

Tehnici de alocare a resurselor pentru creșterea
performanțelor transmiterii datelor în rețelele de
telefonie mobilă de tip GPRS/EGPRS
Teză de doctorat

ing. Doru Todinca

Conducător științific: Prof. dr. ing. Ștefan Holban
Departamentul Calculatoare
Universitatea “Politehnica” Timișoara

2004

Pentru Ioana, pentru fratele meu

©' 2004 Doru Todinca

Mulțumiri

Această teză a fost elaborată sub conducerea domnului profesor dr. ing. Ștefan Holban, căruia doresc să îi mulțumesc și pe această cale. De asemenea, doresc să le mulțumesc tuturor colegilor din departament care au participat la activitatea de pregătire pentru doctorat –examene și referate– și de la care am primit sugestii utile și încurajări.

Un sprijin extrem de important, tehnic și moral, am primit pe durata stagiului de cercetare de la Dublin City University, Performance Engineering Lab, de la dr. John Murphy și dr. Philip Perry, precum și de la colegii Hubert Graja, Gabriel și Cristina Muntean, Ada Diaconescu, Adrian Moș, Octavian Ciuhandu și Valentin Mureșan.

Pentru colaborarea referitoare la studiul performanțelor controlerelor fuzzy, doresc să le mulțumesc lui Radu Emil Precup și domnului profesor Ștefan Preitl, precum și studenților cu care am lucrat în acea perioadă: Mihai Moisa, Ciprian Ghișe, Ruxandra Popeț și Ana-Maria Bădulescu. Le sunt foarte recunoscător actualilor și foștilor colegi din departament care m-au încurajat în studiul logicii fuzzy și cu care am avut numeroase și interesante discuții.

Am fost de nenumărate ori ajutat cu sfaturi, sugestii, instalări de programe, de Marian Boldea, Dan Pescaru și Adrian Petre Mierluțiu, cărora le sunt foarte îndatorat pentru aceste lucruri.

Această teză nu ar fi fost posibilă fără sprijinul permanent al familiei, așa că le mulțumesc în primul rând mamei și fratelui meu, pentru tot ce au făcut pentru mine, iar socrului meu doresc să îi mulțumesc pentru permanentele încurajări din ultimii ani.

Ioana, soția mea, a fost nu doar cel mai asiduu cititor al acestei teze și al articolelor rezultate din munca la teză, ci și cel mai important sfătuitor și sprijin pe toată durata muncii la doctorat.

Această teză este dedicată memoriei tatălui meu.

Timișoara, 12 decembrie 2004.

Cuprins

1	Introducere	1
1.1	Descrierea problemei de cercetare	1
1.2	Structura lucrării	2
2	Alocarea resurselor în GPRS. Probleme actuale de cercetare	4
2.1	Prezentare generală a GPRS	5
2.1.1	Aspecte generale	5
2.1.2	Comparație între GSM și GPRS	6
2.2	Algoritmi de alocare a resurselor pentru rețele cu comutare de pachete	11
2.2.1	Algoritmi de tip Packet Fair Queueing	12
2.2.2	Probleme specifice rețelelor radio	17
2.3	Algoritmi de alocare a resurselor pentru controlul transmiterii datelor în rețele GPRS	19
2.3.1	Abordări analitice	19
2.3.2	Algoritmi de planificare pentru GPRS	20
2.3.3	Simulatoare și emulatoare pentru GPRS	27
2.3.4	Alocarea canalelor între traficul de voce și cel de date	32
2.4	Problema admiterii utilizatorilor în rețele celulare de date	35
2.5	Utilizări ale logicii fuzzy în telecomunicații	38
2.5.1	Mulțimi fuzzy. Operații cu mulțimi fuzzy	39
2.5.2	Logica fuzzy aplicată în rețele ATM	40
2.5.3	Logica fuzzy aplicată în rețele radio	41
2.6	Concluzii și obiectivele tezei	43
3	Model de simulare a alocării resurselor într-o rețea de tip GPRS/EGPRS	46
3.1	Alocarea resurselor în GPRS	46
3.1.1	Arhitectura de protocoale GPRS, fluxul de date și codarea canalelor	46

3.1.2	Managementul mobilității și activarea unui context PDP . . .	50
3.1.3	Calitatea serviciului în GPRS	54
3.2	Descrierea modelului de simulare	56
3.2.1	Simulatorul utilizat	56
3.2.2	Prezentare de ansamblu a modelului	57
3.2.3	Modelul pentru un utilizator GPRS	59
3.2.4	Modulul PCU	66
3.2.5	Generatorul de utilizatori, nodul admission controller și nodul stat	69
3.3	Validarea modelului	73
3.4	Concluzii	75
4	Performanțele algoritmilor pentru controlul transmiterii datelor	78
4.1	Algoritmi pentru controlul transmiterii datelor	78
4.1.1	Algoritmi de tip Packet Fair Queueing	79
4.1.2	Algoritmii de planificare bazați pe cantitatea de date din cozile utilizatorilor	82
4.2	Descrierea condițiilor experimentelor	84
4.2.1	Primul set de experimente	85
4.2.2	Al doilea set de experimente	86
4.3	Compararea performanței algoritmilor de alocare a resurselor	87
4.3.1	Comparație între algoritmi pentru ponderi egale ale utilizatorilor	88
4.3.2	Comparație între algoritmi pentru ponderi diferite ale utiliza- torilor	93
4.3.3	Un rezultat important	96
4.4	Studiul influenței factorilor de pondere asupra performanțelor algorit- milor	98
4.4.1	Utilizarea ponderilor pentru realizarea diferențierii calității ser- viciilor utilizatorilor	99
4.4.2	Utilizarea ponderilor utilizatorilor pentru reducerea congestiilor	102
4.5	Gradul de încredere în rezultatele simulărilor	104
4.5.1	Seturi diferite de numere aleatoare	104
4.5.2	Acuratețea simulărilor	105
4.5.3	Intervale de încredere pentru valorile medii ale întârzierilor fiecărui utilizator	106
4.5.4	Intervale de încredere pentru valorile medii ale întârzierilor pen- tru fiecare clasă de utilizatori	108

4.6	Concluzii	110
5	Soluție fuzzy pentru controlul admiterii utilizatorilor	112
5.1	Logică fuzzy pentru admiterea utilizatorilor: motivație	112
5.2	Performanțele controlerelor fuzzy	114
5.2.1	Variabile lingvistice, inferența fuzzy	114
5.2.2	Controlere fuzzy digitale	116
5.2.3	Implementarea și evaluarea performanțelor controlerelor fuzzy	123
5.3	Algoritmul fuzzy de admitere a utilizatorilor	126
5.3.1	Cerințele algoritmului de admitere a utilizatorilor	126
5.3.2	Formularea matematică a cerințelor algoritmilor de admitere a utilizatorilor	127
5.3.3	Implementarea algoritmului fuzzy de admitere a utilizatorilor .	129
5.3.4	Rezultate în cazul controlerului fuzzy cu o intrare	132
5.3.5	Rezultate în cazul controlerului fuzzy cu două intrari	136
5.3.6	Compararea performanțelor algoritmului fuzzy de admitere a utilizatorilor cu algoritmi bazați pe valori de prag	139
5.4	Concluzii	142
6	Concluzii. Rezumatul contribuțiilor tezei	145
	Bibliografie	150
	Lista figurilor	163
	Lista tabelor	164
A	Glosar	165

Capitolul 1

Introducere

1.1 Descrierea problemei de cercetare

Dezvoltarea din ultimii ani a rețelelor de telefonie mobilă și mai ales trecerea de la rețelele care ofereau doar posibilitatea unor convorbiri telefonice (de exemplu GSM), la rețele care fac posibilă rularea unor aplicații de transfer de date, face ca utilizatorii acestor rețele să își dorească să poată dispune de aceleași servicii atunci când sunt în mișcare ca și în situația în care s-ar afla la locul de muncă sau acasă. Aceasta înseamnă că operatorii de telefonie mobilă trebuie să poată oferi clienților lor posibilitatea să poată rula aceleași aplicații (de la e-mail și transfer de fișiere până la urmărirea unor clipuri video sau audio sau emisiuni tv sau radio), la o calitate apropiată de cea pe care ar avea-o la conectarea de pe un calculator fix.

În acest sens, un aspect extrem de important îl reprezintă **calitatea serviciului** oferit utilizatorilor unei rețele celulare de date. Calitatea serviciului (*Quality of Service – QoS*) este definită ca fiind efectul colectiv al performanței serviciului, care determină gradul de satisfacție al utilizatorilor serviciului [52].

Calitatea serviciului se măsoară prin parametri cum ar fi întârzierile datelor, care se reflectă în timpul de răspuns al aplicațiilor, precum și în capacitatea de trecere a rețelei, așa cum este percepută de utilizator, prin aspecte ce țin de fiabilitatea transferului de date (probabilitatea ca datele să fie pierdute, afectate de erori necorectabile, duplicate, etc), securitatea asigurată informațiilor, precum și prin probabilitatea ca o sesiune a unui utilizator să fie blocată sau întreruptă.

Este important de asemenea ca utilizatorilor să li se poată asigura **niveluri diferite de calitate a serviciului**, în funcție de tipul subscripției, dar și în funcție de aplicațiile rulate, deoarece diverse tipuri de aplicații au cerințe diferite de calitate a serviciului.

Pentru satisfacerea cerințelor de calitate a serviciului utilizatorilor, un rol extrem

de important îl are modul cum se face alocarea resurselor rețelei. În această teză se vor studia **metode de alocare a resurselor la transmiterea datelor în rețele de tip GPRS/EGPRS**.

Problema alocării resurselor la rețele de tip GPRS/EGPRS are două componente principale:

- alocarea resurselor pentru controlul transmiterii datelor (*Transmission Control – TC*)
- admiterea utilizatorilor (*Admission Control – AC*).

Alocarea resurselor pentru controlul transmiterii datelor se referă la tehnicile de partajare a resurselor rețelei, în acest caz resursele radio, între utilizatorii care sunt admiși în rețea și care doresc să transfere date.

Admiterea utilizatorilor se referă la decizia de a admite sau nu un utilizator, atunci când acesta dorește să înceapă o sesiune de transfer de date. Algoritmii de AC trebuie să decidă dacă rețeaua are suficiente resurse pentru a asigura calitatea dorită (și promisă) a serviciilor pentru utilizatorii care deja sunt conectați, ca și pentru utilizatorul care dorește să se conecteze. În rețelele mobile această problemă este mult mai complexă decât în rețelele fixe datorită mobilității utilizatorilor.

În această teză se vor investiga într-un mod unitar ambele componente ale problemei alocării resurselor, atât controlul transmiterii datelor, cât și admiterea utilizatorilor. Investigațiile se vor realiza prin simulare, în acest scop fiind dezvoltat un model de simulare propriu. Se urmărește studiul performanțelor algoritmilor de planificare utilizați la controlul transmiterii datelor și rolul lor în asigurarea calității serviciului, precum și dezvoltarea unei soluții noi, originale, la problema admiterii utilizatorilor în rețele GPRS și EGPRS.

1.2 Structura lucrării

Teza este structurată după cum urmează.

Capitolul 2 prezintă stadiul actual al cunoașterii în domeniul alocării resurselor pentru transmiterea datelor în rețele de tip GPRS și EGPRS. La început se face o prezentare generală a GPRS. În continuare se prezintă stadiul actual al cercetării în domeniul algoritmilor de planificare a resurselor, întâi într-un cadru general, iar apoi în domeniul rețelelor GPRS și EGPRS. Se prezintă de asemenea o serie de algoritmi de admitere a utilizatorilor în rețele celulare de date, precum și utilizări ale logicii fuzzy în telecomunicații. Capitolul se încheie cu definirea obiectivelor acestei teze, raportate la stadiul actual al cunoașterii în domeniu.

Capitolele 3, 4 și 5 reprezintă contribuții de cercetare ale acestei teze.

Capitolul 3 dezvoltă un model de simulare a alocării resurselor într-o rețea de tip GPRS / EGPRS. Acest model este realizat pornind de la descrierea modului în care se face alocarea resurselor într-o rețea GPRS reală. Modelul realizat este capabil să reprezinte ambele aspecte implicate de alocarea resurselor în rețele (E)GPRS, și anume admiterea utilizatorilor și controlul transmiterii datelor.

În capitolul 4 sunt investigați o serie de algoritmi de planificare a resurselor pentru controlul transmiterii datelor. Performanțele lor sunt comparate prin simulare. În acest capitol mai este investigată și problema diferențierii calității serviciului între diferite clase de utilizatori, precum și problema reducerii congestiilor în rețea prin alocarea resurselor pe baza calității legăturii radio a utilizatorilor.

Problema admiterii utilizatorilor este studiată în capitolul 5, unde este propusă o soluție originală de AC, bazat pe logică fuzzy. Soluția propusă cuprinde utilizarea unui controler fuzzy performant și un algoritm de admitere a utilizatorilor exprimat prin reguli fuzzy. Performanțele controlerelor fuzzy au fost studiate prin simulare în aplicații complexe de reglare. Este descris algoritmul fuzzy propus, implementarea sa pe un controler fuzzy, precum și integrarea controlerului fuzzy în cadrul modelului de simulare. Performanțele algoritmului propus sunt studiate în cadrul unui număr de scenarii de simulare și sunt comparate cu performanțele unor algoritmi non-fuzzy de admitere a utilizatorilor.

Lucrarea se încheie cu un capitol de concluzii generale, care conține și rezumatul contribuțiilor personale din teză.

Capitolul 2

Alocarea resurselor în GPRS. Probleme actuale de cercetare

În acest capitol se prezintă și se analizează stadiul actual al cercetărilor din domeniul tezei. Domeniul este cel al alocării resurselor în rețele GPRS și EGPRS, cuprinzând atât algoritmi de alocare pentru controlul transmiterii datelor, cât și problema admiterii utilizatorilor.

Secțiunea 2.1 conține o descriere generală a GPRS.

Secțiunile 2.2 și 2.3 tratează algoritmi de alocare a resurselor pentru controlul transmiterii datelor. În 2.2 se începe cu problema mai generală a alocării resurselor în rețele cu comutare de pachete, apoi se descriu probleme specifice rețelelor radio. În 2.3 se prezintă cercetările actuale privind alocarea resurselor în rețele de tip GPRS și EGPRS. Deoarece metodele de investigare din teză se bazează pe simulare, în subsecțiunea 2.3.3 se prezintă simulatoare și emulatoare pentru GPRS și EGPRS. Această analiză a metodelor existente de alocare a resurselor pentru controlul transmiterii datelor va fi utilizată ca punct de plecare pentru cercetările proprii în această direcție descrise în capitolul 4.

În secțiunea 2.4 sunt prezentate cele mai interesante lucrări referitoare la problema admiterii utilizatorilor în rețelele celulare de date. Deoarece în capitolul 5 teza propune o soluție originală de admitere a utilizatorilor, soluție bazată pe logica fuzzy, în secțiunea 2.5 se discută câteva din aplicațiile logicii fuzzy în domeniul telecomunicațiilor, cu accent pe utilizarea logicii fuzzy în rețele radio.

2.1 Prezentare generală a GPRS

2.1.1 Aspecte generale

Datorită dezvoltării sistemelor de comunicații mobile și a răspândirii pe care acestea au cunoscut-o în ultimii ani, a apărut posibilitatea de a dezvolta noi servicii și aplicații bazate pe rețelele de telefonie celulară, printre care aplicațiile de date vor juca un rol din ce în ce mai important. Deși pentru următorii ani se preconizează că piața comunicațiilor mobile va fi dominată de telefonia mobilă (voce), serviciile bazate pe transferul de date (transfer electronic de fonduri, calculatoare personale conectate în rețelele celulare etc.) vor juca un rol din ce în ce mai important și vor deschide noi piețe.

Evoluția sistemelor de comunicație mobile către generația a treia (*3rd Generation* sau 3G) are la bază, cel puțin în Europa, sistemul GSM (Global System for Mobile communications sau sistem global pentru comunicații mobile) [42]. Primul pas al acestei evoluții este reprezentat de GPRS (General Packet Radio Service – serviciu radio general pentru pachete de date) [17], [15], [12], [41], [44], [5], [4].

În evoluția către 3G, GPRS este continuat de Enhanced GPRS (EGPRS), [46]. EGPRS utilizează tehnologia EDGE (Enhanced Data Rates for Global Evolution), ce permite rate de transfer mai ridicate datorită utilizării unei tehnici de modulare mai performante.

Institutul European pentru Standarde în Telecomunicații (European Telecommunications Standards Institute sau pe scurt ETSI) a dezvoltat specificațiile pentru GPRS ca parte a GSM.

GPRS este un serviciu radio bazat în întregime pe comutarea de pachete, implementat *peste* GSM, care este bazat pe comutarea de circuite.

Avantajele serviciilor cu comutare de pachete față de comutarea de circuite la transferul datelor

Aplicațiile bazate pe transferul de date sunt caracterizate de un caracter neregulat (*bursty*) al traficului [15]. Datorită acestui lucru este mult mai eficient să se utilizeze metoda cu comutare de pachete (*Packet Switching – PS*) decât metoda cu comutare de circuite (*Circuit Switching – CS*).

Din punct de vedere al rețelei, comutarea de pachete este mai eficientă deoarece mai mulți utilizatori pot partaja un canal radio, pe când la comutarea de circuite, un canal este dedicat unui singur utilizator pe întreaga durată a conexiunii. În acest mod, capacitatea de trecere a rețelei va crește și deci resursele radio foarte limitate ale rețelei pot fi folosite de o manieră mult mai eficientă.

Din perspectiva utilizatorului rețelei, o conexiune de tip *PS* elimină timpii lungi care apar înainte de stabilirea conexiunii (stabilirea unei conexiuni *CS* necesită proceduri consumatoare de timp pentru a rezerva resurse în rețea între sursă și destinație). În acest fel utilizatorul poate fi *mereu conectat* (*always connected*), adică, după ce se realizează procedura de atașare la rețea (*attach*), utilizatorul poate transmite sau recepționa date ori de câte ori dorește, iar întârzierile care apar în obținerea resurselor sunt mult mai mici decât în cazul comutării de circuite.

Ratele de transfer a datelor pot fi mult mai mari în cazul *PS* deoarece unui utilizator i se pot aloca mai multe canale pe perioada cât acesta efectuează transferul de date.

De asemenea taxarea se poate face după volumul datelor transferate, nu după durata conexiunii, ca în cazul *CS*. Taxarea după volum apare ca mult mai potrivită cu caracteristicile traficului de date, care este foarte fluctuant (*bursty*), alternând perioadele active cu cele inactive, când nu se transferă date.

E important de menționat că metoda *PS* permite o diferențiere a calității serviciilor oferite utilizatorilor, în funcție de diverse criterii, de exemplu, în funcție de tipul și valoarea subscripției utilizatorului.

În concluzie, GPRS este un serviciu cu comutare de pachete oferit peste GSM, care este o rețea cu comutare de circuite. Comutarea de pachete prezintă multe avantaje față de comutarea de circuite, atât din punctul de vedere al rețelei, cât și din cel al utilizatorului.

O cerință majoră din punctul de vedere al operatorilor de telefonie mobilă este ca noul serviciu (GPRS) să necesite cât mai puține schimbări (și deci cât mai puține investiții) ale structurii GSM existente.

2.1.2 Comparație între GSM și GPRS

Această secțiune prezintă arhitecturile sistemelor GSM și GPRS, scoțând în evidență schimbările apărute prin introducerea GPRS.

GSM

Sistemul GSM este bazat pe conceptul de celulă, care implică un număr de stații de bază (*Base Transceiver Stations – BTS*) [44]. Fiecare stație de bază acoperă o arie geografică numită *celulă*. Celulele sunt de obicei reprezentate ca hexagoane regulate, astfel încât planul poate fi ușor acoperit de o astfel de structură de celule, dar în realitate celulele pot avea o formă neregulată. O altă convenție este că stația de bază

este omnidirecțională și că este situată în centrul celulei, dar în realitate o stație de bază poate servi mai multe celule (de obicei trei celule) folosind o antenă cu sectoare de 120° .

În această lucrare se adoptă convenția celulelor hexagonale având stația de bază în centrul lor deoarece o descriere mai exactă nu este necesară pentru investigațiile din această teză.

Fiecărei celule îi sunt alocate un număr de *frecvențe purtătoare (carriers)*, fiecare frecvență fiind împărțită în opt canale (sloturi de timp).

Deoarece spectrul de frecvențe este limitat, în GSM se reutilizează aceleași benzi de frecvențe, dar în celule diferite, situate la o anumită distanță, cu scopul de a limita interferențele. Acest concept se numește *reutilizarea frecvențelor (frequency reuse)* și este măsurat prin *factorul de reutilizare a frecvențelor – frequency reuse factor*. Operatorul rețelei poate asigna un anumit număr de frecvențe purtătoare fiecărei celule, în funcție de numărul de utilizatori din celula respectivă și din celulele învecinate.

Tabelul 2.1 prezintă principalele caracteristici ale standardului GSM [44].

	GSM 900	GSM 1800
Banda de frecvență	890-915 MHz 935-960 MHz	1710-1785 MHz 1805-1880 MHz
Lățimea unei subbenzi	25 MHz	75 MHz
Banda unui canal radio	200 kHz	200 kHz
Număr de purtătoare	124	374
Intervale temporale per purtătoare	8	8
Acces multiplu	TDMA/FDMA	TDMA/FDMA
Dimensiunea tipică a celulelor	<300 m - 35 km	<100 m - 15 km

Tabelul 2.1: Caracteristicile principale ale GSM

Sistemul GSM folosește o combinație de tehnologii de acces (tehnici care specifică modul de partajare a resurselor radio între utilizatori): acces multiplu cu divizarea frecvenței sau FDMA (*Frequency Division Multiple Access*) și acces multiplu cu divizarea timpului sau TDMA (*Time Division Multiple Access*). Acest lucru înseamnă ca se utilizează frecvențe purtătoare diferite, iar în cadrul aceleiași frecvențe are loc partajarea cu divizarea timpului.

Divizarea frecvenței (FDMA) se referă la faptul că la GSM 900 se utilizează o bandă de frecvențe cuprinsă între 890 MHz și 915 MHz în *uplink* (legătura dinspre stația mobilă MS spre stația de bază) și o bandă de frecvențe între 935 MHz și 960 MHz în *downlink* (de la stația de bază spre stația mobilă), cât și la faptul că

în GSM sunt disponibile 124 de purtătoare în fiecare direcție (*uplink* și *downlink*), cu 200 kHz per purtătoare (banda unui canal radio de 200 kHz).

Divizarea timpului TDMA) constă în faptul ca pentru fiecare frecvență purtătoare se utilizează un număr de 8 intervale temporale (*time slots*).

Arhitectura sistemului GSM este organizată ierarhic și este prezentată în figura 2.1 [17],[12]: stațiile mobile MS (*Mobile Stations*) dintr-o celulă comunică cu nodul BTS (*Base Transceiver Station*) care deservește acea celulă. Mai multe BTS sunt controlate de către controlerul stației de bază BSC (*Base Station Controller*), iar grupul format din BTS și BSC se numește subsistemul stației de bază BSS (*Base Station Subsystem*). În timp ce BTS este responsabil cu interfața radio către MS, BSC controlează resursele radio și gestionează transferurile (*handovers*). Atunci când o stație mobilă se deplasează dintr-o celulă în alta se spune ca are loc *un transfer* (*handover*).

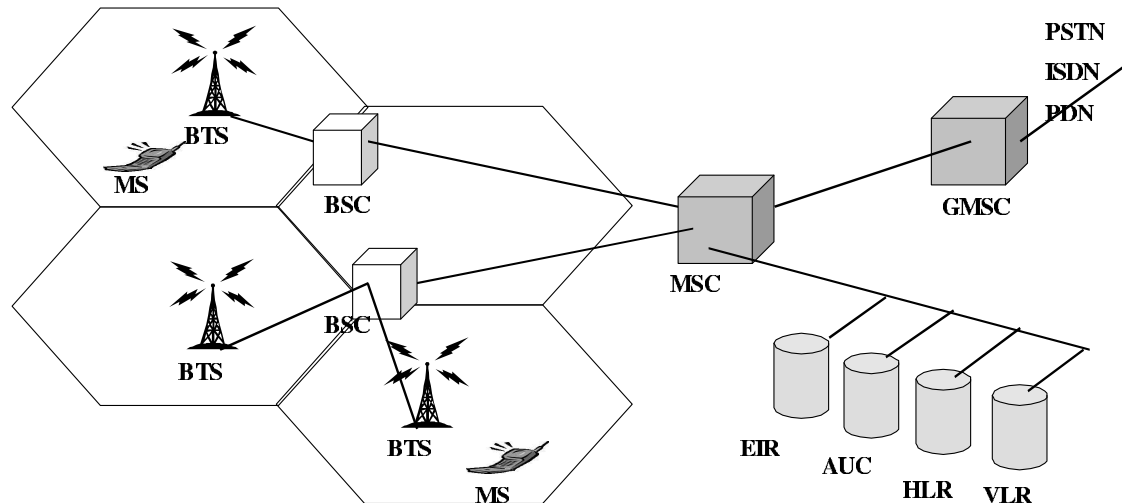
Centrul de comutație pentru servicii mobile MSC (*Mobile Switching Center*) rutează traficul GSM către rețelele externe, cum ar fi PSTN (*Public Switch Telephone Network*), ISDN (*Integrated Services Data Networks*) sau alte rețele cu comutare de circuite sau de pachete (*Packet Data Networks – PDN*). Interfața cu rețelele externe este asigurată de un MSC dedicat, GMSC (*Gateway Mobile Switching Center*).

Celulele radio pot fi grupate, un grup de celule fiind servit de către un BSC. Mai multe grupuri de celule formează o *arie de localizare LA* (*Location Area*), iar una sau mai multe LA formează o *regiune administrativă*, care este asignată unui MSC.

Standardele GSM specifică o serie de baze de date: registrul pentru abonații rezidenți HLR (*Home Location Register*), registrul pentru abonații aflați temporar într-o zonă de acoperire VLR (*Visited Location Register*), centrul de autentificare AUC (*Authentication Center*) și registrul de identitate a echipamentelor EIR (*Equipment Identity Register*).

HLR stochează date permanente (profilul utilizatorilor) și temporare (de exemplu locația curentă a unui utilizator) pentru toți utilizatorii înregistrați la un operator. Un registru VLR stochează date pentru utilizatorii aflați în zona sa, de exemplu identificatorul temporar al fiecărui utilizator. AUC conține informații legate de securitatea rețelei, iar EIR conține date despre echipament, pentru a se preveni utilizarea echipamentului furat.

Câteva din interfețele dintre nodurile unei rețele GSM sunt: Um – interfața dintre MS și BTS, Abis – interfața dintre BTS și BSC, A – interfața dintre BSC și MSC.



BTS = Base Transceiver Station
BSC = Base Station Controller
BSS = Base Station Subsystem (BTS+BSC)
MSC = Mobile Switching Center
GMSC = Gateway MSC
MS = Mobile Station

HLR = Home Location Register
VLR = Visitor Location Register
EIR = Equipment Identity Register
AUC = Authentication Center
PSTN = Public Switched Telephone Network
ISDN = Integrated Services Digital Network
PDN = Packet Data Network

Figura 2.1: Arhitectura sistemului GSM

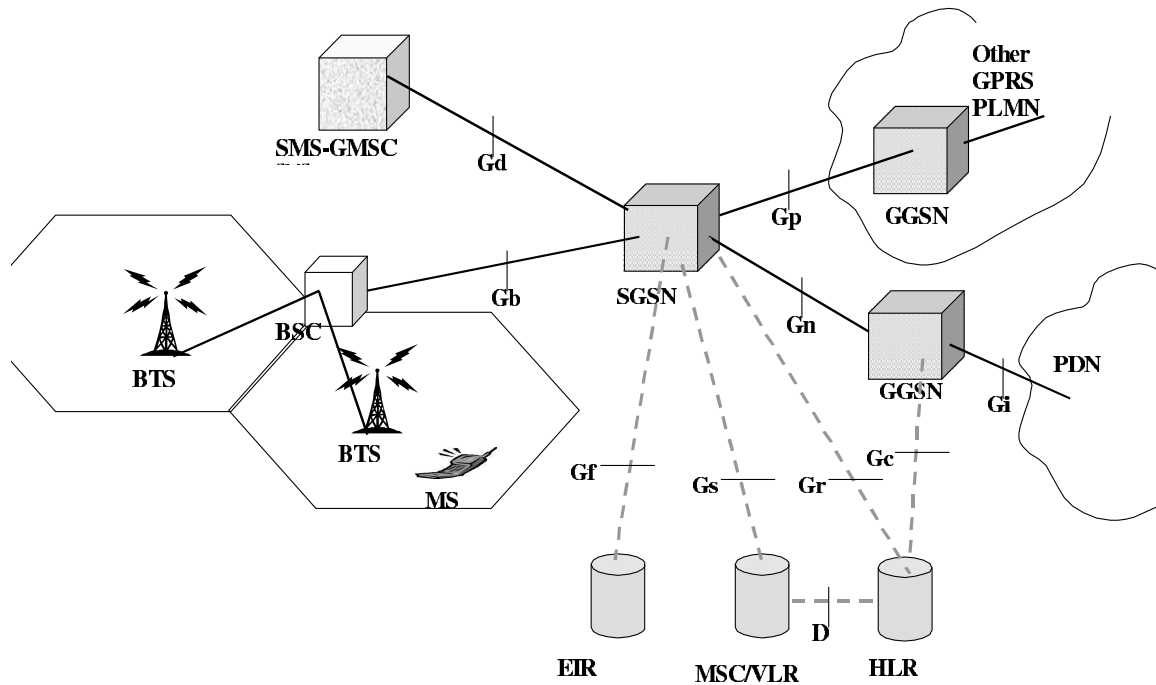
GPRS

Arhitectura sistemului GPRS este prezentată în figura 2.2 [17], [12].

GPRS folosește aceeași interfață radio ca și GSM, ceea ce face ca partea radio a unei rețele GSM poate fi făcută compatibilă cu GPRS fără dificultăți.

Deoarece costul BSS reprezintă 79% din costul echipamentelor hardware ale unei rețele celulare de tip GSM [44], nodurile BTS și BSC din GSM au fost menținute și în GPRS, împreună cu interfețele Um și Abis. La BSC se adaugă un nou echipament hardware, unitatea de control a traficului de pachete PCU (*Packet Control Unit*), care are scopul de a gestiona pachetele GPRS. Chiar dacă hardware-ul din BTS și BSC se păstrează din GSM, ambele noduri au nevoie de modificări ale software-ului pentru a putea opera în GPRS.

Față de nodurile existente în GSM, pentru GPRS se adaugă două noi (tipuri de) noduri: un nod suport GPRS server – SGSN (*Serving GPRS Support Node*) și un nod suport GPRS de tranzit – GGSN (*Gateway GPRS Support Node*). Ele formează



SGSN = Serving GPRS Support Node
GGSN = Gateway GPRS Support Node
SMS-GMSC = SMS-Gateway MSC
PDN = Packet Data Network
PLMN = Public Land Mobile Network

Figura 2.2: Arhitectura sistemului GPRS

o nouă clasă de noduri, nodurile suport GPRS sau GSN (GPRS Support Nodes) și sunt responsabile cu livrarea și rutarea pachetelor de date între stațiile mobile MS și rețelele externe de pachete de date PDN [12], [65].

Nodul SGSN este utilizat pentru rutarea pachetelor în aria sa de servire, pentru managementul mobilității, pentru funcții de autentificare și taxare. El conține un registru de locații care stochează informații pentru toți utilizatorii GPRS care sunt înregistrați la acel SGSN. Aceste informații pot fi de exemplu profilul utilizatorului, celula în care se află utilizatorul, etc.

Nodul GGSN este utilizat ca interfață între rețeaua internă principală (*backbone network*) și rețelele externe de pachete de date. GGSN realizează funcții de autentificare și de taxare, dar cel mai important rol al său este de a converti pachetele GPRS în unități de date PDU (*Protocol Data Units* sau *Payload Data Units*) ale protocolului extern de date PDP (*Packet Data Protocol*), protocol care poate fi IP (*Internet Protocol*) sau X.25. Această conversie (pachete GPRS în PDU) are loc în direcția *uplink*,

iar în *downlink* are loc conversia inversă. Tot în *downlink* GGSN mai realizează conversia adresei PDP a utilizatorului (de exemplu adresa IP) în adresa sa GSM/GPRS. Pachetele externe sunt trimise SGSN-ului la care este conectat utilizatorul destinație.

Structura ierarhică se menține și în GPRS, adică un BSC poate servi mai multe BTS (o celulă este deservită de către un BTS), un SGSN controlează un anumit număr de BSC-uri, dar pe de altă parte, un GGSN poate fi interfața mai multor SGSN-uri cu rețelele de pachete de date externe (PDN), iar un SGSN poate fi conectat la mai multe GGSN-uri, efectuând rutarea pachetelor spre și dinspre mai multe rețele de pachete de date externe.

În interiorul rețelei GPRS informația utilă sub forma unor PDU este încapsulată în GSN-ul sursă și decapsulată la GSN-ul destinație, protocolul utilizat între două GSN-uri fiind IP. Procesul prin care se realizează încapsularea și decapsularea datelor se numește *tunelare (tunneling)*.

În GPRS se mențin nodurile baze de date din GSM, dar li se adaugă noi interfețe și noi funcții, adaptate la cerințele GPRS.

Mesajele de tip SMS (*Short Message Service*) se pot utiliza și în GPRS folosind nodul SMS-GMSC (*SMS Gateway MSC*). Acesta este conectat la SGSN prin interfața Gd.

Interfeța dintre două GSN-uri este Gn dacă ele se află în interiorul aceleiași rețele publice terestre PLMN (*Public Land Mobile Network*) și Gp dacă cele două GSN-uri sunt situate în PLMN-uri diferite (a se vedea figura 2.2). Alte interfețe, așa cum apar ele în figura 2.2 sunt: Gb, între BSC și SGSN, Gf, între SGSN și EIR, Gs și Gr, care interconectează SGSN cu MSC/VLR și respectiv HLR, iar interfața dintre MSC/VLR și HLR se numește D. Un GGSN poate fi conectat cu HLR prin interfața Gc și cu o rețea de pachete de date externă (PDN) prin interfața Gi. Așa cum s-a menționat anterior, interfața radio Um dintre MS și BTS, precum și interfața Abis dintre BTS și BSC sunt aceleași ca și în GSM.

2.2 Algoritmi de alocare a resurselor pentru rețele cu comutare de pachete

Pentru a putea asigura diferențierea calității a serviciului (*QoS – Quality of Service*) într-o rețea de tip GPRS, utilizatorii trebuie să aibă diferite niveluri de precedență sau prioritate, ceea ce implică faptul că lor li se asociază diferite *ponderi*.

Dintre algoritmi de planificare cunoscuți în literatura de specialitate care ar putea satisface cerințele de QoS ale utilizatorilor, având proprietatea că fiecărui utilizator i se asociază o pondere și că resursele rețelei sunt alocate apoi în concordanță cu aceste

ponderi, cei mai cunoscuți sunt algoritmi de tip WFQ (*Weighted Fair Queueing*), numiți și algoritmi de tip PFQ (*Packet Fair Queueing*). În continuare se va descrie mai în detaliu algoritmi din această categorie.

2.2.1 Algoritmi de tip Packet Fair Queueing

Algoritmi de tip PFQ au la bază un algoritm idealizat numit Generalized Processor Sharing (GPS), propus de Parekh și Gallager [73].

Algoritm idealizat GPS

Algoritm GPS face presupunerea că pachetele de date se pot divide oricât de mult (sunt infinit divizibile). Fluxurile de pachete sunt modelate sub forma curgerii unor fluide printr-un canal de capacitate C , algoritmul GPS fiind numit din această cauză și *Fluid Fair Queueing (FFQ)*. Fiecărui flux f îi este asignată o pondere r_f . De-a lungul oricărui interval de timp infinitesimal Δt , unui flux nevid (*backlogged*) îi este alocată o fracțiune din capacitatea canalului egală cu

$$C \cdot \Delta t \cdot \frac{r_f}{\sum_{i \in B_{FFQ}(t)} r_i} \quad (2.1)$$

unde $B_{FFQ}(t)$ este mulțimea fluxurilor care sunt nevide la momentul t .

Singura restricție pentru ponderi este să fie numere reale pozitive, aceasta deoarece importantă nu este atât valoarea ponderilor, ci raportul lor. Astfel, din orice set de coeficienți (ponderi) având valori reale pozitive r'_f se poate obține un alt set de coeficienți r_f prin divizarea coeficienților inițiali la suma lor:

$$r_f = \frac{r'_f}{\sum_{i=1}^N r'_i}$$

unde N este numărul total de cozi într-un server GPS. Suma coeficienților r_f astfel obținuți va fi 1.

Dacă valorile coeficienților r_f sunt interpretate ca și lățimea de bandă a unui flux de date, atunci aceste valori devin în mod evident importante și suma lor trebuie să fie egală cu capacitatea totală a canalului C : $\sum_{i=1}^N r_i = C$.

Așa cum am menționat și mai înainte, GPS este o disciplină de servire idealizată în care pachetele de date nu sunt transmise ca entități, deci acest algoritm nu poate fi implementat în această formă. Parekh și Gallager au propus un algoritm numit GPS pachetizat sau PGPS (*Packetized GPS*) care aproximează GPS și care poate fi implementat practic.

Algoritmul GPS pachetizat (PGPS)

Dacă se notează cu F_p timpul la care pachetul p pleacă (se termină servirea lui) în GPS, atunci PGPS este o schemă de tip *work conserving* care servește pachetele în ordinea crescătoare a valorilor F_p . O schemă de servire este de tip *work conserving* dacă serverul este ocupat (lucrează) atâta timp cât cel puțin o coadă nu este goală (are pachete).

Dacă presupunem că serverul devine liber la momentul t , este posibil ca pachetul care ar trebui să plece conform algoritmului GPS (pachetul cu cea mai mică valoare a lui F_p) să nu fi sosit până la momentul t . În consecință, algoritmul PGPS va fi descris de următoarea regulă: ***serverul preia primul pachet care și-ar fi terminat serviciul în GPS dacă presupunem că nici un pachet nu ar mai urma să sosească după timpul curent t*** [73].

Funcționarea algoritmilor GPS și PGPS va fi ilustrată cu următorul exemplu propus de Parekh și Gallager [73]. Se consideră două sesiuni, numite Sesiunea 1 și Sesiunea 2, având pachete de diferite lungimi care sosesc după cum urmează:

- Sesiunea 1: pachetele p_1^1 , p_1^2 , p_1^3 și p_1^4 sosesc la momentele de timp 1, 2, 3 și 11, având lungimile 1, 1, 2 și respectiv 2.
- Sesiunea 2: pachetele p_2^1 , p_2^2 and p_2^3 sosesc la momentele de timp 0, 5 și 9 și au lungimile 3, 2 and 2.

Cele două sesiuni au ponderi egale $r_1 = r_2 (= 0.5)$. În acest exemplu se consideră capacitatea rețelei ca fiind $C = 1$. Se notează cu p_i^k pachetul cu numărul k din sesiunea i .

La momentul 0 numai sesiunea 2 este nevidă și timpul estimat de terminare a pachetului p_2^1 este 3.

La momentul 1 sesiunea 1 devine nevidă și cele două sesiuni vor avea fiecare o capacitate de 0.5. Deoarece o parte de lungime 1 din pachetul p_1^2 a fost deja trimisă, rezultă că timpul de terminare al acestui pachet va fi 5. Pachetul p_1^1 va fi transmis cu o viteză de 0.5 unități de pachet (de lungime) pe unitatea de timp, rezultând un timp de terminare a transmiterii de 3.

În mod similar se pot obține valorile timpilor de terminare a transmisiei pentru toate pachetele. Timpii de terminare în GPS ai pachetelor din sesiunea 1, adică p_1^1 , p_1^2 , p_1^3 și p_1^4 vor fi 3, 5, 9 și 13, iar timpii de terminare ai pachetelor din sesiunea 2 (p_2^1 , p_2^2 și p_2^3) vor fi 5, 9 și 11.

Sesiunea 1 este vidă în intervalele de timp $[0, 1]$ și $[9, 11]$, iar sesiunea 2 este vidă în intervalul $[11, 13]$ (după momentul 11).

Pachetul p_1^1 are cel mai mic timp de terminare, dar primul pachet planificat să fie transmis la momentul 0 va fi p_2^1 deoarece la momentul 0 pachetul p_1^1 nu sosise încă și deci nu putea nici pleca. Timpul de terminare pentru pachetul p_2^1 este 3. Aplicând regula prezentată mai înainte pentru PGPS, se obține ordinea în care sunt transmise pachetele în PGPS: $p_2^1, p_1^1, p_1^2, p_1^3, p_2^2, p_2^3, p_1^4$, iar timpii de terminare a transmiterii pentru pachetele din cele două sesiuni vor fi 4,5,7 și 13 pentru pachetele din sesiunea 1 și respectiv 3, 9 și 11 pentru pachetele din sesiunea 2.

Rezultatele sunt sintetizate în următorul tabel 2.2:

		Session 1				Session2		
Packet information	Arrival	1	2	3	11	0	5	9
	Size	1	1	2	2	3	2	2
$r_1 = r_2$	GPS	3	5	9	13	5	9	11
	PGPS	4	5	7	13	3	9	11

Tabelul 2.2: Exemplu cu GPS și PGPS [73]

O implementare directă a algoritmului PGPS ar avea de recalculat timpii de terminare pentru *toate* pachetele din GPS atunci când un flux de date (o sesiune) ar deveni vidă sau nevidă, lucru inacceptabil dacă numărul de sesiuni este mare. În lucrarea [73] se introduce noțiunea de timp virtual și se dă o soluție de implementare a PGPS pe baza acestuia.

Pornind de la algoritmi propuși de Parekh și Gallagher, au fost dezvoltati o serie de algoritmi care aproximează algoritmul idealizat GPS (FFQ). Acești algoritmi au fost dezvoltati pornind de la algoritmul idealizat FFQ și sunt cunoscuți sub numele generic de algoritmi de tip PFQ (*Packet Fair Queueing*). În general acești algoritmi încearcă să realizeze un compromis între complexitatea implementării și gradul de corectitudine (*fairness*).

