

# **Alegerea algoritmilor de congestie in functie de cunostintele despre starea retelei**

Grupa 442A

Dobre Marius

Horsa Alexandru

Toma Crina

Uta Cosmin Alexandru

## Table of Contents

Introducere.....	3
Misiune, Metrica si Modele.....	4
The Box is Black .....	6
AIMD-FC .....	6
Mecanisme Binomiale.....	7
SIMD.....	8
Controlul congestiei bazat pe ecuatii.....	9
GAIMD.....	9
RAP.....	9
Ideally Scalable Congestion Control (ISCC).....	10
The box is grey.....	10
TCP – Vegas .....	10
TCP-REAL .....	11
TCP-WESTWOOD .....	11
TFRC .....	11
TCP-JERSEY .....	13
The Box Is Green.....	13
Mecanismul Bimodal.....	13
Controlul congestiei retelei asistate.....	14
Discutii.....	15
Erori.....	17
Bibliografie .....	18
Referinte.....	18

# Introducere

## Toma Crina

O rețea este considerată congestionată când prea multe pachete accesează bufferul aceluiași router rezultând o cantitate de pachete care va fi respinsă. În această stare de încărcare se depășește capacitatea rețelei. În timpul congestiei acțiunile trebuie preluate de ambele protocoale de transmisie și ruterele de rețea pentru a evita colapsul și în plus, pentru a asigura stabilitatea rețelei, eficientizarea alocării resurselor pentru utilizatorii rețelei. Într-adevăr, în timpul colapsului doar o fracțiune din lățimea de bandă este utilizată de trafic în final ajunând la receptor.

Congestia este considerată, în general, ca un eveniment catastrofic. Cu toate acestea, congestia, în sine, este asociată cu diferite proprietăți în funcție de caracteristicile rețelelor care stau la baza, mecanismele de transmisie a protocoalelor, caracteristici de capacitatea fluxurilor, nivel de debit susținut și funcționalitatea routerelor de rețea. Prin urmare, impactul congestiei poate fi temporar și ușor controlabil, sau poate fi catastrofal. Luată în considerare, o rețea de mare viteză care găzduiește un număr de fluxuri concurente care încremențesc sau decrementează. Fereastra fiecărui flux incrementează sau decrementează, de asemenea. Cu toate acestea, spre deosebire de rețelele tradiționale, timpul necesar fluxurilor pentru a exploata lățimea de bandă disponibilă este cu siguranță mai lung; cantitatea de pierdere la congestie este mai mare; iar durata de congestie, în sine, de-a lungul timpului total de comunicație poate fi relativă mai mică. Având în vedere că natura congestiei nu poate fi prescrisă sau, chiar definită cu acuratețe, controlul congestiei devine o sarcină complexă. În plus, complexitatea crește din cauza complexității multiple a algoritmilor de control a congestiei. Ei au nevoie să controleze congestiile și să evite colapsurile, maximizând lățimea benzii de utilizare, garantează stabilitatea rețelei, și de a asigura resurse echitabile de alocare.

Considerând rețeaua ca o cutie neagră care oferă doar o reacție binară pentru fluxurile rețelei deasupra congestiilor această schimbă tot fluxul către utilizatorii finali și cere soluții mai generice și, probabil mai puțin selective. Asta este, indiferent de particularitatea de rețea sau de starea rețelei curente, algoritmul va reacționa în mod similar în toate cazurile. Acest lucru nu este ciudat.

Scopul fiecărui expeditor este să funcționeze independent, dar cu toate acestea, pentru a ajusta rata (sau fereastra) într-o manieră în care lățimea totală a benzii va consuma eficient și echitabil.

Din perspectiva algoritmică problema de mai sus este o provocare deoarece entitățile distribuite (surse) nu au priorități și cunoștințe de starea entității; nici măcar nu știu capacitatea sistemului și numărul de concurenți. Prin urmare, obiectivul de echitate și eficiență inițial pare dificil de atins. Cu toate acestea, în cazul în care sistemul are dreptul la o comportament prescris și entitățile agreează asupra unei tactici de transmisie comune converge ca echitatea să devină admisibilă [32]. AIMD, controlul tradițional congestiv, algoritmic al Internetului care funcționează în cadrul sistemului de aplicare: crește treptat rata de trimitere (după valoarea alfa) în timp ce sistemul ajunge la saturatie. La congestie toți expeditorii scad rata multiplicativ folosind rata de descreștere beta. Pe de altă parte, se pot măsura condițiile de rețea estimând disponibilitatea

latimii de banda sau chiar sustine debitul si obtine informatii despre retea. Cu toate acestea, masuratorile sunt luate la intervale de timp care nu reprezinta neaparat o retea activa sau sa nu corespunde conditiilor totale.

In plus, unele intrebari generice nu pot fi abordate :Cat de des trebuie masurata retea?, Cat de multa incredere putem avea in masuratorile noastre?, Cat de selectiva poate fi strategia de redobandire?,Cum vom asocia masuratorile instantanee ale congestiei cu incarcarea retelei asupra unei lungi dar si suficiente perioade de timp? Acestea fiind spuse, retea nu poate fi o cutie neagra, dar nu este cu siguranta mai buna decat cea gri, care implica un risc ocazional considerabil. Se poate merge dincolo de algoritmi orbi sau de riscul ridicat de estimare cerand retelei 2 ajutor. Bineinteles, precizia implica un anumit cost.

Pe langa problema practica de colaborare se adauga si aceea de a convinge oamenii pentru a adauga functionalitate (si a investi bani) in retele lor, problema strategiilor de recuperare ramane.

## **Misiune, Metrica si Modele**

În contextul muncii noastre, vom defini următoarele unități de măsurare: Fereastra este un mecanism în stratul de transport care limitează numărul de pachete puse în rețea și descrie rata de pachete pe secundă sau biți pe secundă.

Un ciclu este faza dintre doua reactii periodice Prin urmare, un ciclu contine un pas de descrestere activat de congestie si un numar de pasi aditivi crescator.Sistemul este intr-o stare de echilibru atunci cand resursele gratuite utilizate de catre fluxuri sunt echilibrate.

