

Programul: **RESURSE UMANE**
Tipul proiectului: **Proiecte de cercetare pentru tineri doctoranzi**
Cod proiect: **TD-371**

Sinteza lucrării

Controlul congestiei in protocoale de streaming

Cercetarile efectuate in cadrul etapei din anul 2008 au avut ca obiective:

1. Modelarea formala a problemei controlului congestiei pentru aplicatii de streaming multimedia.
2. Dezvoltarea, implementarea si testarea unui algoritm pentru controlul congestiei TCP-friendly si media-friendly.
3. Valorificarea rezultatelor. Scriere articole si finalizarea tezei.

Activitatile efectuate pentru atingerea obiectivului cu numarul 1 sunt cuprinse in cadrul articolelor [5],[7],[8], iar activitatile efectuate in cadrul obiectivului numarul 2 sunt sintetizate in articolele [5] si [6]. Obiectivul 3 a fost atins prin publicarea si acceptarea spre publicare a 4 articole la conferinte internationale si finalizarea tezei de doctorat. In cadrul obiectivului 1 s-a modelat formal controlul congestiei pentru streamuri multimedia, iar pe baza rezultatelor de la obiectivul 1, s-au dezvoltat si testat in cadrul obiectivului 2 mai multi algoritmi de controlul congestiei pentru aplicatii de streaming multimedia. In cele ce urmeaza, descriem pe scurt cateva rezultate obtinute.

Metodologie in doi pasi pentru algoritmi de controlul congestiei TCP-friendly si Media-friendly

AIMD (Additive Increase Multiplicative Decrease), algoritmul de control al congestiei specific TCP-ului, nu este potrivit pentru multimedia streaming [1] datorita debitului de date extrem de fluctuant al acestuia. Prin urmare alti algoritmi de control al congestiei care ofera un debit de date mai constant au fost dezvoltati, probabil unul dintre cei mai cunoscuti fiind TFRC [9]. Toti acesti algoritmi de control al congestiei au un debit de date mai constant pentru ca sunt mai putin agresivi decat TCP-ul in folosirea noii latimi de banda disponibile dar au de asemenea si un timp de raspuns mai mare la aparitia congestiei decat TCP-ul. Pentru ca ofera un debit de date mai constant stream-urile multimedia in special CBR (*Constant Bit Rate*) dar si cele VBR (*Variable Bit Rate*) pot fi mai bine adaptate unei latimi de banda limitate de catre serverele de streaming. Cu toate acestea, desi algoritmi de control al congestiei care ofera un debit de date constant imbunatatesc livrarea stream-urilor multimedia acestia nu reprezinta solutia optima pentru ca nu iau in considerare caracteristicile media ale stream-ului (nu sunt media-friendly).

Pe de alta parte, codec-urile VBR care respecta standardul video MPEG pot varia foarte mult bitrate-ul (rata de biti) stream-ului video la iesire in cazul schimbarii scenei pentru a pastra o calitate relativ constanta a semnalului video (e.g. bitrate-ul poate varia de 20 de ori

de la o secunda la alta) [10]. Cu alte cuvinte bitrate-ul unui astfel de stream nu este in nici un caz constant.

Daca bitrate-ul unui stream nu este constant si, in general, daca caracteristicile media nu sunt uniforme (relativ constante) de-a lungul stream-ului, probabil cea mai buna rata de transmisie pe care o poate obtine un algoritm de control al congestiei nu este neaparat una constanta, ci o rata a transmisiei care urmareste evolutia caracteristicilor media (e.g. bitrate-ul) de-a lungul stream-ului. Desigur aceasta rata a transmisiei trebuie sa respecte caracteristicile specifice ale retelei (sa aiba o forma TCP-friendly).

In cadrul primului obiectiv am dezvoltat urmatoarea tehnica pentru a obtine un algoritm de control al congestiei primar care este TCP-friendly si media-friendly:

- in primul rand construim un algoritm de control al congestiei bazat pe fereastra de congestie care este TCP-friendly, dar care are un debit de date mai constant;
- modelam acest debit de date constant in functie de caracteristicile media ale stream-ului (e.g. bitrate-ul).

Folosind aceasta metodologie vom dezvolta in sectiunea urmatoare doi astfel de algoritmi de control al congestiei: mSQRT si mLOG.

Algoritmi de control al congestiei primari pentru stream-uri multimedia

Folosim modelul de retea dezvoltat in [3] si [4] in care retea este vazuta ca un set de resurse sau legaturi care sunt partajate de un set de surse sau utilizatori. Scopul este sa impartim latimea de banda intre surse intr-un asemenea mod incat sa obtinem un optim social pentru toti utilizatorii care partajeaza retea.