Articolul [122] realizează o trecere în revistă a diferiților algoritmi de tip “*fair queueing*”.

De asemenea, în [118] sunt trecuți în revistă diferiți algoritmi care pot fi utilizați pentru alocarea resurselor în rețele celulare de date. Acești algoritmi se bazează pe algoritmul Fluid Fair Queueing sau pe versiunile sale pachetizate – algoritmi de tipul Packet Fair Queueing (PFQ), cunoscuți și sub numele de Weighted Fair Queueing (WFQ).

Comentarii privind algoritmi de tip Packet Fair Queueing

Algoritmi de tip PFQ sunt utilizați pe larg în routere și switch-uri deoarece pot gestiona atât traficul în timp real, cât și traficul de tip *best effort*.

Algoritmi de tip PFQ alocă fiecărui flux de date o cantitate de serviciu proporțională cu ponderea fluxului respectiv. Atunci când se utilizează și rezervarea resurselor, acești algoritmi garantează o lățime de bandă minimă pentru fiecare sesiune (flux de date), ceea ce înseamnă că algoritmi de tip PFQ pot garanta obținerea unor anumite valori pentru întârzierile *end-to-end* (capăt la capăt) pentru fluxurile supuse unor anumite restricții. Parekh și Gallager [73] au determinat analitic aceste valori garantate ale întârzierilor pentru o rețea compusă dintr-un singur nod dacă traficul de intrare este supus constrângerilor unui algoritm de admisie de tip “găleata găurită” LB (*Leaky Bucket*) la care se cunosc următorii parametri ai traficului de intrare: rata medie sustenabilă (*average sustainable rate*), rata de vârf sau maximă (*peak rate*) și gradul de “burstiness”. Acești parametri sunt importanți și pentru traficul care iese din nod și ei au fost determinați de Parekh și Gallager [73] în cazul algoritmilor GPS și PGPS.

Când algoritmi de tip PFQ sunt utilizați fără a se face rezervarea resurselor, atunci se asigură că lățimea de bandă să fie alocată în mod echitabil între fluxurile aflate în competiție pentru resurse, protejând fluxurile care au un comportament corect față de cele care nu își respectă contractul cu rețeaua. În acest mod, algoritmi de tip PFQ pot fi utilizați și pentru traficul de tip “best effort”.

Algoritmi PFQ au și limitări, așa cum se arată în [69]. Principala problemă a acestor algoritmi este că ei utilizează un singur parametru (ponderea) pentru a aloca resurse, ceea ce duce la un cuplaj între întârziere și lățimea de bandă alocată. Pentru trafic de tip timp real se poate ajunge la o sub-utilizare a rețelei, așa cum se prezintă în exemplul următor [69].

Se poate arăta că întârzierea maximă pe care o suferă un pachet p_i^k este

$$\frac{L_i^k}{r_i} + \frac{L_{max}}{C} \quad (2.2)$$

unde L_{max} este lungimea maximă a pachetelor.

Cum de obicei valoarea lui C este foarte mare comparativ cu L_{max} , al doilea termen din ecuația 2.2 poate fi neglijat și atunci întârzierea maximă a unui pachet poate fi aproximată la

$$\frac{L_i^k}{r_i} \quad (2.3)$$

Pentru un flux audio de 64 Kbps cu pachete de 160 de octeți, dacă dorim să obținem o întârziere pentru cazul cel mai defavorabil egală cu 5ms , atunci rezultă că trebuie să rezervăm o lățime de bandă de 256 Kbps , adică de patru ori mai mult decât lățimea de bandă necesară pentru fluxul audio, de 64 Kbps .

Când sunt utilizați pentru trafic de tip best-effort, algoritmi de tip PFQ tind să favorizeze aplicațiile orientate spre capacitatea de trecere, cum este de exemplu FTP (File Transfer Protocol) în dauna aplicațiilor sensibile la întârzieri, cum ar fi WWW sau telnet.

Un exemplu din [69] ilustrează această idee: o legătură de 2 Mbps este folosită în comun de un flux de date lung, care transferă un fișier de 1MB și de un flux de date având pachete scurte, de 50 KB fiecare. Dacă se presupune că respectivă legătură este servită de un algoritm de tip PFQ și că cele două fluxuri au fiecare o pondere egală cu 1 și dacă pachetele scurte sosesc unul după altul (când se termină transmiterea unui pachet, va sosi următorul), astfel încât cele două fluxuri să fie tot timpul nevide, atunci fiecare dintre cele două sesiuni îi este alocată o capacitate de servire de 1Mbps . În consecință, transferul fișierului lung va dura 8s , iar fiecare din pachetele scurte va avea nevoie de 0.4s pentru a fi transmis. Dacă însă fluxului format din pachete scurte i se asignează o pondere de trei ori mai mare decât a celuilalt flux, el va primi o capacitate de servire de 1.5Mbps și în consecință fiecare pachet va fi servit într-un timp de doar 0.27s , adică cu 33% mai scurt decât în primul caz. Aspectul interesant este că transferul fișierului de 1MB va dura tot 8s , ca și în cazul anterior, deoarece fluxul lung va “umple” golurile rămase între pachetele din celălalt flux în momentele când fluxul format din pachete scurte este vid, utilizând întreaga capacitate de servire în perioadele respective.

Concluzia care se poate trage din acest exemplu e că transferurile de pachete de date mai scurte (și de obicei mai sensibile la întârzieri) pot fi accelerate fără a afecta prea mult performanțele fluxurilor de date mai lungi (senzitive la capacitatea de trecere) printr-o asignare potrivită a ponderilor fiecărui flux. Acest lucru se poate face fie prin estimarea lungimii fiecărui flux, atunci când acesta devine nevid, ceea ce ar putea fi foarte dificil, fie prin reducerea în mod dinamic a ponderii unui flux pe măsură ce transferul se desfășoară, în momentele în care cantitatea de date transferate depășește anumite limite.

Autorii lucrării [69] propun o soluție alternativă, și anume înlocuirea algoritmilor de tip PFQ cu alți algoritmi, cum ar fi *Fair Service Curve (FSC)* – algoritmi având curba serviciului echitabilă, algoritmi care pot îmbunătăți performanțele aplicațiilor sensibile la întârzieri, fără a avea un impact negativ asupra aplicațiilor orientate pe capacitatea de trecere. Ideea algoritmilor FSC este de a asocia fiecărui flux

(fiecărei sesiuni) o curbă a serviciului. O curbă a serviciului este o curbă continuă și nedescrescătoare, compusă de obicei din două segmente având pante diferite: una din pante specifică capacitatea de trecere care îi este garantată fluxului pe termen lung, iar cealaltă pantă corespunde ratei de servire pe timpul perioadelor când transferul de date este mai intens (*bursty periods*).

2.2.2 Probleme specifice rețelelor radio

Problemele specifice transmiterii datelor în rețele radio (*wireless*), comparativ cu problemele rețelelor terestre (*wired*), sunt legate de canalele cu erori (*“bad channels”*) ([61], [70], [11]) și de consumul de putere ([11]).

Conform cu [61], calitatea serviciilor în rețelele fără fir este satisfăcută printr-o combinație de metode de rezervare a resurselor și de planificare a pachetelor. În timp ce rezervarea resurselor are loc la nivel de flux de date și a fost studiată pe larg în literatură, planificarea pachetelor (care are loc la nivel de pachete) a fost mult mai puțin investigată.

Pentru partea de planificare a pachetelor, s-ar părea că algoritmul Fluid Fair Queueing (FFQ) și versiunile sale pachetizate (PFQ) ar trebui să funcționeze și la rețelele fără fir. Totuși, modelul FFQ nu este direct aplicabil rețelelor radio din două motive: *modelului “în salvă” (bursty) al erorilor în canalele radio și variației capacității canalului și a modelului erorilor în funcție de poziția utilizatorului*. Cu alte cuvinte, anumite sesiuni nu pot transmite din cauza erorilor care le afectează legătura radio și acest lucru va avea influență asupra modului în care sesiunile sunt planificate să transmită.

O altă problemă care poate apărea e că stația de bază BS poate să nu cunoască lungimile cozilor din stațiile mobile MS la transmisia în *uplink* (dinspre stația mobilă spre stația de bază), ori mulți din algoritmi de tip PFQ presupun cunoscute lungimile cozilor. Pentru a rezolva aceste probleme, trebuie ca planificarea pachetelor să aibă loc în strânsă legătură cu protocoalele de la nivelul controlului accesului la mediu (MAC).

Au fost propuși diverși algoritmi pentru planificare în rețelele radio, cei mai interesanți fiind, *Idealized Wireless Fair Queueing (IWFQ)* ([61]) și *Channel-condition Independent packet Fair Queueing (CIF-Q)* ([70]).

Pentru fiecare sesiune se definește un *serviciu fără erori* ca fiind serviciul FFQ pentru o sesiune cu un proces de sosire a pachetelor identic cu cel al sesiunii investigate, dar având canalele complet fără erori.

Dacă se notează cu QL_{real} și QL_{FFQ} lungimile cozii reale, respectiv a celei core-

spunzătoare serviciului fără erori, atunci, la un moment de timp t , o sesiune poate fi:

- în urmă (*lagging*), dacă $QL_{real}(t) > QL_{FFQ}(t)$
- în avans (*leading*), dacă $QL_{real}(t) < QL_{FFQ}(t)$
- sincronă (*in sync*) sau satisfăcută, dacă $QL_{real}(t) = QL_{FFQ}(t)$.

Scopul algoritmilor de planificare pentru rețelele radio este “să permită sesiunilor rămase în urmă să își recupereze întârzierea prin determinarea sesiunilor care sunt în avans să renunțe la avansul lor” ([61]).

Conform cu [70], cerințele pe care trebuie să le îndeplinească algoritmi de planificare pentru rețelele radio sunt:

1. Întârzieri maxime și capacitate de trecere garantate pentru sesiunile (fluxurile) fără erori;
2. Corectitudine (*fairness*) pe termen lung garantată pentru sesiunile afectate de erori: dacă o sesiune depășește faza când este afectată de erori, ea va trebui să primească înapoi serviciul pe care l-a “pierdut” pe perioada cât a fost afectată de erori;
3. Corectitudine pe termen scurt: diferența dintre serviciul normalizat primit de oricare două sesiuni din aceeași categorie (rămase în urmă, în avans sau satisfăcute) care au fost tot timpul nevide de-a lungul unui interval de timp trebuie să fie mărginită;
4. Degradare parțială (*graceful degradation*): de-a lungul oricărui interval în care este fără erori, unei sesiuni aflate în avans trebuie să i se garanteze cel puțin o fracțiune minimă din serviciul pe care l-ar fi primit în sistemul (serviciul) fără erori.

Algoritmii menționați diferă prin maniera în care se face transferul resurselor de la sesiunile în avans către cele rămase în urmă.

Autorii lucrării [11] abordează atât problema canalelor cu erori, cât și cea a consumului de putere. Ei propun un formalism matematic care poate cuprinde atât rețelele de tipul TDMA (Time Division Multiple Access) cât și CDMA (Code Division Multiple Access). De asemenea, propun algoritmi de planificare care țin cont

de puterea consumată, acesta fiind un parametru important în rețelele radio și mobile. Fiecărui canal îi este asignat un parametru care reflectă calitatea canalului (din punct de vedere al erorilor), iar canalele sunt clasificate în funcție de acest parametru. Astfel, autorii propun trei clase de canale: *bune*, *cu erori* și *rele*, considerându-se că pentru canalele *rele* este practic imposibil să se transfere date.

Resursele rețelei sunt alocate întâi canalelor bune, iar apoi, dacă mai rămân resurse disponibile, acestea sunt alocate canalelor cu erori, dar resursele nu sunt niciodată alocate canalelor rele. Autorii evaluează prin simulare performanțele algoritmilor propuși.

2.3 Algoritmi de alocare a resurselor pentru controlul transmiterii datelor în rețele GPRS

Alocarea resurselor radio în rețele GPRS și EGPRS este un domeniu de cercetare foarte activ, existând un număr relativ mare de lucrări care abordează această problemă din diverse perspective.

2.3.1 Abordări analitice

Foh și colegii [38] construiesc un model matematic de tip Markov foarte complex pentru un sistem care combină alocarea canalelor între voce și date într-un sistem GSM/GPRS, cu transferul pachetelor în GPRS și cu modelarea caracteristicilor traficului de pachete de date de intrare în GPRS. Abordarea este globală, dar nu se concentrează asupra algoritmilor de planificare pentru transferul datelor.

În [58] este construit un model Markov multi-dimensional cu scopul de a investiga alocarea resurselor radio între traficul de date GPRS (cu comutare de pachete) și traficul de voce GSM (cu comutare de circuite). Sunt studiați patru algoritmi: de alocare fixă a resurselor sau FRA (Fixed Resource Allocation), de alocare dinamică a resurselor – DRA (Dynamic Resource Allocation), alocare fixă a resurselor cu șiruri de așteptare (cozi) – FRAQ (Fixed Resource Allocation with Queue capability) și respectiv alocare dinamică a resurselor cu șiruri de așteptare DRAQ (Dynamic Resource Allocation with Queue capabilities).

Metricile de evaluare a performanțelor sunt ratele de acceptare pentru pachetele GPRS și pentru apelurile telefonice GSM. În FRA, o cerere pentru un anumit număr de canale de date (Packet Data Channels) este acceptată doar dacă toate canalele cerute sunt disponibile la momentul respectiv, pe când în DRA o cerere poate fi

acceptată chiar dacă numărul de canale disponibile este mai mic decât numărul de canale cerute. Cererea e respinsă doar dacă nu mai există nici un canal disponibil. Versiunile cu șiruri de așteptare ale algoritmilor FRA și DRA permit apelurilor telefonice (de voce) să fie puse în așteptare într-o coadă dacă în momentul apelului nu există canale disponibile. Autorii ajung la concluzia că cele mai bune performanțe dintre cei patru algoritmi investigați le are DRAQ.

Pentru a obține un model matematic tratabil, autorii fac o serie de presupuneri care nu sunt foarte realiste:

- sosirile apelurilor GSM precum și ale pachetelor GPRS pot fi reprezentate sub forma unor procese Poisson (această presupunere este foarte discutabilă în ce privește sosirea pachetelor GPRS);
- durata convorbirilor GSM și a transmisiilor pachetelor GPRS au o distribuție de probabilitate exponențială. Acest lucru nu este neapărat valabil în cazul transmisiilor de date în GPRS;
- un canal radio nu poate fi partajat între două sau mai multe pachete GPRS.

Considerăm că aceste presupuneri și în special ultima, pot afecta acuratețea modelului propus.

Alte lucrări (Jiang și colegii [49], Pang și colegii [72], Ajib și Godlewski [2], Chang și colegii [24], Stuckmann și colegii [90], [47] etc.) studiază performanțele algoritmilor de alocare a resurselor în GPRS și EGPRS prin simulare și se vor discuta în secțiunile următoare.

2.3.2 Algoritmi de planificare pentru GPRS

Literatura de specialitate conține o mare varietate de algoritmi utilizați la planificarea resurselor radio în rețele celulare de date. O sistematizare a acestora după principiile care stau la baza lor identifică următoarele categorii:

1. Algoritmi de tip Packet Fair Queueing.
2. Algoritmi de tip Earliest Deadline First.
3. Algoritmi bazați pe cantitatea de date din cozile utilizatorilor.

Algoritmi de tip Packet Fair Queueing

Shankaranarayanan și colegii [49], [50], [87] au investigat algoritmi de tipul PFQ pentru rețele GPRS. Chang și colegii [16], [24] și [57] au studiat algoritmi de planificare de tip PFQ pentru rețele EGPRS.

Algoritmi de tip PFQ pentru rețele GPRS

Lucrarea [49] investighează performanțele algoritmului Weighted Fair Queueing (WFQ) la alocarea resurselor radio, precum și influența pe care protocolul TCP (Transmission Control Protocol) o are asupra performanțelor percepute de utilizatori. Autorii articolului studiază influența ponderilor asupra raportului performanțelor a două clase de utilizatori (utilizatori având prioritate ridicată și utilizatori cu prioritate scăzută), inclusiv cazul limită când în locul algoritmului WFQ se utilizează algoritmul de prioritizare a cozilor (*priority queueing*). În algoritmul de prioritizare a cozilor, cozile de prioritate scăzută nu sunt servite atâta timp cât cozile de prioritate ridicată nu sunt goale. În acest articol canalele sunt considerate fără erori, deoarece autorii se concentrează asupra performanțelor algoritmilor de planificare pentru trafic de date de tip *bursty*, lăsând studiul efectului erorilor în canalele radio pentru cercetări ulterioare. Performanța pentru o clasă de servire se exprimă prin “performanța sa relativă față de alte clase”. Concluzia autorilor este că raportul performanțelor între utilizatorii de clasă ridicată și cei de clasă scăzută este mai mic decât raportul ponderilor celor două clase de utilizatori, apropiindu-se de raportul ponderilor doar atunci când încărcarea rețelei este foarte mare. Explicația dată de autori este legată de caracterul variabil (*bursty*) al traficului de date.

Articolul [50] continuă cercetările din [49]. Pentru a reduce influența pe care alți factori, cum ar fi timpul de propagare și mai ales mecanismul de control al TCP (așa numitul “start lent” – *slow start*), o au asupra performanțelor algoritmilor de planificare, autorii propun înlocuirea ratei de transmisie între cele două capete cu o nouă metrică pentru performanța acestor algoritmi, și anume *rata efectivă de servire*. “Rata efectivă de servire pe care o stație mobilă o primește este calculată ca fiind cantitatea totală de date livrată atâta timp cât coada sa este activă (nevidă), împărțită la lungimea perioadei active” [50].

Autorii imaginează câteva scenarii de simulare în care utilizatorii sunt clasificați după rata lor de transfer (dată de schema de codare folosită). Utilizatorii “buni” au o schemă de codare care permite o rată de transfer dublă în comparație cu utilizatorii “cu probleme”. În lucrare se discută aspectul legat de “corectitudine” (*fair-*

ness) și concluzia la care ajung autorii este că utilizatorii “buni” suferă o degradare a performanței datorită resurselor rețelei consumate suplimentar de utilizatorii “cu probleme” pentru a transfera aceeași cantitate de date ca și utilizatorii “buni”.

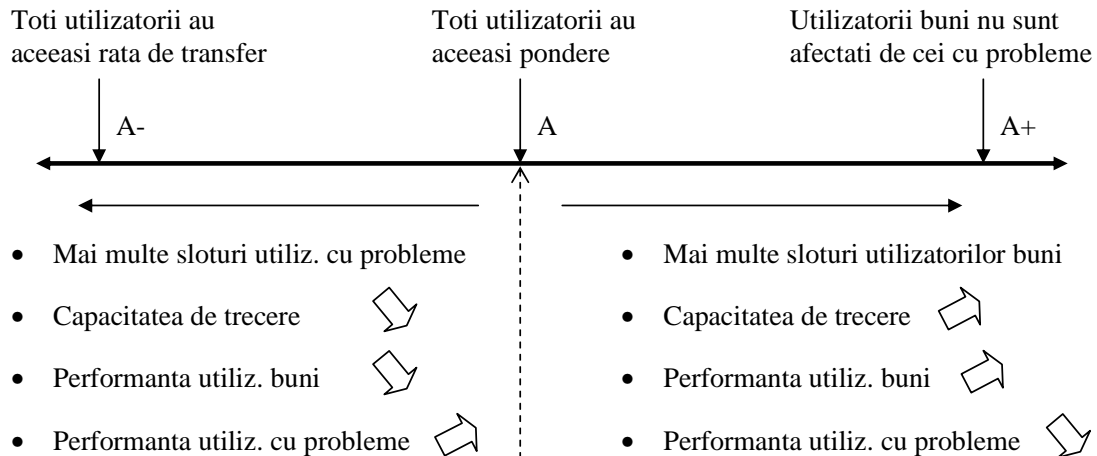


Figura 2.3: Axa de planificare a resurselor [50]

O axă de planificare (*scheduling axis*) ilustrează efectul pe care factorii de pondere (*weighting factors*) dintr-un sistem îl au asupra performanțelor utilizatorilor, în condițiile scenariilor de simulare utilizate (figura 2.3). Mijlocul axei, notat cu A , este punctul în care toți utilizatorii au aceeași pondere. Crescând factorul de pondere al utilizatorilor “buni” se ajunge la un punct notat A^+ , în care performanțele utilizatorilor “buni” (în termeni de rată de servire efectivă) nu sunt afectate de utilizatorii “cu probleme”, adică “utilizatorii buni obțin o performanță ca și cum toți utilizatorii cu probleme ar avea de asemenea legături bune”. În direcția opusă pe axă se află punctul A^- , în care toți utilizatorii au aceeași rată de servire efectivă. Mutând pe axă punctul de funcționare de la stânga, dinspre A^- spre A^+ trecând prin A , capacitatea de trecere a sistemului precum și performanța utilizatorilor buni cresc, pe când performanțele utilizatorilor cu probleme scad.

Algoritmii de planificare propuși asigură o flexibilitate ridicată la alocarea resurselor, astfel încât operatorul rețelei poate modifica cu ușurință punctul de funcționare al sistemului doar prin ajustarea ponderilor utilizatorilor. În acest mod resursele rețelei pot fi alocate utilizatorilor în diferite moduri, obținându-se compromisul dorit între performanță și corectitudine (*fairness*).

Același grup de cercetători analizează în [87] traficul de date într-o rețea radio celulară. Au fost extrase caracteristicile traficului pentru două grupuri de utilizatori:

unii având laptopuri și rulând aplicații de tip Web browsing, iar ceilalți folosind data-telefoane (*dataphones*), adică telefoane capabile să transfere date. Primul grup a inclus și utilizatori care folosesc dispozitive PDA (Personal Data Assistant). Rezultatele au fost obținute prin simularea unor sesiuni de Web browsing (folosind protocolul de transport TCP), iar modelul canalului corespunde unui canal GPRS/EDGE. Scopul acestor investigații a fost înțelegerea distribuțiilor pe care le poate avea încărcarea unui utilizator într-o astfel de rețea.

Algoritmi de tip PFQ pentru rețele EGPRS

Problema algoritmilor de planificare pentru rețele EGPRS este discutată și în [16], [24] și [57]. Primele două lucrări studiază posibilitățile de a oferi diverse clase (tipuri) de servicii în EGPRS utilizând mecanismele de planificare. Diferențierea serviciilor nu se face după clasele de trafic (care sunt conversațională, streaming, interactivă și background, în ordinea descrescătoare a cerințelor de QoS), ci după forma de subscripție a utilizatorilor, în cadrul aceleiași clase de trafic (clasa background în acest caz). Cele două clase de servicii discutate sunt clasa *premium* și clasa *de bază*.

Se consideră două grupuri de utilizatori, în funcție de calitatea legăturii lor radio: grupul G1, cu utilizatorii având legătură radio de la slabă la medie, și grupul G2, cu utilizatori având legătura radio bună. Fiecare grup este modelat ca un lanț Markov discret, grupul G1 având trei stări, iar grupul G2 două stări. Fiecare stare corespunde unei anumite rate de transfer ce poate fi atinsă: 11 kbps, 22 kbps și 33 kbps pentru grupul G1, respectiv 44 kbps și 55 kbps pentru utilizatorii din G2.

Articolele subliniază și diferențele dintre GPRS și EGPRS în ce privește calitatea serviciului. Cel mai remarcabil aspect este că în EGPRS nodul RNC (Radio Network Controller – echivalentul stației de bază din GPRS) participă la procesul de admitere a utilizatorilor (activarea contextului PDP), pe când în GPRS BS nu participă, negocierea profilului de QoS fiind făcută doar de către SGSN. Este însă mai dificil pentru SGSN să gestioneze resursele radio pentru a ști gradul lor de disponibilitate.

Algoritmii de planificare implementați sunt două versiuni ale algoritmului WRR (Weighted Round Robin): un planificator de pachete care cunoaște calitatea legăturii radio și unul convențional. Primul nu va permite utilizatorilor care au o legătură radio slabă (cei cu 11 kbps în acest articol) să folosească resursele rețelei până când nu li se îmbunătățește legătura. Modelul presupune existența a două grupuri de cozi, unul pentru utilizatorii premium, iar celălalt pentru utilizatorii de bază, dar fiecare context PDP activ (fiecare sesiune) are coada proprie.

Algoritmii menționați sunt implementați într-un banc de testare software în timp

real pentru EGPRS [57]. Bancul de test este bazat pe sistemul de operare Linux și folosit pentru evaluarea calității serviciului obținută cu diverși algoritmi de planificare utilizați în RNC sau în partea terestră de rețea. Clasele de QoS considerate sunt *time real*, *date premium* și *date de bază*, fiind o combinație între clasele de trafic și utilizarea diferențierii în funcție de subscripția utilizatorilor.

În [16], implementarea algoritmului WRR constă în alocarea unei anumite cantități de timp de servire al planificatorului RNC fiecărei clase de utilizatori. De exemplu, clasei de bază îi este alocată o secundă, iar clasei premium îi sunt alocate două secunde. Autorii ajung la concluzia neașteptată că algoritmul WRR nu poate implementa diferențierea serviciilor între utilizatori.

La această concluzie eronată s-a ajuns datorită implementării alese pentru algoritmul WRR, în combinație cu caracterul “bursty” al traficului de date. Aceste lucruri împiedică utilizatorii premium să obțină un avantaj semnificativ în urma timpului de servire mai mare ce le este alocat. Autorii constată o creștere de performanță de 20% dacă se utilizează pentru scheduler versiunea care ține cont de calitatea legăturii radio.

Concluzia greșită din [16] este corectată în [24]. De această dată implementarea algoritmului WRR constă în alocarea unui număr diferit de canale (*time slots*) într-un cadru, în funcție de clasa utilizatorilor. Utilizatorilor de bază li se alocă un singur canal, iar utilizatorilor din clasa premium li se alocă două sau patru canale. Concluzia autorilor este că “politica de alocare a mai multor canale utilizatorilor premium nu afectează performanțele utilizatorilor de bază atâta timp cât încărcarea furnizată rețelei (*the offered traffic load*) rămâne aceeași”. Această afirmație nu este întotdeauna adevărată, concluzia autorilor fiind mai degrabă o consecință a caracteristicilor traficului folosit în aceste simulări, care este foarte “bursty” și deci permite utilizatorilor de bază să își extindă lățimea de bandă pe perioadele când cozile utilizatorilor premium nu sunt active (adică sunt goale).

Algoritmi de tip Earliest Deadline First

În [72] sunt investigați trei algoritmi de planificare: FIFO (First-In First-Out), Static Priority Scheduling (SPS), care este de fapt identic cu “priority queueing” și Earliest Deadline First (EDF), concluzia fiind că cele mai bune rezultate se obțin cu EDF. Pe de altă parte algoritmul EDF e prea complex pentru a fi implementat în mod eficient în GPRS, deoarece fiecare pachet primește un marcaj de timp (*time stamp*) și apoi pachetele sunt sortate în ordinea crescătoare a valorilor acestor marcaje, iar “sortarea pachetelor în coadă necesită cel puțin $O(\log N)$ inserții pentru fiecare pachet nou sosit” [72].

O altă lucrare ([86]) studiază prin simulare doi algoritmi de planificare: Static Priority Scheduling (SPS) și Modified Earliest Deadline (MED), ultimul algoritm fiind similar cu EDF. Este comparată problema planificării din GPRS cu cea din ATM, scoțându-se în evidență diferențele dintre cele două tehnologii. Performanțele algoritmilor menționați sunt exprimate prin capacitatea sistemului și prin eficiența sistemului.

Autorii articolului obțin o “metrică a congestiei” cu scopul de a fi folosită pentru admiterea utilizatorilor (*AC – Admission Control*). Ideea de a obține o metrică de congestie care să fie apoi utilizată pentru admiterea utilizatorilor este extrem de interesantă, dar, din păcate, cele două metrici propuse în această lucrare se pot utiliza doar pentru algoritmii SPS și respectiv MED, nu și pentru alți algoritmi de planificare.

Din rezultatele prezentate în [72] și [86] se poate concluziona că algoritmii de tip Earliest Deadline First au performanțe doar cu puțin mai bune decât algoritmul *Static Priority Scheduling* (care este un caz particular de PFQ), dar au o complexitate prea mare pentru a fi implementați în rețele celulare de date de tip GPRS. Din acest motiv, algoritmii de tip EDF nu vor fi investigați în această teză.

Algoritmi bazați pe cantitatea totală de date din cozile utilizatorilor

În [64] Moulsey investighează mai mulți algoritmi de planificare pentru a fi utilizați la transferuri în timp real (de tip streaming) în UMTS, în mod *downlink*. Algoritmii utilizați sunt: *Round Robin*, *Max C/I*, *Max Queue* și *Balanced*. Algoritmul *Max C/I* alocă resursele utilizatorului cu cea mai bună legătură radio, algoritmul *Max Queue* alocă resursele utilizatorului care are cele mai multe date în coadă, iar algoritmul *Balanced* este o combinație între acești doi algoritmi, resursele fiind alocate utilizatorului care are cea mai mare valoare a produsului dintre C/I și lungimea cozii. Cele mai slabe rezultate sunt obținute pentru algoritmul *Max Queue*, iar cele mai bune pentru *Balanced*, dar și algoritmii *Max C/I* și *Round Robin* asigură o capacitate de trecere apropiată de cea obținută cu algoritmul *Balanced*.

Algoritmul denumit în [64] *Max Queue* mai este cunoscut în literatură sub denumirea de *Longest Queue*.

Așa cum s-a menționat și în secțiunea 2.2.2, Baiocchi și colegii ([11]) studiază problema alocării resurselor în canale radio afectate de erori. Ei propun algoritmi de planificare care țin cont de calitatea canalelor radio și de puterea consumată de stațiile mobile și de situația cozilor utilizatorilor. În ce privește cozile utilizatorilor,

sunt studiați doi algoritmi: unul care ia în considerare cantitatea de date din cozi (algoritmul *Longest Queue*) și un algoritm care ține cont de timpul scurs de când un anumit utilizator nu a mai transferat date (algoritmul *Oldest Queue*).

Accentul în acest articol este pus pe alocarea resurselor în funcție de calitatea canalelor radio și autorii arată că algoritmi de planificare care țin cont de calitatea legăturii radio obțin performanțe mult mai bune decât algoritmi care alocă resurse utilizatorilor independent de calitatea legăturii radio a acestora.

Situarea tezei în contextul cercetărilor prezentate

Comparativ cu cercetările prezentate până acum în această secțiune, teza de față se concentrează mai mult asupra algoritmilor de alocare a resurselor folosiți pentru controlul transmiterii datelor decât asupra unor aspecte cum ar fi interacțiunea TCP cu partea radio sau asupra canalelor de semnalizare din GPRS.

În această teză am studiat performanțele algoritmilor de planificare atunci când prioritatea utilizatorilor este în funcție de schema lor de codare, asemănător cu [49], [50], dar am investigat de asemenea problema diferențierii calității serviciilor între utilizatori, lucru ce nu s-a făcut în lucrările menționate mai înainte. În [50], Jiang și colegii încearcă să limiteze influența pe care protocolul TCP o are asupra performanțelor prin alegerea ratei efective de servire ca și metrică a performanței. În această teză am mers mai departe, neincluzând în modelul de simulare protocolul TCP, astfel încât în cazul tezei, performanțele algoritmilor de planificare utilizați la nivel MAC/RLC nu sunt mascate sau influențate de factori cum ar fi mecanismul de control al fluxului din TCP.

În modelul pe care l-am realizat, diferențierea calității serviciilor între utilizatori poate fi făcută și pe baza clasei de trafic, nu doar pe baza subscripției utilizatorilor, ca în [16] și [24], unde doar clasa de trafic background este luată în considerare.

Considerăm că este greșită concluzia din [16], că algoritmul WRR (round robin cu ponderi) nu poate implementa diferențierea QoS între utilizatori. În ce privește concluziile din [24], că utilizatorii având precedență scăzută nu sunt afectați de numărul de canale (*time slots*) alocate utilizatorilor cu precedență ridicată, acestea nu sunt corecte în general, ci mai degrabă rezultatele obținute în [24] se datorează particularităților traficului de date ales în respectiva lucrare.

Algoritmul EDF (Earliest Deadline First), folosit în [72], este prea complex pentru a fi implementat în mod eficient în GPRS, pe când algoritmul WRR, utilizat în această teză, este în același timp simplu de implementat și eficient în asigurarea

diferențierii calității serviciilor.

Sau și Scholefield [86] au propus ideea foarte interesantă a unei metrice de congestie, obținută din algoritmi de planificare utilizați pentru controlul transmiterii datelor, ce urmează să fie utilizată pentru admiterea utilizatorilor. Din păcate, metricele propuse de cei doi autori se pot aplica doar în cazurile particulare ale algoritmilor folosiți în lucrarea respectivă, și nu sunt aplicabile oricărui algoritm de planificare. De asemenea, lucrarea [86] analizează doar doi algoritmi de planificare, Static Priority Scheduling (SPS) și MED (Modified Earliest Deadline).

2.3.3 Simulatoare și emulatoare pentru GPRS

Simulator pentru studiul interfeței radio în GPRS

Lucrările [84] și [83] descriu un model de simulare realizat pentru a investiga performanțele interfeței radio atunci când sistemul GPRS este folosit pentru navigare pe Internet. Autorii au utilizat la realizarea modelului simulatorul bazat pe evenimente Cadence Bones Designer. Algoritmii de planificare implementați sunt *round robin* (*RR*) fără priorități, iar modelarea traficului de tip Internet s-a obținut printr-o combinație de e-mail, WWW și FTP. Intrările modelului de simulare sunt numărul de PDCHs (*Packet Data Channels*), raportul C/I (*carrier to interference ratio*), numărul de stații mobile (MS) și schemele de codare utilizate, iar ieșirile modelului sunt rata de erori a pachetelor și funcția de distribuție cumulativă a întârzierilor pachetelor. Protocoalele de la nivel RLC/MAC sunt modelate foarte detaliat, iar un cadru LLC se consideră echivalent cu un pachet de date (pachet IP). Din păcate autorii nu includ în modelul lor și utilizatorii de tip streaming și mai mult, algoritmul de planificare utilizat (*round robin* fără priorități) este prea simplu pentru a putea suporta diferențierea calității serviciilor utilizatorilor.

Emulatorul POTOMAC pentru GPRS

În [14] și [115] se descrie un emulator de GPRS, utilizat pentru validarea protocoalelor utilizate în rețelele radio. Emulatorul este denumit POTOMAC (Protocol Observation Tool for the Optimisation of MAC layers) în [14] și GPRS Network Emulator (GNE) în [115]. Autorii acestor lucrări consideră că doar o emulare în timp real a protocoalelor poate scoate în evidență eventualele probleme care pot reduce performanțele acestor sisteme. Emulatorul are facilități de a genera date sau poate fi conectat la surse de trafic de date de mare viteză. Emulatorul este construit pe

o platformă Windows NT și are posibilități de vizualizare și control. Platforma emulatorului constă dintr-un server PC și un client PC, fiecare din aceste calculatoare personale fiind conectat la emulator prin legături Ethernet rapide. Autorii realizează diferite experimente și ajung la concluzia că emulatorul s-a dovedit extrem de util la punerea în evidență a unor probleme neașteptate generate de interacțiunea diverselor protocoale și de anumite implementări ale acestora (de exemplu pentru FTP și TCP), la ajustarea unor parametri (de exemplu legați de TCP și interacțiunea acestuia cu partea radio), precum și la testarea în timp real a unor algoritmi de alocare a resurselor.

Emulatorul GPRSim

O atenție aparte merită sistemul GPRSim, dezvoltat de un grup de cercetători de la Aachen University of Technology (RWTH), Chair of Communication Networks (Com-Nets) și care este un simulator/emulator de GPRS. Dintre publicațiile acestui grup de cercetare se pot menționa: [95], [96], [91], [92], [97], [93], [90], [47], [89], [48].

O descriere a simulatorului/emulatorului de (E)GPRS, GPRSim, poate fi găsită în [93] și [47]. GPRSim este un simulator bazat pe evenimente descris în limbajul C++ și care folosește un sistem de biblioteci de programe numit Communication Network Class Library (CNCL). Nodurile care compun o rețea (E)GPRS, și anume stația mobilă MS, stația de bază BS, precum și nodurile suport SGSN (*Serving GPRS Support Node*) și GGSN (*Gateway GPRS Support Node*) sunt descrise în cadrul acestui simulator. Acest lucru se face prin implementarea în detaliu a protocoalelor standardizate din stivele de protocoale GSM și (E)GPRS, aspect care face ca GPRSim să fie mai degrabă un emulator decât un simulator și permite totodată un studiu foarte realist al GPRS și EGPRS. Protocoalele de pe diferite niveluri din stivele de protocoale utilizate în (E)GPRS, cum ar fi TCP/IP, SNDTCP, LLC, RLC/MAC etc., au fost întâi specificate formal în SDL (*Specification and Description Language*) și apoi translatate în C++ cu ajutorul generatorului de cod SDL2CNCL.

GPRSim mai conține de asemenea generatoare de trafic, analizoare statistice și un bloc care modelează canalele radio.

Dintre publicațiile grupului de la Aachen, lucrările [96] și [89] se concentrează mai mult asupra managementului calității serviciilor în rețelele radio, fiind mai interesante în contextul admiterii utilizatorilor (*admission control* sau AC), așa că nu vor fi discutate mai în detaliu în acest paragraf.

O primă versiune a GPRSim este menționată în [15] și apoi în [95]. În [95] se

încearcă determinarea prin simulare a capacității rețelei radio de a asigura calitatea serviciilor pentru utilizatori. Alocarea canalelor pentru GPRS se face la cerere (nu este fixă).

În [97] și [47] se discută aspecte legate de dimensionarea unui sistem GSM/GPRS. Pentru aceasta se descriu modele de simulare pentru diferite tipuri de trafic: e-mail, WWW, WAP (Wireless Application Protocol) și video streaming, modele ce utilizează diferite funcții de distribuție de probabilitate pentru caracteristicile traficului de date. Regulile (procedurile) de dimensionare propuse în ambele articole utilizează *grafuri de dimensionare* corespunzătoare tipurilor de trafic utilizate în fiecare scenariu de dimensionare. În [97] traficul GPRS este o combinație de e-mail, WWW și WAP, pe când în [47], traficul de tip WAP este înlocuit cu mult mai solicitantul video streaming.

Procedurile de dimensionare diferă puțin în funcție de faptul dacă alocarea canalelor pentru GPRS este *fixă* sau *la cerere*, dar în ambele situații criteriul țintă pentru dimensionare este probabilitatea de blocare a convorbirilor telefonice (*voice blocking probability*) pe care operatorul de rețea dorește să o garanteze utilizatorilor. Prima dată se estimează numărul mediu de utilizatori din celulă, iar apoi se calculează cantitatea totală de trafic IP generată. Operatorul își definește parametrul de QoS pe care vrea să i-l garanteze utilizatorului (de exemplu capacitatea de trecere pentru e-mail sau WWW sau timpul de răspuns al unei categorii de aplicații), iar apoi utilizează graful de dimensionare pentru a determina dacă *punctul de operare* al sistemului îndeplinește cerințele referitoare la probabilitatea de blocare a apelurilor telefonice. În cazul utilizării alocării fixe a canalelor pentru GPRS procedura se simplifică, înlocuindu-se graful de dimensionare cu câteva ecuații simple folosite pentru a determina numărul de PDCH (*Packet Data Channels*).

Problema analizei performanțelor diferitelor combinații de trafic, incluzând aplicații multimedia, este tratată în [93]. Caracteristicile traficului pentru WWW, e-mail și WAP sunt descrise prin distribuțiile lor de probabilitate, împreună cu valorile medii și dispersiile lor, iar pentru WAP, autorii folosesc rezultatele din [91]. Pentru video streaming este mult mai greu să se găsească un model statistic adecvat, ca în cazul celorlalte tipuri de trafic, așa că pentru video streaming modelul traficului se bazează pe trei video secvențe în format QCIF (Quarter Common Intermediate Format) propuse de către Video Quality Expert Group (VQEG). Cele trei secvențe, numite *Claire*, *Carphone* și *Foreman* diferă prin gradul mișcării: intensitate redusă pentru Claire și grad ridicat de mișcare pentru Foreman. Traficul video este o combinație constând în 80% Claire, 10% Carphone și 10% Foreman, rezultând un trafic mediu IP de 14.39 kbit/s.

Au fost realizate două seturi de simulări: prima constând într-o combinație de trafic alcătuită din 60% WAP, 28% e-mail și 12% WWW și trimisă unei rețele GPRS, iar a doua simulare conține un trafic IP compus din 10% streaming, 63% e-mail și 27% WWW, trimis unei rețele EGPRS. Este variat numărul de stații mobile și numărul de PDCH disponibile și sunt analizate rezultatele simulării. Indicatorii de performanță folosiți sunt:

- *capacitatea de trecere medie per utilizator (mean throughput per user)* : reprezintă traficul IP măsurat doar în timpul perioadelor active (când se transmit date). Acest parametru reprezintă o caracteristică importantă a calității serviciului, așa cum este aceasta percepută de către utilizatori.
- *timpul mediu de răspuns al aplicațiilor (mean application response time)* : este intervalul scurs între momentul când utilizatorul lansează o cerere și momentul când obiectul cerut este complet recepționat.
- *întârzierea medie a datagramelor IP (mean IP datagram delay)* : întârzierea medie capăt la capăt a datagramelor IP. Acest parametru este considerat extrem de important pentru video streaming.
- *capacitatea de trecere medie pe celulă (mean throughput per cell) sau capacitatea medie a sistemului (system throughput)*: cantitatea totală de date IP transmise împărțită la durata simulării.
- *utilizarea PDCH*: procentajul de PDCH care transportă informații de date sau de control din numărul total de PDCH (inclusiv cele goale – *idle PDCHs*).

Calitatea canalelor radio pentru EGPRS este determinată în funcție de dimensiunea celulei, mobilitatea utilizatorilor și de alți parametri, folosind rezultatele din [92].

