaptive VOD System. În: International Conference on Multimedia Computing and Networking, San Jose, California, 1997.
7. **PRONIOS, N. B., T. BOZIOS:** A Scheme for Multimedia and Hypermedia Synchronization Multimedia Retrieval over Integrated Networks. În: COST 237, vol. 882, 1994, pp. 340-345.
8. **RAMANATHAN, S., P.V. RANGAN:** Adaptive Feedback Techniques for Synchronization Multimedia Retrieval over Integrated Networks. În: IEEE/ACM Transaction on Networking, Vol. 1, No. 2, April 1993, pp. 246-260.
9. **ROTHERMEL, K., T. HELBIG:** An Adaptive Stream Synchronization Protocol. În: Proc. Of the IEEE, Vol. 5, No. 5, 1997, pp. 324-336.
10. **RANGAN, P.V., S. RAMANATHAN, T. KAPPNER:** Performance of Inter-Media Synchronization in Distributed and Heterogeneous Multimedia Systems. În: Computer Networks and ISDN Systems, 1993.
11. **RANGAN, P.V., M.V. HARRICK, S. RAMANATHAN:** Communication Architectures and Algorithms for Media Mixing in Multimedia Conferences. În: IEEE/ACM Transaction on Networking, Vol. 1, No. 1, 1993, pp. 20-30.