Problema alocarii latimii de banda intre fluxuri se reduce la a gasi o solutie pentru urmatoarea problema de optimizare concava:

$$\begin{cases} \max_{x>0} \sum_{s \in S} U_s(x_s) & , x = (x_1, \dots, x_n) \\ & S = \{s_1, \dots, s_n\} \\ \text{subject to: } \sum_{s \in S(l)} x_s \leq c_l \quad \forall l \in L \end{cases} \quad (1)$$

In acest model retea este abstractizata ca un set de legaturi $l \in L$ unde fiecare legatura l are capacitatea c_l . Reteaua este partajata de sursele $s \in S$ si fiecare sursa s transmite date la o rata x_s . Cand sursa s transmite date la rata x_s obtine o utilitate $U_s(x_s)$ care se presupune a fi o functie concava nedescrescatoare de doua ori diferentiabila. De asemenea, avem $S(l)$ setul de surse care folosesc legatura $l \in L$ si $L(s)$ setul de legaturi care sunt folosite de sursa s .

Pentru a construi un algoritm de control al congestiei TCP-friendly si media-friendly trebuie, potrivit metodologiei prezentate anterior, sa construim un algoritm de control al congestiei primar care realizeaza un debit de date mai constant decat AIMD-ul TCP-ului. Pormim de la algoritmul AIMD folosit de TCP care poate fi exprimat ca [2]:

$$\begin{cases} I : W_{t+RTT} = W_t + \alpha & , \alpha = 1 \\ D : W_{t+\delta} = W_t - \beta W_t & , \beta = 1/2 \end{cases}$$

unde W_t este dimensiunea ferestrei de control al congestiei la timpul t , RTT este timpul dus-intors pentru respectiva conexiune si δ este lungimea unui inteval de timp redus care

contine un eveniment de congestie. I reprezinta operatia de crestere si D reprezinta operatia de descrestere. Evolutia fereastrei de congestie a algoritmului TCP poate fi descrisa prin urmatoarea ecuatie care combina cele doua operatii de mai sus:

$$W_{t+\delta} = W_t + \frac{\alpha}{W_t}(1 - q_t) - \beta W_t q_t \quad (2)$$

unde presupunem ca δ este mult mai mic decat RTT si q_t este probabilitatea ca un pachet sa fie eliminat in intervalul de timp $[t, t + \delta]$. Daca inmultim aceasta ecuatie cu $1/RTT$ si consideram ca debitul de date la timpul t este $x_t \approx W_t/RTT$ obtinem:

$$x_{t+\delta} = x_t + \frac{\alpha}{RTT W_t}(1 - q_t) - \beta x_t q_t$$

Apoi considerand δ mic si impartind ambele laturi ale ecuatiei prin δ obtinem urmatoarea evolutie a debitului de date:

$$\dot{x}(t) = \frac{\alpha}{\delta RTT^2 x(t)}(1 - q(t)) - \beta x(t)q(t)$$

Considerand δ egal cu unitatea de timp si dupa eliminarea termenului RTT^2 considerat constant si prin urmare putand fi incorporat in constanta α , interpretam $q(t)$ ca fiind costul aplicat de retea pentru utilizarea a $x(t)$ din resursele sale [3] (in cazul in care congestia nu este prezenta $q(t) = 0$) si obtinem forma finala a evolutiei debitului de date pentru AIMD:

$$\dot{x}(t) = \frac{\alpha}{x(t)} - \beta x(t)q(t) \quad (3)$$

In continuare trebuie sa uniformizam evolutia debitului de date specifica TCP-ului data de ecuatie (3) dar in acelasi timp sa o pastram TCP-friendly. In general cautam sa obtinem un algoritm de control al congestiei primar avand urmatoarea forma:

$$\dot{x}(t) = \frac{\alpha}{I(x(t))} - \beta D(x(t))q(t) \quad (4)$$

unde $I(x)$ este factorul de crestere si $D(x)$ este factorul de descrestere si care are o evolutie a debitului de date mai uniforma decat cea a algoritmului de control al congestiei descris in (3).