Concluzia autorilor este că traficul Internet (e-mail și WWW) și traficul de tip WAP se pot combina fără a-și înrăutăți reciproc performanțele. Dimensiunea mică a pachetelor WAP le permite să “coexiste” foarte bine cu alte tipuri de trafic, chiar în condițiile unei încărcări mari a rețelei.

Situația este complet diferită dacă se combină video streaming cu trafic de tip Internet: chiar dacă procentajul de video streaming din combinație este redus, doar un număr mic de utilizatori de tip streaming pot fi serviți într-o celulă EGPRS. Când sunt disponibile 4 PDCHs, capacitatea de trecere țintă de 14.39 kbit/s se poate obține doar dacă în celulă sunt cel mult 6 stații mobile. Dacă numărul de PDCH disponibile este 8, atunci numărul maxim acceptabil de utilizatori din celulă poate ajunge până

la 10-15. Autorii sugerează acordarea unor privilegii speciale utilizatorilor care au date de tip timp real (cum e și cazul video streaming), realizate printr-un management corespunzător al QoS, aceasta în situațiile în care încărcarea celulei este ridicată.

În [48] se compară o evaluare analitică a performanțelor accesului la Internet prin GPRS cu rezultatele obținute utilizând GPRSim. Concluzia este că, “chiar și abordările analitice care utilizează metode de modelare extrem de complexe nu sunt capabile să descrie în mod cantitativ performanțele GPRS atunci când sursele de trafic implicate au un comportament elastic” [48]. Comportamentul elastic este datorat protocolului TCP, care realizează controlul fluxului de date pentru a evita congestiile. Acest mecanism al TCP va reduce cantitatea de date generată atunci când încărcarea rețelei este mare și deci când capacitatea sistemului este redusă. Când capacitatea sistemului este ridicată, TCP va genera date în loturi (*batch*), iar traficul generat va fi foarte neregulat (*bursty*) și pachetele IP vor suferi întârzieri crescute.

O altă concluzie interesantă a acestui articole este “existența unei anumite ineficiențe în felul în care combinația dintre TCP și GPRS utilizează capacitatea sistemului” [48].

Validarea emulatorului GPRSim prin măsurători într-o rețea GPRS

Lucrarea [90] prezintă în premieră măsurători ale performanțelor unei rețele GPRS, rezultatele măsurătorilor fiind comparate cu performanțele altor tehnologii, cum ar fi ISDN (Integrated Services Data Networks) și modemuri analogice. Măsurătorile au fost realizate în decembrie 2002 la Maastricht (Olanda) într-o celulă radio a rețelei Libertel-Vodafone și se referă la calitatea serviciilor percepute de utilizatori GPRS mobili care rulează diverse aplicații Internet (e-mail, FTP, WWW) pe laptopuri. Deși chiar autorii articolului susțin că aceste măsurători nu ar putea servi la dimensionarea corectă a unei rețele GPRS, ele au fost importante pentru optimizarea protocoalelor și eliminarea unor probleme software și mai ales pentru validarea emulatorului GPRSim.

Măsurătorile au început prin determinarea timpului dus-întors (*round trip time*) până la GGSN, utilizând comanda *ping*). Acest timp măsoară intervalul dintre trimiterea și recepția unui pachet și valoarea sa este foarte importantă pentru durata timpului de stabilire a unei conexiuni TCP, care la rândul său este un parametru important în orice sesiune WWW. Măsurătorile au arătat un management ineficient al fluxurilor temporare de blocuri (*Temporary Block Flows – TBFs*) în cazurile când pachetele sau fișierele transferate au lungimi mici, în sensul că TBF-ul curent era închis

înaintea sosirii unui nou pachet, ceea ce determină întârzieri suplimentare pentru stabilirea unui nou TBF. Pentru a remedia această situație autorii propun conceptul de “*open ended TBFs*”, adică TBF-uri care să rămână deschise câteva secunde după ce datele din cozi au fost transmise, astfel cât la sosirea unui nou pachet de date, acesta să găsească TBF-ul deschis și deci să îl poată reutiliza. În sistemele GPRS mai noi este implementat acest concept sub numele de eliberare întârziată a TBF-ului (*delayed TBF release*) [4].

Următoarele măsurători au vizat transferul prin FTP al unui fișier de 2.7 MB prin Internet. Cum în această situație nu au fost necesare re-stabiliri ale unor TBF-uri, concordanța între simulări și măsurători a fost foarte bună, autorii considerând că aceasta este o foarte bună validare pentru GPRSim. Ultimul set de măsurători a fost făcut pentru sesiuni WWW, folosind doar fișiere de lungimi mici. Simulările au produs performanțe mai bune decât cele măsurate în rețeaua reală, explicația autorilor fiind legată de întârzierile mari cauzate de semnalizări în rețeaua reală, dar autorii au considerat că aceste probleme vor fi eliminate la versiunile ulterioare de software pentru GPRS.

Situarea tezei în contextul cercetărilor prezentate

Comparativ emulatorul POTOMAC dezvoltat de Bourgois și colegii [14], [115] și cu emulatorul GPRSim dezvoltat de Stuckmann și colegii ([96], [91], [92], [97], [93], [90], [47]), în această teză nu am dorit să construim un emulator complex de GPRS sau EGPRS, ci doar un simulator mult mai simplu, dar care să poată fi folosit pentru studiul performanțelor algoritmilor de alocare a resurselor, atât pentru controlul transmiterii datelor (*TC – Transmission Control*), cât și pentru admiterea utilizatorilor (*AC – Admission Control*).

În felul acesta am dorit să evităm interacțiunile complexe care au loc între diversele protocoale folosite în GPRS și să ne concentrăm doar asupra problemelor ce țin de performanțele algoritmilor de alocare a resurselor (*scheduling*). De asemenea, metoda de lucru pe care o propune teza permite abordarea diverselor probleme de cercetare pe rând, una câte una, ceea ce duce la o mai bună înțelegere și interpretare a rezultatelor obținute prin simulare, evitând astfel greșeli de interpretare cum sunt cele din [16] și [24].

2.3.4 Alocarea canalelor între traficul de voce și cel de date

O problemă care apare în GPRS este aceea că resursele radio sunt utilizate în comun de partea de voce (GSM), care este cu comutare de circuite, cu partea de date, cu co-

mutare de pachete. Pentru o convorbire telefonică (voce) se alocă un număr de canale (de obicei un singur canal în fiecare direcție, *uplink* și *downlink*), dar aceste canale trebuie alocate pe toată durata convorbirii, în timp ce traficul de date realizează o alocare mult mai flexibilă a canalelor radio: o sesiune de transfer de date primește de obicei mai mult de un singur canal, dar numai pe durata cât are loc efectiv transferul datelor, nu pe toată durata sesiunii. Cerința de a aloca permanent canale pentru convorbirile telefonice face ca acestea să fie considerate prioritare față de traficul de date la alocarea resurselor (canalelor), putându-se ajunge chiar la situația ca unei sesiuni de date să i se reducă numărul de canale datorită apariției unei convorbiri telefonice, căreia trebuie să i se aloce un canal.

Lindemann și Thümmel au studiat în [59] și [60] alocarea canalelor între traficul cu comutare de circuite (GSM) și cel cu comutare de pachete GPRS. În [59] se prezintă un simulator cu timp discret, iar alocarea canalelor între GSM și GPRS poate fi realizată fie *static*, fie *dinamic*. Pentru a asigura diferențierea calității serviciilor în GPRS, algoritmul utilizat pentru planificare este WFQ (Weighted Fair Queueing). Au fost colectați câțiva indicatori de performanță, cum ar fi traficul mediu transportat (cantitatea medie de date transferate) și probabilitatea de pierdere a pachetelor.

Autorii includ în modelul de simulare și mobilitatea utilizatorilor, prin considerarea unui *cluster* de șapte celule hexagonale. Celula studiată este cea din mijlocul grupului, iar cele șase celule înconjurătoare sunt folosite doar pentru a modela mobilitatea utilizatorilor GSM și GPRS. În acest model, toți utilizatorii GPRS folosesc tot timpul schema de codare CS2.

Încărcarea pentru fiecare utilizator GPRS este generată cu ajutorul unui *proces Poisson modulat Markov MMDP* (*Markov-Modulated Poisson Process*) cu două stări. Cele două stări reprezintă modul *bursty*, respectiv *non-bursty*, caracterizate prin rata de sosire a pachetelor λ_1 și respectiv λ_2 . Timpul mediu cât procesul se găsește în stare *bursty*, respectiv *non-bursty* este $1/\alpha$ și respectiv $1/\beta$, ceea ce înseamnă că rata medie de sosire a pachetelor este $\lambda_{avg} = (\beta \cdot \lambda_1 + \alpha \cdot \lambda_2) / (\alpha + \beta)$, iar gradul de *burstiness* al MMDP este $B = \lambda_1 / \lambda_{avg}$.

Un pachet IP este stocat întâi într-o *coadă de acces* până când va exista un canal disponibil pentru el, după care este mutat într-o *coadă de transfer*, unde este servit în concordanță cu algoritmul WFQ adoptat pentru planificare.

Dacă se adoptă modul fix de alocare a resurselor, atunci se alocă un număr de canale fizice pentru GPRS, iar canalele rămase se alocă pentru GSM. În cazul alocării dinamice a resurselor se alocă un număr de canale pentru GPRS, dar canalele rămase pot fi alocate de asemenea pentru GPRS “la cerere”, dacă nu sunt ocupate de un

apel GSM. Un canal din setul de canale folosit de GSM care este ocupat pe moment de o sesiune GPRS va fi imediat eliberat de GPRS dacă un apel de voce (GSM) cere canalul respectiv.

Concluzia articolului este că metoda de alocare dinamică are performanțe mult mai bune decât alocarea fixă a resurselor.

Problemele descrise mai înainte sunt abordate analitic în [60], folosind o modelare cu lanțuri Markov cu timp continuu. De asemenea se mențin considerațiile inițiale din [59], și anume: traficul GSM și GPRS generate conform unor procese Poisson modulate Markov, durata convorbirilor GSM, respectiv a sesiunilor GPRS precum și timpii de *dwell* pentru GSM și GPRS sunt variabile aleatoare cu distribuție exponențială. Prin *timp de dwell* se înțelege timpul cât o stație mobilă care are o convorbire (sesiune) în desfășurare rămâne în interiorul zonei acoperite de un BSC (rămâne în aceeași celulă), iar *durata convorbirii (a sesiunii)* este timpul cât convorbirea sau sesiunea sunt active, presupunând că aceasta se încheie în mod neforțat de un transfer nereușit dintr-o celulă în alta (*handover failure*) [59], [60]).

Rezultatele se obțin din analiza stării staționare a modelului Markov. Deoarece numărul stărilor modelului Markov este foarte mare (poate ajunge la sute de mii), autorii utilizează metode numerice de rezolvare a problemei, iar rezultatele obținute analitic sunt validate prin simulare.

Ajib și Godlewski realizează studii extensive ale problemei partajării canalelor radio între voce (GSM) și date (GPRS) în [2], [1], [3]. Ei utilizează un model de simulare foarte detaliat al nivelurilor MAC și RLC din stivele de protocoale GPRS. Sunt simulați doi algoritmi de planificare, primul algoritm fiind executat la stabilirea conexiunii și utilizat pentru distribuirea canalelor de trafic între stațiile mobile (MS), iar al doilea algoritm se folosește pentru a determina cum sunt partajate resursele unui astfel de canal de trafic între stațiile mobile care sunt conectate la el.

Lucrarea [6] își propune să studieze prin simulare comportarea sistemelor de comunicații care combină partea de voce cu cea de date, (de exemplu un sistem GSM/GPRS). Autorii implementează un algoritm de alocare a resurselor numit HRRRA (*Hybrid Radio Resource Allocation*), pentru partea de GPRS alocându-se un număr fix de canale, iar parametrii utilizați pentru evaluarea performanțelor algoritmului sunt întârzierile pachetelor și probabilitatea de blocare a convorbirilor telefonice (*voice blocking probability*).

Problema partajării resurselor (în principal a canalelor radio) între traficul bazat pe comutarea de circuite (GSM) și cel cu comutare de pachete (GPRS) nu a fost studiată în mod particular în această teză, dar în modelul de simulare dezvoltat am inclus totuși influența pe care traficul GSM o are asupra traficului de date (GPRS), prin utilizarea unui parametru B , reprezentând numărul de canale disponibile pentru GPRS. Acest parametru poate avea diverse distribuții de probabilitate, rezultate din caracteristicile traficului de voce (GSM) din rețea.

2.4 Problema admiterii utilizatorilor în rețele celulare de date

Pentru a putea asigura utilizatorilor unei rețele parametrul de calitate a serviciului negociați, o condiție este să se evite acceptarea unui număr prea mare de utilizatori în sistem. Pe de altă parte se dorește maximizarea utilizării resurselor rețelei, deci algoritmi de admitere a utilizatorilor trebuie să asigure un compromis cât mai bun între aceste două cerințe.

Algoritmii utilizați pentru admiterea utilizatorilor în rețele mobile de date combină cerințele algoritmilor de admitere a utilizatorilor din rețele fixe de date și ale algoritmilor din rețele mobile utilizate pentru voce.

Problema mobilității utilizatorilor este comună rețelelor mobile de date și de voce, dar traficul de date are cerințe mult mai complexe în ceea ce privește alocarea resurselor rețelei.

Admiterea utilizatorilor în rețele mobile de voce

Principalii parametri pe care îi urmărește un algoritm de admiterea utilizatorilor în rețelele mobile de voce sunt probabilitatea de întrerupere a unei convorbiri (*call dropping probability*) și probabilitatea de blocare a unei convorbiri (*call blocking probability*). Primul indicator se referă la faptul că o convorbire în curs de desfășurare nu mai poate fi continuată și este deci întreruptă de către rețea, datorită faptului că, prin mutarea utilizatorului dintr-o celulă în alta, în noua celulă nu există suficiente resurse pentru a fi alocate respectivei convorbiri. Probabilitatea de blocare a unei convorbiri se referă la situația când un apel telefonic nu este acceptat datorită faptului că nu există suficiente resurse pentru a-i fi alocate. În general se consideră că este mult mai neplăcut pentru un utilizator să i se întrerupă o convorbire în curs de desfășurare decât să i se blocheze un apel nou. Din această cauză algoritmi de AC pentru rețelele mobile de voce încearcă să minimizeze probabilitatea de blocare

a convorbirilor, cu cerința de a menține probabilitatea de întrerupere a convorbirilor sub un anumit prag.

În principiu reducerea probabilității de întrerupere a convorbirilor se realizează într-o celulă radio prin rezervarea unor resurse pentru utilizatorii care vin din alte celule. Cu cât se rezervă mai multe resurse radio pentru utilizatorii care urmează să vină din alte celule, cu atât crește probabilitatea de blocare a unei convorbiri inițiate în celula respectivă.

Admiterea utilizatorilor în rețele celulare de date

În cazul AC pentru rețele celulare de date, probabilitatea de întrerupere și probabilitatea de blocare a convorbirilor joacă de asemenea un rol foarte important. În plus față de rețelele mobile pentru voce, complexitatea este crescută în primul rând de faptul că resursele cerute de aplicațiile de date variază puternic în timp, spre deosebire de cazul când se transmite voce, situație în care se alocă aceeași cantitate de resurse de-a lungul întregii durate a convorbirii. Cerințele de QoS ale aplicațiilor de date sunt mult mai complexe decât în cazul vocii, diverse aplicații având cerințe diferite în ce privește întârzierile acceptate, pierderile, nivelurile de precedență, etc. Caracteristicile traficului de date sunt de asemenea foarte complexe, putând exista trafic de tip conversațional (voce), streaming, interactiv (WWW) sau background (e-mail, FTP).

Este important de menționat faptul că, în cazul rețelelor celulare de date, algoritmi de admitere a utilizatorilor trebuie ca, în plus față de menținerea probabilității de întrerupere a convorbirilor și a probabilității de blocare a convorbirilor în limitele dorite, să asigure utilizatorilor admiși calitatea serviciului pe care aceștia au negociat-o cu rețeaua.

În ce privește *rețelele fixe de date*, algoritmi de AC au fost intens studiați pentru cazul rețelelor de tip ATM (Asynchronous Transfer Mode), în [75] fiind prezentată o sinteză a acestora. În afară de faptul că algoritmi dezvoltați pentru rețele fixe de date nu au de a face cu problema mobilității utilizatorilor, algoritmi de AC utilizați pentru rețelele ATM nu pot fi utilizați în mod direct la AC în rețelele mobile de date deoarece *resursele critice ale rețelei diferă în cele două cazuri*: dacă la rețelele ATM cel mai critic parametru este gradul de ocupare a bufferelor de date, la rețelele radio parametrul critic este reprezentat de resursele radio.

Conform autorilor articolului [63], *cerințele principale* ale unui algoritm de AC pentru rețele celulare de date sunt:

- maximizarea utilizării canalelor radio (a resurselor radio ale rețelei) într-un mod echitabil pentru utilizatori,
- minimizarea, în același timp, a probabilității de întrerupere și a probabilității de blocare a convorbirilor.
- algoritmi trebuie să evite sau să minimizeze reducerea serviciului acordat conexiunilor existente.

În continuare se prezintă câteva din soluțiile cele mai interesante existente în literatură la problema admiterii utilizatorilor în rețele celulare de date.

Moorman și colegii ([63]) propun o metodă numită *threshold access sharing* (partajarea accesului cu praguri) care aplică diverse strategii de AC în funcție de încărcarea rețelei. Încărcarea rețelei (*network load*) poate fi *mică (small)*, *medie (medium)* și *mare (large)*, dar dificultatea constă tocmai în stabilirea pragurilor dintre aceste domenii de încărcare (mică, medie și mare). Autorii asignează pur și simplu valori acestor praguri: o încărcare până la 50% este mică, între 50% și 100% este medie, iar o încărcare peste 100% este mare.

Stuckmann și colegii [89], [94] au dezvoltat o metodă de admitere a utilizatorilor în care lățimea de bandă disponibilă este împărțită în trei regiuni, fiecare regiune fiind partajată între două clase de trafic, dacă e necesar. Când rețeaua este foarte încărcată, sesiunile care au prioritate mare pot prelua din lățimea de bandă alocată sesiunilor cu prioritate mai mică, dar numai până la o limită, astfel încât fiecare clasă de QoS are asigurată o lățime de bandă minimă garantată. Autorii nu precizează însă care sunt valorile acestor lățimi de bandă minim garantate.

Kim și colegii [53] descriu un alt algoritm de AC bazat pe praguri, combinat cu un algoritm de adaptare a lățimii de bandă (*BAA - Bandwidth Adaptation Algorithm*). Fiecare utilizator face parte dintr-o clasă de trafic multimedia și algoritmul de AC admite doar un număr de utilizatori din fiecare clasă încât să nu se depășească o valoare de prag asociată fiecărei clase. Totuși, un utilizator dintr-o anumită clasă de trafic poate fi admis chiar și în situația în care numărul de utilizatori din acea clasă a atins sau chiar a depășit valoarea de prag dacă utilizatorul vine dintr-o altă celulă (este un utilizator de tip handoff). În astfel de situații se poate ajunge la supraîncărcarea rețelei și atunci se utilizează algoritmul de BAA pentru reducerea congestiei. Algoritmul de BAA constă în reducerea lățimii de bandă alocate utilizatorilor din clasele

de trafic cu prioritate redusă. Lăţimea de bandă a acestor utilizatori este redusă de la o valoare maximă necesară până la o valoare minimă acceptabilă, procesul durând până când lăţimea de bandă totală alocată sesiunilor în curs de desfăşurare este mai mică sau cel mult egală cu lăţimea de bandă disponibilă.

Şi în cazul acestui algoritm de AC, problema dificilă constă în determinarea valorilor pragurilor. Autorii utilizează o metodă de programare neliniară pentru a rezolva problema.

O altă abordare interesantă a problemei admiterii utilizatorilor în reţelele celulare de date apare în [45]. Autorii propun o metodă de decizie teoretică bazată pe procese de decizie Markov (*Markov Decision Processes MDP*). Principala dificultate întâmpinată este că numărul stărilor modelului Markov devine foarte mare pentru probleme reale, ceea ce face ca timpul de rezolvare a soluţiei problemei MDP să devină inacceptabil de mare.

2.5 Utilizări ale logicii fuzzy în telecomunicaţii

În ultimii ani logica fuzzy si-a găsit o serie de aplicaţii în telecomunicaţii. Lucrările [40], [62] conţin sinteze de literatură privitoare la utilizarea logicii fuzzy în telecomunicaţii.

În [40] Ghosh şi colegii subliniază faptul că logica fuzzy poate asigura un cadru matematic robust pentru tratarea impreciziei existente în “lumea reală” şi prezintă câteva domenii de aplicare a logicii fuzzy existente în momentul apariţiei articolului (1998): teoria aşptării, managementul bufferelor de date, rutare, evitarea congestiilor în reţele de mare viteză, etc. Un concept interesant care apare în acest articol este cel de praguri fuzzy (*fuzzy thresholds*), menţionat în contextul teoriei aşteptării, dar care în această teză va fi extins la problema admiterii utilizatorilor (capitolul 5).

Făcând o analogie cu domeniul conducerii proceselor, în care logica fuzzy s-a impus cu foarte mult succes, Mandyam şi Srinath [62] subliniază potenţialul uriaş al aplicării logicii fuzzy în telecomunicaţii. Dintre domeniile de aplicare a logicii fuzzy cei doi autori menţionează detecţia semnalelor în canalele de comunicaţie non-gaussiene (inclusiv canale radio), egalizarea canalelor, controlul puterii consumate (de asemenea cu aplicaţii în reţelele radio), rutarea datelor, generarea de secvenţe pseudo-aleatoare şi localizarea vehiculelor, prioritizarea apelurilor într-o aplicaţie militară, etc. Autorii relevă faptul că aplicaţiile logicii fuzzy în telecomunicaţii sunt doar la început, dar că perspectivele de dezvoltare sunt foarte promiţătoare.

În această secţiune se prezintă mai întâi noţiuni de bază privind mulţimile fuzzy

și logica fuzzy ([123], [67]), iar apoi sunt trecute în revistă aplicații ale logicii fuzzy în rețele ATM și în rețele celulare radio.

2.5.1 Mulțimi fuzzy. Operații cu mulțimi fuzzy

În teoria clasică a mulțimilor, o mulțime $A \subset X$ este definită ca o colecție de elemente x care aparțin unei mulțimi X , numită universul discursului. O astfel de mulțime se numește mulțime clasică sau non-fuzzy (*crisp*). O mulțime *crisp* poate fi descrisă prin funcția sa caracteristică, care, pentru fiecare element $x \in X$ are valoarea 1 dacă $x \in A$ și 0 dacă x nu aparține mulțimii A ($x \notin A$).

Pentru mulțimi fuzzy, funcția de apartenență are orice valori în intervalul de numere reale $[0, 1]$, nu doar în mulțimea discretă $\{0, 1\}$.

Definiție:

Dacă X este o colecție de obiecte notate generic cu x atunci o mulțime fuzzy A în X este o mulțime de perechi ordonate:

$$A = \{(x, \mu_A(x)) | x \in X\} \quad (2.4)$$

$\mu_A(x)$ este numită funcția de apartenență sau gradul de apartenență a lui x în A , care mapează pe X în spațiul de apartenență M (uzual $M = [0, 1]$). Dacă M conține doar cele două valori 0 și 1 atunci A nu este mulțime fuzzy și $\mu_A(x)$ este identică cu funcția caracteristică a unei mulțimi clasice. [123], [67]

Domeniul funcției de apartenență este o submulțime a numerelor reale nenegative al cărei suprem este finit (în mod obișnuit intervalul închis $[0, 1]$).

Pentru o mulțime fuzzy se definește *suportul* său astfel:

Suportul unei mulțimi fuzzy A , notat $S(A)$ este mulțimea clasică alcătuită din toate elementele x care aparțin lui X pentru care $\mu_A(x) > 0$.

La fel ca și în cazul mulțimilor clasice, și pentru mulțimile fuzzy se pot defini operații de reuniune, intersecție și complementul unei mulțimi.

Funcția de apartenență $\mu_C(x)$ pentru intersecția C a două mulțimi fuzzy A și B , $C = A \cap B$ este dată de relația:

$$\mu_C(x) = \min(\mu_A(x), \mu_B(x)) \quad (2.5)$$

Funcția de apartenență $\mu_D(x)$ pentru reuniunea D a două mulțimi fuzzy A și B , $D = A \cup B$, este dată de relația:

$$\mu_D(x) = \max(\mu_A(x), \mu_B(x)) \quad (2.6)$$

Funcția de apartenență a complementului unei mulțimi fuzzy A , notată cu $\mu_{CA}(x)$, este definită de relația:

$$\mu_{CA}(x) = 1 - \mu_A(x) \quad (2.7)$$

2.5.2 Logica fuzzy aplicată în rețele ATM

Logica fuzzy, uneori în combinație cu rețelele neuronale, a fost aplicată cu succes în rețele ATM la admiterea utilizatorilor sau la controlul parametrilor (*UPC - Usage Parameter Control*). Rețelele de tip *Asynchronous Transfer Mode (ATM)* sunt descrise în [31], [116], [98].

Catania și colegii, în [18], [21], [37], [7], [19], [20] utilizează pentru controlul parametrilor un mecanism cu ferestre de dimensiuni variabile, ajustând dimensiunea ferestrelor cu ajutorul regulilor fuzzy.

Cheng și Chang în [27], [28] propun o schemă de admitere a utilizatorilor în rețele ATM utilizând o combinație de logică fuzzy și rețele neuronale. Schema lor de AC se bazează pe ideea de a estima comportamentul viitor al utilizatorilor în ce privește intensitatea traficului pe baza tipului și a intensității de până atunci a traficului acestora.

Yaghmae și colegii ([119]) realizează o extensie fuzzy a algoritmului *Leaky Bucket (LB)* pentru problema controlului parametrilor în rețele ATM. Ei utilizează ideea de a extinde cu ajutorul logicii fuzzy pragurile folosite la algoritmul LB.

Lee și Hou ([55]) propun un sistem neuro-fuzzy pentru controlul congestiilor în rețele ATM, iar Ng și Tham utilizează în [68] un controler integrat neuro-fuzzy pentru admiterea utilizatorilor în rețele ATM. Kandel și colegii prezintă în [51] un switch ATM fuzzy pentru managementul traficului și controlul congestiilor.

Razouqi și colegii propun o metodă de rerutare a celulelor ATM pentru a se evita pierderea de date într-o rețea ATM, simulând o astfel de rețea alcătuită din 50 de switchuri ATM ([82], [81]).

Liang și colegii propun utilizarea de mulțimi fuzzy de ordinul 2 (mulțimi fuzzy ale căror funcții de apartenență sunt de asemenea mulțimi fuzzy) pentru AC în ATM, dar utilizarea acestor mulțimi poate duce la creșterea complexității computaționale într-un mod care ar face greu de justificat beneficiile obținute prin utilizarea mulțimilor

fuzzy de ordinul 2. Atlasis și colegii utilizează algoritmi de învățare statistică pentru AC în ATM [8], pentru a determina lățimea de bandă echivalentă utilizată în algoritmi de AC.

În [56], lățimea de bandă efectivă (*effective bandwidth*) este determinată cu ajutorul logicii fuzzy, prin ajustarea fuzzy a unui coeficient de pondere care face legătura dintre parametrii de QoS ai conexiunilor și traficul măsurat. Lățimea de bandă efectivă este apoi utilizată pentru a lua decizia de a admite sau nu noi utilizatori în rețea. Datorită folosirii logicii fuzzy, acest algoritm de AC nu utilizează nici metode analitice complexe și nici descrieri a priori ale traficului.

În [25] se combină logica fuzzy cu algoritmi autoregresivi pentru a realiza modelarea traficului de date.

2.5.3 Logica fuzzy aplicată în rețele radio

În rețelele celulare, logica fuzzy a fost aplicată mai ales pentru determinarea momentului când un utilizator schimbă celula, pentru a prezice direcția de mișcare a unui utilizator mobil, cu scopul de a i se rezerva resurse în celulele în care acesta urmează să se deplaseze, precum și la admiterea utilizatorilor.

Edwards și colegii utilizează în [34] și [33] algoritmi bazați pe logica fuzzy pentru a determina momentul când un utilizator mobil comută dintr-o celulă în alta (*handoff*). Algoritmii fuzzy sunt comparați cu algoritmi clasici, bazați pe histereză și autorii subliniază performanțele superioare (în termeni de viteză și precizie) ale algoritmilor fuzzy.

Chen, Gong și Chao, în [26] utilizează logica fuzzy, mai precis matrice fuzzy, pentru a prezice celulele în care se va deplasa un utilizator mobil și pentru a putea astfel rezerva resurse în celulele respective.

Shen și Mark ([88]) utilizează de asemenea logica fuzzy pentru a face predicții despre modul de deplasare al unui utilizator radio, cu scopul de a i se rezerva resurse în celulele în care acesta urmează să se deplaseze.

O abordare complementară celor din [26] și [88], în care logica fuzzy era utilizată pentru a prezice viitoarea celulă ocupată de un utilizator mobil, este prezentată în [54]. De această dată nu se mai încearcă estimarea în avans a celulei ce urmează a fi ocupată de utilizatorul mobil, ci, atunci când apare un nou apel într-o celulă, se rezervă resurse atât în celula curentă, cât și în celulele învecinate, pe baza unui algoritm numit DCAC (*Distributed Call Admission Control*). Cu ajutorul unui controler fuzzy (*FLC – Fuzzy Logic Controller*) se determină cantitatea de resurse radio (canale) alocată pentru utilizatorii din celula curentă, precum și resursele care sunt rezervate pentru

utilizatorii din celulele vecine, în eventualitatea că ei s-ar deplasa în celula curentă. Algoritmul obținut, denumit FDCAC (Fuzzy DCAC) are performanțe mai bune decât algoritmi care nu utilizează logica fuzzy.

Un articol foarte interesant este și [22], care prezintă problema managementului mobilității într-o rețea eterogenă alcătuită dintr-un segment GPRS, un segment UMTS (*Universal Mobile Telecommunications Systems*) și un segment de telefonie mobilă prin satelit bazat pe rețeaua EuroSkyWay (M-ESW). Cele trei segmente sunt interconectate cu ajutorul protocolului Mobile IP. Se presupune că terminalele utilizatorilor se pot conecta la toate cele trei segmente, iar decizia de a comuta de pe un segment pe altul (*handover*) se ia atunci când calitatea serviciului oferită unei aplicații scade sub o anumită limită. Atât decizia de a schimba segmentul, cât și alegerea noului segment se iau cu ajutorul logicii fuzzy, pe baza calității semnalului primit de stația mobilă, a calității serviciului percepută de aplicație, precum și a unor factori cum ar fi costul, fiabilitatea, starea bateriilor, lățimea de bandă, rata erorilor, etc. În acest mod se va putea ajunge la situația în care utilizatorului mobil i se asigură cel mai bun serviciu disponibil într-o anumită situație.

Fernandez și colegii propun în [36] o metodă de a converti cerințele de QoS în parametri ce permit configurarea unui domeniu de tip DiffServ (*Differentiated Services* – servicii diferențiate), astfel încât să fie satisfăcute cerințele de calitate a serviciului pentru utilizatorii domeniului respectiv. Deși lucrarea nu se referă în mod explicit la rețelele celulare, metoda propusă prezintă interes și în cazul rețelelor de tip GPRS sau EGPRS, deoarece asigurarea calității serviciilor în rețele de tip GPRS/EGPRS prin integrarea acestor rețele în domenii de tip DiffServ este o etapă de viitor în dezvoltarea rețelelor (E)GPRS, sugerată și de [46] și [52].

În [23] este utilizată logica fuzzy pentru admiterea utilizatorilor într-o rețea mobilă de tip CDMA prin ajustarea cu ajutorul unei memorii asociative fuzzy (un controler fuzzy) a resurselor rezervate pentru utilizatorii care vin în celula curentă din celulele învecinate. Ca și parametri de QoS, autorii acestui articol se concentrează doar asupra menținerii probabilității de întrerupere a apelurilor în curs de desfășurare.

Dini și Guglielmucci prezintă în excelentul articol [32] o strategie de admitere a utilizatorilor bazată pe logica fuzzy aplicată în sisteme WCDMA (*Wideband CDMA* – CDMA de bandă largă). Ideea prezentată în acest articol este oarecum similară cu ideea prezentată în această teză, în capitolul 5, de aplicare a logicii fuzzy la admiterea utilizatorilor în GPRS. Ideea din această teză a fost prezentată în lucrările proprii [102], [106], [103] și [108], concomitent și independent de [32]. Parametrii care determină decizia de a admite sau nu un utilizator sunt foarte diferiți la tehnologia WCDMA față de GPRS sau EGPRS, dar faptul că ideea dezvoltată în această teză, de a aplica

logica fuzzy la admiterea utilizatorilor, a mai apărut legat de un alt tip de rețea celulară de date este o confirmare indirectă a valabilității metodei pe care o propune teza.

2.6 Concluzii și obiectivele tezei

Lucrarea de față își propune să investigheze și să îmbunătățească tehnicile de alocare a resurselor pentru creșterea performanțelor transmiterii datelor în rețele de telefonie mobilă de tip GPRS/EGPRS.

Obiectivele științifice ale acestei teze, raportate la stadiul actual al realizărilor din acest domeniu, pot fi rezumate astfel:

1. Realizarea unui model de simulare care să permită investigarea tehnicilor de alocare a resurselor în rețele de tip GPRS și EGPRS.

Se alege investigarea prin simulare deoarece, după cum am discutat și în 2.3.1, abordările analitice din literatură se bazează pe o serie de presupuneri simplificatoare care nu sunt în concordanță cu situația reală în ce privește rețelele celulare de date. În plus, metodele matematice utilizate în aceste abordări sunt extrem de complexe. O prezentare a stadiului actual privind modelarea și simularea rețelelor GPRS și EGPRS a fost făcută în 2.3.3.

Modelul propus trebuie să îndeplinească următoarele cerințe:

- Să permită investigarea unitară a
 - algoritmilor de alocare a resurselor între utilizatorii care sunt admiși în rețea (problema controlului transmiterii datelor sau *Transmission Control - TC*).
 - algoritmilor de admitere a utilizatorilor în rețea (problema admiterii utilizatorilor sau *Admission Control - AC*)
- Să permită realizarea unor scenarii de simulare complexe, în conformitate cu situația dintr-o rețea GPRS sau EGPRS reală.
- Să asigure un grad de abstractizare al modelării unei rețele GPRS sau EGPRS reale care să permită atât o simulare eficientă (din punct de vedere al duratei simulării și al resurselor de calcul cerute de simulare), cât și posibilitatea de a aborda probleme punctuale, astfel încât interpretarea rezultatelor în cazul acestor probleme să nu fie influențată de factori care nu țin în mod direct de problema respectivă (de exemplu rezultatele privind

performanțele algoritmilor utilizați la problema controlului transmiterii datelor nu trebuie să fie influențate de mecanismele de control ale unor protocoale de rețea, cum ar fi TCP).

Modelul de simulare propus de această teză este prezentat în capitolul 3.

2. Studiul prin simulare al performanțelor algoritmilor de planificare utilizați la alocarea resurselor radio între utilizatorii admiși într-o rețea de tip GPRS (*Transmission Control*).

După cum am prezentat în 2.2 și 2.3, există în literatură numeroase propuneri de algoritmi de planificare pentru controlul transmiterii datelor. Teza își propune să determine care dintre aceștia sunt cei mai potriviți pentru GPRS.

Studiul algoritmilor de planificare utilizați pentru *transmission control* este prezentat în capitolul 4 și își propune să rezolve următoarele aspecte:

- Compararea performanței diferitor algoritmi de planificare prin prisma valorilor întârzierilor pe care fiecare din algoritmi investigați le asigură datelor utilizatorilor. Sunt comparate performanțele a două categorii de algoritmi: o categorie care aproximează algoritmul idealizat numit Generalized Processor Sharing (GPS), descris în secțiunea 2.2 și care sunt reprezentați de algoritmi *Weighted Round Robin (WRR)* și *Oldest Queue (OQ)*, și o categorie de algoritmi care alocă resursele rețelei pe baza cantității de date din cozile utilizatorilor. Aceștia din urmă sunt reprezentați prin algoritmi *Longest Queue (LQ)*, *Total Queue Length (TQL)* și *Total File Length (TFL)*.
 - Studiul capacității algoritmilor investigați de a asigura diferențierea calității serviciilor oferite utilizatorilor GPRS și EGPRS pe baza clasei de calitate a serviciului căreia îi aparține fiecare utilizator.
 - Studiul posibilității de a utiliza diferențierea între utilizatorii din aceeași clasă de calitate a serviciului pe baza calității legăturii radio a utilizatorilor, cu scopul de a reduce și chiar elimina congestiile din rețea.
3. Dezvoltarea unei soluții originale, bazată pe logică fuzzy, de admitere a utilizatorilor în rețele de tip GPRS și EGPRS.

Acest lucru se face pornind de la problemele algoritmilor actuali de admitere a utilizatorilor în rețelele celulare de date, probleme relevate în secțiunea 2.4. Soluția își propune să depășească problemele algoritmilor actuali de admitere a utilizatorilor în următoarele aspecte:

- o probabilitate de întrerupere a sesiunilor de date cât mai aproape de zero
- o probabilitate de blocare a apelurilor foarte scăzută
- un grad foarte ridicat de asigurare a calității serviciului pentru utilizatorii admiși
- un grad ridicat de adaptare la modificarea condițiilor din rețea
- flexibilitate în ce privește politicile de admitere a utilizatorilor aplicate de operatorul de rețea.

Problema admiterii utilizatorilor este tratată în capitolul 5 și cuprinde următoarele aspecte:

- Identificarea neajunsurilor prezentate de algoritmi actuali de AC, aplicați la rețelele celulare de date.
- Dezvoltarea unei soluții originale de admitere a utilizatorilor, bazate pe logica fuzzy.
- Investigarea performanțelor controlerelor fuzzy
- Implementarea algoritmului de admitere a utilizatorilor într-un controler fuzzy și integrarea în modelul de simulare
- Validarea soluției originale fuzzy prin comparare cu algoritmi non-fuzzy de admitere a utilizatorilor.

Capitolul 3

Model de simulare a alocării resurselor într-o rețea de tip GPRS/EGPRS

În acest capitol se prezintă modul cum se face alocarea resurselor în GPRS. Pornind de la acesta, s-a dezvoltat în această teză un model de simulare pentru studiul performanțelor algoritmilor de alocare a resurselor în rețele GPRS sau EGPRS, model care este descris în acest capitol. Am modelat o celulă (E)GPRS în care există un număr de utilizatori ce doresc să transfere date. Modelul realizat este capabil să reprezinte ambele aspecte implicate de problema transferului datelor, și anume: admiterea utilizatorilor (*AC – Admission Control*) și controlul transmiterii datelor (*TC – Transmission Control*).

3.1 Alocarea resurselor în GPRS

3.1.1 Arhitectura de protocoale GPRS, fluxul de date și codarea canalelor

Arhitectura de protocoale GPRS

În această secțiune va fi discutat numai planul transmisiei la GPRS, deoarece planul de semnalizare nu face obiectul acestei lucrări.

Planul transmisiei apare în figura 3.1 [12].

În rețeaua terestră a GPRS (*GPRS backbone* sau coloana vertebrală [66]), protocolul de tunelare GPRS (*GTP – GPRS Tunneling Protocol*) realizează tunelarea pachetelor de date și a informației de semnalizare între nodurile GSN. Așa cum s-a menționat în secțiunea 2.1.2, în partea referitoare la arhitectura GPRS, datele utilizatorilor sunt încapsulate și decapsulate de către nodul GSN transmițător, respectiv

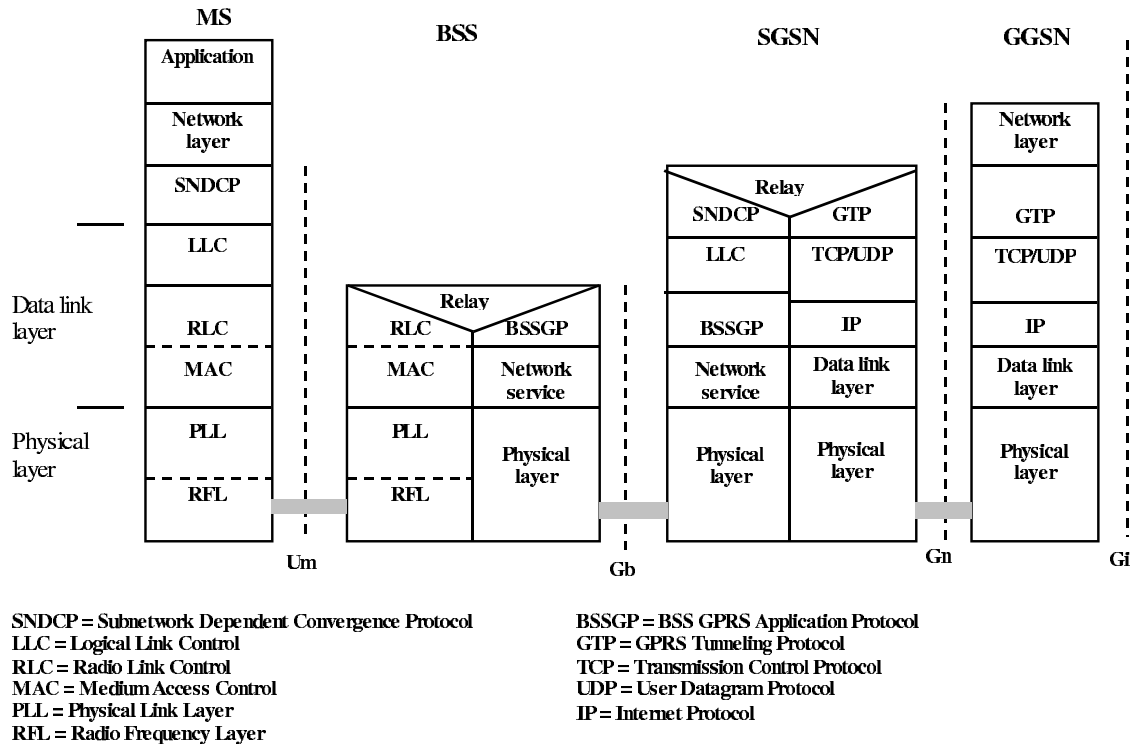


Figura 3.1: Arhitectura de protocoale GPRS. Planul transmisiei datelor

receptor. Sub GTP se utilizează un protocol de transport (*TCP – Transmission Control Protocol* sau *UDP – User Datagram Protocol*), iar sub protocolul de transport este folosit protocolul Internet IP (*Internet Protocol*). Sub IP se pot folosi diverse protocoale, cum ar fi Ethernet, ATM sau ISDN. În concluzie, stiva de protocoale GPRS în rețeaua terestră este IP/X.25 peste GTP peste TCP/UDP peste IP.