Misiunea stabilita in procesul de evaluare de evitare a congestiei respectiv de algoritim de control sunt:

- Atingerea latimii de banda mare.
- Pentru a converge mai rapid,.
- Pentru a minimiza lungimea oscilatiilor
- Pentru a mentine capacitatea de reactie mare
- Pentru a fi compatibil cu traditionalele protocoale AIMD

## Sistemul de control sincronizat pentru m utilizatori comuni dintr-o retea

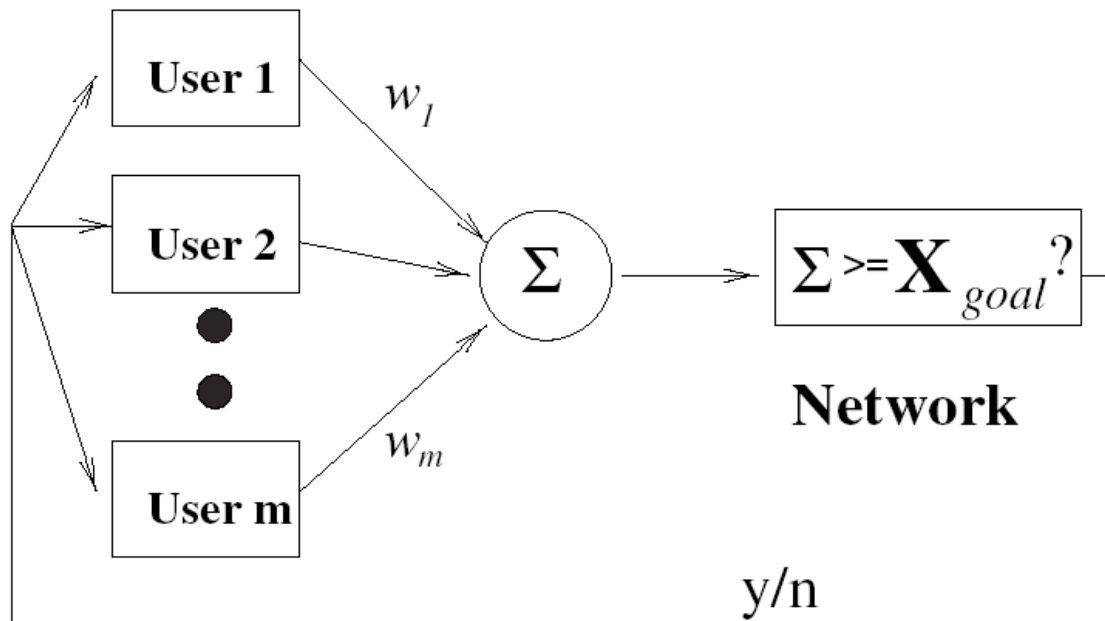


Figura 1

Masuratorile noastre pentru performanta sunt dupa cum urmeaza:

Eficienta este media tranzitata pe fluxurile de pas (sau pe RTT) atunci cand sistemul este in echilibru.

Corectitudinea caracterizeaza distributia corecta a resurselor dintre fluxuri. O metrice bine-cunoscuta este [7]:

Viteza de convergenta- descrie timpul care a trecut de la starea de echilibru.

Netezimea este reflectata de amplitudinea oscilatiilor in timpul scaderii multiplicative. Aceasta depinde de marimea oscilatiilor. Capacitatea de raspuns este masurata de numarul de pasi (RTT) pentru a ajunge la echilibru.

Un model de sistem de sincron al reactiei este aratat in figura 1. In controlul congestiei schimbarea multimii este un eveniment ce a avut loc. Evenimentul este o reactie binara. Modelul sincron este caracterizat de o generatie sincronizata de raspunsuri in congruenta cu [7]. Sistemul de raspuns este 1 atunci cand latimea de banda este disponibila si 0 atunci cand latimea de banda este epuizata. Instructiunile entitatilor (surselor) sistemului este de a creste sau a descreste rata de date. Retineti ca, in retelele reale, comportamentul receptiv al sistemului nu este administrat de catre orice autoritate centralizata cu informatii aditionale a retelei dinamice- este doar un numar redus de pachete din cauza congestionarii care in mod natural se intampla atunci cand latimea de banda este depasita. Sistemul are  $m$  utilizatori (fluxuri) si rata de transfer instantanee pentru fluxul  $W_i$ . Limitarile sistemului sunt derivate din dinamica pachetelor de retea:

- Latimea de banda  $B$  este limitata.
- Fluxurile nu sunt in legatura cu rata de transfer a alor fluxuri (dimensiunile ferestrei).
- Fluxurile nu sunt in legatura cu numarul competitorilor de pe canal.
- Niciun flux nu este in legatura cu marimea latimii de banda  $B$ .

Deși modelul sincron este adoptat pe scară largă, el este asociat cu un număr de ipoteze sau/si simplificări care nu pot fi detinute de rețelele reale.

## **The Box is Black**

(Starea rețelei este necunoscută)

### **Horsa Alexandru**

Algoritmul „Additive Increase Multiplicative Decrease” este folosit pentru ajustarea ferestrei TCP. Pe baza analizei făcută de Chiu și Jain, algoritmul atinge stabilitatea și converge la corectitudine în situațiile în care cererile fluxurilor concurente depășesc lățimea de bandă a canalului [7].

Controlul congestiei la TCP-ul tradițional se bazează pe ideea de bază a algoritmului AIMD. În cazul TCP-Tahoe și TCP-Reno creșterea aditivă a fazei coincide cu cea de la AIMD, când protocoalele sunt în faza de evitare a coliziunilor. În cazul refuzării unui pachet, în locul unei descreșteri multiplicative se aplică o tactică mai conservativă în TCP-Tahoe. Fereastra congestiei se resetează și protocolul intră din nou în faza „slow-start”. Pe de altă parte, la TCP-Reno, când emitorul primește 3 „DACKs” (Duplicate Acknowledgments), este utilizată o descreștere multiplicative în pragul ferestrei cât și în pragul „slow-start”. În acest caz, protocolul rămâne în faza de evitarea a coliziunilor. La expirarea timpului de „timeout”, TCP-Reno intră în faza „slow-start” ca și TCP-Tahoe.

### ***AIMD-FC***

O îmbunătățire recentă adusă algoritmului AIMD este „Additive Increase Multiplicative Decrease with Fast Convergence” (AIMD-FC) [31]. Această îmbunătățire are impact asupra eficienței și a corectitudinii. Optimizarea are loc asupra procedurii de convergență a algoritmului AIMD ce oferă un timp mai scurt de convergență și o eficiență sporită. AIMD-FC crește lățimea de bandă față de AIMD de la  $\frac{3}{4}$  la  $\frac{5}{6}$ .