Algoritmul de control al congestiei mSQRT

Algoritmul de control al congestiei mSQRT este un algoritm de control al congestiei prezentat in [2] TCP-friendly si cu un debit de date relativ stabil avand urmatoarea forma:

$$\dot{x}(t) = \frac{\alpha}{\sqrt{x(t)}} - \beta \sqrt{x(t)}q(t)$$

Algoritmul mSQRT este mai uniform decat TCP pentru ca creste debitul de date mai putin agresiv decat acesta (i.e. $\frac{\alpha}{\sqrt{x(t)}}$), dar il si descreste mai putin drastic decat TCP-ul (i.e. $\beta \sqrt{x(t)}$).

In continuare trebuie sa modifcam debitul de date al algoritmului mSQRT pentru a tine cont de caracteristicile media ale stream-ului. Vom folosi aici doar bitrate-ul si valoarea buffer-ului la client, dar pot fi folosite si alte caracteristici media: media bitrate-ului pe scene,

metrici de calitate, valori PSNR etc. Versiunea finala a mSQRT, care este TCP-friendly dar de asemenea si media-friendly, formulata ca un algoritm de control al congestiei primar este:

$$\begin{cases} \dot{x}_s(t) = m_s(t) * \frac{\alpha}{\sqrt{x_s(t)}} - \beta \sqrt{x_s(t)} q_s(t) \\ \text{unde } m_s(t) = \frac{b(t)}{b_{avg}} \left(1 + \frac{1}{\Delta(t)}\right) \end{cases} \quad (5)$$

$$q_s(t) = \sum_{l \in L(s)} p_l \left(\sum_{r: r \in S(l)} x_r(t) \right) \quad (6)$$

unde $x_s(t)$ este debitul de date al sursei s , $b(t)$ este bitrate-ul masurat in bytes pentru o secunda a stream-ului curent, b_{avg} este bitrate-ul mediu pentru intregul stream si $q_s(t)$ este dat in forma din [3] ca fiind costul total de utilizare a resurselor retelei din $L(s)$; $p_l(y)$ este o functie continua, crescatoare si ne-negativa a y , reprezentand costul utilizarii resursei l . Daca $y < c_l$ atunci $p_l(y) = 0$. In acest fel algoritmul mSQRT creste debitul de date atunci cand nu este congestie in functie de necesitatile de bitrate ale stream-ului si de asemenea creste debitul de date cand buffer-ul $\Delta(t)$ este prea mic pentru a evita golirea buffer-ului la client. Functia $m_s(t)$ cuantifica utilitatea sau necesarul de latime de banda al aplicatiei de streaming.

Se poate demonstra ca algoritmul de control al congestiei descris in (5) este stabil si urmatoarea propozitie demonstreaza aceasta afirmatie.

Propozitia 1. Functia strict concava

$$V_1(x) = \sum_{s \in S} \alpha m_s(t) \log(x_s(t)) - \beta \sum_{l \in L} \int_0^{\sum_{r: r \in S(l)} x_r(t)} p_l(y) dy$$

este o functie Lyapunov pentru sistemul dinamic (5)-(6), deci sistemul este stabil in jurul echilibrului.

Algoritmul de control al congestiei mLOG

Algoritmul de control al congestiei mLOG este urmatorul algoritm de control al congestiei uniform si TCP-friendly:

$$\dot{x}(t) = \frac{\alpha}{\log(x(t))} - \beta \log(x(t)) q(t)$$

Algoritmul mLOG este mai uniform decat TCP pentru ca creste debitul de date mai putin agresiv decat acesta (i.e. $\frac{\alpha}{\log(x(t))}$), dar il si descreste mai putin drastic decat TCP-ul (i.e. $\beta \log(x(t))$). Luand in considerare caracteristicile media ale stream-ului cum am facut si pentru mSQRT obtinem forma primara a algoritmului de control al congestiei mLOG:

$$\begin{cases} \dot{x}_s(t) = m_s(t) * \frac{\alpha}{\log(x_s(t))} - \beta \log(x_s(t)) q_s(t) \\ \text{unde } m_s(t) = \frac{b(t)}{b_{avg}} \left(1 + \frac{1}{\Delta(t)}\right) \end{cases} \quad (7)$$

$$q_s(t) = \sum_{l \in L(s)} p_l \left(\sum_{r: r \in S(l)} x_r(t) \right) \quad (8)$$

Propozitia 2. Functia strict concava

$$V_2(x) = \sum_{s \in S} \alpha m_s(t) \left(li(x_s(t)) - \frac{x_s(t)}{\log(x_s(t))} \right) - \beta \sum_{l \in L} \int_0^{\sum_{r: r \in S(l)} x_r(t)} p_l(y) dy$$

unde $li(x_s(t))$ este integrala logaritmica ¹ si este o functie Lyapunov pentru sistemul dinamic (7)-(8), deci sistemul este stabil in jurul echilibrului.