Transferul pachetelor de date între MS și SGSN este realizat prin SNDCP (*Subnetwork Dependent Convergence Protocol* – protocol de convergență dependent de subrețea), care multiplexează mai multe conexiuni de la nivel rețea într-o singură conexiune logică a LLC (*Logical Link Control Layer* – Nivelul legătură logică). De asemenea, SNDCP realizează compresia și decompresia datelor utilizatorului și a headerelor.

Interfața *Um* constă din *nivelul legătură de date (data link layer)* și din *nivelul fizic (physical layer)*.

Nivelul legătură de date este la rândul său alcătuit din două subniveluri: subnivelul legăturii logice (LLC), între MS și SGSN, și din RLC/MAC (*Radio Link Control / Medium Access Control* – controlul legăturii radio / controlul accesului la mediu), între MS și BSS.

LLC asigură o legătură logică de înaltă fiabilitate între MS și SGSN, care include funcții cum ar fi: controlul secvenței, furnizarea datelor în ordinea corectă, controlul fluxului, detecția erorilor la transmisie, retransmisia utilizând ARQ (*Automatic Repeat reQuest* – cerere de repetare automată) și cifrare. LLC poate lucra atât în mod cu confirmare (*acknowledged mode*), cât și fără confirmare (*unacknowledged mode*).

Nivelul RLC/MAC asigură următoarele două funcții principale:

- RLC stabilește o legătură fiabilă între MS și BSS și realizează segmentarea și reasamblarea cadrelor LLC în *blocuri de date RLC* sau *radio blocuri*, de asemenea realizează retransmiterea blocurilor ce conțin erori necorectabile utilizând o schemă de tip ARQ. RLC poate lucra atât în mod cu confirmare, cât și fără confirmare.
- Nivelul MAC controlează accesul mai multor stații mobile la resursele radio comune (canalele radio), utilizând algoritmi de planificare pentru a asigura pentru fiecare utilizator calitatea negociată a serviciilor.

Nivelul fizic (*physical layer*) dintre MS și BSS are două subiveluri: *Physical Link Layer (PLL)*, care asigură un canal fizic între MS și BSS și care este responsabil de codificarea canalelor, întrețeserea datelor (*interleaving*) și de detecția congestiilor la nivel fizic ([12]), și subnivelul *Radio Frequency Layer RFL*, care realizează modularea și demodularea.

Interfața dintre BSS și SGSN se bazează pe BSSGP (*BSS GPRS Application Protocol* – protocolul de aplicații BSS GPRS), care face rutarea informațiilor legate de QoS între cele două noduri (BSS și SGSN). Sub BSSGP, protocolul de servicii de rețea se bazează pe protocolul *Frame Relay*.

BSS și SGSN trebuie să implementeze funcții de releu (*internetworking functions*) între RLC și BSSGP, respectiv între SNDCP și GTP.

Fluxul de date

Pornind de la planul transmisiei datelor, se poate descrie fluxul de date în GPRS, așa cum este el prezentat în figura 3.2 [17]: unitățile de date la nivel de rețea (*N-PDU* – *Network-layer Protocol Data Units*), de exemplu pachetele IP, sunt transformate de către SNDCP în cadre LLC. Transformarea include în mod opțional prezența headerului sau/și a datelor, segmentare și criptare. Cadrele LLC, conținând suplimentar câte un header (*Frame Header*) și biți de control ai cadrului (*Frame Check Sequence bits*), sunt transmise între MS și SGSN prin interfața Um.

Cadrele LLC sunt segmentate în blocuri RLC, în funcție de schema de codare a utilizatorului (a se vedea explicația la pagina 49). Un bloc RLC conține headerul blocului

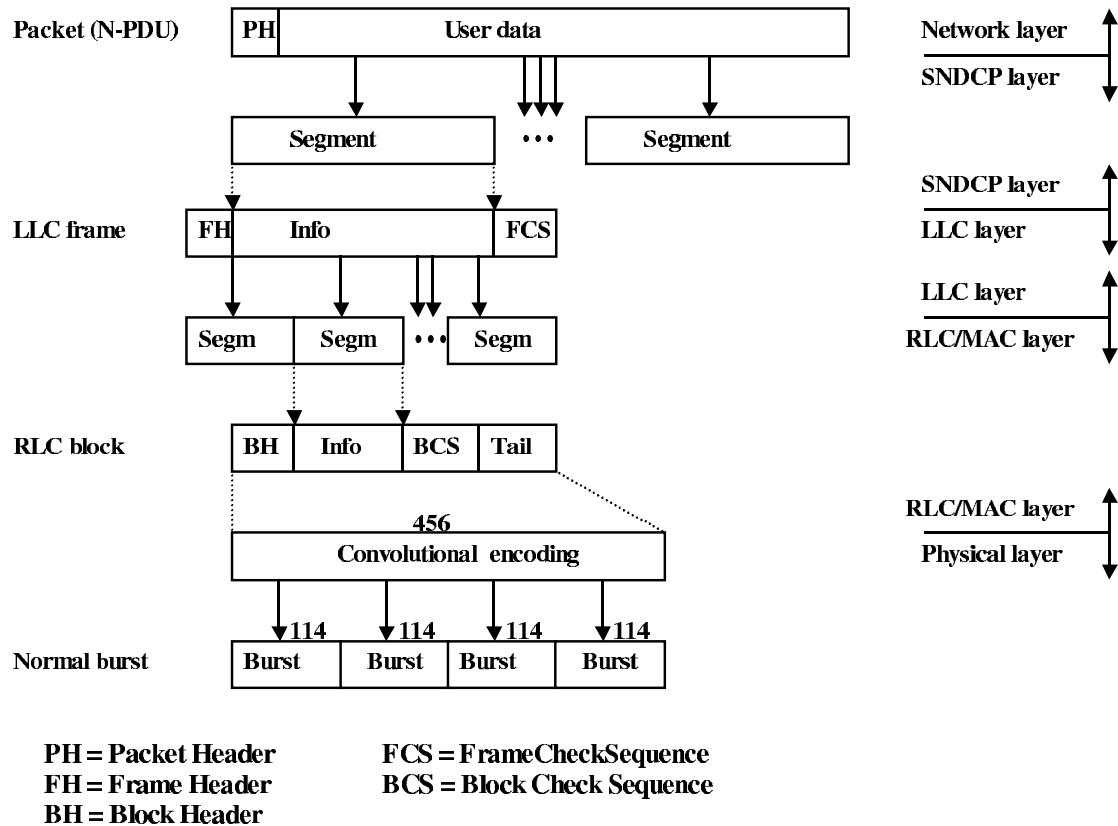


Figura 3.2: Fluxul de date în GPRS

(*Block Header*), biții de informație, biții de control ai blocului (*BCS - Block Check Sequence bits*) și opțional *tail bits* (biți de coadă), folosiți ca umplutură (*padding*), până la atingerea lungimii blocului. Întregul bloc este codificat folosind coduri convoluționale, obținând, eventual după eliminarea unui număr de biți la anumite scheme de codare (*puncturing*), 456 de biți, care sunt apoi împărțiți în patru grupe (*bursts*) a 114 biți fiecare și transmiși în patru cadre TDMA consecutive la nivel fizic.

Codificarea canalelor

Codificarea canalelor se folosește pentru a proteja datele transmise de erorile ce pot apărea la transmiterea prin interfața aeriană (*Um*) [12]. Tehnicile de codare din GPRS sunt asemănătoare cu cele din GSM. În GPRS se folosesc patru scheme de codare (*Coding Schemes*), notate CS-1 la CS-4. În EGPRS se utilizează nouă scheme de modulare și codare (*Modulation and Coding Schemes*) MCS-1 la MCS-9. Se selectează o schemă de codare sau alta în funcție de calitatea legăturii radio dintre stația mobilă MS și BTS, în concordanță cu tehnica de adaptare a legăturii (*Link Adaptation* –

LA) utilizată. Calitatea legăturii radio se exprimă prin *channel to interference ratio* – raportul între canal și interferență, notat C/I.

3.1.2 Managementul mobilității și activarea unui context PDP

Procedurile GPRS de atașare și detașare

Înainte de activarea unei sesiuni GPRS trebuie să aibă loc o procedură de *atașare* (*attach procedure*). Atunci când o stație mobilă se înregistrează la un SGSN, MS lansează o *cerere de atașare* (*attach request*). Rețeaua verifică întâi dacă utilizatorul este autorizat printr-o *procedură de autentificare* (*authentication procedure*), procedură asemănătoare cu cea din GSM, doar că rolul MSC (din GSM) este înlocuit de SGSN în dialogul cu HLR din procesul de autentificare. După autentificarea inițială, de-a lungul întregii sesiuni GPRS, nu mai sunt necesare alte proceduri de autentificare [44].

O *procedură de actualizare a locației* (*location update procedure*) este necesară pentru a memora în HLR SGSN-ul curent al stației mobile. Dacă procedura de atașare se încheie cu succes (adică dacă SGSN răspunde cu *attach accept*), atunci stației mobile îi este asignată o identitate temporară a legăturii logice (*TLLI – Temporary Logic Link Identity*).

O sesiune GPRS se încheie cu o *procedură de detașare* (*detach procedure*), care poate fi inițiată de către MS sau de către SGSN sau care poate fi apelată implicit atunci când stația mobilă este inactivă pentru o perioadă suficient de lungă încât timerul STANDBY să expire. Când are loc o procedură de detașare se șterg toate contextele PDP. Contextele PDP sunt descrise în paragraful “Activarea și dezactivarea unui context PDP” de la pagina 52.

Managementul mobilității

Pentru a ști tot timpul locația curentă a MS, managementul mobilității, numit și managementul locației, combină mesaje de actualizare a locației de la MS (mesaje prin care stația mobilă informează nodul SGSN despre locația sa curentă), cu procedura de *paging*. Procedura de *paging* se folosește în modul de transfer *downlink*, atunci când poziția exactă a MS nu este cunoscută.

A fost definit un model de stare pentru managementul locației în GPRS. Figura 3.3 ([44], [12]) prezintă acest model pentru stația mobilă (există un model similar și pentru SGSN).

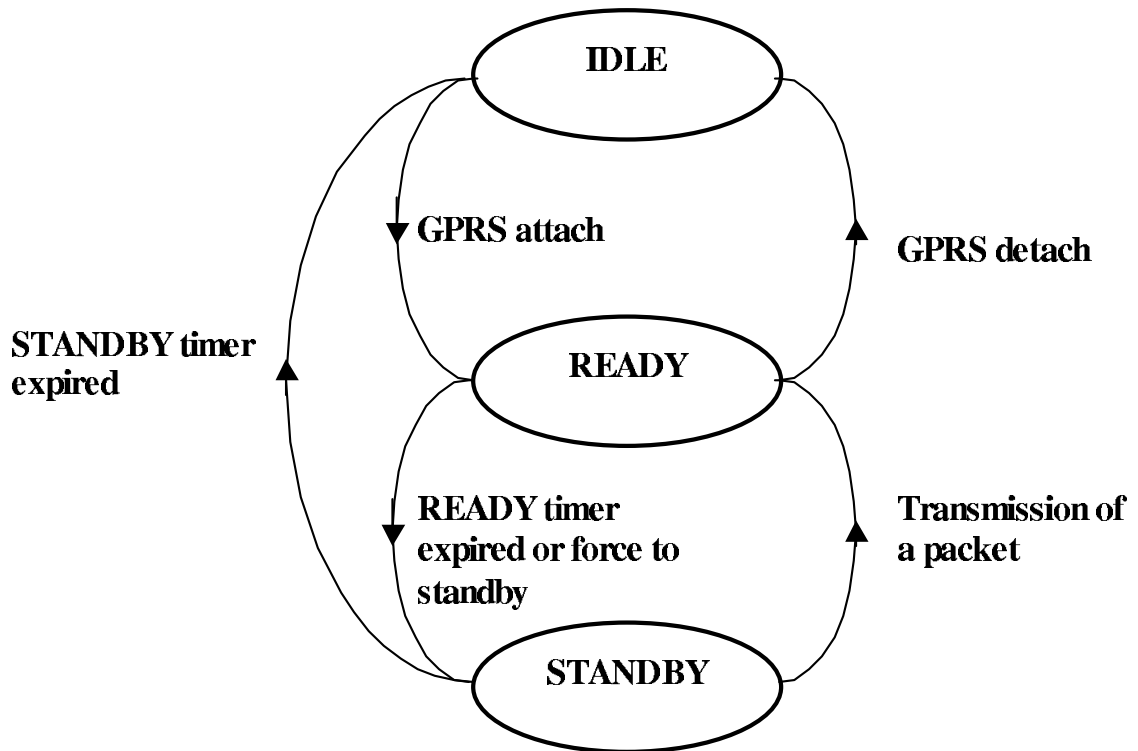


Figura 3.3: Stările managementului mobilității pentru o stație mobilă

În starea IDLE, utilizatorul nu este atașat la rețea, nu au loc proceduri de actualizare a locației, iar stația de mobilă nu poate fi contactată. O stație mobilă trece din starea IDLE în starea READY atunci când a parcurs cu succes procedura de atașare.

În starea READY, MS informează SGSN-ul la care este arondată despre fiecare schimbare de celulă. O arie de locație GSM (*GSM Location Area LA*) conține una sau mai multe arii de rutare GPRS (*GPRS Routing Area – RA*), iar o RA conține mai multe celule. O stație mobilă va informa nodul SGSN atunci când se deplasează într-o altă celulă. Dacă noua celulă face parte dintr-o altă RA, atunci MS trimite o cerere de actualizare a ariei de rutare (*routing area update request*) către SGSN. Dacă noua RA este deservită de același SGSN ca și RA anterioară, atunci se spune că are loc o actualizare de RA intra-SGSN, iar dacă noua RA este deservită de către un alt SGSN, atunci are loc o actualizare de RA inter-SGSN.

În starea READY, o stație mobilă activează unul sau mai multe contexte PDP pentru a transmite sau pentru a recepționa date. Starea READY are un timer care monitorizează activitatea fiecărei MS și dacă timerul expiră (din cauză că MS nu a transmis sau recepționat date în intervalul respectiv), MS va trece din starea READY

în starea STANDBY. Pentru ca să treacă din starea READY în starea IDLE, trebuie să aibă loc o procedură de detașare inițiată de MS sau de SGSN.

Atunci când se află în starea STANDBY, o stație mobilă va informa SGSN-ul despre poziția sa doar atunci când se deplasează într-o nouă RA, nu și atunci când se deplasează într-o altă celulă aflată în aceeași RA. Din această cauză, atunci când SGSN dorește să știe poziția (celula) unei MS aflate în starea STANDBY, el va trebui să efectueze o procedură de *paging* a ariei de rutare. Procedura de actualizare a RA în starea STANDBY poate fi, ca și în cazul stării READY, de tip inter- sau intra-SGSN. Când MS aflată în starea STANDBY transmite sau recepționează date, ea va trece în starea READY.

Activarea și dezactivarea unui context PDP

După ce o stație mobilă a executat cu succes o procedură de atașare, pentru a putea transmite sau recepționa date spre sau de la o rețea externă de date PDN (*Packet Data Network*), ea va trebui să obțină o adresă de la acea rețea externă de date. O astfel de adresă se numește adresă PDP (*PDP address*), de la Packet Data Protocol. Pentru fiecare sesiune de transfer de date se crează un așa numit context PDP (*PDP-context*) [12]. Contextul PDP conține tipul de PDP (de exemplu IPv4 sau IPv6), adresa PDP (de exemplu adresa IP), cerințele de calitate a serviciului (QoS) și adresa nodului GGSN care este utilizat ca și punct de acces la rețeaua de date externă.

Procedura de activare a unui context PDP este prezentată în figura 3.4 ([12]). Mai precis, figura 3.4 conține procedura de activare a contextului PDP inițiată de către MS, deoarece și rețeaua poate să inițieze o astfel de procedură, dacă există date de transmis spre MS (în mod *downlink*). În acest caz rețeaua va cere MS să inițieze o procedură de activare a contextului PDP, iar mai departe lucrurile decurg așa cum sunt prezentate în figura 3.4.

Procedura începe atunci când MS trimite mesajul *activate PDP context request*, de cerere de activare a contextului PDP, informând SGSN că dorește să activeze un context PDP. Acest mesaj include conținutul contextului PDP dorit. Alocarea adresei PDP se poate face static sau dinamic, în cazul alocării dinamice câmpul de adresă PDP din mesaj fiind lăsat gol.

Apoi are loc autentificarea utilizatorului, folosind funcții de securitate. Dacă autentificarea se desfășoară cu succes, atunci SGSN trimite nodului GGSN corespunzător mesajul *create PDP context request* (cerere de creare a contextului PDP), iar GGSN își actualizează tabela de rutare astfel încât să poată ruta date între SGSN și rețeaua de date externă (PDN). SGSN poate reduce sau chiar respinge cererea de calitate a serviciului (QoS) primită de la MS dacă resursele disponibile ale rețelei nu pot sat-

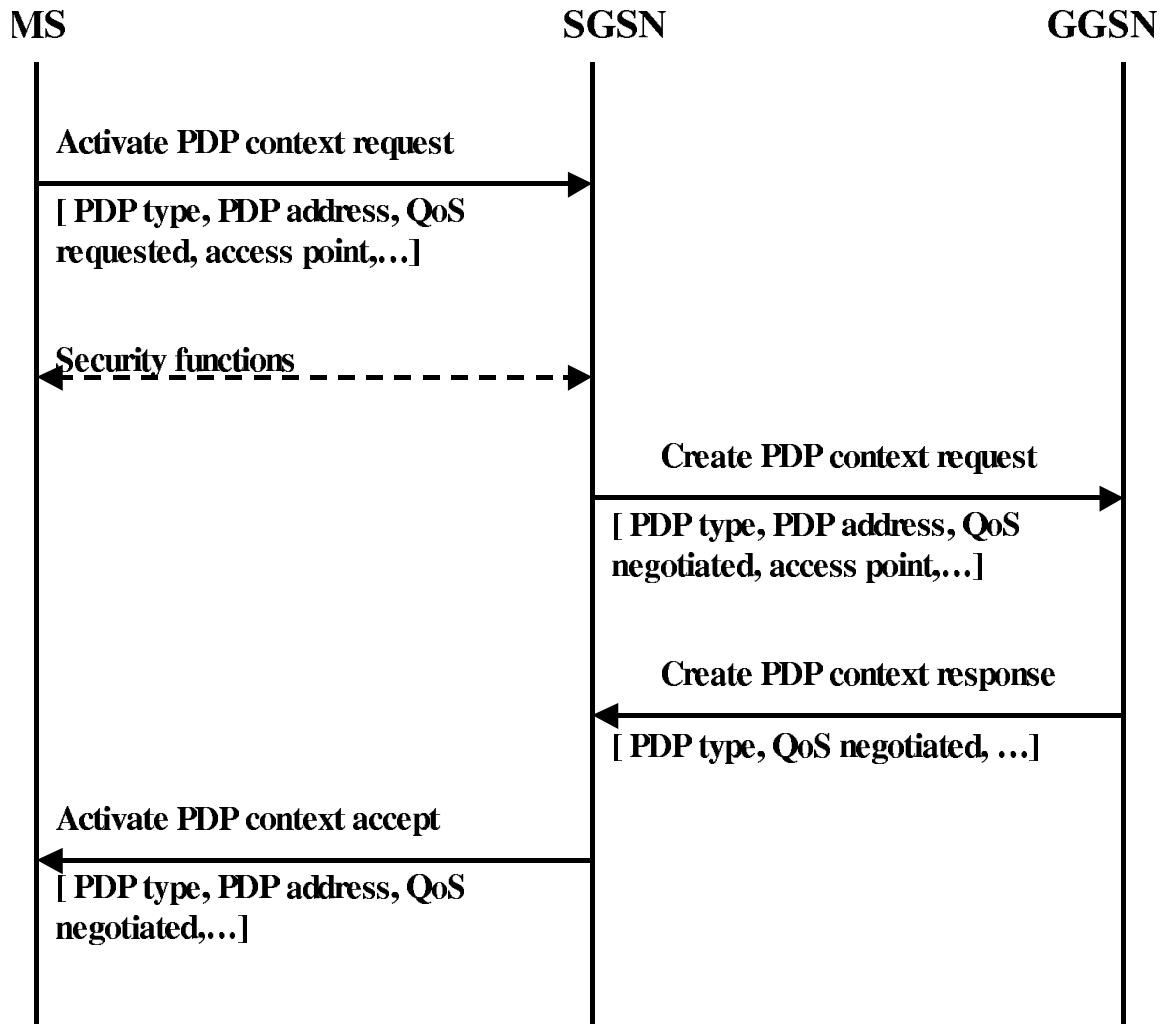


Figura 3.4: Procedura de activare a contextului PDP

isface cerințele utilizatorului. GGSN returnează un mesaj de confirmare către SGSN (*create PDP context response*), mesaj ce conține și adresa de PDP, în cazul în care aceasta a fost alocată dinamic. Și GGSN poate respinge cererea sau poate să reducă și mai mult calitatea serviciului, în cazul unei încărcări mari a rețelei.

SGSN își actualizează tabela de contexte PDP și informează MS că procedura de activare a contextului PDP a avut succes, prin trimiterea mesajului *activate PDP context accept*.

Pentru a dezactiva contextul PDP se utilizează o procedură de dezactivare a contextului PDP. Această procedură poate fi inițiată de MS, SGSN sau GGSN.

SGSN poate iniția și o procedură de modificare a contextului PDP, dacă sunt

necesare schimbări în profilul de QoS. MS poate respinge contextul PDP modificat, caz în care trebuie să lanseze o nouă procedură de activare a contextului PDP.

Resursele radio sunt alocate în GPRS doar la cerere. Pentru a transmite sau pentru a recepționa date, între MS și BSS trebuie să se desfășoare o altă procedură, numită *Temporary Block Flow (TBF) establishment* (stabilirea unui flux temporar de date). Unei MS i se alocă efectiv resurse, sub forma unor sloturi de timp (canale radio) prin intermediul unui TBF. Atunci când un utilizator a terminat de transmis sau de recepționat datele, are loc închiderea fluxului temporar de date, astfel încât să nu se risipească resursele radio, dar contextul PDP rămâne în continuare activ o anumită perioadă, permițând utilizatorului să transfere date din nou, fără a mai fi necesară activarea unui alt context PDP. Se poate deci spune că în GPRS există două niveluri de *admission control*: activarea unui context PDP și stabilirea unui TBF .

3.1.3 Calitatea serviciului în GPRS

Calitatea serviciului (QoS) în GPRS este descrisă în [52] [85], [30] și este definită ca fiind efectul colectiv al performanței serviciului, care determină gradul de satisfacție al utilizatorilor serviciului [52]. Diferențierea între utilizatori se face prin calitatea diferită a serviciilor oferite acestora.

În cadrul procedurii de activare a contextului PDP, profilul de QoS este negociat pe baza profilului de QoS subscris de utilizator (stocat în *Home Location Register – HLR*), a profilului de QoS cerut de MS și a resurselor pe care rețeaua le are disponibile la momentul respectiv. La negociere participă MS, SGSN și GGSN, rolul determinant avându-l SGSN.

Calitatea serviciului în GPRS Release 1997

Calitatea serviciului în GPRS a fost definită în GPRS faza 1, release 1997, așa cum apare în specificațiile ETSI.

În GPRS Release 1997, profilul de QoS este o combinație a următoarelor atribute:

1. *precedența serviciului*: indică prioritatea relativă a menținerii serviciului pentru conexiunile existente și specifică ordinea în care rețeaua va elimina pachete în situația apariției unor condiții critice (de exemplu congestie în rețea). Precedența serviciului poate fi *mare*, *normală* sau *mică*.
2. *întârzieri*: există patru clase de întârzieri care specifică valoarea medie a întârzierii și valoarea întârzierii a 95% din datele utilizatorilor pentru o unitate de informație

a utilizatorului numită unitate standard de date (*SDU – Standard Data Unit*). Întârzierile se consideră doar în interiorul rețelei GPRS. Valorile întârzierii medii a unei SDU de 128 de octeți sunt cuprinse între 0.5 secunde pentru cea mai performantă clasă predictivă și 50 de secunde pentru cea mai puțin performantă clasă predictivă. Pentru o SDU de 1024 octeți, valorile medii ale întârzierii sunt între 2 și 75 de secunde, iar valorile a 95% din întârzieri sunt între 7 și 375 de secunde. Valorile pentru clasa de întârziere *best effort* nu sunt precizate.

3. *fiabilitate*: definește probabilitatea ca o SDU să fie pierdută, duplicată, coruptă sau recepționată în afara secvenței corecte. Aplicațiile au diferite grade de sensibilitate la erori.
4. *capacitatea de trecere*: acest atribut indică ce capacitate de trecere cere utilizatorul. Capacitatea de trecere poate fi specificată prin rata de transfer maximă sau medie (*maximum* sau *medium bit rate*).

Calitatea serviciului în GPRS Release 1999

Calitatea serviciului, așa cum a fost definită în GPRS release 1997, suferă de câteva probleme majore:

- nu este clar cum se combină atributele (parametrii) de QoS pentru a forma clase de QoS.
- BSS nu este informată despre profilul de QoS negociat, ceea ce îi afectează capacitatea de a aloca în mod eficient resursele radio
- numai SGSN poate renegocia profilul de QoS, ceea ce înseamnă ca nici MS și nici GGSN nu pot influența profilul de QoS, chiar în situația în care ar detecta o schimbare a condițiilor de mediu (o legătură radio cu probleme pentru MS sau congestie în rețeaua externă în cazul GGSN).
- cea mai gravă problemă în GPRS QoS release 1997 este că este permis doar un singur profil QoS pentru fiecare adresă PDP. Deoarece o adresă PDP identifică o aplicație și nu un flux de date, înseamnă că un context PDP conține profilul de QoS pentru o anumită aplicație și nu pentru un flux de date, ceea ce face dificilă combinarea aplicațiilor care aparțin aceluiași utilizator, dar cu cerințe de QoS diferite (de exemplu streaming și FTP).

GPRS faza 2, release 1999, corectează aceste probleme, prin definirea atributelor de QoS prin coordonare cu specificațiile UMTS.

Calitatea serviciilor se bazează pe clasele de trafic, așa cum se arată în tabelul 3.1 [44].

Clasa de trafic	Exemple de aplicații	Caracteristici fundamentale
Clasa conversațională	Voce și video telefonie	Prezervarea întârzierilor, întârzieri reduse, tipar conversational
Clasa streaming	Timp real video streaming	Păstrează variația întârzierilor, puține retransmisii
Clasaa interactivă	Navigare pe internet	Prezervarea conținutului, retransmisii, tipar cerere-răspuns
Clasa background	Descărcare de e-mail și fișiere	Insensitivă la întârzieri, păstrarea conținutului, retransmisii

Tabelul 3.1: Clasele de QoS în GPRS release 1999

În release 1999, BSS este informată despre profilul QoS, ușurându-se astfel alocarea aresurselor radio. De fiecare dată când există un flux de pachete fie în *uplink*, fie în *downlink*, SGSN îi transmite BSS profilul QoS asociat contextului PDP respectiv.

De asemenea, în GPRS faza 2, procedura de modificare a contextului PDP poate fi inițiată de către MS, SGSN sau GGSN.

Cel mai important lucru este că, în release 1999, este posibil să existe mai multe contexte PDP pentru o adresă PDP. Fiecare adresă PDP este asociată cu un *TFT* (*Traffic Flow Template*), care identifică un flux de trafic (*traffic flow*). În acest fel, un utilizator poate avea mai multe fluxuri active simultan, fiecare cu propriul profil de QoS.

3.2 Descrierea modelului de simulare

În această secțiune se prezintă modelul de simulare realizat în această teză pentru studiul alocării resurselor într-o rețea de tip GPRS. Următoarele subsecțiuni prezintă simulatorul utilizat, prezentarea de ansamblu a modelului și apoi se descrie fiecare modul în detaliu.

3.2.1 Simulatorul utilizat

Primele versiuni ale modelului dezvoltat în această teză [109], [110], [111], [43] au fost realizate la Dublin City University, utilizând programul de simulare SES/ *Workbench*

al companiei Hyperformix. Deoarece Workbench este un program comercial de modelare și simulare, care necesită o licență de utilizare de care nu dispunem la Universitatea “Politehnica” Timișoara, am continuat cercetările folosind un program de modelare și simulare “open-source”, OMNeT++, care este un simulator bazat pe evenimente, specializat pentru modelarea rețelelor, produs și menținut de András Varga [114].

Translatarea modelului în OMNeT++ din modelul elaborat inițial în Workbench a fost relativ ușor de realizat, deoarece ambele simulatoare au la bază o platformă C/C++, ceea ce a permis ca o mare parte din codul sursă scris pentru Workbench să poată fi folosit și în OMNeT++.

Inițialele OMNeT++ provin de la Objective Modular Network Testbed in C++. Un model de simulare în OMNeT++ constă dintr-un număr de *module* (sau *noduri*) care comunică între ele prin *mesaje*. Un modul poate crea, procesa sau distruge mesaje sau le poate transmite altor module (de obicei după procesarea lor). Codul unui modul este activat de apariția unui *eveniment* în sistem, iar un eveniment constă în sosirea unui mesaj la un modul. Există și situația în care un modul își trimite mesaje lui însuși (această facilitate este utilizată mai ales pentru implementarea timerelor). Un mesaj poate avea un număr de *attribute* (*parametri*), care pot fi fie predefiniți în OMNeT++ (de exemplu lungimea mesajului sau felul său), fie creați de utilizatorul programului. De exemplu un nod poate să îi adauge unui mesaj un parametru reprezentând momentul când a fost creat sau distrus sau momentul când mesajul a sosit la acel nod.

Un modul în OMNeT++ poate fi *simplic* sau *compus*. Modulele compuse reprezintă aspectul *structural* al modelării în OMNeT++. Prin compunerea modulelor se pot realiza descrieri structurale ierarhice ale unui sistem, numărul de niveluri în ierarhie nefiind limitat de utilitarul OMNeT++. Pentru descrierile structurale a fost dezvoltat un limbaj, numit NED (*NEtwork Description*), care este integrat în OMNeT++.

La baza oricărei ierarhii de module stau modulele simple, care nu sunt descrise structural, ci comportamental, utilizând limbajul C++ și o serie de funcții predefinite în OMNeT++ (în special funcții pentru procesarea mesajelor).

3.2.2 Prezentare de ansamblu a modelului

Figura 3.5 prezintă o descriere de ansamblu a modelului de simulare pe care l-am realizat pentru a studia problema alocării resurselor în rețele GPRS și EGPRS.

Fiecare utilizator i din celulă este modelat printr-un modul compus *user i*. În model există de asemenea câte un modul (nod) pentru: *unitatea de control a pachetelor*

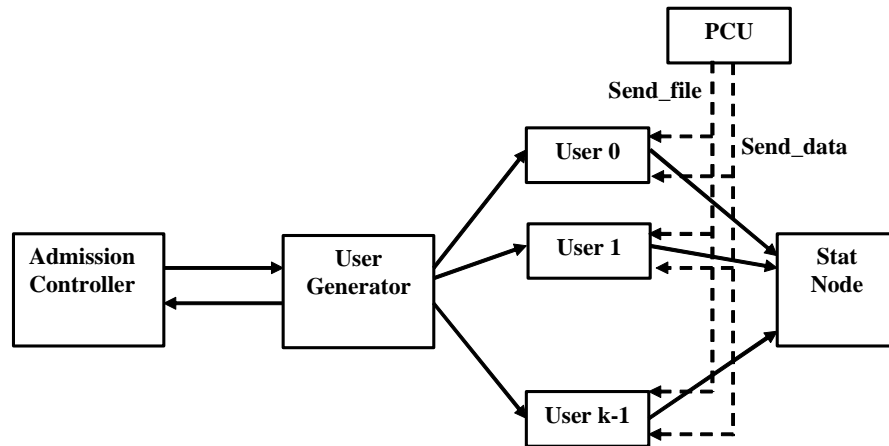


Figura 3.5: Modelul de simulare

PCU (*Packet Control Unit*), *generatorul de utilizatori* (*User Generator*), un nod pentru colectarea rezultatelor și realizarea statisticilor privind simularea, numit *stat node* și *admission controller-ul* – modulul care realizează admiterea utilizatorilor în rețea. Fiecare din aceste module va fi detaliat în subsecțiunile următoare. Sunt modelate de asemenea calitatea legăturii radio și partajarea resurselor radio între partea de date (GPRS) și cea de voce (GSM).

În model există un număr fix de *module utilizator*, fiecare din acestea reprezentând un loc ce poate fi ocupat de un utilizator GPRS, numărul maxim de module utilizator fiind N_MAX (fiecare modul utilizator poate conține cel mult un utilizator. Un utilizator GPRS mai este denumit *nod mobil* (*mobile node MN*), iar un modul utilizator va mai fi numit modul MN. Dacă un modul utilizator nu conține nici un MN, atunci acesta este un modul liber, iar dacă el conține un MN, atunci modulul este ocupat. Prin utilizarea unui număr fix de module utilizator am evitat crearea și distrugerea în mod dinamic de module utilizator, care este teoretic posibilă în OMNeT++, dar care în cazul modulelor compuse (așa cum e și modulul utilizator) impune multe restricții de lucru. Pe de altă parte, la soluția adoptată în teză, trebuie avut grijă ca numărul utilizatorilor admiși în celulă să nu depășească N_MAX .

După ce un nou MN este creat de către modulul *generator de utilizatori*, acesta cere să fie admis în celulă, cererea sa fiind adresată modulului *admission controller*. Dacă modulul *admission controller* decide să admită noul MN, atunci acestuia trebuie să i se găsească un modul utilizator liber. În caz că toate modulele utilizator sunt ocupate la momentul respectiv, se va genera un mesaj de eroare. În mod normal

nu se ajunge în această situație deoarece algoritmul de AC nu permite admiterea utilizatorului dacă rețeaua este atât de încărcată.

Dacă utilizatorul nou creat este admis în celulă și dacă există un modul utilizator liber, atunci el va ocupa acest modul (dacă există mai multe module libere, îl va ocupa pe primul disponibil, adică modulul având indicele cel mai mic). La sosirea MN în modulul utilizator, acesta va începe prin a activa funcțiile de *reset* pentru toate sub-modulele care compun modulul utilizator. Fiecare astfel de sub-modul are funcția sa de reset, care va șterge toate mesajele din cozile sub-modulului respectiv și în plus va aduce sub-modulul într-o stare inițială cunoscută.

Un MN eliberează un modul utilizator în două situații: fie când se încheie sesiunea de transfer de date (datele au fost transferate integral între MS și BSS), fie atunci când utilizatorul părăsește celula înainte de încheierea sesiunii de transfer de date. În fiecare din aceste situații utilizatorul va trebui să informeze modulul generator de utilizatori, deoarece acesta gestionează situația locurilor libere în modulele utilizator.

3.2.3 Modelul pentru un utilizator GPRS

Fiecărui utilizator GPRS (MN) îi corespunde în modelul de simulare realizat în OM-NeT++ un *modul utilizator* sau *modul MN*.

Un *modul utilizator* este un modul compus, a cărui structură internă este reprezentată în figura 3.6. Acesta constă din: o *sursă interactivă (source)* sau *generator de date interactiv*, un *buffer de fișiere (file buffer)*, un *buffer pentru blocuri de date (data packet buffer)*, un *nod de bulcare (loop node)* și un *modul de ieșire sau de scurgere (sink node)*. Prezent în componența unui *nod mobil*, dar nereprezentat în figura 3.6 este și un nod numit *handoff node*, care modelează transferul MN în altă celulă, mai precis faptul că MN se deplasează din celula curentă în altă celulă, înainte ca sesiunea de transfer de date să se fi încheiat.

Nodul sursă

După primirea comenzii *start user* de la nodul generator de utilizatori, sursa va crea unul sau mai multe “fișiere” la anumite intervale de timp, fiecare fișier având o anumită lungime. Unui “fișier” din model îi poate corespunde un fișier real sau un grup de fișiere transferate într-o sesiune GPRS (un fișier în cazul unei sesiuni FTP, mai multe fișiere reprezentând conținutul unei pagini Web în cazul unui transfer WWW) sau o parte dintr-un fișier, de exemplu un cadru în cazul unei sesiuni de video streaming.

Nodul sursă poate lucra fie în mod *interactiv* fie în mod *streaming*. În mod interactiv, nodul *sink* anunță nodul sursă atunci când un fișier a fost transferat cu

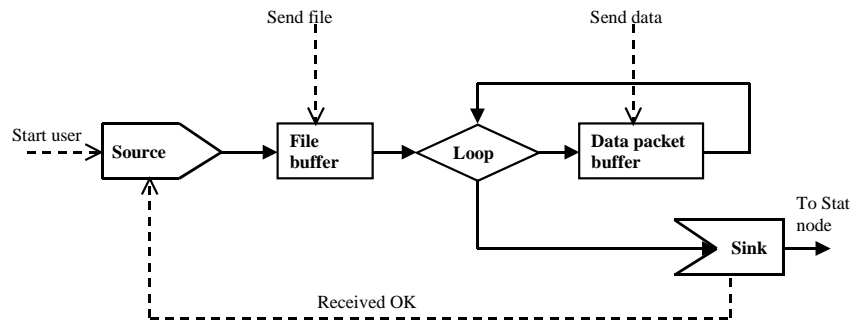


Figura 3.6: Modulul utilizator

succes prin transmiterea comenzii *received OK*, iar sursa poate genera un alt fișier doar după recepționarea acestei comenzi. În acest fel, în modul de lucru interactiv, sursa nu poate genera date noi până când precedentul fișier nu a fost transferat cu succes. Modul interactiv modelează de exemplu o sesiune WWW, când utilizatorul nu va transfera conținutul unei noi pagini Web înainte de a recepționa (și eventual citi) conținutul paginii precedente, dar un comportament similar apare și în cazul unui transfer de fișiere (FTP) sau de e-mailuri.

Dacă se dorește modelarea unei sesiuni de audio sau video streaming, atunci sursa este pusă în mod streaming și în acest caz ea poate genera noi date indiferent dacă precedentul “fișier” a fost complet transferat sau nu.

Un mod particular de lucru al sursei de date, numit *generare continuă de date* se obține din modul de lucru interactiv atunci când sursa va genera un nou fișier imediat ce fișierul precedent a fost transferat cu succes (a ajuns la nodul sink). Deși acest mod de lucru (generare continuă de date) nu corespunde în mod normal unei situații de interes practic (care să modeleze comportamentul unui utilizator GPRS în cazul unui anumite aplicații), el este foarte util pentru a modela algoritmi de planificare idealizați (de exemplu algoritmul FFQ prezentat în secțiunea 2.2.1) sau pentru a testa funcționarea corectă a modelului de simulare și a algoritmilor de planificare implementați.

Din punct de vedere al simulatorului OMNeT++, un “fișier” este un mesaj, care poate avea diverse atribute (parametri): lungimea fișierului, momentul creerii fișierului sau momentul când fișierul a ajuns la un anumit nod sau când a părăsit acel nod, numărul de ordine al fișierului respectiv în cadrul sesiunii de lucru a utilizatorului, inclusiv un atribut care specifică dacă respectivul fișier este ultimul din acea sesiune de lucru.

Nodul *file buffer*

Dupa crearea sa, un fișier este stocat în bufferul de fișiere (*file buffer*), unde acesta așteaptă să fie transferat (trimis sau recepționat de utilizatorul GPRS) atunci când utilizatorul are dreptul să transfere date.

Fiecare *file buffer* are un parametru reprezentând capacitatea maximă a bufferului (în biți), iar fișierele care sosesc la bufferul de fișiere vor fi eliminate dacă această capacitate a fost depășită prin sosirea noului fișier. Informația despre cantitatea de date eliminate datorită depășirii capacității de stocare a bufferului de fișiere este înregistrată de către nodul pentru colectarea statisticilor (*stat node*) și afișată la terminarea simulării.

Pentru a se asigura transferul corect al datelor într-un mediu de transmisie puternic afectat de erori, așa cum este cazul canalelor radio, în GPRS și EGPRS se utilizează codificarea datelor. Așa cum s-a arătat și în secțiunea 3.1.1, partea referitoare la fluxul de date în GPRS (pagina 48) și cum este ilustrat în figura 3.2, un bloc de date, de exemplu un pachet IP (care poate fi echivalat cu un cadru LLC), este segmentat în unități de date mai mici care sunt apoi codificate utilizând coduri convoluționale, informația codificată fiind numită *bloc radio*. Un bloc radio are în GPRS o lungime fixă, de 456 de biți. În funcție de schema de codare utilizată, numărul de biți de informație (biți utili sau ai utilizatorului) din cadrul unui radio bloc poate varia între 181 de biți în cazul schemei de codare CS1, până la 428 de biți pentru schema de codare CS4. Spre deosebire de GPRS, unde există patru scheme de codare, în EGPRS se utilizează 9 astfel de scheme, numite scheme de modulare și codare MCS1 la MCS9 (*Modulation and Coding Schemes*).

Procesul segmentării unui pachet de date al utilizatorului este modelat în felul următor: atunci când bufferul de fișiere recepționează comanda *send file* de la PCU, fișierul este împărțit în blocuri radio, în concordanță cu schema de codare a utilizatorului, iar aceste blocuri sunt transferate în nodul *data packet buffer*. Momentul în care fișierul părăsește nodul *file buffer* este memorat ca și parametru (atribut) al mesajului OMNeT++ care reprezintă respectivul fișier. Acest parametru va fi utilizat la calculul *timpului de așteptare (waiting delay)* al unui fișier, care reprezintă intervalul de timp petrecut de fișierul respectiv în bufferul de fișiere.