Observațiile de la care au plecat autorii algoritmului sunt:

- În timpul fazei de creștere aditivă, resursele sistemului sunt alocate în mod egal fluxurilor. Dimensiunea 'k' este cunoscută fiecărui flux din sistem.
- AIMD afectează atât fereastra inițială, cât și cantitatea resurselor sistemului „k”, ce vor fi alocate în mod echitabil, în timpul fazei descreșterii multiplicative. Manipularea ferestrelor inițiale necunoscute este scopul real pentru a atinge corectitudinea.
- Distanța dintre linia care delimitează lățimea de bandă și linia eficienței, când sistemul se află la echilibru depinde numai de factorul de descreștere multiplicative [7].
- Doi algoritmi ar avea nevoie de același număr de cicluri pentru a converge la corectitudine: spre exemplu, două variante de AIMD cu rate diferite de creștere aditivă dar același raport de descreștere multiplicative. Numărul pașilor determină eficiența relativă a algoritmului de a converge la corectitudine.

Practic, corectitudinea este atinsa in AIMD-FC prin eliberarea (cu ajutorul ajustarilor ferestrelor) resurselor initiale a fluxurilor, deoarece in faza de crestere aditiva fluxurile isi maresc consumul de resurse uniform. Astfel, devine vizibil ca diferenta dintre AIMD si AIMD-FC se concentreaza pe portiunea ferestrei de congestie care este afectata de descresterea multiplicativa (denumita „decrease window”).

In plus, exista cateva probleme legate de AIMD-FC:

- Eficienta limitelor algoritmului AIMD nu a fost exploatata.
- Alte modificari pot fi facute pentru a favoriza capacitatea de reactie sau finetea.

Aceeiasi autori au imbunatatit performanta [31], finetea si corectitudinea algoritmului AIMD-FC propunand un nou algoritm denumit AIMD-FC+.

### ***Mecanisme Binomiale***

Bansal si Balakrishnan au prezentat in [5] o noua clasa de algoritmi neliniari de control al congestiei denumiti Algoritmi Binomiali de Control al Congestiei. Acesti algoritmi contin denumirea de binomiali deoarece controlul se bazeaza pe implicarea a doi termeni algebrici traditionali cu exponenti diferiti.

In timp ce algoritmul de control AIMD poate fi exprimat sub forma:

$$\begin{aligned} \text{Increase} : W_{t+R} &\leftarrow W_t + a; a > 0 \\ \text{Decrease} : W_{t+\delta t} &\leftarrow (1 - \beta)W_t; 0 < \beta < 1 \end{aligned}$$

Autorii au generalizat regulile AIMD astfel:

$$\begin{aligned} \text{Increase} : W_{t+R} &\leftarrow W_t + \frac{a}{W_t^k}; a > 0 \\ \text{Decrease} : W_{t+\delta t} &\leftarrow W_t - \beta W_t^l; 0 < \beta < 1 \end{aligned}$$

unde I se refera la cresterea in fereastra ca rezultat al primirii unei ferestre de acknowledgments in timpul unui singur mesaj dus-intors (RTT – Round-Trip Time), iar D se refera la descresterea in fereastra dupa detectarea congestiei de catre emitator,  $W_t$  - dimensiunea ferestrei la momentul t, R – timpul de comunicare dus-intors al fluxului (Round-Trip Time), si a, b, k, l constante.

Spre exemplu, pentru  $k=0, l=1$ , se obtine AIMD. Autorii au propus, in spatiul (k,l), doua variatii ale AIMD (compatibile cu TCP). Algoritmii IIAD (avand  $k=1$  si  $l=0$ ) si SQRT ( $k=1/2$  si  $l=1/2$ ). Primul se numeste „Inverse Increase Additive Decrease” (IIAD) deoarece regula sa de crestere este invers proportionala cu fereastra curenta. Cel de-al doilea se numeste SQRT deoarece atat cresterea, cat si descresterea sa sunt invers proportionale cu radacina patrata a ferestrei curente.

Concluziile autorilor din [5]:

- Un algoritm binomial este compatibil TCP daca si numai daca  $k+l=1$  si  $l \leq 1$  pentru valori a si b convenabile.

- Acesti algoritmi pot concura incorect printr-un gateway „drop-tail” (refuza pachetele de la intrare in coada). Utilizarea unei scheme de management activ al cozii „RED” (Random Early Detection) amelioreaza aceasta problema.
- AIMD este agresiv in cautarea latimii de banda disponibila pentru a si b cunoscute si este cel mai eficient si mai potrivit algoritm binomial pentru aplicatiile care pot tolera o mare ajustare a latimii de banda.
- Datorita faptului ca algoritmi binomiali sunt neliniari, pentru doua fluxuri date cu ferestrele initiale  $x_1$  si  $x_2$ , cea mai mica valoare din cele doua creste mai mult. Acest lucru duce la o corectitudine mai buna in alocarea resurselor.
- Parametrul „k” reprezinta agresivitatea cautarii, iar „l” reprezinta conservativitatea raspunsului la congestie. Astfel, exista un compromis intre „k” si „l” pentru ca un protocol binomial sa ajunga la o anumita valoare de transfer (la o rata de pierderi data).

## ***SIMD***

Un alt algoritm nelinier de control al congestiei este SIMD [24]. SIMD este primul algoritm de control al congestiei care utilizeaza informatii din trecut.

Regulile de control ale algoritmului sunt definite astfel:

$$\begin{aligned} \text{Increase} : W_{t+R} &\leftarrow W_t + a\sqrt{W_t - W_0}; a > 0 \\ \text{Decrease} : W_{t+\delta t} &\leftarrow W_t - \beta W_t; 0 < \beta < 1 \end{aligned}$$

unde  $W_0$  este dimensiunea ferestrei dupa ultima descrestere si  $W_t$  este aproximarea continua a dimensiunii ferestrei la momentul  $t$  (in Round Trip Time) din momentul in care dimensiunea ferestrei a inceput sa creasca. Autorii arata ca:

$$w(t) = w_0 + a^2 \frac{t^2}{4}$$

Acesta foloseste descresteri multiplicative ca AIMD, dar SIMD foloseste o regula de crestere diferita fata de cele folosite de AIMD si algoritmi binomiali. Regula se bazeaza pe informatiile din trecut.

In cazul in care doua fluxuri SIMD sunt in competitie, fluxul cu cea mai mica dimensiune a ferestrei este mai agresiv datorita naturii neliniare a algoritmului. Astfel se obtine o convergenta mai buna. Autorii arata in [24] ca SIMD converge mai rapid decat algoritmi fara memorie, AIMD si cei binomiali.