Detalii ale implementarii si probleme legate de echitabilitate

Primul pas in obtinerea unei implementari practice pentru mLOG si mSQRT este discretizarea ecuatiilor diferentiale (5) si (7). Obtinem urmatoarele versiuni discrete ale ecuatiilor:

$$\begin{aligned} mSQRT : x[t+1] &= x[t] + m(t) * \frac{\alpha}{\sqrt{x(t)}} - \beta \sqrt{x(t)} q(t) \\ mLOG : x[t+1] &= x[t] + m(t) * \frac{\alpha}{\log(x(t))} - \beta \log(x(t)) q(t) \\ unde \quad m(t) &= \frac{b(t)}{b_{avg}} \left(1 + \frac{1}{\Delta(t)} \right) \end{aligned}$$

Dupa ce consideram $q(t)$ ca fiind probabilitatea pierderii de pachete in intervalul de timp $[t, t+1]$ obtinem urmatoarele caracterizari ale ferestrei de congestie pentru mLOG si mSQRT pe baza carora se poate realiza o implementare la nivel de pachet:

mSQRT:

$$\begin{cases} I : W_{t+RTT} = W_t + m(t) * \frac{\alpha}{sqr(t(W_t))} \\ D : W_{t+\delta} = W_t - \beta sqr(t(W_t)) \end{cases}$$

mLOG:

$$\begin{cases} I : W_{t+RTT} = W_t + m(t) * \frac{\alpha}{\log(W_t)} \\ D : W_{t+\delta} = W_t - \beta \log(W_t) \end{cases}$$

Folosind caracterizarile ferestrei de congestie prezentate mai sus putem specifica regulile de crestere pentru mSQRT si mLOG care sunt similare celor folosite de AIMD in cadrul TCP-ului. Pentru fiecare pachet confirmat mSQRT incrementeaza fereastra de congestie cu $m(t) * \frac{\alpha}{W_t sqr(t(W_t))}$ iar dupa detectarea congestiei decrementeaza fereastra de congestie cu $\beta sqr(t(W_t))$. Pentru mLOG fiecare pachet confirmat duce la o incrementare a ferestrei de congestie cu $m(t) * \frac{\alpha}{W_t \log(W_t)}$ iar detectarea congestiei duce la o decrementare a ferestrei cu decrements it by $\beta \log(W_t)$.

References

- [1] V. Jacobson, *Congestion avoidance and control*, ACM Comput. Commun. Rev., vol. 18, pp. 314-329, 1988.
- [2] D. Bansal, H. Balakrishnan, *Binomial Congestion Control Algorithms*, IEEE Infocom 2001.

¹ $li(x) = \int_0^x \frac{dt}{\log(t)}$

- [3] F. P. Kelly, A. K. Maulloo, and D. K. H. Tan, *Rate control for communication networks: Shadow prices, proportional fairness and stability*, J. Oper. Res. Soc., vol. 49, no. 3, pp. 237-252, Mar. 1998.
- [4] S. H. Low and D. E. Lapsley, *Optimization flow control I: Basic algorithm and convergence*, IEEE/ACM Transactions on Networking, vol. 7, pp. 861-874, Dec. 1999.
- [5] A. Sterca, *Primal Congestion Control Algorithms for Multimedia Streams*, Proc. of 10th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, Timisoara, septembrie 2008 (ISI Proceedings), in curs de publicare.
- [6] A. Sterca, *Window-based Congestion Control Algorithms for Multimedia Streams*, 6th International Conference on Applied Mathematics, Baia-Mare, septembrie 2008, in curs de publicare.
- [7] A. Sterca, *Optimized adaptation of video streams in streaming servers*, Proc. of 7th Int. RoEduNet Conf., Cluj-Napoca, August 2008, pp. 113-116.
- [8] A. Sterca, *An Adaptation Model for Video Streaming Servers in Best-Effort Networks*, 7th Int. Joint Conference on Mathematics and Computer Science, book of abstracts, Cluj-Napoca, iulie 2008, pp.74.
- [9] S. Floyd, M. Handley, J. Padhye, J. Widmer, *Equation-Based Congestion Control for Unicast Applications*, ACM SIGCOMM 2000.
- [10] F. Pereira, T. Ebrahimi, *The MPEG-4 Book*, Prentice Hall PTR, ISBN 0130616214, 9780130616210, 2002.