Nodul *data packet buffer* și nodul de buclare

Nodul *data packet buffer* stochează blocurile de date (blocurile radio) rezultate din segmentarea unui "fișier". De fiecare dată când utilizatorul are dreptul să transfere date (prin interfața radio), adică atunci când bufferul de date primește comanda *send*

data de la PCU, se vor transfera un număr de blocuri radio între stația mobilă MS și stația de bază BS prin interfața radio Um. Blocurile de date afectate de erori pot fi retransmise (dacă se utilizează modul de lucru cu confirmare și retransmisie).

Pentru a modela procesele de transmitere și retransmitere este utilizat un nod de buclare (*loop node*).

Așa cum s-a arătat în secțiunea 3.1.1, într-un sistem GPRS, un *cadru LLC* (*Logical Link Control frame*) este fragmentat în radio blocuri și apoi trimis nivelului RLC/MAC. Nivelul RLC/MAC trimite radio blocurile prin interfața radio (între MS și BSS) și retransmite blocurile afectate de erori [12], [15], [17], [41], [46], [99]. La receptor cadrul LLC este reasamblat din blocurile radio primite (a se vedea și figura 3.2 din secțiunea 3.1.1).

Se poate utiliza un algoritm de *adaptare a legăturii* (*link adaptation*) pentru a găsi un compromis optim între numărul de blocuri retransmise și schema de codare utilizată. Algoritmul pornește de la o schemă de codare care permite o capacitate de trecere mai mare, iar dacă au loc prea multe retransmisii, atunci se modifică schema de codare, folosind o schemă cu redundanță mai puternică.

Pentru a simplifica modelul de simulare (și deci pentru a crește viteză simulării) și deoarece în această teză nu se investighează problema algoritmilor de *link adaptation*, am decis să modelăm procesul de segmentare și reasamblare a cadrelor LLC în și din blocuri radio în următoarea manieră. Atunci când se transferă un număr de blocuri radio dintr-un fișier, lungimea f_l a fișierului (a cadrului LLC) aflat în *data packet buffer* se modifică în conformitate cu următoarea ecuație:

$$f_l = f_l - (no_of_data_packets - no_of_err_data_packets) \cdot u_cs \quad (3.1)$$

În ecuația (3.1) *no_of_err_data_packets* reprezintă numărul de radio blocuri care sunt retransmise, iar *u_cs* reprezintă numărul de biți utili (ai utilizatorului) conținuți într-un bloc radio. Numărul de biți utili este determinat de schema de codare, de exemplu 181 biți pentru CS1 sau 428 biți pentru CS4. Numărul de blocuri retransmise reprezintă un procent din numărul total de blocuri transmise, procent dat de parametrul *BLER* – *Block Error Rate* (rata de blocuri eronate).

Transferul blocurilor radio se face atunci când PCU transmite comanda *send data* la *data packet buffer*, specificând și numărul de blocuri radio care se vor transfera în ciclul curent de 20 de milisecunde prin setarea variabilei *no_of_data_packets*. Acest ciclu de 20 ms se numește *block period*.

După ce i se calculează noua lungime, fișierul este trimis la nodul de buclare, unde lungimea sa se compară cu zero: dacă lungimea fișierului este pozitivă, atunci mesajul OMNeT++ care reprezintă fișierul este trimis din nou la nodul *data packet buffer*,

ceea ce înseamnă că transferul fișierului nu s-a încheiat, iar dacă lungimea fișierului este zero, atunci fișierul va fi trimis la *sink*, iar momentul când se întâmplă acest lucru va fi înregistrat.

Nodul “sink”

În primele versiuni ale modelului de simulare nodul *sink* era utilizat pentru a distruge “fișierele” (de fapt mesajele OMNeT++ care reprezintă fișiere) și pentru a colecta informații statistice despre utilizatori pe durata simulării.

În versiunea actuală a modelului, nodul *sink* nu mai distruge mesajele OMNeT++ reprezentând fișiere, ci doar le transmite mai departe modulului *stat node*, care va colecta statisticile referitoare la simulare. Această modificare a modelului a fost făcută pentru a colecta statisticile despre toți utilizatorii într-un singur nod, nu în fiecare modul utilizator ca în primele versiuni. Această culegere centralizată a statisticilor îmbunătățește performanțele simulării.

Așa cum s-a menționat la descrierea nodului sursă, nodul *sink* informează sursa atunci când un fișier a ajuns la *sink*, iar dacă sursa lucrează în mod interactiv, ea nu va genera noi date până când nu primește această informație de la *sink*.

De asemenea, nodul *sink* informează nodul *handoff* al modulului utilizator atunci când ultimul fișier din sesiunea de transfer de date a fost transferat cu succes (deci, atunci când ultimul fișier a ajuns la *sink*), ceea ce înseamnă că sesiunea a fost încheiată fără ca utilizatorul să părăsească celula GPRS curentă. Tot în această situație nodul *sink* informează modulul generator de utilizatori că s-a eliberat modulul utilizator respectiv.

Nodul “handoff”

Nodul *handoff* modelează situația în care un utilizator (MN) părăsește celula GPRS curentă înainte de încheierea sesiunii de transfer de date. Deoarece modelul de alocare a resurselor din această teză se concentrează asupra celulei GPRS curente, se dorește modelarea comportamentului unui utilizator GPRS doar atunci când el se află în celula studiată și utilizatorul nu mai prezintă interes după ce părăsește celula.

Atunci când un nou nod mobil MN ocupă un modul utilizator care era liber în prealabil, el va porni un timer care expiră atunci când timpul de *dwel* s-a scurs. Timpul de *dwel* reprezintă timpul cât utilizatorul care are o sesiune în desfășurare rămâne în celula curentă. Valoarea timpului de *dwel* este setată de generatorul de utilizatori la crearea fiecărui utilizator. Dacă sesiunea de transfer de date se încheie înainte de expirarea timpului de *dwel*, atunci timerul respectiv este oprit, iar în caz

contrar, adică dacă timpul de *dwell* expiră în cursul desfășurării sesiunii de transfer de date, atunci utilizatorul părăsește celula, ceea ce, din punct de vedere al modelului de simulare, înseamnă că trebuie să elibereze modulul utilizator.

Procedura de eliberare a modulului utilizator se desfășoară astfel: întâi nodul *handoff* anunță celelalte module, care pot interacționa cu modulul utilizator, despre faptul că utilizatorul va elibera modulul respectiv și le cere să nu îi mai trimită mesaje. În modelul din această teză, mesajele OMNeT++ pot veni la modulul utilizator de la modulul PCU sau de la propriile sale submodule, care își trimit mesaje unul altuia. Din punctul de vedere al sistemului GPRS modelat, aceste mesaje OMNeT++ pot fi date sau comenzi. Pentru a împiedica PCU să îi mai trimită mesaje, sub-modulul *handoff* al modulului utilizator în cauză informează modulul PCU că utilizatorul curent nu mai cere resurse de la rețea, astfel încât sub-modulele nodului PCU nu vor mai trimite comenzi *send file* sau *send data* către acest utilizator.

Pentru a bloca trimiterea de mesaje între nodurile interne ale unui modul utilizator, nodul *handoff* setează o variabilă globală numită *handoff_mode* și fiecare nod va avea un anumit comportament în cazul setării acestei variabile, comportament diferit de cel normal. De obicei diferența de comportament constă în aceea că nodurile vor șterge mesajele pe care în mod normal le-ar fi trimis altor noduri.

Următorul pas va fi că nodul *handoff* așteaptă un timp suficient de lung pentru a fi sigur că modulul utilizator din care face parte nu va mai primi nici un mesaj, apoi anunță modulul generator de utilizatori că modulul utilizator curent a devenit liber și deci disponibil pentru a fi ocupat de un nou utilizator. Este însă responsabilitatea utilizatorului nou creat să șteargă mesajele care ar fi putut rămâne în cozile nodurilor ce compun modulul utilizator și acest lucru este realizat de către funcțiile de reset activate la sosirea unui nou utilizator de către nodul sursă.

Timpul cât nodul *handoff* trebuie să aștepte, de la declanșarea procedurii de părăsire a celulei și până la declararea modulului utilizator ca fiind liber este maximul dintre perioada PCU (de 20 ms) și valoarea curentă a perioadei nodului sursă (timpul cât nodul sursă așteaptă înainte de a crea un nou fișier).

Implementarea nodului *handoff* a fost oarecum mai dificilă, în primul rând datorită comportamentului său care nu este sincron cu al celorlalte noduri din sistem (expirarea timpul de *dwell* este similar cu primirea unei întreruperi de către celelalte noduri din sistem).

Implementarea modelului în OMNeT++

Figura 3.7 prezintă o captură de ecran cu un utilizator, captură efectuată cu simulatorul OMNeT++ lucrând în mod grafic. Se observă nodurile componente ale unui

modul utilizator: în partea stângă a figurii este generatorul de date, reprezentat sub forma unui robinet, celelalte noduri din figură fiind nodul *file buffer* cu eticheta *fb*, nodul de buclare cu eticheta *loop_n*, *data packet buffer* cu eticheta *db*, nodul *hand-off*, cu eticheta *hoff* și nodul *sink*, reprezentat în colțul din dreapta jos al figurii. Se observă că intrările pentru generatorul de date provin de la *sink* (comanda *receive OK*, din exterior (comanda *start user*, precum și de la nodul *handoff*). Ieșirea nodului sursă va fi intrare pentru bufferul de fișiere, care mai are înca o intrare (comanda *send file* de la PCU) și o ieșire, spre nodul de buclare. Nodul de buclare comunică bidirecțional cu bufferul de date, acesta din urmă având și o intrare de comandă de la PCU (*send data*). De asemenea, o ieșire a nodului de buclare merge la *sink*.

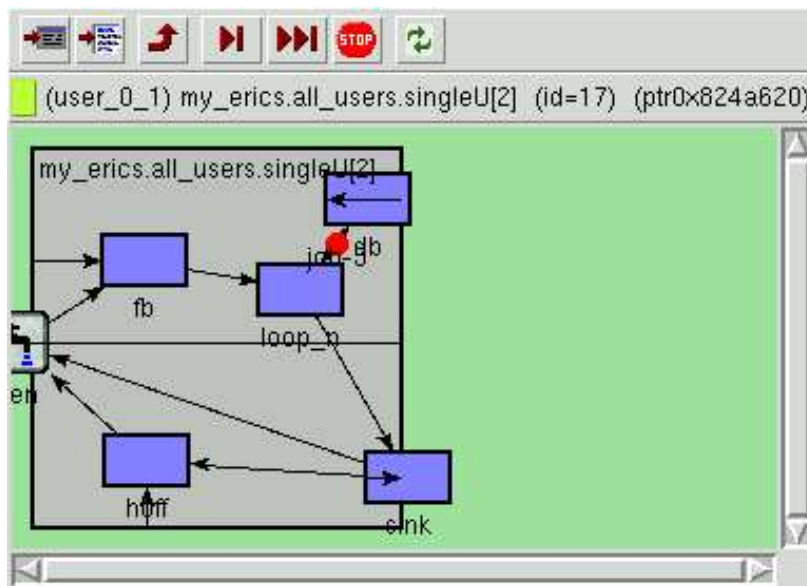


Figura 3.7: Captură de ecran pentru modulul utilizator în timpul simulării

Figura 3.7 surprinde modulul utilizator la timpul de simulare 5280ms, când are loc trimiterea unui mesaj de la *loop node* la *data packet buffer*, iar figura 3.8 arată că la momentul 5280 nodul *db* (*data packet buffer*) din componența modulului utilizator cu eticheta *singleU* și indicele 2 generează un eveniment (trimite un mesaj). Se observă că evenimentul precedent a constat în trimiterea unui mesaj de către PCU (submodulul *transmission controller*, având eticheta *tc*, mesaj care a reprezentat comanda *send data* transmisă bufferului de date al modulului utilizator cu indicele 2. Tot în figura 3.8 se observă că modulul utilizator cu indicele 2 este la rândul său parte componentă a modulului reprezentând toți utilizatorii, cu eticheta *all_users*, din întregul sistem (cu eticheta *my_erics*).

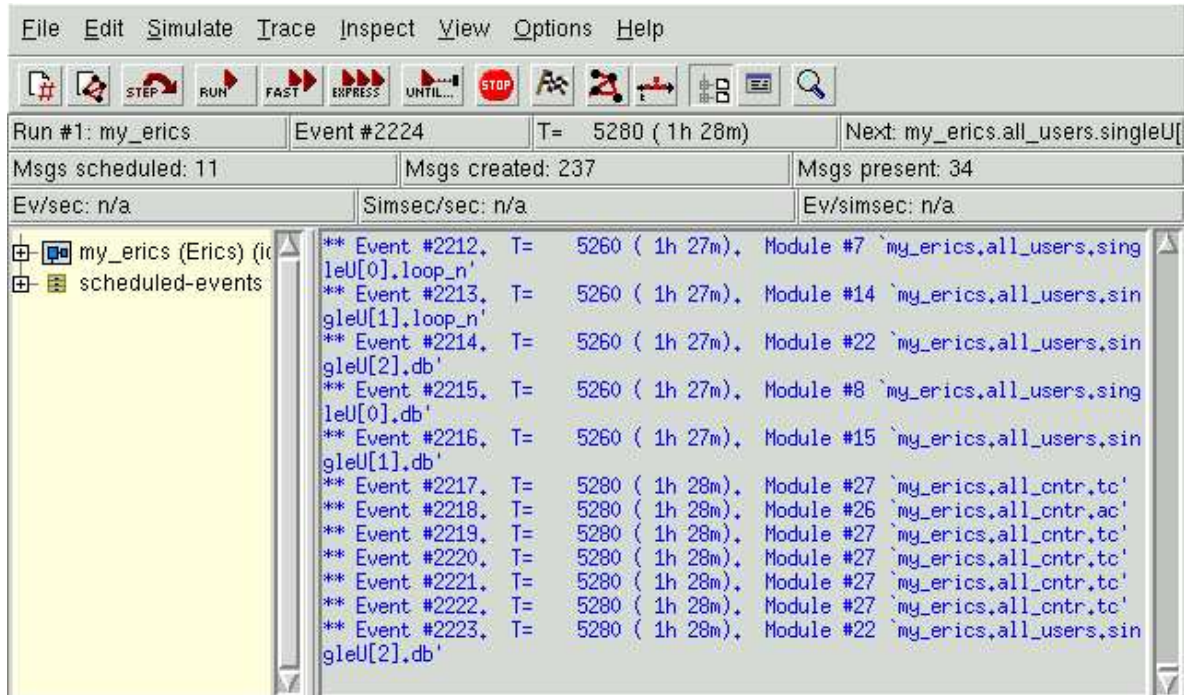


Figura 3.8: Captură de ecran conținând evenimentele de la timpul de simulare 5280ms

3.2.4 Modulul PCU

Modulul PCU (Packet Control Unit – unitatea de control a pachetelor) este unul dintre cele mai importante module hardware (alături de nodurile suport GPRS), adăugat în GPRS la configurația de bază GSM. Așa cum se menționează și în secțiunea 2.1, la pagina 9, PCU este responsabilă cu controlul traficului de pachete. În modelul de simulare realizat în cadrul acestei teze, modulului GPRS PCU îi corespunde nodul OMNeT++ PCU, a cărui structură este prezentată în figura 3.9. Nodul OMNeT++ PCU, ca și omologul său GPRS, este responsabil cu algoritmi de planificare utilizați pentru controlul transmiterii datelor (*Transmission Control – TC*).

Modulul PCU se compune dintr-un *nod sursă (source)*, un nod *file admission controller* și un nod *transmission controller*. *Sursa* generează un singur mesaj, care este apoi “ciclat” între nodurile *file admission controller* și *transmission controller*, durata ciclului fiind de 20 ms (numită și *block period*). Această durată corespunde intervalului la care modulul GPRS PCU alocă utilizatorilor care au date în cozi resursele radio ale rețelei, resurse reprezentate de câte 8 canale (sloturi de timp) pentru fiecare frecvență purtătoare.

Captura de ecran corespunzătoare modulului PCU este prezentată în figura 3.10. Imaginea surprinde momentul când nodul *file admission controller* transmite un mesaj

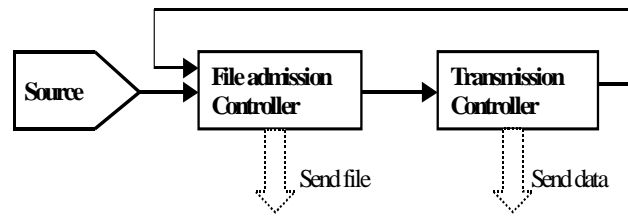


Figura 3.9: Modulul PCU

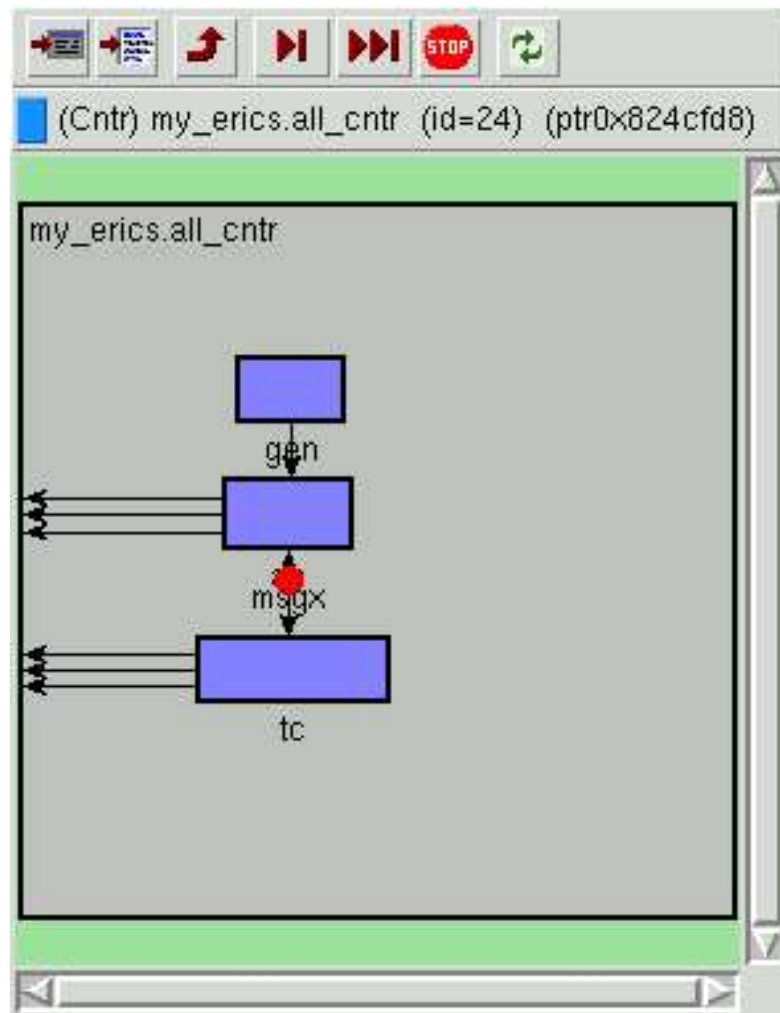


Figura 3.10: Reprezentarea OMNeT++ a modulului PCU

nodului *transmission controller*. Se observă că fiecare din aceste două noduri are câte 3 ieșiri, reprezentând liniile pe care se transmit comenzile *send file* și respectiv *send data* spre cele trei module utilizator din componența sistemului. Pentru mai multă

claritate, captura de ecran a fost realizată pentru situația când în sistem există doar 3 module utilizator.

Nodul *file admission controller*

Acest nod monitorizează utilizatorii care au date în cozi și determină care dintre ei pot deveni *activi*, adică au dreptul să transfere date, iar apoi le transmite comanda *send file*.

Conform [30] și celor prezentate în secțiunea 3.1.2, pagina 54, într-o rețea (E)GPRS există mai multe niveluri de *admission control (AC)*. În această teză se consideră că admiterea utilizatorilor (*admission control*) se realizează la activarea unui context PDP, când are loc negocierea cerințelor de QoS ale utilizatorului și când se ia decizia de admitere sau respingere a respectivului utilizator. Aspectele legate de acest nivel de AC vor fi discutate în capitolul 5. Un alt nivel de AC în (E)GPRS are loc la stabilirea unui flux de date temporar (TBF), atunci când are loc de fapt alocarea resurelor fizice ale rețelei pentru transferul datelor.

Nodul *file admission controller* modelează tocmai acest al doilea nivel de AC, și anume stabilirea unui TBF, care este o tehnică utilizată în (E)GPRS pentru a preveni risipirea resurselor (radio ale) rețelei prin evitarea alocării de canale radio utilizatorilor care nu au date în coadă. Acest nivel de *admission control* se mai numește *link level admission control* deoarece are loc la nivelul legătură de date (*data link level*) din stiva de protocoale.

În actuala implementare a nodului *file admission controller* sunt admiși toți utilizatorii care au date în coadă, deoarece considerăm că dacă algoritmi de activare a unui context PDP sunt eficienți, atunci rețeaua va dispune de resursele fizice necesare fiecărui utilizator care dorește să își stabilească un TBF. De altfel, în special pentru traficul de tip timp real (*streaming*) se utilizează conceptul de eliberare întârziată a TBF (*delayed TBF release* [4]) sau echivalent TBF cu sfârșit deschis (*open ended TBF* [90]). Această idee se bazează pe măsuratori experimentale și constă în menținerea deschisă a unui TBF chiar și atunci când utilizatorul respectiv nu mai are date în coadă, astfel încât la apariția următorului bloc de date, acesta să găsească conexiunea deschisă și să nu se mai piardă timp cu stabilirea unui nou TBF.

Nodul *transmission controller*

Acesta rulează algoritmi de planificare care determină pentru fiecare din utilizatorii activi (care au date în coadă) câte blocuri de date are fiecare dreptul să transfere în timpul perioadei curente a PCU (de 20 ms). Nu toți utilizatorii activi transferă date

în timpul unui astfel de perioade de 20ms. Numărul de blocuri de date pe care un utilizator are dreptul să le transfere la un moment dat este determinat de ponderea sa, a cărei valoare este în funcție de clasa de QoS a utilizatorului, de calitatea legăturii sale radio etc.

În capitolul 4 se vor prezenta diferiți algoritmi de planificare pe care i-am implementat în această teză, cum ar fi *Oldest Queue (OQ)*, *Longest Queue (LQ)*, *Total File Length (TFL)*, *Total Queue Length (TQL)* și *Iterative* sau *Weighted Round Robin (IRR* sau *WRR)*.

3.2.5 Generatorul de utilizatori, nodul admission controller și nodul stat

Această secțiune detaliază modulele din figura 3.5, care nu au fost discutate până acum.

Nodul generator de utilizatori

Modulul *generator de utilizatori – user generator* este folosit pentru a crea utilizatori de-a lungul simulării. Procesul de generare de utilizatori se desfășoară după anumite funcții de distribuție de probabilitate, fiecare utilizator având la creare un set de caracteristici: profilul QoS (exprimat prin clasa de trafic), timpul de *dwell*, precum și un parametru care specifică dacă utilizatorul este *nou* (dacă își începe sesiunea de transfer de date în celula curentă) sau dacă el vine din altă celulă, având deja o sesiune de transfer de date în plină desfășurare. Acest ultim parametru este foarte important pentru algoritmul de admitere a utilizatorilor (AC) deoarece se consideră că este mult mai neplăcut pentru un utilizator al unei rețele de telefonie mobilă să i se întrerupă o convorbire sau o sesiune în curs de desfășurare decât să îi fie respins un apel pentru o convorbire sau o sesiune nouă și deci utilizatorii care vin din alte celule vor avea în general șanse mult mai mari de a fi admiși de algoritmul de *admission control*.

Generatorul de utilizatori este alcătuit din două submodule: *un generator* și un modul numit *disconnect user*. Cele două submodule accesează o variabilă globală numită *adm_users*, care este un tablou unidimensional de numere întregi, dimensiunea tabloului fiind *N_MAX* (numărul maxim de module utilizator). Fiecare element al tabloului corespunde unui modul utilizator. Dacă modulul utilizator este liber (neocupat de un utilizator), atunci valoarea corespunzătoare din tablou este 0, iar în caz contrar, valoarea elementului tabloului este dată de clasa de trafic a utilizatorului care ocupă modulul utilizator respectiv: 3 pentru utilizator de tip streaming, 2 pentru

utilizator interactiv (WWW) și 1 pentru utilizator din clasa background.

Când un utilizator își încheie sesiunea de transfer de date sau când părăsește celula, el va anunța modulul *disconnect user* că a eliberat locul respectiv, astfel încât acesta să poată pune pe 0 valoarea corespunzătoare din tabloul *adm_users*.

Atunci când un utilizator creat de generatorul de utilizatori este admis în sistem de către algoritmul de *admission control*, modulul generator va căuta în tabloul *adm_users* și va asigura primul loc liber (elementul cu indicele cel mai mic având valoare 0) aceluia utilizator. Dacă nu există nici un loc liber se va genera un mesaj de eroare. Dacă există loc liber, atunci modulul generator de utilizatori va trimite un mesaj OMNeT++ (*start user*) conținând caracteristicile noului utilizator modulului utilizator corespunzător, iar acest mesaj va declanșa procedura de resetare a submodulelor componente ale modulului utilizator și de asemenea va porni timerul corespunzător timpului de *dwell* al utilizatorului.

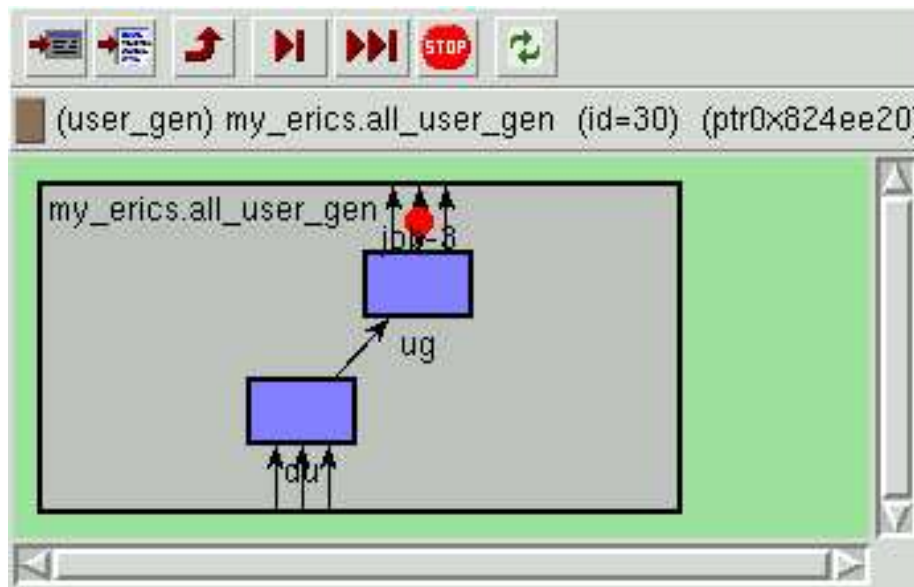


Figura 3.11: Reprezentarea OMNeT++ a modulului generator de utilizatori

Reprezentarea grafică a modulului generator de utilizatori în OMNeT++ este dată în figura 3.11. Se observă cele două submodule: generatorul, cu eticheta *ug* și *disconnect users*, cu eticheta *du*. Generatorul are câte o ieșire pentru fiecare din cei 3 utilizatori din sistem, cărora le transmite comanda *start user*, iar nodul *disconnect users* are 3 intrări, câte o intrare corespunzătoare fiecărui modul utilizator, prin care utilizatorii anunță că au încheiat sesiunea și vor elibera modulul utilizator. La momentul când a fost capturată imaginea, submodulele generator trimite o comandă *start user*, reprezentată printr-un mesaj OMNeT++ unui modul utilizator. Captura

de ecran a fost făcută pentru cazul când în sistem există doar 3 module utilizator.

Modulul *admission controller*

În GPRS admiterea utilizatorilor este realizată la activarea unui context PDP. În modelul de simulare din această teză, pentru admiterea utilizatorilor este utilizat un modul dedicat numit *admission controller*

La crearea unui utilizator de către modulul generator de utilizatori se trimite o cerere către modulul *admission controller* sub forma unui mesaj OMNeT++ conținând caracteristicile utilizatorului, iar acest modul va decide dacă să admită sau nu cererea respectivă. Dacă cererea este respinsă se va înregistra acest lucru (în scopul colectării statisticilor despre simulare). Dacă cererea a fost respinsă, atunci mesajul OMNeT++ este șters, iar dacă cererea a fost acceptată, atunci modulul generator de utilizatori urmează procedura descrisă mai înainte (caută un loc liber în tabloul *adm_users* și apoi trimite comanda *start user* nodului MN respectiv). Mesajul primit de la modulul generator de utilizatori și care conține cererea îi va fi returnat acestuia, însă se adaugă (sub forma de atribut al mesajului) și decizia de acceptare sau respingere.

Algoritmii dezvoltați pentru *admission control* sunt prezentați în capitolul 5.

Modulul pentru colectarea statisticilor *stat node*

Fiecare “fișier” transferat cu succes de utilizatorul GPRS ajunge la modulul pentru colectarea statisticilor despre simulare, numit *stat node*.

Modulul *stat* calculează, pentru fiecare utilizator, valorile medii ale următorilor parametri:

- timpul de așteptare (*waiting delay*) – reprezintă timpul pe care un fișier îl petrece în bufferul de fișiere, așteptând să fie transferat
- timpul de transfer (*sending delay*) – este timpul scurs din momentul când fișierul este transferat din *file buffer* în *data packet buffer* până în momentul când fișierul ajunge la *sink*
- timpul total (*total delay*) – este suma dintre *waiting delay* și *sending delay*
- valorile medii ale acestor timpi *per bit*
- cantitatea totală de date eliminate (datorită depășirii capacității bufferului de fișiere)

- numărul de blocuri retransmise
- durata sesiunii de transfer de date.

Menționăm că *waiting delay* este mai important în cazul în care utilizatorul generează date în mod streaming, deoarece atunci este mai probabil să se acumuleze mai multe fișiere în *file buffer*, situație când timpul mediu de așteptare în bufferul de fișiere poate crește. De asemenea, în mod streaming se poate ajunge mai ușor la situația de a se depăși capacitatea de stocare a bufferului de fișiere și la eliminarea unor fișiere.

Toate informațiile colectate despre procesul de control al transmiterii datelor (*transmission control*), precum și statisticile referitoare la procesul de admitere a utilizatorilor (*admission control*) – acestea din urmă colectate de nodul *admission controller*, sunt înregistrate în fișiere text, care pot fi ulterior procesate (trasare de grafice etc).

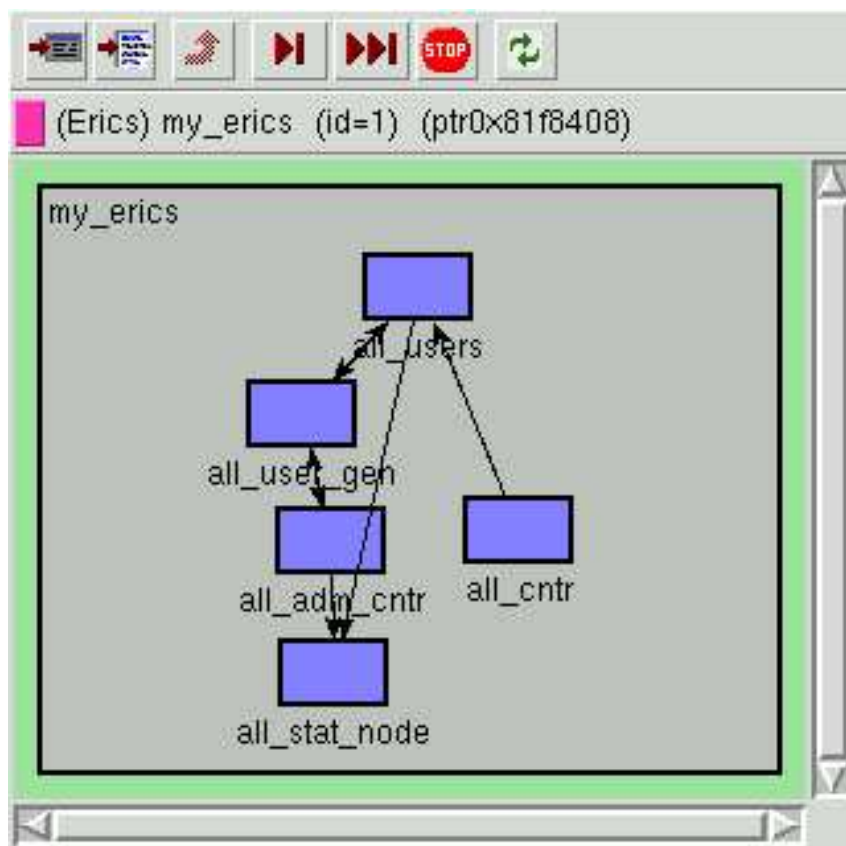


Figura 3.12: Reprezentarea OMNeT++ a întregului sistem

Figura 3.12 conține reprezentarea în OMNeT++ a întregului model de simulare, model alcătuit din: grupul modulelelor utilizatorilor, cu eticheta `all_users`, PCU (`all_cntr`), modulul generator de utilizatori (`all_user_gen`), modulul *admission controller* având eticheta `all_adm_cntr` și nodul *stat* cu eticheta `all_stat_node`.

Alte funcții ale modelului de simulare

Modelul de simulare mai realizează și alte funcții, care nu sunt reprezentate în figura 3.5. Calitatea legăturii radio poate fi modelată ca și în [43], unde am folosit un model al legăturii radio în funcție de mobilitatea utilizatorilor GPRS, pentru diferite tipuri de celule urbane din zone urbane sau rurale și pentru diferite viteze de deplasare a utilizatorilor.

Partajarea canalelor de trafic între voce (GSM) și date (GPRS) este modelată în felul următor: numărul de canale alocate pentru GSM se modelează cu ajutorul unor funcții de distribuție de probabilitate (de exemplu funcții de tip exponențiale trunchiate), iar canalele rămase sunt folosite pentru traficul de date.

3.3 Validarea modelului

O modalitate de a valida modelul de simulare dezvoltat în această teză a fost de a “emula” algoritmul idealizat de planificare *Fluid Fair Queueing (FFQ)* (prezentat în secțiunea 2.2.1), [73]. Pentru emularea acestui algoritm idealizat vom folosi *WRR (Weighted Round Robin)* ca și algoritm de planificare. Algoritmul FFQ poate fi folosit pentru a testa corectitudinea rezultatelor simulării (și implicit, corectitudinea modelului de simulare) deoarece se știe că, în cazul FFQ, raportul între timpii de transfer (*sending delays*) a doi utilizatori este egal cu raportul ponderilor lor (a se vedea 2.2.1).

Pentru a emula FFQ, au fost separate fluxurile de date ale utilizatorilor, lucru pe care l-am realizat punând sursele de date ale utilizatorilor în modul de generare continuă, adică generând următorul fișier imediat ce precedentul fișier a fost transferat cu succes (a ajuns la *sink*).

Pentru a determina analitic valoarea parametrului *sending_delay* al unui fișier de lungime FL se utilizează formula (2.1), care exprimă fracțiunea din capacitatea canalului pe care o primește un flux de date într-un interval infinitezimal de timp și se obține rata de servire $service_rate_f$ de care beneficiază un flux de date nevid f la orice moment de timp t [122]:

$$service_rate_f = C \cdot \frac{r_f}{\sum_{i \in B_{FFQ}(t)} r_i} \quad (3.2)$$

unde $B_{FFQ(t)}$ este mulțimea fluxurilor nevide la momentul t .

În cazul GPRS, capacitatea rețelei constă în numărul de canale radio alocate traficului de date într-un ciclu PCU. Parametrilor r_i din (3.2) le corespund ponderile W_i alocate fiecărei clase i de QoS. Notăm cu k_i numărul de utilizatori din clasa i de QoS. Se presupune că toate cele $B = 8$ canale radio sunt alocate traficului de date, adică $C = B$.

Întârzierea unui fișier din clasa i de QoS ($sending_delay_i$) este dată de raportul dintre lungimea fișierului și cantitatea serviciului primită într-o unitate de timp (adică rata serviciului), înmulțit cu durata intervalului de alocare a serviciului Δt :

$$sending_delay_i = \frac{FL}{service_rate_i} \cdot \Delta t \quad (3.3)$$

În relația (3.3) intervalul de timp Δt este egal cu durata unui ciclu PCU, $cntr_cycle$, acesta fiind intervalul de timp în care se alocă resursele rețelei, și are valoarea de 20ms.

Rezultă că parametrul $sending_delay$ pentru un utilizator din clasa de QoS i are valoarea:

$$sending_delay_i = \frac{1}{W_i} \cdot \frac{FL \cdot \sum_i (k_i \cdot W_i)}{B} cntr_cycle \quad (3.4)$$

În tabelul 3.2 se prezintă o comparație între valorile obținute prin simulare și cele calculate ale parametrului $sending_delay$ pentru trei clase de QoS (notate HP, MP și LP), atunci când fiecare utilizator generează date în mod continuu. Durata simulării a fost de 500000 ms, iar valorile parametrilor sunt: $FL = 360$ blocuri radio, $k_{LP} = 2$, $k_{MP} = 5$, $k_{HP} = 3$, $W_{LP} = 1$, $W_{MP} = 2$, iar W_{HP} ia valorile 3, 4, 10 și 20.

Clasa de QoS	$W_{HP} = 3$		$W_{HP} = 4$		$W_{HP} = 10$		$W_{HP} = 20$	
	sim	calc	sim	calc	sim	calc	sim	calc
HP	6299.7	6300	5399.6	5400	3799.7	3800	3239.6	3240
MP	9449.8	9450	10799.8	10800	18900.0	18900	32399.3	32400
LP	18900.0	18900	21600.0	21600	37800.0	37800	64800.0	64800

Tabelul 3.2: Comparație între valorile obținute prin simulare și valorile calculate pentru $sending_delay$

Rezultatele testelor, prezentate în tabelul 3.2, arată o concordanță perfectă între timpii de transfer ($sending_delays$) obținuți prin simulare și cei calculați pentru algoritmul FFQ. Așa cum se observă din tabel, raportul între întârzieri este egal cu raportul ponderilor utilizatorilor.

A fost testat și generatorul de utilizatori, caz în care am obținut fluxurile continue de date prin crearea unui nou utilizator (având același profil QoS și deci aceeași pondere ca și utilizatorul precedent) imediat după ce utilizatorul precedent și-a încheiat sesiunea de transfer de date (evident, toți utilizatorii au folosit modul de generare continuă de date pentru sursele lor de date). Și în acest caz rezultatele simulărilor au coincis cu cele calculate.

Merită menționat faptul că felul în care am modelat transferul de date (stabilirea fluxurilor temporare de blocuri –TBF– și alocarea canalelor între utilizatori) este mult mai simplu decât în cazul modelelor prezentate de Ajib și Godlewski [2] sau de Stuckmann și colegii [94], modele care conțin numeroase detalii tehnice privitoare la modul de alocare a resurselor radio. Faptul că am evitat includerea în modelul de simulare din această teză a unor detalii tehnice inutile ne-a permis să ne concentrăm mai mult asupra performanțelor algoritmilor de planificare. În acest fel am putut să explicăm de exemplu de ce algoritmi bazați pe cantitatea de date din cozile utilizatorilor (care alocă mai multe resurse utilizatorilor cu mai multe date în coadă) nu sunt eficienți pentru problema controlului transiterii datelor (transmission control). Explicațiile pentru această problemă au fost prezentate pe larg în [110], [101] și vor fi tratate de asemenea în capitolul 4.

3.4 Concluzii

Acest capitol prezintă modul cum se face alocarea resurselor în GPRS și EGPRS, iar în partea a doua, modelul de simulare dezvoltat în cadrul acestei teze pentru studiul problemei alocării resurselor în (E)GPRS.

Secțiunea 3.1.1 prezintă, pornind de la arhitectura de protocoale GPRS, fluxul de date și codificarea canalelor în GPRS. Spre deosebire de alte simulatoare și emulatoare de GPRS, cum ar fi GPRSim, realizat de Stuckmann și colegii ([90], [47], [89], [48],[96] prezentate în secțiunea 2.3.3), care implementează la nivel de detaliu stiva de protocoale GPRS, în modelul de simulare pe care l-am realizat acest lucru se face simplificat, deoarece de fapt dorim să modelăm fluxul de date în GPRS, mai degrabă decât fiecare protocol din stiva GPRS. Relativ la figura 3.2, care descrie fluxul de date în GPRS, în teză ne axăm pe modelarea alocării resurselor la nivel MAC/RLC, fără a studia influența pe care protocoalele de nivel mai înalt (de exemplu TCP) o au asupra performanțelor algoritmilor de alocare a resurselor. În consecință, am simplificat modul de reprezentare a fluxului de date astfel: un utilizator generează la un moment dat unul sau mai multe “fișiere”, care sunt apoi segmentate în radio

blocuri la transmițător, iar la capătul receptor se face reasamblarea blocurilor radio în “fișiere”. Dependent de nivelul de detaliu pe care îl urmărim, “fișierele” din model pot fi considerate fie cadre LLC (sau echivalent, pachete IP), și atunci un fișier real este compus în mai multe astfel de cadre, fie un “fișier” din model corespunde direct unui fișier real, sau chiar unui grup de fișiere (de exemplu, conținutului unei pagini de web), situație în care simplificăm felul în care are loc în realitate segmentarea unui fișier real până la nivel de blocuri radio, comasând toate nivelurile reale de segmentare într-unul singur.