## ***Controlul congestiei bazat pe ecuatii***

### ***GAIMD***

General AIMD Congestion Control (GAIMD) este un protocol parametrizat, apropiat de TCP, care generalizeaza controlul congestiei AIMD prin parametrizarea valorii aditive de crestere „a” si raportul multiplicativ de descrestere „b”. Autorii lucrarii [58] au extins transferul ecuatiei standard-ului TCP, propus in lucrarea [42], pentru a include parametrii  $\alpha$ ,  $\beta$ :

$$T_{a,b}(p, RTT, T_0, b) = \frac{1}{RTT \sqrt{\frac{2b(1-\beta)}{\alpha(1+\beta)}} p + T_0 \min(1, 3\sqrt{\frac{(1-\beta^2)b}{2\alpha}} p) p (1 + 32p^2)}$$

unde  $p$  este rata pierderilor;  $T_0$  este timpul intre retransmisii; „b” este numarul de pachete recunoscute („acknowledged”) de fiecare ACK. Debitul total apartinand protocoalelor TCP-friendly ( $a$ ,  $b$ ) este limitat de catre debitul mediu al protocolului TCP standard ( $a=1$ ,  $b=0.5$ ), ceea ce inseamna ca ecuatia (11), care este derivata din (10) ar putea juca rol de ghid pentru a obtine apropierea.

$$T_{a,b}(p, RTT, T_0, b) = T_{1,0.5}(p, RTT, T_0, b) \quad (10)$$

Autorii lucrarii [58] deriva din (1) si (2) o relatie simpla pentru a si b:

$$a = \frac{4(1 - b^2)}{3} \quad (11)$$

In urma experimentelor, acestia au propus  $b=7/8$  ca valoare adecvata pentru reducerea ferestrei. Pentru  $b=7/8$ , relatia (3) ofera o valoare  $a=0.31$ .

### ***RAP***

Rate Adaptation Protocol (RAP) [48] este un protocol de transport bazat pe rata de transfer, apropiat de TCP. Acesta foloseste algoritmul cu descresteri multiplicative si cresteri aditive. Decupleaza controlul congestiei retelei de la nivelul fiabilitatii aplicatiei. Cu toate acestea, in comparatie cu TFRC, finetea nu este un criteriu in design-ul RAP.

## ***Ideally Scalable Congestion Control (ISCC)***

Metoda ISCC se bazează pe ideea scalabilității ideale. O schemă are scalabilitate ideală, dacă  $S_n$  este constant pentru toate fluxurile. Parametrul  $S_n$  precizează cât de repede crește pierderea pachetelor când mai multe fluxuri partajează o legătură comună și se referă în mod direct la abilitatea schemei de a suporta un număr mare de fluxuri.

### **The box is grey**

(Algoritmii se bazează pe estimări și măsurători)

Uta Cosmin Alexandru

Standardul TCP se bazează pe pierderi de pachete pe baza de estimări și măsurători, sau un semnal de de la link-uri supraîncărcate. Cu toate acestea, pierderea de pachete nu este un indiciu suficient de congestie, în sine, pentru un număr de motive:

- pierderea de pachete poate fi cauzată de corupția biților aleatorii atunci când lățimea de bandă este încă disponibilă.
- confirmarea pe bază de detectare a pierderilor de la marginea expeditorului poate fi afectată de „cross-traffic” pe calea inversă.
- pierderi de pachete, ca un feedback binar, nu pot indica nivelul de conținut înainte de apariția congestiei.

Prin urmare, o tactică eficientă de ajustare a fereastră ar trebui să se adapteze la rețeaua de diverse condiții de muncă, care nu pot fi semnalate pur și simplu sub formă de „pachet drops”. Mai multe protocoale de măsură bazate pe transport se bazează pe informații precise privind condițiile de rețea.

### ***TCP – Vegas***

Un bine cunoscut conceput de măsură bazat pe mecanismul de evitare a congestiei este TCP Vegas [6]. Vegas definește BaseRTT să fie minim de toate RTTs măsurate și ExpectedRate care urmează să fie raportul a ferestrei de congestie BaseRTT. Expeditorul măsoară ActualRate bazat pe esanționul RTTs. În cazul în care diferența între ExpectedRate și ActualRate este mai mică decât o limită inferioară, fereastra de congestie crește liniar în cursul următorului RTT ; în cazul în care diferența depășește o limită superioară, TCP Vegas scade liniar în cursul următorului RTT. Vegas realizează ratele de transport mai bine decât Reno și Tahoe. Cu toate acestea [6], Vegas nu poate garanta corectitudinea. În plus [20], nu se pot distinge natura erorii. Din perspectiva cercetării curente, este important să se ia în considerare faptul că autorii lui Vegas au demonstrat eficient că în modul de măsurare bazat pe ajustarea ferestrei este un mecanism viabil.

## ***TCP-REAL***

TCP-Real [59,55] are un receptor orientat și un măsurator de bază a congestie bazat pe un mecanism de control care îmbunătățește semnificativ performanța TCP peste rețele eterogene (cu fir / fără fir) și peste cai asimetrice. TCPReal merge dincolo de limitarea feedback-ului binar ACK-based. acesta estimează 13 nivel de afirmație și a distinge motivul pierderilor de pachete. TCPReal se bazează pe:

- Receiver-orientat pentru detectare congestie, care preia impactul fals al evaluărilor expeditorului, datorită pierderii r sau întârzierii recunoașterii pe o cale inversa. Receptorul măsoara condiția rețelei și acordă rezultatele pentru ACK-uri trimise înapoi la expeditor.
- Măsurători bazate pe modele de val care disting natura unei pierderi de pachete (din cauza congestionării sau erorilor tranzitorii fără fir)

Un val [54] constă dintr-un număr segmente de date fixe trimise de „back-to-back” , potrivite caracteristicii inerentă a TCP pentru a trimite pachete de „back-to-back”. Receptorul calculează rata de date de primire a unui val, care reflectă nivelul de susținere la link-ul de strangulare. Dacă un „packet drop” este din cauza unei erori fără fir, rata de date primite nu va fi afectată de decalajul de lipsă pachete deoarece dimensiunea valului este publicată la receptor. Fereastra de congestie este mult redusă numai atunci când o picătură este asociată cu congestia.