Utilizatorii GPRS din model, aidoma celor reali, au posibilitatea de a genera date în mod interactiv sau în mod *streaming*, aparțin unor clase diferite de calitate a serviciului (lucru modelat prin ponderea pe care o are fiecare utilizator), aceste clase de QoS putând fi bazate fie pe clasele de trafic (ca în GPRS release 1999, a se vedea pagina 55), fie pe baza subscripției utilizatorului. De asemenea, în model sunt utilizate schemele de codare din GPRS sau schemele de modulare și codare din EGPRS. Se poate modela de asemenea procesul de retransmitere a blocurilor de date afectate de erori, cu ajutorul *nodului de buclare* și al *nodului data packet buffer* din structura unui modul MN.

Referitor la managementul mobilității dintr-o rețea GPRS reală, acesta este simplificat în model, nefiind incluse procedurile de *attach* și *detach*, dar utilizatorii GPRS din model, la fel ca și cei reali, se pot deplasa dintr-o celulă în alta. Pentru situația când un MN părăsește celula înainte de încheierea sesiunii de transfer de date este utilizat nodul *handoff* (pagina 63), iar apariția în celula curentă a unui MN dintr-o celulă învecinată este modelată, în cazul de interes în care respectivul MN are o sesiune în curs de desfășurare, cu ajutorul modulului generator de utilizatori, care atribuie fiecărui MN creat un atribut care stabilește dacă acesta provine dintr-o altă celulă sau inițiază o sesiune de lucru nouă.

Activarea (și dezactivarea) unui context PDP în cazul unui utilizator real este realizată în model cu ajutorul algoritmilor de admitere a utilizatorilor, respectiv al modulului *admission controller*. Algoritmii propuși în această teză pentru admiterea utilizatorilor vor fi detaliați în capitolul 5. De asemenea este modelat și procesul de stabilire a unui flux temporar de date (TBF), prin nodul *file admission controller*, component al modulului OMNeT++ PCU.

Modulul PCU din model îndeplinește aceleași funcții ca și corespondentul său real, având aceeași perioadă de lucru (*block period*) de 20 de milisecunde, iar algoritmi de planificare a resurselor pe care îi rulează PCU pentru a partaja resursele rețelei între utilizatorii activi vor fi detaliați în capitolul 4.

Contribuțiile din acest capitol pot fi rezumate astfel:

1. Realizarea unui model de simulare care să permită studiul tehnicilor de alocare a resurselor în rețele de tip GPRS/EGPRS.
2. Caracteristicile acestui model, comparativ cu modele similare prezentate în literatură sunt:
 - gradul de abstractizare al modelului, care permite pe de o parte simularea unor situații complexe, de tip real, iar pe de altă parte evită detaliile de implementare care nu se referă la problemele investigate.
 - permite studiul algoritmilor de alocare a resurselor la controlul transmiterii datelor (TC), fără ca rezultatele privind performanțele algoritmilor să fie influențate de factori cum ar fi mecanismele de control ale protocoalelor TCP sau LLC, sau ale interferențelor complexe dintre protocoalele din stiva de protocoale (E)GPRS
 - permite de asemenea studiul performanțelor algoritmilor de admitere a utilizatorilor (AC).
3. Modelul a fost validat prin compararea rezultatelor simulărilor cu rezultatele analitice obținute pentru algoritmul idealizat *Fluid Fair Queueing*, atunci când utilizatorii generează fluxuri continue de date.
4. Modelul a fost proiectat și implementat în întregime de către autorul tezei.
5. Descrieri ale acestui model au fost publicate în lucrările [43] și [100].

Capitolul 4

Performanțele algoritmilor pentru controlul transmiterii datelor

În acest capitol se prezintă o serie de algoritmi de planificare la nivel MAC/RLC pentru problema controlului transmiterii datelor în rețele celulare de date de tip GPRS și EGPRS și se investighează performanțele lor prin prisma întârzierilor datelor utilizatorilor (E)GPRS. Algoritmii sunt descriși în secțiunea 4.1.

Investigațiile se realizează prin simulare, condițiile în care se realizează simulările fiind prezentate în secțiunea 4.2, în timp ce rezultatele simulărilor și interpretarea acestora alcătuiesc secțiunea 4.3.

Secțiunea 4.4 se ocupă cu studiul influenței factorilor de pondere ai utilizatorilor și tratează situația când ponderile sunt folosite pentru diferențierea calității serviciului oferit utilizatorilor, cât și cazul în care ponderea e asociată calității legăturii radio a utilizatorilor, caz folosit pentru reducerea congestiilor din rețea.

Metodele folosite pentru a asigura un grad ridicat de încredere în rezultatele simulărilor sunt prezentate pe larg în 4.5.

Capitolul se încheie cu o secțiune de concluzii, care sumarizează și contribuțiile personale ale autorului.

4.1 Algoritmi pentru controlul transmiterii datelor

În secțiunile 2.2 și 2.3 am discutat diverși algoritmi utilizați în literatură pentru controlul transmiterii datelor.

În această teză am investigat două categorii de algoritmi:

- algoritmi de tip Packet Fair Queueing: WRR (Weighted Round Robin) și Oldest Queue (OQ)

- algoritmi bazați pe cantitatea de date din cozile utilizatorilor: Longest Queue (LQ), Total Queue Length (TQL) și Total File Length (TFL).

Acești algoritmi sunt utilizați la nivel MAC/RLC și au fost implementați în modulul PCU, descris în 3.2.4.

4.1.1 Algoritmi de tip Packet Fair Queueing

Pentru pachete de lungime constantă algoritmul WRR este o aproximare foarte bună a WFQ ([61]). În cazul GPRS, locul pachetelor este luat de blocurile radio, care au o lungime fixă, deci se poate considera că algoritmul WRR este o bună aproximare a FFQ. Un avantaj major al algoritmului WRR comparativ cu alți algoritmi de tipul PFQ este complexitatea sa mult mai redusă, aspect foarte important în GPRS și EGPRS, unde algoritmul de scheduling implementat în PCU (Packet Control Unit) trebuie să furnizeze rezultate la fiecare ciclu al PCU, care are o durată de 20 de milisecunde.

În cazul algoritmului **Round Robin** controlerul trebuie să știe doar care utilizatori cer acces la resursele rețelei (canale radio). Fiecare utilizator este lăsat să transfere un singur bloc de date la un moment dat, astfel încât dacă există B PDTCH disponibile (de exemplu $B = 8$) și doar $K < B$ utilizatori activi (care au date în cozi), atunci doar K din cele B canale sunt utilizate, rămânând $(B - K)$ canale nefolosite, ceea ce înseamnă că resursele radio ale rețelei nu sunt utilizate eficient (sunt risipite). Avantajele algoritmului RR sunt simplitatea sa și faptul că este echitabil (*fair*). Dar din cauza lipsei sale de eficiență, am utilizat algoritmul doar în primele stadii ale cercetării, iar apoi l-am înlocuit cu IRR (WRR). Un alt dezavantaj al RR este că nu poate asigura diferențierea calității serviciilor (QoS) între utilizatori.

Algoritmul **WRR** elimină aceste dezavantaje ale RR. În prima iterație a algoritmului WRR fiecare utilizator primește un anumit număr de canale (PDTCH), în concordanță cu nivelul său de prioritate. Dacă mai rămân canale disponibile după ce fiecare utilizator a transferat numărul de blocuri de date ce i-a fost asignat, atunci algoritmul va mai parcurge încă o iterație, împărțind resursele (canalele) rămase utilizatorilor care mai au blocuri de date în coadă. Algoritmul se oprește fie atunci când toate cele B PDTCH disponibile au fost alocate utilizatorilor, fie atunci când toate cozile utilizatorilor sunt goale.

Prioritatea utilizatorilor este modelată printr-un factor de pondere (*weighting factor*), notat cu W în model (sau cu X în primele versiuni ale modelului [110], [109],

[111]): un utilizator cu prioritatea W va transfera W blocuri de date într-un ciclu PCU de 20ms, dacă la acel moment există cel puțin W PDTCH disponibile. Dacă numărul de canale disponibile este mai mic decât ponderea, atunci utilizatorul va transfera în ciclul curent atâtea blocuri de date câte canale sunt disponibile, urmând ca în ciclul următor să transfere restul de blocuri (rămase până la W).

Descrierea în pseudocod a algoritmului WRR (IRR) este prezentată în figura 4.1.

```

/* WRR (IRR) initial data:
   B - the bandwidth;
   weight[i]- the weighting factor for user i (a positive
   integer value);
   user[i].queue_length - the number of data block in the
   queue (data packet buffer) of user[i]

Calculated data:
   send - the number of data blocks sent by a user at a
   certain iteration
   not_sent[i] - the number of data blocks that user[i]
   could not send because there were not enough channels
   available: they will be send in the next cycle(s)
*/

no_of_available_channels=B;
i = initial_value;
while((no_of_available_channels > 0) and (there is one
not empty user queue)){
  if( user[i].queue_length >0){
    send = min(weight[i],no_of_available_channels,
    user[i].queue_length);
    not_sent[i]=min(0, min(weight[i],
    user[i].queue_length) -no_of_available_channels);
    user[i].queue_length = user[i].queue_length - send;
    no_of_available_channels = no_of_available_channels
    - send;

    }//end if
  if (not_sent[i]==0)
    i++; //go to next user in the next iteration;
  //else user[i] will continue to send in the next iteration
} //end while

```

Figura 4.1: Descrierea în pseudocod a algoritmului WRR (IRR)

Înainte de parcurgerea iterațiilor algoritmului WRR, are loc inițializarea unor variabile: B , $weight[i]$ și $user[i].queue_length$.

B reprezintă lățimea de bandă, adică numărul de canale disponibile în ciclul PCU curent de 20 ms. Dacă presupunem că toate cele 8 canale sunt disponibile pentru traficul GPRS, atunci valoarea variabilei B este fixă de-a lungul simulării și egală cu 8. Dacă luăm în considerare și traficul GSM (voce), atunci valoarea lui B se va

modifica de la o rulare a algoritmului la alta, în funcție de activitatea părții de voce și de politica operatorului de alocare a resurselor radio între partea de voce și cea de date (de obicei partea de voce are prioritate față de cea de date).

Parametrul $weight[i]$, factorul de pondere al unui utilizator (*weighting factor*) reprezintă ponderea asignată aceluși utilizator în funcție de nivelul său de prioritate sau/și precedentă. Valoarea factorului de pondere poate rămâne fixă de-a lungul simulării sau poate fi ajustată dinamic de către un algoritm de adaptare a lățimii de bandă (*BAA – Bandwidth Adaptation Algorithm*).

Lungimile cozilor utilizatorilor, mai precis numărul de blocuri de date pe care fiecare utilizator le are în *data packet buffer*, este exprimat prin $user[i].queue_length$.

Valoarea variabilei $queue_length$ este actualizată pentru fiecare utilizator atunci când acesta trimite date din coadă sau atunci când un nou fișier este segmentat în blocuri de date care sunt transferate în *data packet buffer*.

Numărul de blocuri de date pe care un utilizator le va transfera în ciclul curent de 20ms (parametrul $send$) este calculat pentru fiecare utilizator. Este posibil ca un utilizator să fie planificat să transfere un anumit număr de blocuri de date, dar la acel moment să nu fie suficiente canale disponibile. Parametrul $not_sent[i]$ memorează, pentru fiecare utilizator, numărul de blocuri de date care nu au putut fi transferate în ciclul PCU curent datorită numărului insuficient de canale disponibile. Utilizatorul respectiv va fi planificat să trimită aceste blocuri de date în ciclul următor. Parametrul $not_sent[i]$ este calculat ca minimum dintre factorul de pondere al utilizatorului și lungimea cozii acestuia, exprimată în blocuri de date.

Un ciclu PCU (ciclu al controlerului) are o durată de 20ms și poate consta în una sau mai multe iterații ale algoritmului WRR.

Un ciclu de controler începe prin inițializarea valorii variabilelor B (numărul de canale alocate pentru GPRS) și i (utilizatorul curent, adică cel care este planificat să transfere date). Utilizatorul curent este determinat în urma procedurii de round robin.

Atâta timp cât mai sunt canale disponibile și cât mai există date în cozile utilizatorilor, algoritmul va continua să parcurgă iterații printre utilizatori. Dacă un utilizator care a fost planificat să transmită nu mai are date în coadă, algoritmul va trece la următorul utilizator, iar în caz contrar (când coada nu e goală), algoritmul va calcula parametrul $send$, ca fiind valoarea minimă între factorul de pondere al utilizatorului, lungimea cozii utilizatorului (exprimată în blocuri de date) și numărul de canale disponibile.

Înainte de a se trece la următorul utilizator, se va actualiza lungimea cozii utilizatorului curent i , precum și numărul de canale rămase disponibile. Dacă utilizatorul

$user[i]$ avea în coadă un număr de blocuri de date mai mic decât numărul de canale disponibile, atunci se va trece la următorul utilizator, care va încerca să transfere date în concordanță cu ponderea sa.

Algoritmul Oldest Queue va fi explicat în secțiunea 4.1.2 deoarece implementarea sa este asemănătoare cu a algoritmilor bazați pe cantitatea de date din cozile utilizatorilor.

4.1.2 Algoritmii de planificare bazați pe cantitatea de date din cozile utilizatorilor

Acești algoritmi utilizează un criteriu de selecție a utilizatorului planificat să transmită, criteriu care este evaluat de către controler (PCU). Utilizatorii sunt ordonați descrescător după acel criteriu și utilizatorul care are valoarea cea mai mare pentru criteriul respectiv va primi toate canalele (PDTCHs) disponibile. Dacă utilizatorul respectiv nu are suficiente date în coadă, atunci i se vor aloca resurse și următorului utilizator (rezultat în urma procesului de ordonare) și așa mai departe până când fie nu mai sunt resurse (canale) disponibile, fie nu mai sunt date în cozile utilizatorilor.

Criteriul după care sunt ordonați descrescător utilizatorii este în funcție de algoritmul utilizat, astfel:

- pentru algoritmii bazați pe cantitatea de date din cozile utilizatorilor, pentru LQ criteriul este numărul de blocuri de date din *data packet buffer*, iar TQL și TFL consideră pentru fiecare utilizator cantitatea totală de date din cele două buffere: *data packet buffer* și *file buffer*. Pentru TQL cantitatea totală de date se consideră exprimată în număr de blocuri de date, iar pentru TFL se exprimă în biți.
- pentru OQ criteriul este timpul scurs de când utilizatorul respectiv a fost servit ultima dată. Se subliniază că OQ este tot un algoritm de tip round robin, similar cu WRR, dar cu un ciclu mai lung. Considerăm *ciclul algoritmului* ca fiind intervalul necesar ca toți utilizatorii să fie serviți (planificați să transmită) cel puțin o dată, presupunând că toți utilizatorii continuă să aibă date în coadă.

Descrierea în pseudocod a algoritmilor OQ, LQ, TQL și TFL este dată în figura 4.2.

Toți utilizatorii au o *valoare actuală* (*actual value*) și o *valoare efectivă* (*effective value*) pentru criteriul respectiv. Valoarea actuală reprezintă valoarea reală a criteriului (un număr pozitiv), iar valoarea efectivă se obține din cea actuală prin înmulțire cu factorul de pondere (*weighting factor*) al utilizatorului respectiv. De exemplu,


```

/* The other algorithms:
OQ: actual_value[i] = simulation time elapsed since user[i] was
served last time
LQ: actual_value[i] = user[i].queue_length (the number of data
blocks in data packet buffer)
TQL: actual_value[i] = total amount of data waiting in both file
buffer and data packet buffer, expressed in radio blocks
TFL: actual_value[i] = total amount of data waiting in both file
buffer and data packet buffer, expressed in bits
K = no of active users (that request network resources) */

// Initialize effective_value[]
for(i=0; i<K; i++)
    effective_value[i] = actual_value[i] * weight[i];
//all actual_value[i] and effective_value[i] >= 0;

// the auction:
order_users_function();
/* arranges users in the decreasing order of
effective_value[i] and stores them in the array
ordered_users[]

"winner" is the user with the highest value of
effective_value[i], i.e. user[winner] is ordered_users[0] */

j = 0; // start with the "winner"
while (no_of_available_channels>0) and (there is at least one not
empty user queue){
    if( ordered_users[j].queue_length >0){
        send = min(no_of_available_channels,
            ordered_users[j].queue_length);
        no_of_available_channels = no_of_available_channels - send;
        ordered_users[j].queue_length =
            ordered_users[j].queue_length - send;
    }//end if
    if(no_of_available_channels > 0) {
        j++ //next user
    }
    //else all the available channels are used and the cycle is finished;
//another auction has to take place
}

```

Figura 4.2: Descrierea în pseudocod a algoritmilor OQ, LQ, TQL și TFL

pentru LQ, *valoarea efectivă a cozii* = (*valoarea actuală a cozii*) · *factor de pondere*.

$$\text{effective queue length} = (\text{actual queue length}) \cdot \text{weighting factor}.$$

Factorii de pondere sunt numere întregi pozitive. Pentru un utilizator cu prioritate scăzută factorul de pondere va fi 1 sau o valoare apropiată de 1, iar pentru un utilizator

cu prioritate ridicată, factorul de pondere va avea o valoare considerabil mai mare decât 1.

Valoarea luată în considerare la ordonarea descrescătoare a utilizatorilor este cea efectivă, avantajând astfel utilizatorii cu prioritate ridicată față de cei cu prioritate scăzută deoarece primii au șanse să câștige mult mai des licitația pentru resursele rețelei.

Licitarea este realizată de funcția *order_users_function*. Această funcție primește un tablou conținând valorile efective pentru toți utilizatorii și calculează un alt tablou, numit *ordered_users*, în care utilizatorii sunt așezați în ordinea descrescătoare a valorilor efective. În acest fel, câștigătorul licitației va fi *ordered_user[0]*.

Algoritmul începe să acorde resursele rețelei câștigătorului licitației și se va opri atunci când toate canalele au fost alocate sau când toate cozile utilizatorilor sunt goale.

Dacă utilizatorul care a fost planificat să transmită are coada nevidă, atunci el va transfera un număr de blocuri de date dat de minimul dintre numărul de canale disponibile și lungimea cozii sale (numărul de blocuri de date din *data packet buffer*). După ce se calculează această valoare, se vor actualiza lungimea cozii și numărul de canale disponibile.

Dacă utilizatorul care a câștigat licitația are în coadă mai puține blocuri de date decât numărul de canale disponibile, atunci următorul utilizator (în ordinea descrescătoare a valorilor efective) este planificat să transmită și procedura se repetă până la alocarea tuturor canalelor sau până la golirea tuturor cozilor utilizatorilor (*data packet buffers*).

Licitarea se repetă la fiecare ciclu al controlerului.

4.2 Descrierea condițiilor experimentelor

Pentru a compara performanțele algoritmilor de alocare a resurselor și pentru a studia influența factorului de pondere asupra performanțelor acestor algoritmi au fost efectuate două seturi de simulări, în cadrul fiecărui set putând exista unele subseturi, obținute prin modificarea unor parametri.

În ambele seturi de experimente au fost folosiți 10 utilizatori, aceștia putând avea prioritate mare (*HP - High Priority*), medie (*MP - Medium Priority*) sau scăzută (*LP - Low Priority*). Performanțele algoritmilor de alocare a resurselor sunt exprimate prin trei parametri:

- timpul de așteptare a fișierelor în bufferul de fișiere (*waiting delay*)

- timpul de transfer al fișierelor (*sending delay*), care reprezintă durata transferului unui fișier care a fost deja segmentat în blocuri radio și se găsește în *data packet buffer*
- întârzierea totală a unui fișier (*total delay*) sau timpul total de transfer al unui fișier, care este suma celorlalți doi timpi. Cum timpul total de transfer este parametrul perceput de utilizatorul rețelei GPRS, acesta exprimă cel mai bine performanțele transferului de date.

De asemenea, în ambele seturi de experimente numărul de canale disponibile pentru date este de 8 canale la fiecare 20 milisecunde.

4.2.1 Primul set de experimente

În primul set de experimente s-a modificat încărcarea rețelei, mai precis gradul de ocupare a canalelor (*channel occupancy*) de la 10% la 95% prin modificarea intervalului la care utilizatorii generează date. Utilizatorii generează date la anumite intervale sub forma unor “pagini” care constau din unul sau mai multe fișiere. În aceste experimente, pentru fiecare utilizator o pagină constă fie dintr-un singur fișier având lungimea uniform distribuită în intervalul (2700, 3300) octeți, fie din 3 fișiere cu lungimea uniform distribuită în intervalul (900, 1100) octeți.

Intervalele de generare a datelor sunt obținute fie printr-o distribuție de probabilitate triunghiulară în jurul valorii medii *inter_page_delay*, fie printr-o distribuție exponențială având valoarea medie *inter_page_delay*.

În cazul distribuției triunghiulare se generează date în jurul valorii medii, într-un interval cuprins între [0.9, 1.1] din valoarea medie, conform relației

$$triang(0.9, 1, 1.1) \cdot inter_page_delay$$

Această distribuție asigură o împrăștiere redusă a valorilor intervalului de generare a datelor în jurul valorii medii.

Datele generate cu distribuția exponențială au un caracter mult mai neregulat (*bursty*) decât în cazul distribuției triunghiulare, deoarece rezultatul expresiei *exponential(inter_page_delay)* poate avea orice valoare pozitivă, pe când datele obținute din distribuția triunghiulară nu pot avea valori în afara intervalului *inter_page_delay*·[0.9, 1.1]. În acest mod, la distribuția exponențială se pot genera succesiv date la intervale mult mai scurte decât intervalul mediu, ceea ce duce, mai ales în situația când încărcarea rețelei este mare (90% sau 95%) la perioade când utilizatorii generează mai multe date decât pot fi procesate pe moment. Totuși, până la urmă, toate datele vor putea fi transmise.

Valoarea parametrului *inter_page_delay* se modifică de la aproximativ 33 de secunde pentru o încărcare de 10% la aproximativ 3.5 secunde pentru o încărcare de 95%.

Toți utilizatorii încep să genereze date simultan, dar ei tind să se desincronizeze pe parcurs. Durata simulării este de 5000 de secunde, iar algoritmi studiați sunt WRR, OQ, TFL, TQL și LQ.

În cadrul acestui set de experimente distingem două subseturi de experimente:

- *Subsetul A*: toți utilizatorii au aceeași pondere, factorul de pondere W fiind egal cu 1. În cadrul acestui subset sunt comparate performanțele tuturor algoritmilor de planificare investigați în acest capitol (WRR, OQ, LQ, TQL și TFL).
- *Subsetul B*: primii 3 utilizatori sunt de tip HP, următorii 5 de tip MP, iar ultimii 2 de tip LP. Ponderea utilizatorilor de tip LP este $W_{LP} = 1$, a celor MP poate fi $W_{MP} = 2$ sau 4, iar a celor HP este W_{HP} cu valorile 3 sau 8, dar întotdeauna $W_{HP} \geq W_{MP} \geq W_{LP}$. Practic, seturile de valori pentru cele 3 ponderi pot fi: $\{1,2,3\}$, $\{1,2,8\}$ și $\{1,4,8\}$. Acest subset de experimente a fost folosit pentru studiul influenței factorilor de pondere asupra performanțelor algoritmilor de alocare a resurselor în scopul realizării diferențierii calității serviciilor oferite utilizatorilor. Diferențierea calității serviciului se realizează în acest caz pe baza tipului de subscripție a utilizatorilor.

4.2.2 Al doilea set de experimente

În acest set de simulări media intervalului de generare a datelor este fixă și are valoarea de o secundă, iar încărcarea rețelei (gradul de ocupare a canalelor) are o valoare de 90%. S-a ales o valoare așa de mare a încărcării rețelei deoarece este foarte important să se cunoască eficiența algoritmilor de alocare a resurselor atunci când încărcarea rețelei este mare și există pericolul congestionării rețelei. Fiecare utilizator generează o pagină constând dintr-un singur fișier având lungimea de 36 de blocuri de date (ceea ce rezultă într-o încărcare medie a rețelei de 90%). Lungimea fișierelor este exprimată de această dată în blocuri radio, nu în biți, ceea ce face ca algoritmi TFL și TQL să se comporte identic, fiind deci simulat doar unul din ei (TQL). Pentru toți utilizatorii a fost folosită schema de codare CS1, ceea ce înseamnă că fiecare bloc radio are o lungime de 181 de biți, dar rezultatele nu diferă dacă se folosește o altă schemă de codare.

Intervalele dintre momentele de generare a paginilor respective sunt fie exponențial, fie triunghiular distribuite în jurul valorii medii, funcțiile de distribuție fiind la fel ca și la 4.2.1.

La aceste experimente se vor folosi doar două categorii de utilizatori, LP și HP, ponderea utilizatorilor LP fiind întotdeauna egală cu 1, iar a celor HP poate lua valorile 1, 2, 4, 8 sau 10.

La aceste experimente se modifică numărul utilizatorilor cu prioritate ridicată, notat cu n_{HP} , de la 1 la 10, studiindu-se în primul rând performanțele utilizatorilor de tip HP, în funcție de numărul lor (raportat la numărul total de utilizatori) și de ponderea pe care o au. De asemenea, au fost comparate și performanțele celor patru algoritmi de alocare a resurselor.

Pentru a reduce gradul de corelare a utilizatorilor, momentul la care fiecare utilizator începe să genereze date este uniform distribuit în intervalul $[0, 100]$ milisecunde.

Algoritmii simulați au fost WRR, OQ, TQL și LQ, iar durata simulării a fost de 5000 de secunde, la fel ca și la primul set de experimente.

La fel ca și la primul set de experimente (4.2.1), atunci când se compară performanțele diversilor algoritmi, condițiile în care se desfășară simularea sunt identice, inclusiv în ce privește numerele aleatoare generate de către simulator. Atunci când se dorește, se pot schimba numerele (pseudo)aleatoare generate de către simulator prin schimbarea valorii inițiale (*seed*) de la care pornește generarea numerelor aleatoare, indiferent de distribuția lor de probabilitate. Așa cum se va prezenta în 4.5, majoritatea simulărilor s-au efectuat cu 5 valori diferite pentru *seed*.

4.3 Compararea performanței algoritmilor de alocare a resurselor

În această secțiune se vor compara prin simulare performanțele algoritmilor de planificare WRR, OQ, LQ, TQL și TFL în două situații:

1. încărcarea rețelei, exprimată prin gradul de ocupare a canalelor radio, variază între 10% și 95%, iar toți cei 10 utilizatori au ponderi egale (au ponderea $W=1$). Aceste simulări corespund setului 1 A descris în secțiunea 4.2.1 și vor fi descrise în 4.3.1.
2. încărcarea rețelei este mare, fiind întotdeauna de 90%, iar utilizatorii au ponderi variabile. Se consideră două categorii de utilizatori, LP, cu pondere 1, și HP, cu ponderea având valorile 2, 4, 8 și 10. Aceste simulări corespund setului 2 de experimente (a se vedea secțiunea 4.2.2) și vor fi prezentate în subsecțiunea 4.3.2.

În subsecțiunea 4.3.3 se vor interpreta rezultatele obținute prin simulare și se va trage o concluzie importantă privind performanțele algoritmilor bazați pe cantitatea

de date din cozile utilizatorilor.

4.3.1 Comparație între algoritmi pentru ponderi egale ale utilizatorilor

Figura 4.3 prezintă valorile medii ale duratei transmisiei (*sending delay*) în funcție de încărcarea rețelei pentru algoritmi studiați în cazul când fiecare utilizator generează 3 fișiere pe pagină cu lungimea uniform distribuită în intervalul [900, 1100] octeți, iar rata de generare are o distribuție triunghiulară (primul set de experimente). Se observă că algoritmi WRR și OQ au cele mai scăzute valori pentru *sending delay*, urmați de LQ, iar TQL și TFL au cele mai slabe rezultate. De asemenea, valorile pentru WRR și OQ sunt aproape identice, cele două curbe fiind aproape imposibil de distins.

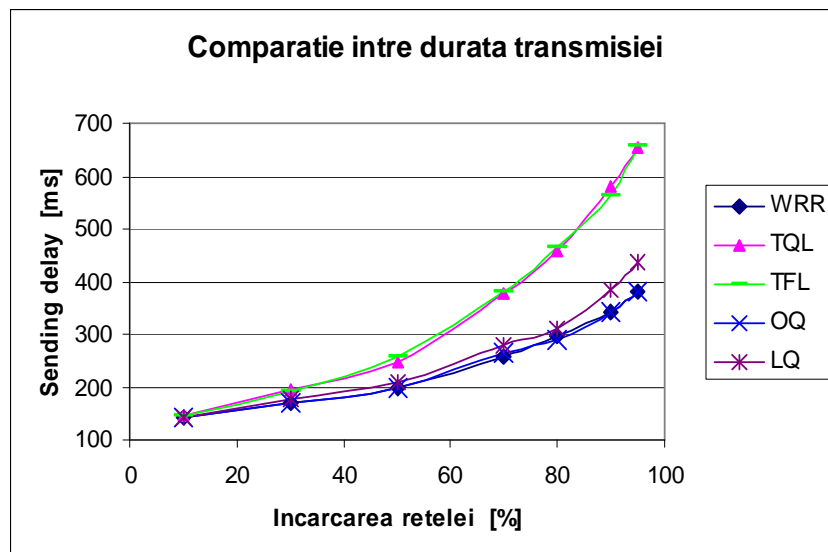


Figura 4.3: Valorile medii pentru *sending delay*, distribuție triunghiulară a intervalelor de generare a datelor, 3 fișiere pe pagină

Figura 4.4 prezintă valorile medii ale întârzierii totale (*total delay*) pentru aceeași situație (aceleași condiții de simulare) și concluziile sunt evident aceleași: WRR și OQ au cele mai bune performanțe, urmați de LQ, pe când performanțele algoritmilor TQL și TFL sunt foarte slabe.

Figura 4.5 prezintă valorile medii ale întârzierii totale atunci când utilizatorii generează trei fișiere pe pagină, cu lungimi ca în figurile anterioare, dar perioada de generare a datelor are o distribuție exponențială. Se observă în primul rând că valorile medii ale

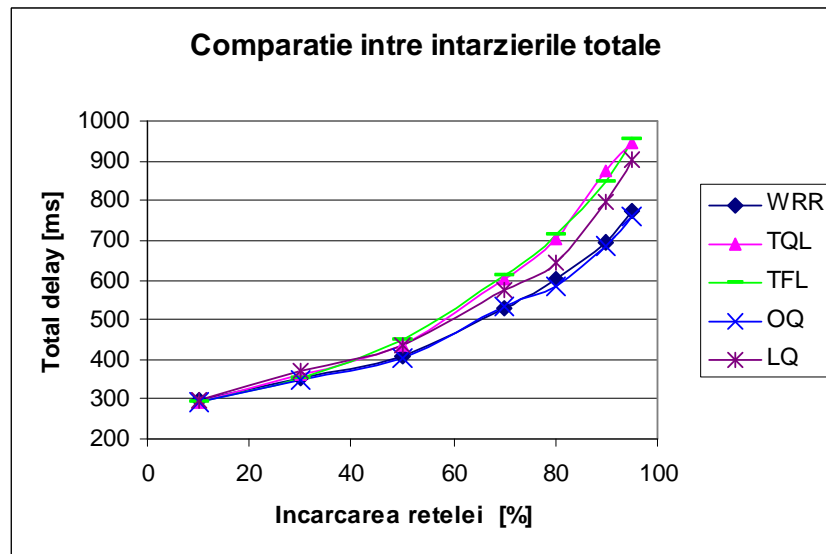


Figura 4.4: Valorile medii pentru *total delay*, distribuție triunghiulară a intervalelor de generare a datelor, 3 fișiere pe pagină

întârzierilor sunt mult mai mari decât în cazul distribuției triunghiulare, mai ales când încărcarea rețelei depășește 90%, datorită caracterului mult mai neregulat (*bursty*) al procesului de generare a datelor. Pentru claritate, în figură s-au reprezentat doar încărcări ale rețelei mai mari de 70%. Pentru încărcări foarte mari performanțele algoritmului OQ se înrăutățesc sensibil, iar performanțele medii ale algoritmului LQ sunt comparabile cu ale algoritmului WRR.

În acest caz se impune o analiză mai atentă a rezultatelor. În primul rând, la un grad de ocupare a canalelor de 95% și o distribuție exponențială a intervalelor de generare a datelor, rețeaua se află foarte aproape de limita de instabilitate, adică atunci când încărcarea este mai mare de 100% și când rețeaua nu poate procesa toate datele generate de utilizatori. Într-o astfel de situație cozile utilizatorilor cresc mereu și, dacă se consideră că ele au o capacitate infinită și deci nu se elimină fișiere (așa cum este cazul în aceste simulări), atunci rezultatele simulării nu sunt concludente, deoarece întârzierile fișierelor depind de lungimea simulării, ele crescând cu cât simularea este mai lungă.

Un indicator foarte util al gradului de congestie al rețelei se obține dacă urmărim numărul de fișiere stocate în bufferele de fișiere ale utilizatorilor. În cazul distribuției triunghiulare, acesta nu va depăși niciodată 3 fișiere, pe când în cazul distribuției exponențiale se ajunge până la valori de ordinul (50 - 60) de fișiere, atunci când se generează 3 fișiere pe pagină, ceea ce explică valorile medii extrem de mari ale

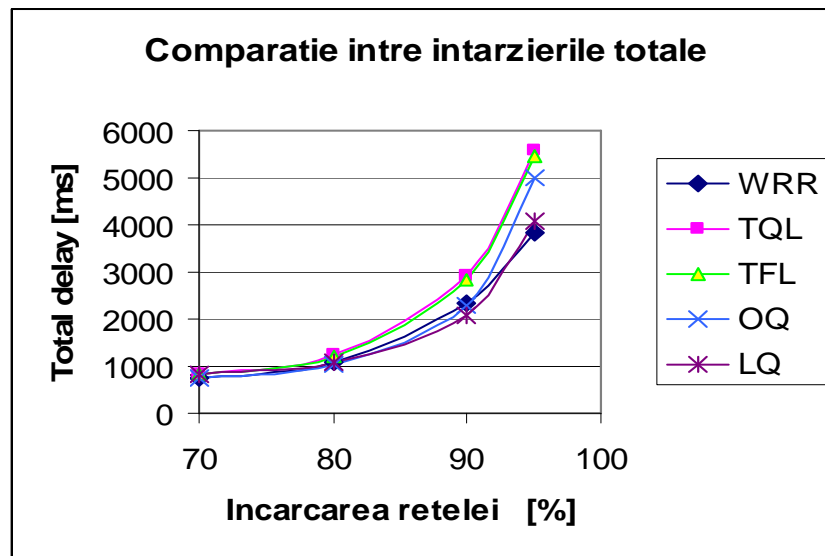


Figura 4.5: Valorile medii ale *total delay*, distribuție exponențială a intervalelor de generare a datelor, 3 fișiere pe pagină

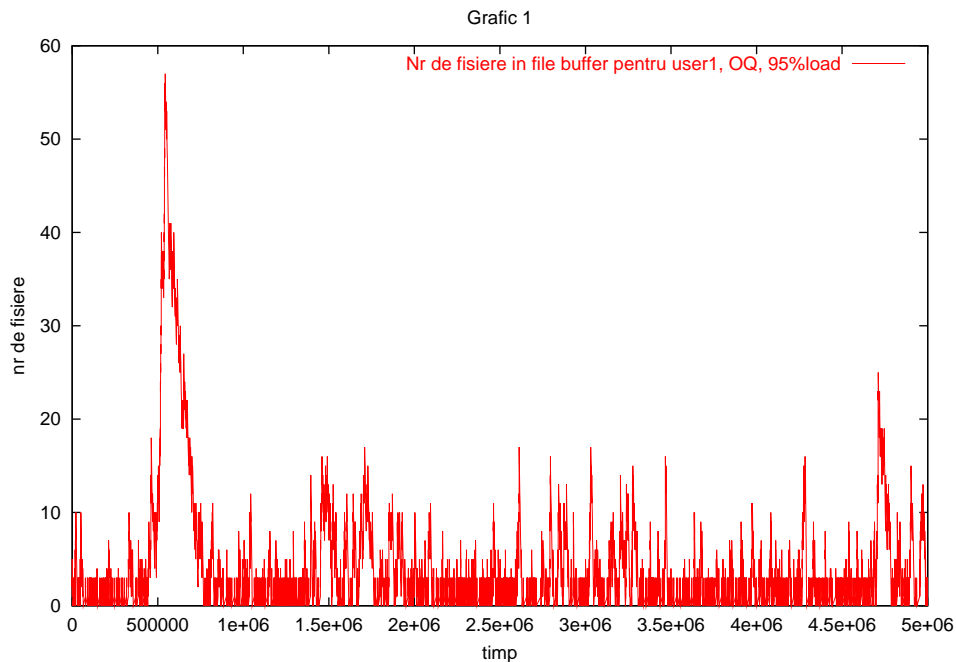


Figura 4.6: Evoluția în timp a numărului de fișiere din bufferul de fișiere al utilizatorului $user_1$, distribuție exponențială, 3 fișiere, încărcare 95%, algoritm OQ

întârzierii totale a fișierelor, valoare influențată decisiv de timpii de așteptare extrem de mari. Totuși, se constată că până la urmă cozile fișierelor scad până la zero sau

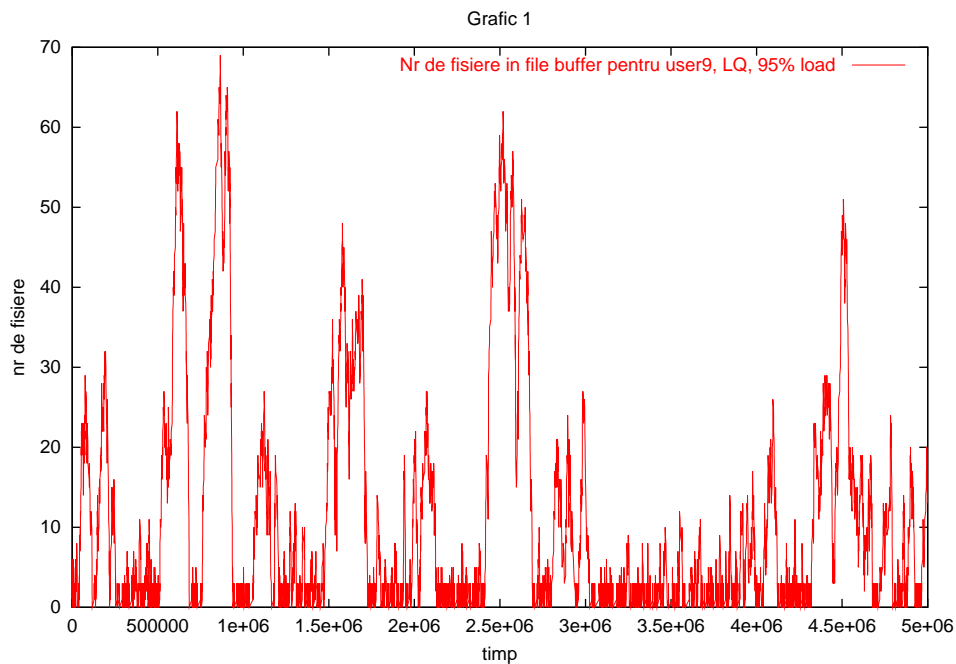


Figura 4.7: Evoluția în timp a numărului de fișiere din bufferul de fișiere al utilizatorului $user_9$, distribuție exponențială, 3 fișiere, încărcare 95%, algoritm LQ

la valori mici (în jur de 3 fișiere), chiar și pentru o încărcare medie a rețelei de 95%. Acest lucru este valabil pentru toți algoritmii, cu excepția algoritmului LQ, la care cozile utilizatorilor cu număr mare de ordine ($user_9$ în acest caz) par să crească în timp, atingând sau fiind aproape de situația de instabilitate descrisă mai înainte, pe când utilizatorii cu număr mic de ordine obțin timpi extrem de buni, care coboară media între cei 10 utilizatori, obținând astfel performanțe aparent foarte bune.

Explicația acestui comportament al algoritmului LQ va fi dată în secțiunea 4.3.3, dar merită reținut faptul că acest comportament face ca utilizarea algoritmului LQ să devină inacceptabilă pentru încărcări mari ale rețelei, în ciuda performanțelor aparent extrem de bune.

Figura 4.6 prezintă evoluția în timp a numărului de fișiere din bufferul de fișiere (*file buffer*) al unui utilizator pentru algoritmul OQ, iar figura 4.7 prezintă evoluția numărului de fișiere pentru utilizatorul cu cea mai mare valoare a indicelui ($user_9$) pentru algoritmul LQ. În ambele situații încărcarea rețelei este de 95%. În figura 4.6 se observă că numărul de fișiere din bufferul de fișiere poate ajunge până aproape de 60, dar numai foarte rar, în mod normal nefiind mai mare de 20, iar valoarea cea mai frecventă a numărului de fișiere din *file buffer* este 3. Din figură se poate observa de asemenea că acest număr rămâne în general stabil, ceea ce denotă faptul că rețeaua

reuşeşte să transfere toate datele, așa cum era de așteptat.

Nu la fel stau lucrurile în figura 4.7, unde numărul de fișiere din bufferul de fișiere al utilizatorului *user₉* depășește în mod frecvent valoarea 20, ajungând chiar până aproape de 70, dar cel mai important este că se observă o oarecare tendință de creștere a numărului de fișiere din buffere spre sfârșitul simulării.

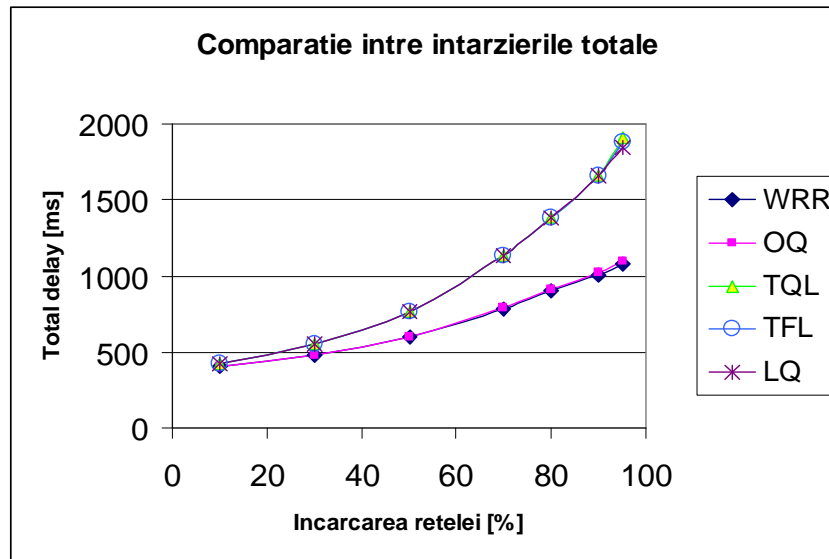


Figura 4.8: Valorile medii ale *total delay*, pentru o distribuție triunghiulară a intervalului de generare a datelor, 1 fișier pe pagină

Figurile 4.8 și 4.9 arată valorile medii ale întârzierii totale în funcție de încărcarea rețelei pentru algoritmi studiați atunci când fiecare pagină generată de utilizatori conține un singur fișier. Figura 4.8 corespunde unei distribuții triunghiulare a intervalului de generare a datelor, iar figura 4.9 unei distribuții exponențiale. În figura 4.8 s-au reprezentat valorile medii ale întârzierilor pentru o încărcare a rețelei cuprinsă între 10% și 95%, iar în figura 4.9, pentru o încărcare a rețelei cuprinsă între 10% și 90%.

Așa cum era de așteptat, în cazul distribuției triunghiulare algoritmul LQ are rezultate practic identice cu TFL și TQL, deoarece în acest caz în buferele de fișiere ale utilizatorilor nu se acumulează mai multe fișiere și deci TQL și LQ funcționează identic. În cazul distribuției exponențiale, pe anumite perioade se pot acumula mai multe fișiere în *file buffer* la utilizatori, și deci există diferențe între LQ și TQL. În ambele situații WRR și OQ au performanțe mult mai bune decât LQ, TQL și TFL.

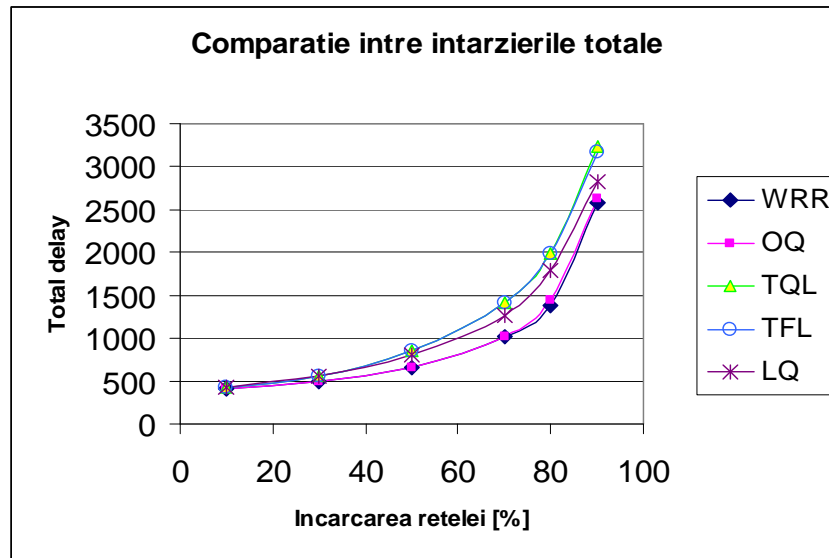


Figura 4.9: Valorile medii ale *total delay*, pentru o distribuție exponențială a intervalelor de generare a datelor, 1 fișier pe pagină

4.3.2 Comparație între algoritmi pentru ponderi diferite ale utilizatorilor

Figurile 4.10 la 4.13 se referă la al doilea set de experimente, când încărcarea medie a rețelei este de 90%, există doar două categorii de utilizatori (de prioritate mare - HP, și de prioritate scăzută, respectiv LP), ponderea utilizatorilor LP este egală cu 1, iar ponderea utilizatorilor HP, W_{HP} are valorile 2, 4, 8 și respectiv 10. Intervalul mediu de generare a datelor este 1000 ms (o secundă), pe abscisă este reprezentat numărul de utilizatori HP, care variază între 1 și 10, iar pe ordonată apare valoarea medie a întârzierilor pentru utilizatorii cu prioritate mare. Deși s-au efectuat simulări atât pentru distribuția triunghiulară, cât și pentru cea exponențială a intervalelor de generare a datelor, aici se vor prezenta doar rezultatele pentru distribuția exponențială, cazul distribuției triunghiulare fiind similar, doar cu valori mai mici ale întârzierilor totale.

În toate figurile se observă că rezultatele algoritmilor WRR și OQ sunt foarte apropiate și mult mai bune decât în cazul TQL. Algoritmul LQ are uneori rezultate medii mai bune decât WRR și OQ, dar acest lucru se datorează faptului că el avantajează utilizatorii care au un indice mic. Cum în aceste experimente utilizatorii HP au întotdeauna indice mai mic decât cei LP, se explică în acest fel rezultatele aparent mai bune ale algoritmului LQ.

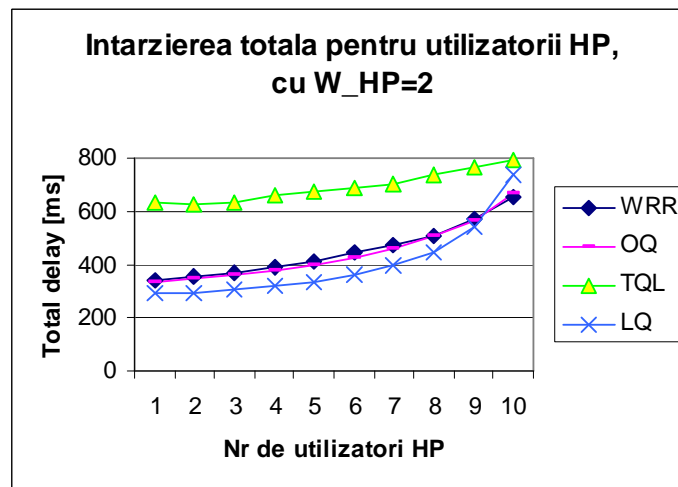


Figura 4.10: Valorile medii pentru *total delay*, distribuție exponențială, 1 fișier pe pagină, $W_{HP}=2$, $W_{LP}=1$, setul 2 de simulări

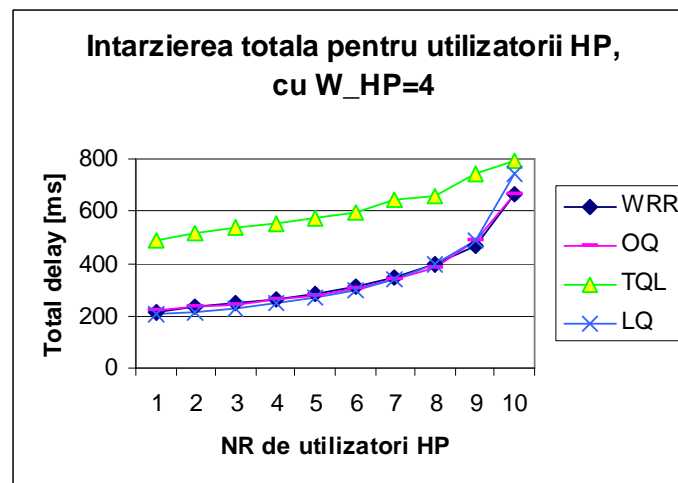


Figura 4.11: Valorile medii pentru *total delay*, distribuție exponențială, 1 fișier pe pagină, $W_{HP}=4$, $W_{LP}=1$, setul 2 de simulări

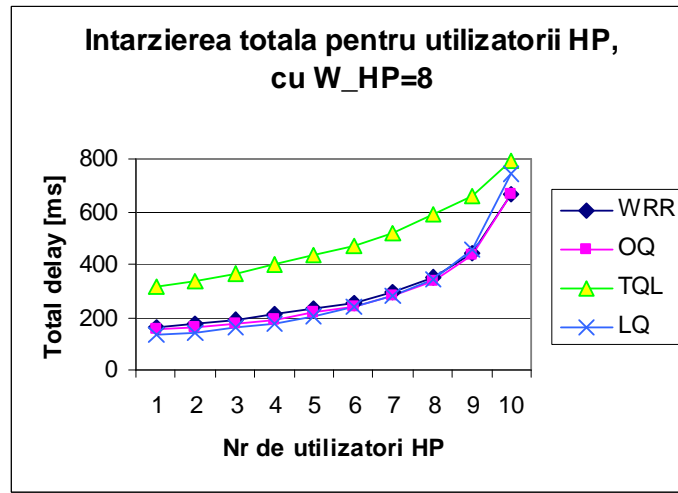


Figura 4.12: Valorile medii pentru *total delay*, distribuție exponențială, 1 fișier pe pagină, $W_{HP}=8$, $W_{LP}=1$, setul 2 de simulări

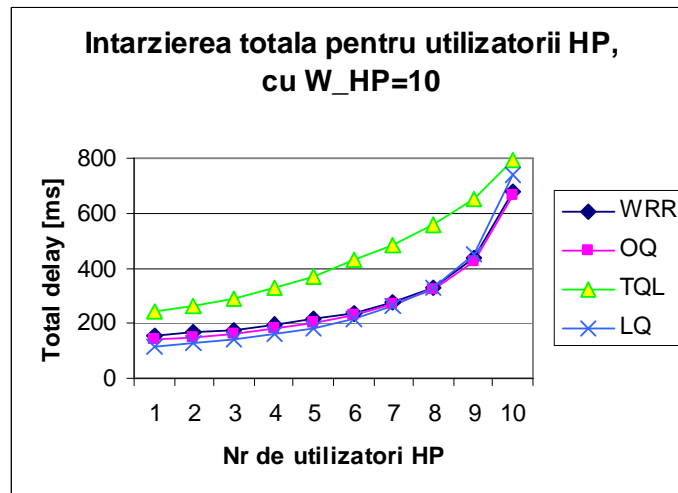


Figura 4.13: Valorile medii pentru *total delay*, distribuție exponențială, 1 fișier pe pagină, $W_{HP}=10$, $W_{LP}=1$, setul 2 de simulări

În secțiunea 4.4 vom folosi aceste rezultate pentru studiul influenței valorilor ponderii utilizatorilor cu prioritate ridicată.

Se poate trage concluzia că cele mai bune performanțe le au algoritmi WRR și OQ, ceilalți algoritmi nefiind suficient de eficienți pentru a fi utilizați la alocarea resurselor rețelei pentru controlul transmiterii datelor.

4.3.3 Un rezultat important

În mod intuitiv, utilizatorii care au o cantitate mare de date în cozi ar trebui avantajați în timpul procesului de alocare a resurselor, ceea ce înseamnă că algoritmi de alocare a resurselor bazați pe cantitatea de date din cozile utilizatorilor, adică LQ, TFL și TQL, ar trebui să fie eficienți.

Astfel de algoritmi pot avea performanțe foarte bune în alte tipuri de rețele, de exemplu în situațiile în care obiectivul principal al algoritmilor de planificare a resurselor ar fi evitarea eliminării datelor datorită umplerii bufferelor de date, așa cum se întâmplă la rețelele ATM, dar rezultatele din această lucrare arată că algoritmi bazați pe cantitatea de date din cozile utilizatorilor, cum sunt LQ, TQL și TFL, nu sunt eficienți pentru controlul transmiterii datelor (*transmission control*) în rețelele celulare de date. Pentru a explica acest lucru vom prezenta o situație în care performanțele acestor algoritmi sunt foarte slabe.

Este important de remarcat faptul că situația pe care o prezentăm nu este una extremă, a cărei probabilitate de apariție e foarte scăzută, ci are loc aproape de fiecare dată când un nou utilizator intră în sistem și începe să transfere date.

Să considerăm următorul scenariu de simulare, reprezentat în figura 4.14: un număr N de utilizatori, notați $user_0$ la $user_{N-1}$, transmit aceeași cantitate de date. Numărul de canale disponibile îl considerăm constant, $B = 8$. Utilizatorii transmit date în mod periodic, cu aceeași perioadă T , dar ei încep să transmită la momente de timp diferite, astfel încât numărul de utilizatori care transmit date la un moment dat variază de la 1 la N . Momentul la care un utilizator începe să transmită (devine activ) este ales astfel încât $user_i$ începe să transfere date la momentul $t_i = i \cdot t$, unde t satisface relația $N \cdot t < T$. Alegem o valoare pentru t astfel încât atunci când utilizatorul $i + 1$, notat $user_{i+1}$ începe să transmită să aibă loc înainte ca utilizatorul i să fi terminat de transmis, adică utilizatorul i mai are în coadă un număr n de blocuri de date.

Așa cum se observă în figura 4.14, la momentul t_0 utilizatorul notat $user_0$ începe să transmită. La momentul $t_1 = t_0 + t$, când următorul utilizator, $user_1$, dorește să transmită date, primul utilizator mai are în bufferul de date n blocuri de date.

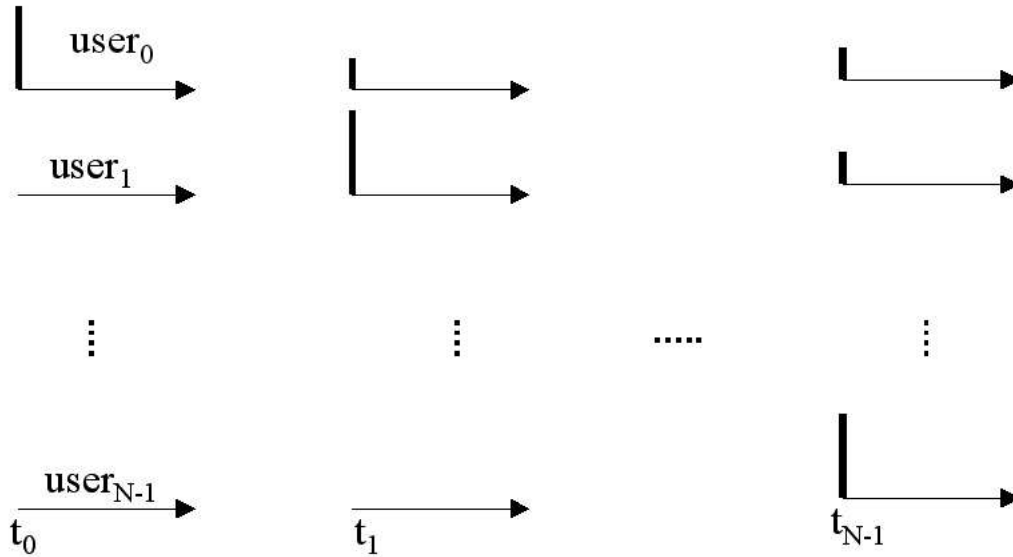


Figura 4.14: Scenariu de simulare, cu utilizatorii generând date la momentele t_0, t_1, \dots, t_{N-1}

În figura 4.14 cantitatea de date a fiecărui utilizator este reprezentată printr-o bară verticală proporțională cu cantitatea de date. În continuare doar $user_1$ va transfera date, deoarece cantitatea sa de date este mai mare decât a utilizatorului notat $user_0$.

Situația se repetă atunci când $user_2$ dorește să transmită și așa mai departe, până când și ultimul utilizator, notat $user_{N-1}$ ajunge să devină activ la momentul t_{N-1} .

Aceasta înseamnă că $user_0$ va putea din nou să transmită date doar după ce ultimul utilizator ($user_{N-1}$) va avea mai puțin de n blocuri de date în coadă. Deci primul utilizator trebuie să aștepte un interval mai mare decât $(N-1) \cdot t$ până când va putea din nou să transfere date, lucru ce explică performanțele slabe ale acestor algoritmi. Atunci când un utilizator generează mai multe fișiere de fiecare dată, intervalul de așteptare va fi mai mare în cazul algoritmilor TQL și TFL decât pentru LQ, deoarece și cantitatea de date este mai mare. La TFL și TQL $user_0$ trebuie să aștepte până când $user_{N-1}$ transmite toate fișierele, mai puțin ultimul, plus timpul cât transmite din ultimul fișier până când rămân cel mult n blocuri în coadă, pe când la LQ contează doar fișierul aflat în bufferul de date.

Pe de altă parte algoritmul LQ poate prezenta alte probleme: în situația în care toți utilizatorii generează fișiere de aceeași lungime, dacă mai mulți utilizatori generează date simultan (sau mai exact, dacă fișierele intră în bufferul de date în același moment de timp), atunci utilizatorii vor avea cozi egale. Presupunând că ei

au aceeași schemă de codare și aceeași prioritate, înseamnă că va fi servit primul utilizatorul cu indicele cel mai mic.

Dacă acesta reușește să își transfere următorul fișier în bufferul de date înainte ca ceilalți să termine de transferat datele din bufferul de date sau dacă reușește să transfere în bufferul de date următorul fișier înainte de începerea următorului ciclu al controllerului, înseamnă că el va avea coada cea mai lungă și deci va fi servit din nou, în timp ce ceilalți trebuie să aștepte. În acest fel utilizatorii cu un indice mai mare riscă să acumuleze întârzieri foarte mari. În situația când rețeaua este foarte încărcată, este foarte probabil ca utilizatorii să se “sincronizeze” și să apară situația descrisă mai înainte. De altfel, în simulările din setul 1 A, la o încărcare de 95% a canalelor, apare singura situație când un utilizator (cel de-al 9-lea) riscă să devină instabil, adică să nu reușească să transfere toate datele pe care le-a generat și coada lui de fișiere să crească mereu.

Rezultatul privind ineficiența algoritmilor bazați pe cantitatea de date din cozile utilizatorilor pentru problema controlului transmiterii datelor în rețelele radio de date este important, deoarece există autori (de exemplu [11], [64]) care folosesc algoritmi de acest tip pentru controlul transmiterii datelor.

4.4 Studiul influenței factorilor de pondere asupra performanțelor algoritmilor

În această secțiune se va studia influența pe care o au valorile ponderilor utilizatorilor asupra întârzierilor fișierelor utilizatorilor la controlul transmiterii datelor. Pentru acest lucru ne vom concentra asupra algoritmului WRR, deoarece este cel mai potrivit candidat pentru implementarea transferului datelor, așa cum a rezultat din studiul realizat în secțiunea 4.3. Menționăm că și în cazul algoritmului OQ se obțin rezultate apropiate cu cele obținute pentru WRR.

Alocând ponderi diferite utilizatorilor în funcție de clasa de QoS (calitate a serviciului) din care aceștia fac parte se poate obține diferențierea calității serviciului utilizatorilor pe baza ponderilor lor, așa cum se arată în subsecțiunea 4.4.1. Dacă asociem ponderea utilizatorilor cu schema lor de codare, care reflectă calitatea legăturii radio a utilizatorilor, atunci putem utiliza ponderi diferite pentru a favoriza utilizatorii care au o legătură radio mai bună, lucru ce poate fi folosit la reducerea congestiilor în rețea și va fi prezentat în subsecțiunea 4.4.2.

4.4.1 Utilizarea ponderilor pentru realizarea diferențierii calității serviciilor utilizatorilor

Figurile 4.15, 4.16 și 4.17 prezintă întârzierile totale medii pentru cele trei clase de utilizatori în condițiile din primul set de experimente, subsetul B, utilizând distribuție triunghiulară pentru intervalele la care utilizatorii generează date, aceste date constând într-un singur fișier cu lungime uniform distribuită în intervalul [2700, 3300] octeți. Ponderile celor 3 clase de utilizatori sunt $\{W_{HP}, W_{MP}, W_{LP}\}=\{3,2,1\}$ în figura 4.15, $\{8,4,1\}$ în figura 4.16 și $\{8,2,1\}$ în figura 4.17. În toate cele trei figuri au fost reprezentate și valorile medii ale întârzierii totale atunci când nu se face diferențiere între utilizatori, situație notată pe grafice cu w111 (toți utilizatorii au pondere 1).

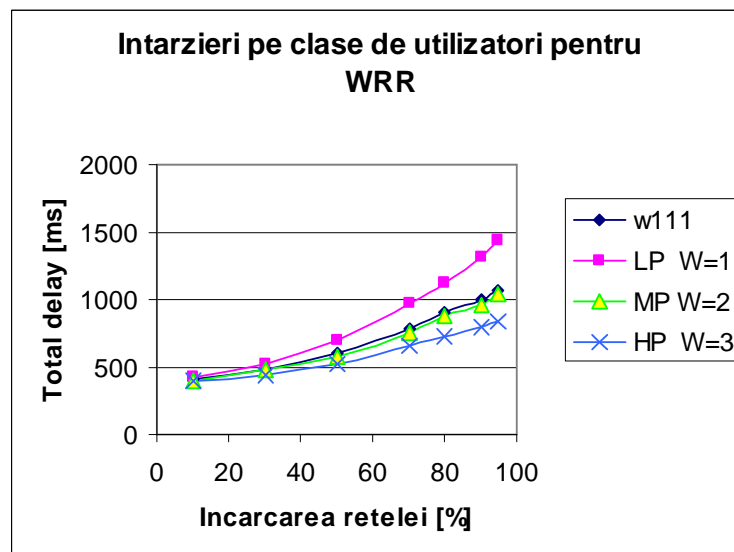


Figura 4.15: Comparația valorilor medii ale întârzierilor totale pentru clase diferite de utilizatori, algoritm WRR și ponderi 3, 2, 1

În toate cele trei cazuri se observă o diferențiere foarte clară între cele 3 categorii de utilizatori, diferențierea fiind cu atât mai pronunțată cu cât încărcarea rețelei este mai mare. Acest lucru este important, deoarece atunci când rețeaua este foarte încărcată se dorește ca utilizatorii cu prioritate ridicată să fie protejați și, pe cât posibil, să li se garanteze o anumită calitate a serviciului. Așa cum era de așteptat, în toate aceste cazuri întârzierile utilizatorilor LP sunt mai mari decât în cazul în care toți utilizatorii au pondere 1, iar cele ale utilizatorilor HP sunt întotdeauna considerabil mai mici decât în cazul când nu se face diferențiere între utilizatori (toți au pondere 1). Astfel, utilizatorii HP sunt avantajați în comparație cu celelalte categorii de

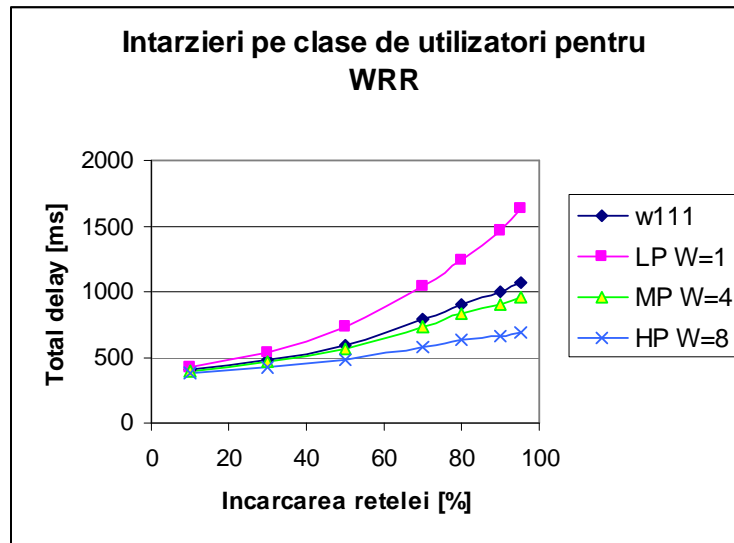


Figura 4.16: Comparația valorilor medii ale întârzierilor totale pentru clase diferite de utilizatori, algoritm WRR și ponderi 8, 4, 1

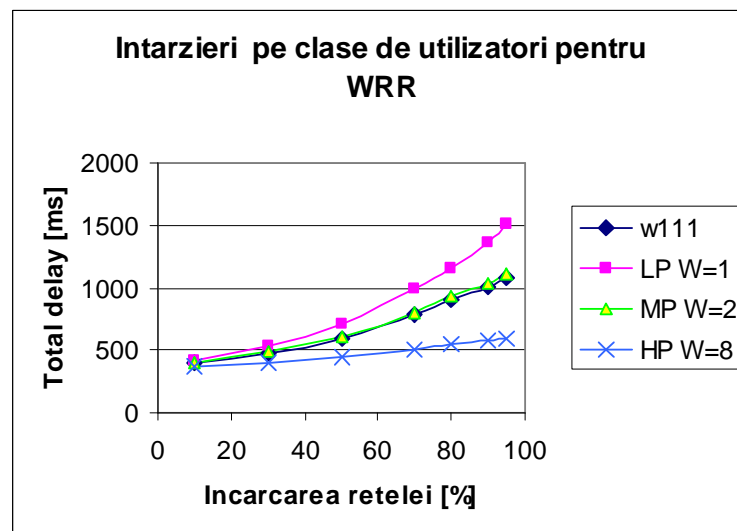


Figura 4.17: Comparația valorilor medii ale întârzierilor totale pentru clase diferite de utilizatori, algoritm WRR și ponderi 8, 2, 1

utilizatori, iar până la urmă avantajele pe care aceștia le primesc se vor repercuta asupra întârzierilor utilizatorilor LP.

Utilizatorii MP au aproximativ aceleași întârzieri ca și în situația în care toți utilizatorii ar avea ponderi egale, fiind puțin mai mici în situațiile când ponderile celor

trei clase de utilizatori sunt $\{3,2,1\}$, și respectiv $\{8,4,1\}$, și puțin mai mari în situația ponderilor $\{8,2,1\}$. Astfel, în funcție de situația de moment (de exemplu, în funcție de încărcarea rețelei), operatorul de rețea poate avantaja și mai mult utilizatorii HP, eventual chiar afectând puțin calitatea serviciilor oferite utilizatorilor MP. De exemplu, în cazul din figura 4.17 – dacă încărcarea rețelei ar fi foarte mare operatorul de rețea va încerca să garanteze performanțele utilizatorilor HP chiar cu prețul unui mic sacrificiu de calitate pentru utilizatorii MP.

Dar este foarte important de remarcat că, pentru această distribuție a numărului de utilizatori (jumătate dintre utilizatori sunt MP, iar din cei rămași, aproximativ jumătate sunt HP și ceilalți sunt LP), se poate oferi o calitate mai bună utilizatorilor HP prin modificarea ponderilor acestora, lucru care nu afectează în mod semnificativ performanțele utilizatorilor MP, ci doar ale utilizatorilor LP. Dacă se asociază utilizatorii HP celor din clasa *premium*, care au plătit mai mult pentru servicii mai bune, utilizatorii de tip MP celor din categoria *standard*, iar utilizatorii LP sunt cei din clasa *economic*, atunci se observă foarte clar cum se poate realiza *diferențierea calității serviciilor pe baza subscripției utilizatorilor*, diferențiere care se realizează foarte simplu, doar prin valorile ponderilor utilizatorilor. Acest mod de a realiza diferențierea utilizatorilor pe baza clasei lor de QoS a fost publicat de autorul tezei în [110].

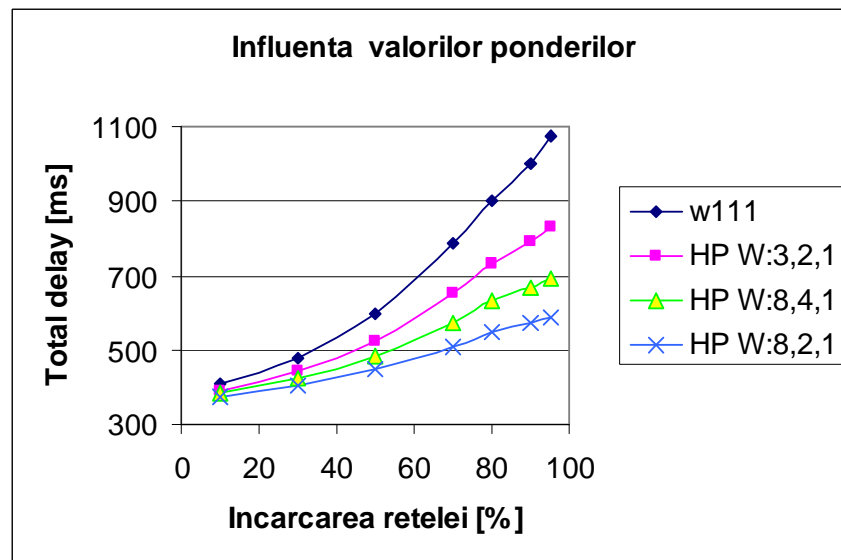


Figura 4.18: Influența valorilor ponderilor asupra întârzierii totale pentru WRR, distribuție triunghiulară, setul 1 B de simulări

Figura 4.18 prezintă valorile medii ale întârzierii totale ale utilizatorilor HP pentru

valori diferite ale ponderilor celor 3 categorii de utilizatori (seturile de valori $\{1,1,1\}$, $\{3,2,1\}$, $\{8,4,1\}$ și $\{8,2,1\}$), în aceleași condiții experimentale ca și cele din figurile 4.15, 4.16 și 4.17, fiind într-un fel o sinteză a celor 3 situații prezentate în figurile anterioare. Se observă că la o încărcare a rețelei de 95%, valoarea medie a întârzierii totale scade de la 1100ms, cât este în cazul când toți utilizatorii au pondere 1, la ceva mai puțin de 600ms, în situația ponderilor $\{8,2,1\}$.

4.4.2 Utilizarea ponderilor utilizatorilor pentru reducerea congestiilor

Figurile 4.19 și 4.20 ilustrează influența valorilor ponderilor utilizatorilor atunci când există doar două categorii de utilizatori, respectiv HP și LP, în condițiile celui de-al doilea set de experimente. Datele sunt generate la intervale distribuite exponențial în jurul valorii medii de 1000ms, iar un utilizator generează un singur fișier având lungimea de 36 de blocuri de date (cu CS1). Numărul utilizatorilor HP se modifică între 1 și 10, iar ponderea lor are valorile 2, 4, 8 și 10, în timp ce ponderea utilizatorilor LP va fi tot timpul egală cu 1. Rezultatele se prezintă doar pentru WRR, dar cu algoritmul OQ se obțin rezultate foarte apropiate.

În figura 4.19 sunt prezentate valorile medii ale întârzierii totale pentru utilizatorii HP, iar în figura 4.20 se prezintă valorile medii ale întârzierii totale pentru utilizatorii LP. În ambele figuri, pe abscisă se reprezintă numărul de utilizatori HP.

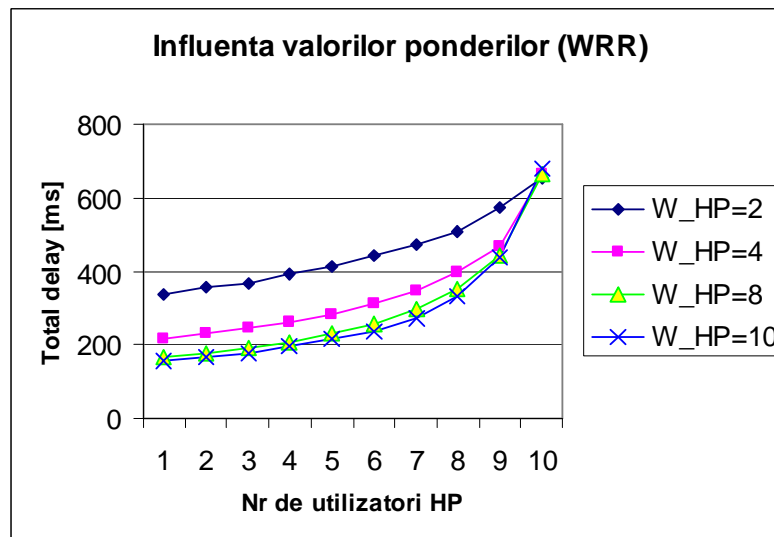


Figura 4.19: Influența valorilor ponderilor asupra întârzierii totale a utilizatorilor HP pentru WRR, distribuție exponențială, setul 2 de simulări

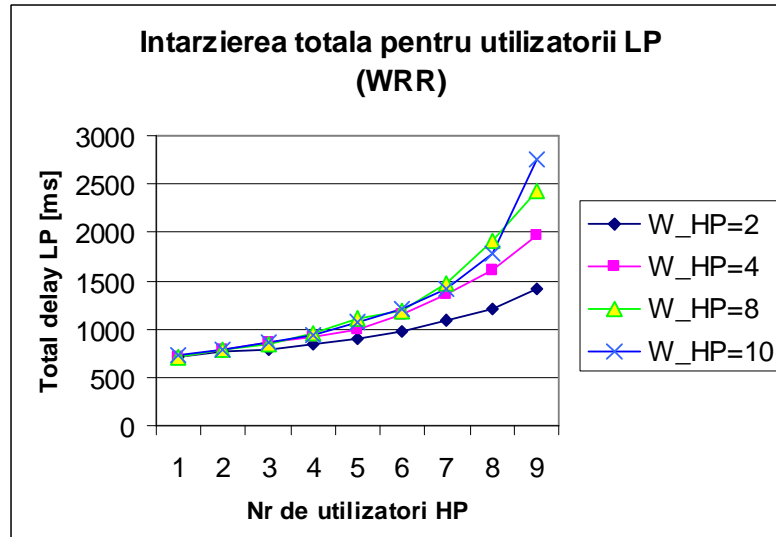


Figura 4.20: Influența valorilor ponderilor asupra întârzierii totale a utilizatorilor LP pentru WRR, distribuție exponențială

Se observă că atât rezultatele utilizatorilor HP, cât și ale celor LP sunt mai bune cu cât numărul utilizatorilor HP este mai scăzut. La creșterea ponderii utilizatorilor HP cresc performanțele acestora și scad performanțele utilizatorilor LP.

Dacă *diferențierea între utilizatori se face în funcție de calitatea legăturii lor radio*, atunci acest lucru poate fi folosit pentru reducerea congestiilor, așa cum am arătat și în [109] și [111]: dacă se consideră, pentru simplitate, doar două scheme de codare pentru utilizatori, CS1 și CS2, atunci utilizatorii cu o legătură radio mai bună vor utiliza schema CS2, iar cei cu o legătură radio mai slabă vor fi obligați să folosească schema CS1, care asigură o codificare mai puternică a datelor. Dacă asociem nivelele de prioritate cu schema de codare, utilizatorii având schema CS2 vor fi în categoria HP, iar cei cu CS1 în categoria LP. Astfel sunt avantajați utilizatorii cu schema de codare CS2 și aceștia vor putea transfera un volum mai mare de date, contribuind la reducerea congestiilor. Mai mult, utilizatorii care au o legătură radio cu probleme sunt în pericol de a avea o rată mare a erorilor, lucru ce va necesita un număr mare de retransmisii. În cazul când rețeaua este încărcată, acest lucru ar duce la o risipă a resurselor radio și deci ar putea favoriza accentuarea congestiilor.

4.5 Gradul de încredere în rezultatele simulărilor

Pentru a asigura un grad ridicat de încredere în rezultatele simulărilor au fost realizate mai multe lucruri:

1. Majoritatea rezultatelor au fost obținute prin utilizarea a 5 seturi diferite de numere aleatoare
2. A fost utilizată facilitatea simulatorului OMNeT++ de a detecta acuratețea simulărilor
3. Pentru valorile medii ale întârzierilor totale ale fișierelor *fiecărui utilizator* au fost determinate *intervalele de încredere* pentru un *grad de încredere de 95%*
4. La procesarea rezultatelor simulărilor, atunci când s-au determinat valorile medii ale timpului de transfer al fișierelor și ale întârzierii totale a fișierelor pentru *o anumită categorie de utilizatori* (utilizatori de prioritate scăzută, de prioritate medie și de precedență ridicată, adică respectiv LP, MP și HP) au fost de asemenea determinate *intervalele de încredere* pentru un *grad de încredere de 95%*.

În cele ce urmează se va detalia fiecare din aceste aspecte.

4.5.1 Seturi diferite de numere aleatoare

Simulatorul OMNeT++, ca de altfel marea majoritate a simuloarelor pentru rețele, utilizează numere pseudoaleatoare, care sunt generate prin metode matematice pornind de la o anumită valoare inițială, numită *seed* în limba engleză, astfel încât, dacă se rulează mai multe simulări, numerele aleatoare generate de către simulator vor fi întotdeauna aceleași. Acest lucru este util deoarece asigură un grad ridicat de determinism și permite de exemplu o mai corectă comparare a performanțelor unor algoritmi, deoarece toți algoritmi rulează în condiții identice. Indiferent de distribuția de probabilitate pe care o formează un set de numere (pseudo)aleatoare, generarea lor pleacă de la o distribuție normală, care la rândul ei este obținută pornind de la o anumită valoare inițială (*seed*).

Simulatorul furnizează un program, numit *seedtool*, care permite specificarea unor valori inițiale (*seed*) diferite și a “distanței” dintre aceste valori inițiale. *Distanța* se definește ca fiind diferența între pozițiile numerelor pseudoaleatoare generate în cadrul șirului respectiv, nu între valorile numerelor.

O distanță suficient de mare permite generarea unor serii de numere aleatoare care nu se suprapun, deci care nu sunt corelate. În cazul nostru s-a estimat că în oricare din seturile de simulări, numărul de numere aleatoare generate pe simulare este mai mic de 55000, deci distanța aleasă între oricare două valori inițiale este de 55000. Această distanță este mai mult decât acoperitoare în cazul nostru, deoarece numărul maxim de cicluri de generare de numere aleatoare se obține pentru setul 2 de simulări, când vor exista 5000 astfel de cicluri (fiecare din cei 10 utilizatori generează date o dată la o secundă, iar durata simulării este de 5000 de secunde) și la fiecare ciclu se generează 10 numere aleatoare (valoarea intervalului de generare a datelor pentru fiecare din cei 10 utilizatori, deci în total 10 valori).

În cazul primului set de simulări, pentru situația când fiecare utilizator generează un singur fișier pe pagină, deci un singur fișier la fiecare interval de generare a datelor, numărul de numere aleatoare generate va fi de cel mult 28662, adică se înmulțește numărul de utilizatori (10) cu numărul de numere aleatoare generate în fiecare interval (două numere, unul pentru lungimea fișierului și unul pentru lungimea intervalului) și cu durata simulării (5000 secunde) și se împarte la intervalul cel mai mic dintre două generări de date (3.489 secunde pentru o încărcare de 95%): $2 \cdot 5000 / 3.489 = 28661.5$.

În situația primului set de simulări, atunci când s-au generat 3 fișiere pe pagină, s-a folosit un singur *seed* pentru fiecare simulare.

4.5.2 Acuratețea simulărilor

A fost utilizată facilitatea simulatorului OMNeT++ de a detecta acuratețea simulărilor. Această facilitate constă în activarea unei funcții care, după un moment de timp specificat, colectează un număr n de valori ale parametrului a cărui acuratețe este urmărită (în cazul nostru acest parametru este întârzierea totală a fișierelor fiecărui utilizator), calculează deviația standard *stddev* a setului constând din cele n valori și apoi compară raportul $stddev/n^2$ cu parametrul *acc* reprezentând acuratețea dorită. Dacă $stddev/n^2 \leq acc$ atunci funcția respectivă returnează valoarea 1, iar dacă inegalitatea nu este satisfăcută, atunci valoarea returnată de funcție este 0.

Dimensiunea setului de valori este $n = 6$ (specificat în codul funcției), valoarea parametrului ce reprezintă acuratețea a fost aleasă $acc = 0.01$, iar momentul de declanșare al funcției a fost ales egal cu momentul de sfârșit al simulării minus $3 \cdot inter_page_delay$ (de trei ori valoarea medie a intervalului de generare a datelor de către utilizatori). Au fost contorizate numărul de valori 1 (acuratețea dorită este respectată) și numărul de valori 0 (când acuratețea dorită nu este respectată) returnate de funcție.

Au existat un număr foarte mic de situații când, pentru unii utilizatori, funcția de detectare a acurateții nu s-a activat din cauză că utilizatorul respectiv nu a reușit să transfere cu succes cel puțin 6 fișiere în intervalul respectiv. Acest lucru este posibil dacă încărcarea rețelei este mare și dacă utilizatorul respectiv are prioritate mică. În marea majoritate a situațiilor funcția de detectare a acurateții s-a activat cel puțin o dată pentru fiecare utilizator.

În primul set de simulări, singurele cazuri când acuratețea nu a fost respectată pentru distribuție exponențială a intervalelor de generare a datelor, au apărut în situația când încărcarea rețelei a fost de 95% și numai pentru utilizatorii de prioritate scăzută (LP), atunci când s-au folosit valori diferite ale ponderilor pentru utilizatori, în cazul algoritmilor WRR și OQ. Explicația pe care o propunem este că acești utilizatori au în această situație întârzieri extrem de mari, dar și fluctuația valorilor întârzierilor este mare (se poate întâmpla ca unele fișiere ale utilizatorilor LP să fie procesate relativ repede în comparație cu celelalte, atunci când pentru scurte perioade de timp, încărcarea rețelei scade datorită faptului că datele din cozile majorității utilizatorilor HP și MP au fost deja transferate). La distribuția triunghiulară a intervalelor de generare a datelor a fost întotdeauna respectată acuratețea dorită.

În setul 2 de simulări nu apare nici o situație în care acuratețea să nu fi fost respectată, doar un număr foarte mic de cazuri în care nu s-a activat funcția de detecție a acurateții.

4.5.3 Intervale de încredere pentru valorile medii ale întârzierilor fiecărui utilizator

Ca o verificare suplimentară față de funcția de detecție a acurateții, în unele simulări au fost determinate și intervalele de încredere pentru întârzierile totale ale fișierelor utilizatorilor.

Conform cu [76] (secțiunea 5.6.6, pagina 150), “dacă x_1, x_2, \dots, x_n ($n \geq 30$) sunt valori de selecție dintr-o distribuție de probabilitate arbitrară de medie m și dispersie σ^2 , atunci un interval de încredere pentru medie, de nivel aproximativ $1 - \alpha$, este:

$$I_m = (\bar{x} - z_{1-\alpha/2}d, \bar{x} + z_{1-\alpha/2}d) \quad (4.1)$$

unde, pentru cazul când dispersia σ este necunoscută, $d = s/\sqrt{n}$, iar s este dispersia selecției (x_1, x_2, \dots, x_n), iar $z_{1-\alpha/2}$ este cvantila distribuției normale $N(0,1)$ corespunzătoare lui $1 - \alpha/2$, cu $0 < \alpha < 1$.