## ***TCP-WESTWOOD***

În TCP-Westwood (TCPW) [38], expeditorul măsoara continuu rata de conexiunea prin monitorizarea ratei de ACK-uri ce se întorc. La trei duplicate recunoscute sau timeout, pragul de start lent și congestionările fereastrei sunt stabilite în consistență cu lățimea de bandă efectivă utilizată la momentul pierderilor de pachete este un experimentat. Nici un mecanism specific nu există pentru a sprijini eroarea clasificată sau tactici corespunzătoare de recuperare pentru rețele cu fir / fără fir, deși mecanismul propus pare a fi eficient peste link-uri wireless simetrice, datorită controlului său al congestiei eficiente. O versiune optimizată a TCP-Westwood este TCP-Westwood + [39].

## ***TFRC***

TFRC este un TCP, protocol pe baza ritmului de control a congestie, care intenționează să concureze pentru lățimea de bandă cu fluxurile TCP. Rata de trimiterea a datelor este ajustată în răspuns la nivelul de congestie așa cum este indicat de rata de pierdere. Această ajustare este "blândă", care este tranzitată instantaneu are , în general, o variație mult mai mică în timp, în comparație cu TCP. Uniformizarea decalajelor de transport adecvate face TFRC necesar pentru streaming aplicații media, de telefonie sau de altă natură care necesită o rată de bună trimitere. Cu

toate acestea, netezimea are prețul său propriu: protocolul devine mai puțin sensibil în funcție de disponibilitatea de lățime de bandă [60].

În plus, TFRC este proiectat pentru aplicații care folosesc pachete de dimensiuni fixe. În cazul cererilor cu o variație în mărimea de pachete (ec. unele aplicații audio), mecanismul TRFC de control al congestiei nu poate fi utilizat. Un TFRC numit TFRC-PS (pentru TFRC-PacketSize) poate fi utilizat în locul lui. Acolo nu este disponibil încă pentru proiectul de TFRC-PS, dar mai mulți cercetători lucrează la aceste aspecte.

TFRC introduce "evenimentul de pierdere" în loc de pierderi de pachete tradiționale. O pierdere de evenimente este definită ca una sau mai multe pachete pierdute sau marcate dintr-o fereastră de date (un pachet marcat se referă la o indicație de congestionare a congestiei explicită). TFRC utilizează un mecanism bazat pe receptor pentru calcularea ratei de pierdere a evenimentului. Un astfel de mecanism este adecvat pentru controlul congestiei multicast și, de asemenea, se potrivește în cazul unui server cu multe cereri de soluționare concurente de la clienți cu mai multă memorie și ciclurile CPU disponibile. Receptorul măsoară rata evenimentului de pierdere și apoi trece aceste informații către expeditor. Expeditorul o calculează folosind o ecuație tranzitată care încorporează evenimentul în rata de pierdere, timp tur-retur și dimensiunea pachetului.

În concluzie, mecanismul de control al congestiei TFRC funcționează după cum urmează:

- Receptorul măsoară rata evenimentului de pierdere (bazată pe pachetele pierdute sau marcate de la ECN, într-o fereastră unică) și apoi trimite această informație înapoi, la expeditor.
- Expeditorul utilizează această informație, de asemenea, pentru a măsura timpul de tur-retur(RTT).
- Evenimentul ratei de pierdere, timpul tur-retur și dimensiunea pachetului sunt utilizate în funcția de calcul. Expeditorul își ajustează rata de trimitere de date care se potrivesc cu rata calculată.

$$X = \frac{s}{R \cdot \sqrt{\frac{2 \cdot \beta \cdot p}{3}} + (t_{RTO} \cdot (3 \cdot \sqrt{\frac{3 \cdot \beta \cdot p}{8}}) \cdot p \cdot (1 + 32 \cdot p^2))}$$

Unde:

- X este rata de transmitere în bytes / secunda.
- S este dimensiunea pachetului în octeți.
- R este timpul de călătorie dus-întors în secunde.
- p este evenimentul ratei de pierdere, între 0 și 1,0, de numărul de evenimente o pierdere ca fracțiune din numărul de pachete transmise.
- t\_RTO este valoarea TCP în secunde.
- b este numărul de pachete recunoscute printr-o confirmare TCP singură.

Ar putea fi util pentru a asocia fiecare informație trimisă înapoi la expeditor, cu o cifră statistică, ceea ce indică posibilitatea unei decizii greșite. Apoi, pe baza acestor date, o strategie agresivă sau conservatoare poate fi aleasă.

## ***TCP-JERSEY***

Autorii [56] au propus recent un nou sistem de TCP, numit TCP-Jersey. Ei s-au concentrat pe capacitatea de mecanism de transport pentru a distinge pierderile de pachete de congestie fara fir. TCP-Jersey introduce estimarea disponibila lăţimii de bandă (ABE), algoritmul si avertizarea congestionării (CW) a router-ului de configurare. ABE realizeaza estimări continuu a lăţimii de bandă disponibilă și direcţionează expeditorului pentru a ajusta rata de transmitere în conformitate cu estimarea. „CW-configured-routers” marcheaza pachetele atunci când există un semn al unei incipient de congestioneare notificand expeditor, care, clasifică erorile în consecinţă.

Performanţa mecanismelor de transport de mai sus este strans cuplata cu robusteţea estimatorilor lor. Mai multe anchete [22] au fost efectuate cu privire la corectitudinea estimatorilor propuse.

### **The Box Is Green**

(Starea reţelei este cunoscuta)

Dobre Marius

### ***Mecanismul Bimodal***

Mecanismul bimodal de control si de evitare a congestiei masoara partea pusa in comun din lungimea de banda care ar fi trebuit sa fie alocata fiecarui debit, in orice moment, in timpul executiei sistemului. Daca aceste parti ar fi fost cunoscute, atunci sursele ar fi putut evita congestia prin ajustarea imediata dupa ce partea comuna a fost descoperita, la o noua stare in care lungimea de banda alocata pentru fiecare debit. Cu toate acestea, disponibilitatea lungimii de banda nu este doar o problema a capacitatii canalului, ci este si dependenta de numarul de debite participante si de comportamentul transmis de la sursa. Deci punerea in comun poate fi masurata doar intr-o stare de echilibru.

Autorii au propus in [3] un mecanism bimodal, bazat pe ideea ca ajustajele in sus si in jos trebuie sa opereze in asociatie cu starea sistemului. Actiunea este determinata chiar daca sistemul este in echilibru (partea distribuita este cunoscuta) sau nu (partea comuna nu este este necunoscuta). Cand partea distribuita este cunoscut, algoritmul se comporta ca AIMD, pana cand doi ciclii de concestie au trecut, fiind sufficient sa recalculeze FAIRSHARE. Algoritmul seteaza apoi lungimea de banda alocata pentru debitul  $f$  la  $(1-\epsilon)f$  si comuta intr-un mod de distributie cunoscut. Asadar, algoritmul bimodal de control a congestiei calculeaza explicit distributia si converge in doi ciclii de congestie la distributie. In acest mod, algoritmul continua sa foloseasca cresteri aditive si scaderi multiplicative, doar ca factorul de scadere multiplicative este  $\alpha$  in loc de  $\beta$ .

Acest algoritm este diferentiat de clasa algoritmilor TCP. Algoritmii TCP favorizeaza netezimea , reducand corectitudinea. Acest algoritm calculeaza partea distribuita in mod explicit,

deci corectitudinea nu este compromisa. Mai mult, din moment ce acest algoritm restrictioneaza debitul pentru a folosi doar partea pusa in comun, poate fi utilizat in stransa legatura cu orice alt protocol de transport (spre exemplu standardul AIMD) fara sa monopolizeze pentru el cea mai mare parte a lungimii de banda. Acest fapt este in contrast cu protocoalele TCP, care incearca sa capteze toata lungimea de banda disponibila. Desigur, acest algoritm nu va functiona bine in legatura cu alte protocoale care in mod agresiv (si necinstit) capteaza o parte disproporcionada a lungimii de banda pentru debitele lor.

Exista de asemenea probleme nerezolvate legate de mecanismul bimodal:

- Cu cat castigul este mai mare in goodput (nivelul de aplicatie care descrie rata de success a transmiterii unui mesaj), cu atat spatiul liber pentru debitele care intra este mai micatunci cand controversa creste.

- Investigand valoarea optima a lui  $\epsilon$  in stransa legatura cu dinamica unor medii specifice este un subiect pentru un studiu viitor.

- Modificarea algoritmului pentru un scenariu ansincron prin integrarea RTT intr-un calcul de distributie comuna. Spre exemplu, lungimea de banca alocata unui debit poate fi marita cand pachetele RTT scad, si poate fi scazuta cand RTT-ul creste.

- Integrarea unor asemenea idei cu o retroactiune orientate spre receptionare.

## Controlul congestiei retelei asistate

Red gates ("Portiile rosii" [12]) pierd pachetele cand o congestie este pe cale sa se produca. RED pierd pachetele, activand scaderi multiplicative in unele debite cand lungimea cozii de asteptare depaseste un prag predeterminat. RED poate functiona fara a necesita o schimbare in nivelul infrastructurii de transport.

Ramakrishnan si Floyd in [46] propun o Notificare Explicita a Congestiei (ECN) care sa fie adaugata protocolului IP pentru a active controlul congestiei TCP. Spre deosebire de RED, ECN permite ruterelor sa marcheze probabilistic un bit in header-ul IP, in loc sa piarda pachetul, sa informeze gazdele care asteapta o congestie cand lungimea cozii de asteptare depaseste un prag. Gazdele reduc multiplicativ congestia cand se primesc pachete cu bitul ECN setat, inainte ca buffer-ul ruterului sa fie supraincarcat si pachetul sa fie inevitabil pierdut. O dualitate este servita prin ECN: performanta TCP poate fi imbunatatita prin mijloace de evitare a pierderilor de date cauzate de spatiul limitat al buffer-ului si prabusirea congestiei poate fi evitata.

Studiile recente prezinta in [40] o discutie critica a performantelor asteptate de la RED. O observatie interesanta despre RED si ECN este ca acestea pot, cumva, sa limiteze evolutia viitoare. Daca ar fi sa ne imaginam o TCP mai sofisticata care sa faca deosebire intre congestie si pierderile wireless. Din moment ce RED pierde pachete proportional cu rata de trimitere, este neclar cat de cinstit ar fi RED in TCP-urile mai sofisticate care nu neaparat se dau inapoi in cazul unei pierderi wireless accidentale [27].

In [10] s-au introdus mecanisme bazate pe identificarea debitelor cu lungime mare de banda din istoricul de pierderi al RED. Algoritmul RED-PD (RED cu pierderi preferentiale [37]) foloseste mecanisme de pierdere preferentiale pentru fiecare debit. Doua alte abordari care folosesc pierderile preferentiale cu programare FIFO (first in first out) sunt coada Core-Stateless Fair (CSFQ [51]) si detectarea timpurie a debitului aleator (FRED [34]). CSFQ macheaza pachetele cu o estimare a ratei de trimitere curente. Ruterul foloseste aceste informatii impreuna cu estimarea FAIRSHARE a debitului pentru a decide daca un pachet trebuie aruncat sau nu. FRED mentine o stare, cu toate ca doar pentru debitele care au pachete in coada de asteptare. Debitele cu multe pachete in buffer au o probabilitate mai mare sa fie pierdute.

Autorii din [43] au continuat abordarile CSFK si CHOKe din [44]. Mecanismul propus de acestia pastreaza o mostra a traficului primit. Un debit cu mai multe pachete in acea mostra are o probabilitate crescuta sa fie pierdute. Stochastic Fair Blue (SFB [8]) foloseste nivele multiple de dispersare pentru a identifica debitele cu o lungime de banda marita. Asa cum afirma autorii, mecanismul lor functioneaza bine doar cu putine debite cu lungime mare de banda. Anjum and Tassiulas au propus in [2] un mechanism care sa piarda pachetele, bazat pe gradul de ocupare a bufferului al unui debit, in timp ce ERUF ("A. Rangarajan. Early regulation of unresponsive flows" [47]) foloseste source quench (un protocol de internet pentru mesaje control) pentru a avea pachete nelivrabile pierdute la ruterele periferice. Pe de alta parte, SRED (SRED: Stabilized RED [41]) prinde debitele recente pentru a determina debitele cu lungime mare de banda.

## Discutii

Pe de alta parte, aplicatiile in sesi genereaza tipare diferite de trafic si ca atare sunt asociate cu diferite tipare de congestie. Cum se poate justifica modul practic de realizare a experimentelor doar cu trafic FTP? Cat de diferita este congestia datorata unui numar mare de debite (mai mare decat capacitatea retelei) fata de congestia datorata unui numar limitat de debite cu ferestre mari? Asadar, care este contributia si inter-relatia pentru fiecare timeout si mecanismul ferestrei de gestionare?

Studiul cu privire la TCP discutat mai sus ridica o alta intrebare importanta: unde este locul potrivit pentru a adauga functionalitatea ceruta? Aceasta intrebare nu are un raspuns concret; controlul erorii nu este exclusiv o problema manageriala a ruterului sau a statiei de baza, dar nici exclusiva asignata nivelului de transport. O abordare acceptata unanim afirma ca putem implementa doar o functie la un strat inferior, daca stratul poate executa complet functia. Din moment ce nivelul inferior nu poate avea suficienta informatie despre cerintele aplicatiei, protocolul parametrilor si conditiile dispozitivului, nu poate implementa integral functiile de control a erorii; el poate fi folosit doar sa optimizeze functia unui strat superior.

In mod cert, poate fi benefic pentru emitator ca acesta sa stie cu precizie ca va avea loc congestia. Cu toate acestea, in contextul retelelor cu fire eterogene sau wireless, contributia ECN poate fi limitata: prin neprimirea unei notificari explicite, emitatorul TCP nu va putea sa afirme in mod sigur ca o pierdere detectata nu a fost cauzata de congestie. Precizia dorita a emitatorilor TCP capabili ECN poate fi mai bine estimata daca nivelul congestiei poate de asemenea indicat pentru a permite TCP-ului sa implementeze o recuperare mai sofisticata. Precizia, de altfel, vine cu un cost:

functionalitatea rutelor este mai complexa, modificari sunt necesare in rutare cat si in TCP, si, in final, rezultatul poate fi mai scazut datorita neomogenitatii. De exemplu, cateva rutere nu sunt capabile ECN.

Putem trece dincolo de criteriile de optimizare arhitecturale si sa ne ingrijoram de potentiale strategii false datorate deciziilor locale ale ruterului. Spre exemplu, cand un ruter intampina o congestie sau un proces catre congestie, actioneaza. Actiunea este luata in asociere cu o presupunere inclusa: acea portiune a retelei care urmeaza nu va schimba caracteristicile de trafic ale debitelor – ceva nu neaparat adevarat atunci cand o retea wireless este montata la sfarsitul receptorului. In mod cert, debitul TCP permite presupunerea unui comportament monoton. Cu toate acestea, cand presupunerea noastra este gresita, nu avem nicio indicatie ca strategia noastra este corecta inca.

Pornind de la aceasi idee, am explicat anterior logica din spatele protocoalelor bazate pe masurari. Exista totusi o diferenta semnificativa; rezeroarele au intr-adevar capacitatea sa masoare debitul de trafic prin intreaga retea, asadar nu fac aprecieri gresite datorate vederii limitate a retelei.

Tocmai de aceea, cand multe debite concureaza pentru o lungime de banda limitata, o strategie “window back-off” nu va produce un castig semnificativ. Spre exemplu, un numar mare de debite peste o retea de 10 Mbps va opera probabil cu “single-pachet windows” care nu vor permite ingustari ulterioare. Evident, sub o asemenea controversa a pachetelor, eficienta algoritmilor nu este chiar o problema. Doar un timeout specific poate permite tuturor debitelor folosirea corecta a retelei.

In final, evaluarea controlului congestiei este o problema importanta. Catiiva autori au propus modificari mecanismelor existent; cu toate acestea, putini dintre acestia au obtinut o implementare minora. Acest lucru poate fi justificat in primul rand pe baza a 4 observatii diferite, rezumate mai jos:

- (1) Natura problemei; complexitatea acesteia datorata unui numar mare de parametri care au un impact asupra performantei sistemului nu permite totdeauna o evaluare efectiva
- (2) Natura congestiei in internetul modern nu este chiar cunoscuta; spre exemplu, cum putem caracteriza nivelele diferite de congestie? Cat de des apar acestea? Care este o conditie extrema a congestiei, cat dureaza aceasta si ce procent din pachet este pierdut?
- (3) Pozitia congestiei nu este bine determinata; unde se intampla o congestie in internet? Spre exemplu, congestia poate aparea la baza gateway-ului sau la un ruter local. Pozitia este asociata cu numarul de debite care traverseaza ruterul congestionat.
- (4) Obiectivul controlului congestiei se schimba. In retelele congestionate din zilele noastre, prabusirea nu este principala problema; netezimea si raspunsul devin mult mai importante.



# Erori

Toma Crina

Efortul de distingere a erorilor asociate congestiei a erorilor fara fir. Acest efort investit in principal in distingerea pozitiei erorilor [4,50,17] mai degraba decat concentrandu-se asupra dinamicii de combinatie a celor doua erori. De exemplu, congestiile persistente care pot fi calificate in unele rutere dintr-o retea cu fir se pot schimba intr-una mai tranzitorie atunci cand erorile fara fir sunt introduse pe ultima suta de metri, unde unele receptoare fara fir ocupa un loc.

In plus, disputele inalte pot fi tolerate sub aceleasi circumstante ale congestiei. Intr-un context similar intr-o congestiei de retea de viteza mare poate cauza pierderi din cauza faptului ca ferestrele congestiei pot creste pana la valori ridicate, insa aceasta poate dura mai putin si, potential, aceasta va aparea mult mai putin frecvent [1,9,23,26,49]. Dinamica de fir si a erorilor fara fir par a fi problem mai importante decat localizarea geografica a erorii si aplicarea tehnicilor bine-cunoscute.

Congestiile accidentale combinate cu disputele inalte nu pot apela reconstituiri ample.

## Bibliografie

1. Approaches to Congestion Control in packet networks, L. Mamatras, V. Tsaoussidis, Chi Zhang

## Referinte

- [1] I. Akyildiz, G. Morabito, and S. Palazzo. TCP Peach: A New Congestion Control Scheme for Satellite IP Networks. *IEEE/ACM Transactions on Networking*, 9(3):307–321, June 2001.
- [2] F. M. Anjum and L. Tassiulas. Fair bandwidth sharing among adaptive and non-adaptive flows in the internet. In *IEEE InfoCom 99*, March 1999.
- [3] P. C. Attie, A. Lahanas, and V. Tsaoussidis. Beyond AIMD: Explicit fair-share calculation. In *Proceedings of ISCC 2003*, June 2003.
- [4] H. Balakrishnan, S. Seshan, E. Amir, and R. H. Katz. Improving TCP/IP Performance over Wireless Networks. In *Proceedings of ACM Mobicom '95*, November 1995.
- [5] D. Bansal and H. Balakrishnan. Binomial Congestion Control Algorithms. In *Proceedings of the IEEE INFOCOM'01*, 2001.
- [6] L. Brakmo and L. Peterson. TCP Vegas: End to End Congestion Avoidance on a Global Internet. *IEEE Journal on Selected Areas of Communications*, October 1995.
- [7] D. Chiu and R. Jain. Analysis of the Increase/Decrease Algorithms for Congestion Avoidance in Computer Networks. *Journal of Computer Networks and ISDN*, 17(1):1–14, June 1989.
- [8] W. Feng, D. Kandlur, D. Saha, and K. G. Shin. BLUE: A new class of active queue management algorithms. Technical Report CSE-TR-387-99, April 1999.
- [9] S. Floyd. Highspeed tcp for Large Congestion Windows. RFC 3649, December 2003.
- [10] S. Floyd and K. Fall. Promoting the use of end-to-end congestion control in the Internet. *IEEE/ACM Transactions on Networking*, 7(4):458–472, 1999.
- [12] S. Floyd and V. Jacobson. Random early detection gateways for congestion avoidance. *IEEE/ACM Transactions on Networking*, 1(4):397–413, 1993.
- [17] T. Goff, J. Moronski, and D. Phatak. Freeze-TCP: A True end-to-end Enhancement Mechanism for Mobile Environments. In *Proceedings of the INFOCOM, (Israel), 2000*, 2000.
- [20] U. Hengartner, J. Bolliger, and T. Cross. TCP Vegas Revisited. In *Proceedings of IEEE INFOCOM 2000*, March 2000.
- [22] M. Jain and C. Dovrolis. Ten fallacies and pitfalls on end-to-end available bandwidth estimation. In *Internet Measurement Conference 2004*, pages 272–

277, 2004.

- [23] C. Jin, D. Wei, and S. Low. Fast TCP: motivation, architecture, algorithms, performance. In Proceedings of the IEEE Infocom, March 2004.
- [24] S. Jin, L. Guo, I. Matta, and A. Bestavros. TCP-friendly SIMD Congestion Control and Its Convergence Behavior. In Proceedings of the ICNP'2001, November 2001.
- [26] D. Katabi, M. Handley, and C. Rohrs. Congestion Control for High Bandwidth-Delay Product Networks. In In the proceedings on ACM Sigcomm 2002, 2002.
- [27] M. Khanna, C. Zhang, and V. Tsaoussidis. Experimental evaluation of RED in Heterogeneous Environments. The International Journal of Communication Systems IJCS.
- [31] A. Lahanas and V. Tsaoussidis. Additive Increase Multiplicative Decrease - Fast Convergence (AIMD-FC). In Proceedings of Networks 2002, August 2002.
- [34] D. Lin and R. Morris. Dynamics of random early detection. In SIGCOMM '97, pages 127–137, Cannes, France, september 1997.
- [37] R. Mahajan, S. Floyd, and D. Wetherall. Controlling high-bandwidth flows at the congested router, 2001.
- [38] S. Mascolo, C. Casetti, M. Gerla, M. Sanadidi, and R. Wang. TCP Westwood: Bandwidth Estimation for Enhanced Transport over Wireless Links. In Proceedings of the MobiCom'01, July 2001.
- [39] S. Mascolo, L. A. Grieco, R. Ferorelli, P. Camarda, and G. Piscitelli. Performance evaluation of westwood+ tcp congestion control. In Internet performance symposium (IPS 2002), volume 55, pages 93–111, 2002.
- [40] M. May, T. Bonald, and J. Bolot. Analytic Evaluation of RED Performance. In INFOCOM 2000, August 2000.
- [41] T. J. Ott, T. V. Lakshman, and L. H. Wong. SRED: Stabilized RED. In Proceedings of INFOCOM, volume 3, pages 1346–1355, 1999.
- [42] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose. Modeling TCP Throughput: A Simple Model and its Empirical Validation. In Proceedings of the ACM SIGCOMM, 1998.
- [43] R. Pan, L. Breslau, B. Prabhakar, and S. Shenker. Approximate fairness through differential dropping, 2001.
- [44] R. Pan, B. Prabhakar, and K. Psounis. CHOKe - A stateless queue management scheme for approximating fair bandwidth allocation. March 2000.
- [46] K. Ramakrishnan and S. Floyd. A Proposal to add Explicit Congestion Notification (ECN) to IP. RFC 2481, January 1999.
- [47] A. Rangarajan. Early regulation of unresponsive flows. Technical Report

TRCS99-26, July 1999.

[48] R. Rejaie, M. Handley, and D. Estrin. RAP: An end-to-end rate-based congestion control mechanism for realtime streams in the internet. In INFOCOM (3), pages 1337–1345, 1999.

[49] S. Shalunov. Tcp Armonk (tcpar). Technical report, September 2002.

[50] P. Sinha, T. Nandagopal, N. Venkitaraman, R. Sivakumar, and V. Bharghavan. WTCP: A reliable transport protocol for wireless wide-area networks. *Wireless Networks*, 8(2-3):301–316, 2002.

[51] I. Stoica, S. Shenker, and H. Zhang. Core -stateless fair queueing: Achieving approximately fair bandwidth allocations in high speed networks. In SIGCOMM, pages 118–130, 1998.

[54] V. Tsaoussidis, A. Lahanas, and C. Zhang. The Wave and Probe Communication Mechanisms. *The Journal of Supercomputing*, Kluwer Academic Publishers, June 2001.

[55] V. Tsaoussidis and C. Zhang. Tcp-real: Receiver-oriented congestion control. *Computer Networks Journal (Elsevier)*, 40(4), November 2002.

[56] K. Xu, Y. Tian, and N. Ansari. TCP-jersey for wireless IP communications. *IEEE JSAC*, 22(4):747–756, May 2004.

[58] Y. Yang and S. Lam. General AIMD Congestion Control. In *Proceedings of the IEEE International Conference on Network Protocols*, November 2000.

[59] C. Zhang and V. Tsaoussidis. TCP-Real: Improving Real-time Capabilities of TCP over Heterogeneous Networks. In *Proceedings of the 11th IEEE/ACM NOSSDAV*, June 2001.

[60] C. Zhang and V. Tsaoussidis. The interrelation of TCP Responsiveness and Smoothness. In *Proceedings of 7th IEEE ISCC*, July 2002.