Interpretarea intervalului de încredere în cazul unei populații statistice este ([76]) că, dacă extragem sute de eșantioane din populația în studiu, având media m și

dispersia σ^2 , pentru fiecare eșantion de volum n (x_1, x_2, \dots, x_n), calculând intervalul de încredere conform ecuației (4.1), atunci $(1 - \alpha)$ din aceste intervale vor conține media m .

În cazul nostru populația statistică este înlocuită cu rezultatele simulării, iar pentru un grad de încredere de 95% rezultă că $(1 - \alpha) = 0.95$, deci $\alpha = 0.05$. Valoarea cvantilei $z_{1-\alpha/2}$ este în acest caz 1.96. Nivelul de încredere utilizat a fost de 95%, iar intervalele de încredere le exprimăm prin procente din valoarea medie a întârzierii totale. De exemplu, dacă obținem o valoare medie de 1000ms și limite ale intervalului de încredere de $\pm 6\%$, atunci, cu o probabilitate de 95%, valoarea medie a întârzierii totale se situează în intervalul $[1000 - 0.06 \cdot 1000, 1000 + 0.06 \cdot 1000]$.

Se dorește ca valorile limitelor intervalelor de încredere să fie cât mai mici. În general, niște valori prea mari ale acestor limite semnaleză faptul că probabil durata simulării a fost prea scurtă și deci gradul de încredere în rezultatele simulării este redus.

Pentru primul set de simulări, atunci când se generează 3 fișiere pe pagină, cu o distribuție exponențială a intervalelor de generare a datelor și când toți utilizatorii au aceeași pondere, intervalul de încredere nu variază cu mai mult de 5% din valoarea medie în jurul valorii medii a întârzierii totale a fișierelor fiecărui utilizator, atunci când se utilizează algoritmul WRR. Singurele excepții au loc pentru cazul când încărcarea rețelei este de 10%, când intervalele de încredere sunt de $\pm 8\%$. Pentru distribuția triunghiulară a valorilor intervalelor de generare a datelor, intervalele de încredere sunt mai mici decât la distribuția exponențială. Facem observația că valorile mai mari ale intervalelor de încredere pentru o încărcare foarte scăzută a rețelei sunt probabil datorate numărului mai mic de eșantioane pentru care au fost determinate. Oricum, valorile întârzierilor la o încărcare atât de scăzută a rețelei nu sunt nici pe departe la fel de importante ca și în cazul unor încărcări mai mari ale rețelei.

Tot în cazul primului set de simulări și tot în situația când toți utilizatorii au aceeași pondere, dar pentru cazul când se generează un singur fișier pe pagină, fluctuația intervalelor de încredere pentru o distribuție exponențială a intervalelor de generare a datelor este de până la 12% pentru o încărcare a rețelei de 10%, în jur de 5% și nu mai mare de 6.5% pentru încărcări între 30% și 80% inclusiv, în jur de 6% și mai mică de 8% pentru o încărcare de 90% sau 95%. Pentru distribuție triunghiulară a valorilor intervalelor de generare a datelor, valorile limitelor intervalului de încredere sunt similare cu cele de la distribuția exponențială doar pentru o încărcare de 10% a rețelei, fiind mai mici cu cel puțin 1-2% pentru celelalte valori ale încărcării rețelei. Aceste intervale sunt valabile pentru algoritmi WRR și OQ și sunt doar puțin mai mari pentru ceilalți algoritmi (în jur de 6-7%), dar nu depășesc totuși

8% pentru încărcări mai mari de 10%.

Pentru cazul când, în cadrul primului set de simulări, ponderile utilizatorilor sunt diferite, valorile limitelor intervalelor de încredere pentru utilizatorii HP sunt de asemenea în jur de 5-6%.

Pentru setul al doilea de simulări, în condițiile în care toți utilizatorii au aceeași pondere, valorile limitelor intervalelor de încredere sunt situate pentru fiecare utilizator sub 3% din valoarea medie a întârzierii totale a fișierelor sale.

Considerăm că valorile obținute pentru limitele intervalelor de încredere sunt suficient de mici, indicând deci un grad ridicat de încredere în rezultatele simulărilor.

4.5.4 Intervale de încredere pentru valorile medii ale întârzierilor pentru fiecare clasă de utilizatori

Formula (4.1) este valabilă doar pentru un volum al datelor $n \geq 30$, condiție care nu este întotdeauna îndeplinită în acest caz. Deoarece acum seturile de valori pentru care se vor determina intervalele de încredere sunt rezultatele unor măsurători, se poate considera că aceste valori corespund unei distribuții normale și deci pentru calculul intervalului de încredere se poate folosi propoziția 5.6.5 din [76]:

“fie (x_1, x_2, \dots, x_n) o realizare a selecției aleatoare de volum n , (X_1, X_2, \dots, X_n) , $X_i \sim N(m, \sigma^2)$, $\forall i = \overline{1, n}$, unde media m și dispersia σ^2 sunt necunoscute. Atunci un interval de încredere pentru media m cu nivelul de încredere $(1 - \alpha)$ este:

$$I_m = (\bar{x} - t_{1-\alpha/2}d, \bar{x} + t_{1-\alpha/2}d) \quad (4.2)$$

unde \bar{x} este media selecției, $d = s/\sqrt{n}$, s^2 este dispersia selecției, iar $t_{1-\alpha/2}$ este cvantila distribuției Student $t(n - 1)$ corespunzătoare lui $(1 - \alpha/2)$ ”.

Și în această secțiune valoarea gradului de încredere a fost de 95%, iar valorile cvantilelor distribuției student $t(n - 1)$ pentru acest grad de încredere le-am obținut utilizând utilitarul Octave din distribuția Linux.

Pentru primul set de experimente, pentru toți algoritmi cu excepția algoritmului LQ, dacă toți utilizatorii au aceeași pondere și dacă intervalele de generare a datelor au distribuție exponențială, atunci valorile limitelor intervalelor de încredere pentru întârzierea totală a utilizatorilor sunt mai mici de 2% din valoarea mediei dacă încărcarea rețelei nu depășește 80% și sunt de aproximativ 5% pentru cazul când încărcarea rețelei este de 90% sau de 95%. S-a considerat un grad de încredere de 95% și cazul când utilizatorii generează un singur fișier pe pagină. În cazul tuturor algoritmilor, inclusiv LQ, valorile limitelor intervalelor de încredere sunt mult mai mici (mai mici de 2% indiferent de încărcarea rețelei) dacă intervalele de generare a datelor au o distribuție triunghiulară.

Valorile pentru LQ sunt mult mai mari (de ordinul 25%) dacă încărcarea rețelei este de 90% sau 95% și dacă distribuția valorilor intervalelor de generare a datelor este exponențială. Pentru distribuție triunghiulară toate valorile sunt mai mici de 2%.

Dacă utilizatorii generează tot un singur fișier, dar au ponderi diferite (setul 1B de simulări), au fost determinate limitele intervalului de încredere pentru algoritmi WRR și OQ. Pentru utilizatorii HP, aceste limite nu depășesc 2.5%, indiferent de încărcarea rețelei și de tipul distribuției (exponențiale sau triunghiulare), cu excepția situației când ponderea utilizatorilor HP este doar $W_{HP} = 3$, situație în care limitele intervalelor pot crește până la 4% pentru distribuția exponențială.

Pentru utilizatorii MP limitele intervalelor de încredere nu depășesc 5%. Pentru utilizatorii LP, aceste limite sunt mai mari, mai ales în cazul distribuției exponențiale, când pot atinge chiar valori în jur de 20% pentru o încărcare a rețelei de 95%, fiind în jur de 10-15% pentru o încărcare de 90% și sub 10% pentru încărcări mai mici ale rețelei. Pentru distribuția triunghiulară valorile sunt sub 6%.

Pentru al doilea set de experimente, atunci când ponderile utilizatorilor sunt egale, limitele intervalului de încredere sunt mai mici de 1.5% pentru toți algoritmi cu excepția algoritmului LQ, indiferent de distribuția intervalelor de generare a datelor, iar pentru algoritmul LQ limitele sunt de aproximativ 25% pentru distribuție exponențială și de doar 3% pentru distribuție triunghiulară. Valoarea de 25% este foarte mare, dar acest lucru se datorează anomaliilor acestui algoritm, care avantajează utilizatorii cu un indice mic, în defavoarea celor cu indice mare, ducând la diferențe extrem de mari ale valorilor întârzierilor în funcție de indicele utilizatorului (valori foarte mici pentru $user_0$ și foarte mari pentru $user_9$). Pentru mai multe detalii se poate consulta secțiunea 4.3.3. Restul valorilor limitelor intervalelor de încredere sunt foarte bune, fiind o dovadă a corectitudinii rezultatelor simulărilor.

Pentru algoritmi OQ și WRR, aceste valori ale limitelor intervalelor de încredere sunt valabile și pentru utilizatorii HP, atunci când utilizatorii au ponderi diferite. Chiar și valorile corespunzătoare utilizatorilor LP sunt mai mici de 5-6%, cu excepția situației când sunt doar 1 sau 2 utilizatori LP, și restul până la 10 sunt HP. Algoritmul TFL are valori similare cu OQ și WRR, cu excepția situației când există doar unul sau doi utilizatori dintr-una din cele două clase (fie HP, fie LP). Așa cum era de așteptat, limitele intervalelor de încredere pentru algoritmul LQ sunt foarte mari (de ordinul 25%).

Diferența acestor valori în cazul utilizatorilor cu pondere LP poate fi pusă pe seama faptului că valorile întârzierilor sunt mult mai mari în cazul lor și de asemenea

diferențele între întârzieri sunt mai mari. De asemenea, această situație apare atunci când numărul de utilizatori LP este mic (1 sau 2).

Din punctul de vedere al performanței algoritmilor, este mult mai important să se asigure nu doar performanțe mai bune utilizatorilor cu prioritate mai mare, ci și ca gradul de încredere în aceste rezultate să fie mai mare, lucru care este obținut în toate simulările descrise în această lucrare.

4.6 Concluzii

Acest capitol realizează o investigație asupra diversilor algoritmi de planificare utilizați pentru controlul transmiterii datelor la nivel MAC/RLC (*transmission control*) în rețele celulare de tip GPRS și EGPRS. În urma simulărilor rezultă că algoritmi Oldest Queue (OQ) și mai ales Weighted Round Robin (WRR), algoritmi ce aproximează algoritmul idealizat Packet Fair Queueing, sunt mult mai eficienți pentru a fi utilizați la *transmission control* decât algoritmi Longest Queue (LQ), Total Queue Length (TQL) și Total File Length (TFL), algoritmi care alocă resursele rețelei pe baza cantității de date din cozile utilizatorilor.

Pentru realizarea acestei investigații s-au implementat algoritmi menționați în simulatorul dezvoltat și prezentat în capitolul 3 și s-au realizat două seturi de simulări, în funcție de condițiile experimentale, în cadrul fiecărui set existând eventual subseturi. S-a acordat o atenție deosebită gradului de încredere în rezultatele simulărilor, care a fost obținut prin tehnici cum ar fi: utilizarea de seturi diferite de numere (pseudo)aleatoare pentru distribuțiile de probabilități utilizate, utilizarea facilităților oferite de simulator pentru a determina dacă o simulare satisface sau nu un anumit grad de acuratețe, dorit de utilizator, și metode statistice, bazate pe grade de încredere și intervale de încredere.

Prin investigarea influenței pe care o au ponderile alocate utilizatorilor în performanțele acestora s-a demonstrat o metodă simplă de a asigura atât diferențierea calității serviciului pentru utilizatori, cât și reducerea congestiilor din rețea. Această metodă constă în asocierea ponderii utilizatorilor fie cu clasa lor de calitate a serviciului, în primul caz, fie cu calitatea legăturii radio, în al doilea caz. Performanțele utilizatorilor sunt reprezentate de întârzierile suferite de fișierele acestora.

În urma acestor investigații se crează premisele utilizării rezultatelor din acest capitol la problema admiterii utilizatorilor (*admission control*), care va fi prezentată pe larg în capitolul 5. Exprimarea întârzierilor utilizatorilor în funcție de încărcarea rețelei, de valorile ponderilor utilizatorilor, precum și de numărul utilizatorilor din fiecare clasă de QoS va permite realizarea unui algoritm eficient pentru admiterea

utilizatorilor.

Contribuțiile acestui capitol sunt:

1. Un studiu sistematic prin simulare al performanței a două categorii de algoritmi de planificare a resurselor radio în rețele GPRS și EGPRS: o categorie de algoritmi ce aproximează algoritmul idealizat Fluid Fair Queueing, categorie reprezentată de algoritmi WRR și OQ și o categorie de algoritmi la care resursele se alocă pe baza cantității de date din cozile utilizatorilor (cu reprezentanții LQ, TQL și TFL). În urma simulărilor efectuate s-a arătat că algoritmi WRR și OQ sunt mai eficienți decât cei bazați pe cantitatea de date din cozile utilizatorilor. Rezultate legate de aceste aspecte au fost publicate în [110], [109], [111].
2. Interpretarea și explicarea originală, pe un caz general, a rezultatului obținut prin simulare, adică a faptului că algoritmi bazați pe cantitatea de date din cozile utilizatorilor nu sunt eficienți pentru implementarea controlului transmiterii datelor în rețele (E)GPRS. Această concluzie este originală și importantă în domeniul controlului transmiterii datelor în rețele celulare, deoarece există autori care au utilizat algoritmi bazați pe cantitatea de date din cozile utilizatorilor. Acest rezultat a fost publicat în [110].
3. Studiul influenței factorilor de pondere ai utilizatorilor, atunci când acești factori de pondere sunt asociați clasei de calitate a serviciului (QoS) a utilizatorilor ([110]).
4. Studiul efectului factorilor de pondere, atunci când aceștia sunt asociați cu calitatea legăturii radio a utilizatorilor. Prin favorizarea utilizatorilor cu o legătură radio mai bună se pot reduce congestiile din rețea. Investigații pe această temă au fost prezentate în [109] și [111].
5. Utilizarea unor combinații de metode statistice pentru a asigura un grad ridicat de încredere al rezultatelor obținute prin simulare.

Capitolul 5

Soluție fuzzy pentru controlul admiterii utilizatorilor

În acest capitol se propune o soluție originală, bazată pe logică fuzzy, pentru controlul admiterii utilizatorilor în rețele GPRS și EGPRS. Soluția propusă cuprinde utilizarea unui controler fuzzy performant și un algoritm de admitere a utilizatorilor exprimat prin reguli fuzzy.

Secțiunea 5.1 prezintă motivația acestui mod de abordare a problemei admiterii utilizatorilor. Secțiunea 5.2 prezintă rezultatele obținute prin simulare ale unui studiu al performanțelor controlerelor fuzzy în aplicații complexe de reglare, iar secțiunea 5.3 prezintă algoritmul de admitere a utilizatorilor și validarea sa prin simulare, incluzând și compararea performanțelor algoritmului fuzzy cu algoritmi de admitere a utilizatorilor bazați pe valori de prag.

5.1 Logică fuzzy pentru admiterea utilizatorilor: motivație

Multe dintre metodele de admitere a utilizatorilor (*Admission Control – AC*) pentru rețelele celulare de date propuse în literatura de specialitate și care au fost prezentate în secțiunea 2.4 au o serie de dificultăți care pot fi sintetizate astfel:

1. Pentru a putea exprima problema admiterii utilizatorilor într-o formă tratabilă matematic, se face de multe ori presupunerea că funcțiile de distribuție de probabilitate pentru o serie de parametri de genul durată convorbirilor, probabilitatea de *handover*, caracteristicile traficului generat de utilizatori, etc, sunt de tip Poisson sau ca pot fi descrise prin procese Poisson modulate Markov (MMPP - Markov Modulated Poisson Processes). Astfel de presupuneri sunt

corecte în ce privește traficul de voce, dar nu mai corespund în general complexității traficului de date, ceea ce pune la îndoială corectitudinea unor soluții care au la bază astfel de presupuneri.

2. Pentru probleme reale, complexitatea algoritmilor de AC devine foarte mare, în principal datorită numărului foarte mare de stări care descriu problema, mai ales în cazul utilizării de modele Markov pentru AC ([45]).
3. Pentru algoritmi de AC bazați pe praguri ([63], [89], [94], [53]), este foarte dificil să se determine sau să se atribuie valori pragurilor.

Considerăm că logica fuzzy aplicată la admiterea utilizatorilor poate duce la depășirea acestor probleme deoarece regulile fuzzy de tip *if-then* utilizează variabile lingvistice, care folosesc o reprezentare a cunoștințelor apropiată de cea umană și nu au nevoie de presupuneri privind funcțiile de distribuție de probabilitate ale caracteristicilor traficului, ale duratei convorbirilor, etc.

Un alt avantaj al utilizării logicii fuzzy ar fi complexitatea computațională mult redusă a inferenței fuzzy, comparativ cu metodele bazate pe modele Markov.

Metodele de admitere a utilizatorilor bazate pe valori de prag pot fi extinse în mod natural cu ajutorul logicii fuzzy. Înlocuirea pragurilor fixe care delimitează diverse zone cu reginui care se suprapun parțial și astfel realizează o tranziție graduală de la o zonă la altă va asigura o serie de avantaje, cel mai important fiind acela că **dispare necesitatea de a determina cu precizie valoarea de prag**. Astfel se evită ca pentru determinarea valorilor de prag să se utilizeze metode matematice relativ complexe (cum ar fi programarea neliniară în [53]) sau să se asigneze valori pur euristice ca în [63]. Un alt dezavantaj al utilizării unor valorilor de prag este acela că aceste valori, determinate în anumite condiții conform unor proceduri de calcul (de exemplu programare neliniară) nu mai sunt valabile dacă se schimbă condițiile de lucru și deci procedura de calcul va trebui reluată. Acest dezavantaj ar dispărea în cazul unor metode bazate pe logica fuzzy, care prezintă chiar prin natura lor o toleranță ridicată la modificări ale condițiilor de lucru.

În cazul unei metode de AC de genul celei prezentate în [63], când se aplică strategii de AC diferite în funcție de regiunea în care se lucrează, poate fi un avantaj ca trecerea de la o regiune la alta să se facă gradat, ca și în cazul logicii fuzzy, nu brusc, ca în situația valorilor de prag.

Procesul de inferență fuzzy constă din combinarea unor reguli de tipul *if-then* cu *fapte* (valori măsurate ale unor parametri, de exemplu încărcarea rețelei), obținând concluziile, care sunt folosite pentru a efectua anumite acțiuni în sistem (de exemplu

admiterea sau respingerea unui utilizator). Un alt avantaj important al logicii fuzzy este acela că regulile utilizate la inferența fuzzy sunt exprimate în limbaj natural, aceste reguli putând fi formulate de exemplu de către un expert în domeniu. În acest fel, experiența operatorilor de rețea poate fi utilizată în mod direct, prin încorporarea ei în regulile fuzzy, fără a mai fi necesar să se apeleze la metode matematice complexe, de genul teoriei așteptării sau al modelelor Markov.

Așa cum se menționează în [40] “abilitatea de a modela rețelele în continuumul logicii fuzzy, nu în tradiționalele metode cu valori discrete, combinată cu simulări extensive, oferă un compromis rezonabil între modelarea analitică riguroasă și simulările pur calitative”.

5.2 Performanțele controlerelor fuzzy

5.2.1 Variabile lingvistice, inferența fuzzy

L.A. Zadeh a introdus în anul 1973 noțiunea de *variabile lingvistice*, acestea devenind un instrument de bază în logica fuzzy.

O *variabilă lingvistică* are un nume, un univers al discursului, precum și o mulțime de termeni, care sunt mulțimi fuzzy, definite prin funcțiile lor de apartenență. Variabilele lingvistice iau ca valori termeni dintr-un limbaj (natural sau artificial), și nu valori numerice, ca în cazul variabilelor clasice [123].

În logica fuzzy variabilele lingvistice sunt utilizate pentru *raționamentul nuanțat sau aproximativ (approximate reasoning)*, în cadrul inferenței fuzzy.

Inferența fuzzy a fost definită de către Zadeh, Mizumoto și Mamdani prin generalizarea tautologiei *modus ponens*. În logica clasică, *modus ponens* se descrie formal prin $A \wedge (A \Rightarrow B) \Rightarrow B$ semnificația fiind următoarea: fiind dată premiza “*A este adevărat (A is true)*” și implicația (regula) “*dacă A, atunci B (if A then B)*”, atunci concluzia va fi “*B este adevărat (B is true)*”. *Modus ponens* generalizat în logica fuzzy se exprimă astfel:

Premise	$x \text{ is } A'$
Implication	If $x \text{ is } A$ then $y \text{ is } B$
Conclusion	$y \text{ is } B'$

Fiind dat un fapt A' și o regulă $R_{A \rightarrow B}$, inferența fuzzy înseamnă compoziția dintre fapt și regulă ($A' \bullet R_{A \rightarrow B}$), cu scopul de a obține noi fapte, respectiv concluzia B' , dată de relația $B' = A' \bullet R_{A \rightarrow B}$.

În această teză se va utiliza pentru inferența fuzzy procedura computațională propusă de Zadeh [123], [67], [78].

Conform acestei proceduri, funcția de apartenență a mulțimii B' va fi:

$$\mu_{B'}(y) = \max_x \min(\mu_{A'}(x), \mu_{R_{A \rightarrow B}}(x, y)) \quad (5.1)$$

unde: $\mu_{R_{A \rightarrow B}}(x, y) = \min(\mu_A(x), \mu_B(y))$, $x \in U$, $y \in V$ și $\mu_A(x)$, $\mu_{A'}(x)$, $\mu_B(y)$ sunt funcțiile de apartenență pentru mulțimile fuzzy A , A' și B , definite pe U și V și luând valori în intervalul de numere reale $[0, 1]$. A este un termen în premiză, A' este faptul, B , este un termen în concluzie, iar B' este consecventul (consecința) regulii.

După transformări, din (5.1) se obține (5.2)

$$\mu_{B'}(y) = \min \max_{x \in U} [\min(\mu_{A'}(x), \mu_A(x)), \mu_B(y)] = \min(a, \mu_B(y)) \quad (5.2)$$

unde

$$a = \max_{x \in U} \min(\mu_{A'}(x), \mu_A(x)) \quad (5.3)$$

Atunci când un fapt va activa mai multe reguli, rezultatele tuturor regulilor activate se vor combina astfel:

$$B' = B_1' \cup B_2 \cup \dots \cup B_n' \quad (5.4)$$

unde B_i' , $i = 1, 2, \dots, n$ sunt rezultatele fiecărei reguli activate.

Dacă premiza este compusă, atunci gradele de activare ale tuturor subpremiselor (a , în (5.3)) sunt combinate folosind operatorii logici corespunzători (AND, OR, NOT). În această lucrare se va folosi doar operatorul logic AND (ȘI), implementat prin operația de minim în logică fuzzy.

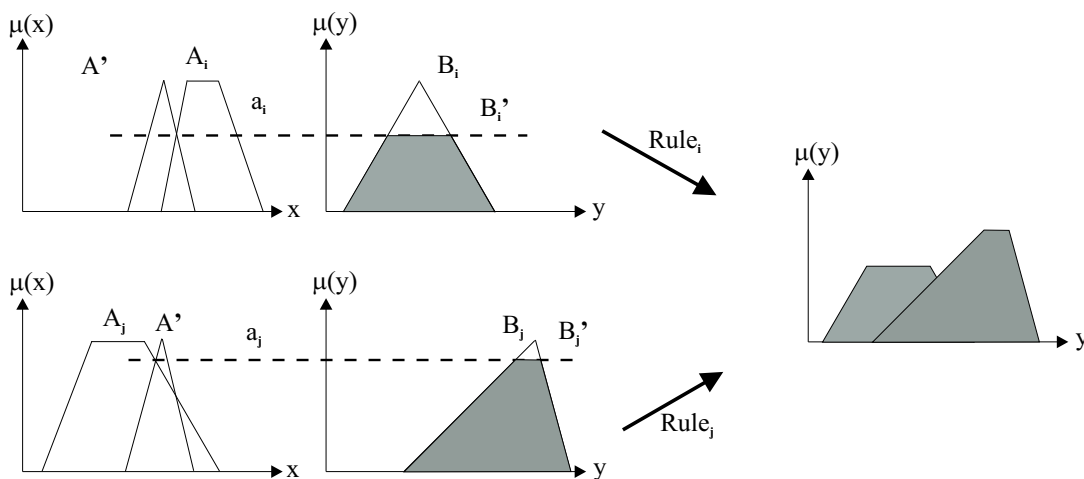


Figura 5.1: Mecanismul de inferență fuzzy

De cele mai multe ori se dorește ca rezultatul operației de inferență fuzzy să fie o valoare de tip *crisp* (adică nu o mulțime fuzzy). Această valoare *crisp* se obține

printr-o operație numită defuzzificare. Cea mai utilizată procedură de defuzzificare este metoda centrului de greutate.

Figura 5.1 ilustrează procesul de inferență fuzzy pentru cazul unei premise simple, când faptul A' intersectează doi termeni lingvistici, A_i și A_j din premiza regulii și deci activează două reguli, i și j , gradele de activare fiind a_i și a_j .

Consecințele B'_i și B'_j ale celor două reguli sunt combinate în partea dreaptă a figurii.

5.2.2 Controlere fuzzy digitale

Figura 5.2 [74] prezintă schema generală a unui proces de reglare fuzzy, elementele principale fiind *procesul condus* (*Controlled System / Process*) și *controlerul fuzzy* (*FLC*), reprezentat cu linie punctată în figură. FLC este alcătuit din următoarele blocuri: fuzzificator (*Fuzzifier*), mașina de inferențe fuzzy (*Decision Making Logic*), baza de cunoștințe (*Knowledge Base*) și defuzzificatorul (*Defuzzifier*).

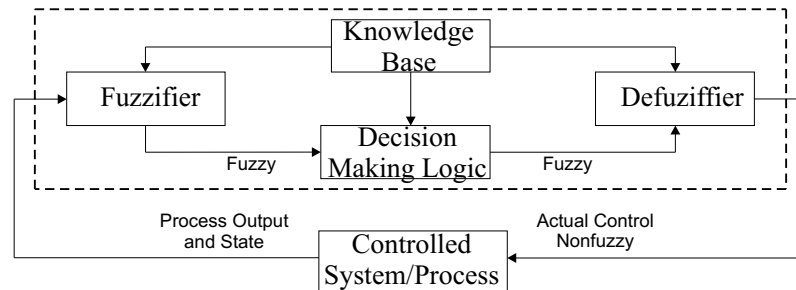


Figura 5.2: Schema generală a unui proces de reglare fuzzy

Informația de intrare în FLC, preluată de la procesul condus, constă într-un set de valori *crisp*, care sunt mai întâi transformate în mulțimi fuzzy de către fuzzificator, constituind *faptele* în cadrul procesului de inferență fuzzy. Inferența fuzzy este declanșată de apariția faptelor, care vor activa anumite reguli din baza de cunoștințe. Procesul de inferență fuzzy este implementat de mașina de inferență fuzzy. Rezultatele inferenței fuzzy sunt defuzzificate de către blocul de defuzzificare, pentru a se obține mărimile de control, de tip *crisp*, care vor acționa asupra procesului condus.

Operația de fuzzificare este descrisă în figura 5.3. În partea stângă a figurii este prezentată valoarea de intrare, non-fuzzy (*crisp value*), iar în partea dreaptă a figurii apare rezultatul defuzzificării, care este o mulțime fuzzy centrată în jurul valorii *crisp* inițiale. Pe abscisă este reprezentat domeniul de valori ale intrării în FLC, în partea stângă, respectiv reprezentarea internă în FLC a acestui domeniului (în partea dreaptă a figurii), iar pe ordonată apar valorile funcției de apartenență ale mulțimii

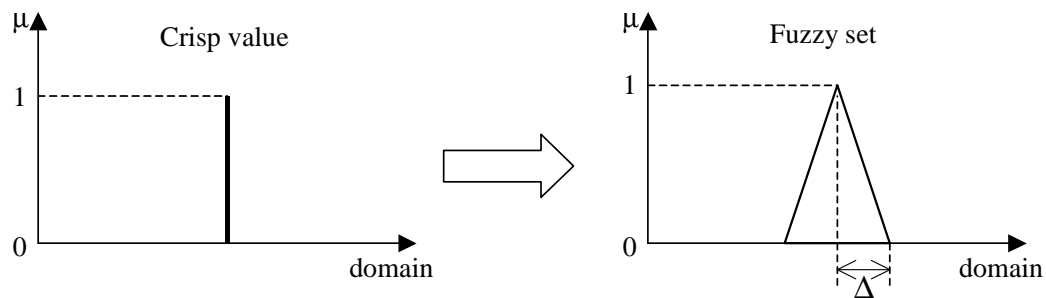


Figura 5.3: Operația de fuzzificare

fuzzy rezultate în urma fuzzificării. S-a notat cu Δ gradul în care se face fuzzificarea: dacă $\Delta > 0$ atunci rezultatul fuzzificării este o mulțime fuzzy propriu-zisă, iar dacă $\Delta = 0$, atunci rezultatul fuzzificării este un *singleton*, adică practic valoarea intrării nu este modificată.

Există numeroase situații practice în care se cere ca asupra procesului reglat să se acționeze în timp real, la intervale scurte, și atunci este necesară implementarea hardware a controlerului fuzzy.

Primul circuit integrat fuzzy a fost realizat de către Togai și Watanabe în 1986 ([113]), iar de atunci s-au implementat numeroase tipuri de circuite fuzzy, atât în tehnologie digitală (de exemplu [117]), cât și analogică (de exemplu [120]).

Compararea performanțelor controlerelor fuzzy (indiferent că sunt implementate hardware sau nu) constituie o problemă complexă, datorită faptului că aceste FLC-uri sunt realizate în tehnologii foarte diferite, iar implementările lor sunt de asemenea foarte diverse, astfel încât fiecare producător de controlere fuzzy scoate în evidența calitățile produsului său.

În cele ce urmează se va prezenta un studiu al performanței controlerelor fuzzy, studiu realizat prin simulare. În cadrul acestui studiu se au în vedere câteva implementări hardware în tehnologie digitală a FLC-urilor, dar rezultatele sunt mai generale, nefiind legate de aspectul implementării hardware.

Caracteristicile FLC

Una din cele mai importante caracteristici ale unui FLC îl constituie viteza sa de operare, care comercial se reprezintă fie prin numărul de inferențe fuzzy pe secundă (*Fuzzy Inferences Per Second – FIPS*), fie prin numărul de operații elementare (minim, maxim, etc) pe secundă. În primul caz inferența este definită ambiguu: fie ca operația descrisă de o singură regulă, fie doar o parte a unei reguli.

Patyra ([74]) precizează că acești parametri nu pot măsura suficient de bine viteza unui FLC și introduce timpul de răspuns intrare-ieșire, notat θ_{IN-OUT} . Acesta este definit ca întârzierea totală din momentul în care datele (faptele) sunt prezente la intrarea FLC până în momentul când se generează valorile *crisp* la ieșire.

În studiul său care dorește să sistematizeze compararea performanțelor FLC [74] Patyra propune un set de parametri care să caracterizeze un FLC.

Parametrii fundamentali ai unui FLC sunt:

- numărul de variabile de intrare (K)
- numărul de variabile de ieșire (L)
- numărul maxim de reguli lingvistice din baza de reguli (N)
- numărul maxim de funcții de apartenență pentru universul discursului la fiecare intrare (MB_{IN})
- numărul maxim de funcții de apartenență pentru universul discursului la ieșiri (MB_{OUT})
- numărul de vectori binari ce caracterizează funcțiile de apartenență (n), adică numărul de biți pe care se reprezintă domeniul de valori pentru fiecare intrare
- numărul de biți dintr-un vector binar (m), care determină precizia de reprezentare a valorilor funcțiilor de apartenență
- timpul de răspuns intrare-ieșire (θ_{IN-OUT})

Pentru un FLC cu K intrări, dacă fiecare variabilă de intrare are maximum MB_{IN} termeni, rezultă că numărul maxim de reguli fuzzy este $N = (MB_{IN})^K$.

În continuare vom considera, fără a restrânge generalitatea problemei, controlere fuzzy cu o singură ieșire, adică $L = 1$.

FLC clasic

Tot în [74] se propun două scheme generale pentru controlere fuzzy, din care ne vom referi în continuare doar la una din ele, FLC clasic, care este prezentată în figura 5.4 pentru cazul $K = 2$ și $L = 1$, numit DISO (*Double Input Single Output*). Cazul general se poate cu ușurința deduce din acesta.

Fiecare din valorile de intrare de la *Input 1* și *Input 2* este fuzzificată, rezultând un vector de $n * m$ biți, stocat în memoriile A1, respectiv A2. Primul nivel de circuite MIN va face intersecția dintre vectorul de intrare și fiecare din termenii fuzzy ai

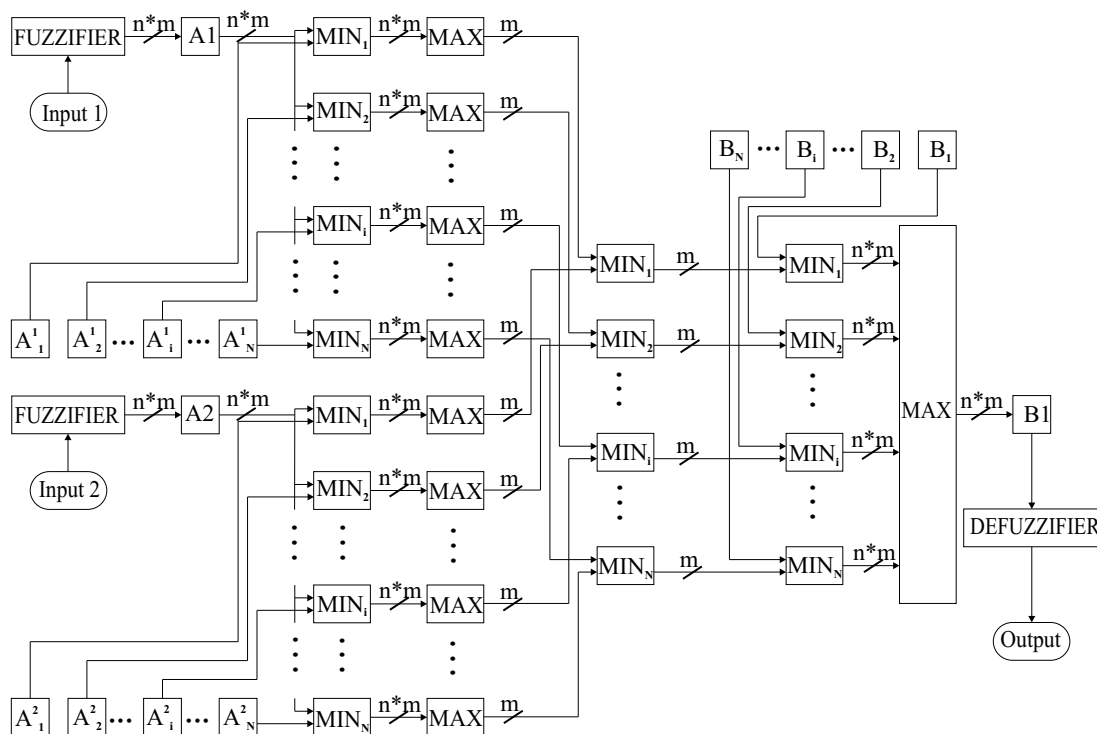


Figura 5.4: FLC clasic cu 2 intrări

respectivei variabile de intrare (stocați în memoriile $A_1^1 \dots A_N^1$ pentru intrarea 1 și respectiv $A_1^2 \dots A_N^2$ pentru intrarea 2). Ieșirea fiecărui bloc MIN din acest prim nivel este de asemenea un vector de $n \cdot m$ biți. Nivelul următor, format din blocuri MAX, determină pentru fiecare din aceste intersecții de mulțimi valoarea maximă, sub forma unei mărimi scalare pe m biți. În figura 5.1 aceste valori sunt descrise prin a_i și a_j și expresia lor apare în ecuația (5.3). Aceste valori reprezintă gradul în care fiecare din faptele de intrare se potrivește cu fiecare din variabilele lingvistice din premise, adică gradul în care fiecare din premise este adevărată.

Pentru a determina gradul de activare al fiecărei reguli fuzzy mai trebuie combinate gradele de adevăr ale premizelor, lucru realizat de următorul nivel de blocuri MIN, care implementează funcția ȘI logic (*AND*) care leagă premisele regulii. Pe figura 5.4 se observă că aceste blocuri MIN au ambele intrări și ieșirea pe m biți. Cu gradul de activare a regulii astfel obținut pentru fiecare regulă (acest grad de activare se mai numește *fire strength*) se va face operația de minim cu toți termenii din concluzii (ultimul nivel de blocuri MIN din figură), rezultând câte un vector de n valori a m biți. Pe figura 5.1 acești vectori sunt mulțimile B_i' și B_j' . Combinarea tuturor consecințelor regulilor active este realizată de ultimul bloc MAX (care implementează operatorul reuniune din ecuația (5.4)), iar rezultatul final va fi în cele din

urmă defuzzificat.

Pe baza pe schemei din figura 5.4, se poate estima timpul de răspuns intrare-ieșire ([74]):

$$\Theta_{IN-OUT} = \tau_{FUZZ} + 3\tau_{MIN} + (n - 1)\tau_{MAX} + (N - 1)\tau_{MAX} + \tau_{DEFUZZ} \quad (5.5)$$

FLC simplificat

Chiueh a propus în [29] o simplificare a arhitecturii controlerelor fuzzy, simplificare posibilă dacă se îndeplinesc următoarele condiții:

1. intrările în FLC nu mai sunt fuzzificate, utilizându-se direct valorile *crisp* de intrare
2. termenii variabilelor lingvistice de intrare se aleg astfel încât să nu se intersecteze mai mult de doi termeni. În figura 5.5 se prezintă o astfel de situație, iar termenii variabilelor lingvistice sunt împărțiți în *pari* și *impari*. Astfel, partea de sus a figurii 5.5 prezintă toți termenii unei variabile lingvistice de intrare, notați *nl* – *negative large*, *nm* – *negative medium*, *ns* – *negative small*, *zr* – *close to zero*, *ps* – *positive small*, *pm* – *positive medium*, *pl* – *positive large*, pe rândul următor sunt arătați termenii impari (*nl*, *ns*, *ps* și *pl*), iar în partea de jos a figurii apar termenii pari: *nm*, *zr* și *pm* (se consideră că numerotarea termenilor începe de la 1 și se face de la stânga spre dreapta).

Se notează cu *MOF* (*Maximum Overlap Factor*) gradul maxim de suprapunere între termenii variabilelor lingvistice din premise și cu *MNSAR* (*Maximum Number of Simultaneously Active Rules*) numărul maxim de reguli fuzzy active simultan. Pentru un număr de K intrări ale FLC, se obține:

$$MNSAR = MOF^K \quad (5.6)$$

Numărul maxim de reguli active simultan va determina gradul de paralelism al unei implementări hardware a FLC simplificat. Pentru $MOF = 2$ și $K = 3$ se obține $MNSAR = 8$.

În [107], [112], [105] am detaliat schema de principiu prezentată de Chiueh în [29] pentru cazul FLC cu două și cu trei intrări, iar circuitul rezultat l-am denumit FLC simplificat.

Arhitectura FLC simplificat cu 3 intrări este prezentată în figura 5.6 ([112], [105]).

Pentru fiecare intrare există o memorie ce conține funcțiile termenilor lingvistici ai variabilei de intrare respective, digitizați pe $n = 64$ de biți. Fiecare locație de

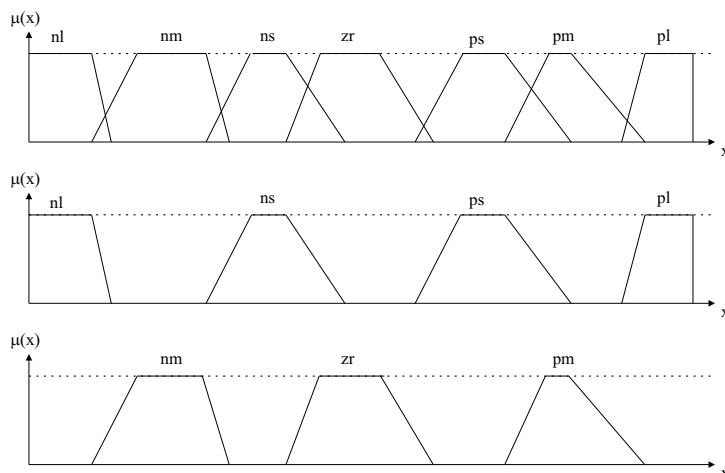


Figura 5.5: Termenii variabilei de intrare, împărțiți în pari și impari

memorie conține simbolul ($S - Symbol$) și gradul de apartenență ($D - Degree$) a cel mult doi termeni lingvistici: termenul par și cel impar. Valoarea *crisp* a intrării va fi folosită ca adresă în memoria de intrare, iar din memorie vor fi citite perechile ($Symbol, Degree$) corespunzătoare celor doi termeni. Dacă într-o anumită zonă de memorie există doar un singur termen lingvistic, atunci în locul celuilalt se stochează în memorie codul pentru *no symbol*. Fiecare pereche ($symbol, degree$) are doi indici: primul indice corespunzător intrării (de la 1 la 3 în figură), iar al doilea indice indică dacă termenul respectiv e par (indicele este 1) sau impar (indicele are valoarea 2).

Prin concatenarea codurilor tuturor combinațiilor simbolurilor corespunzătoare celor 3 intrări se obțin adresele în memoriile de reguli notate RM_{111} , la RM_{222} din figura 5.6. Astfel simbolurile impare din cele 3 intrări (adică S_{11} , S_{21} și S_{31}) vor fi adrese în memoria de reguli (*Rule memory*) RM_{111} , simbolurile S_{11} , S_{21} și S_{32} vor fi adrese pentru RM_{112} și așa mai departe. Conținutul locației adresate de cele 3 simboluri va fi codul termenului lingvistic al variabilei de ieșire din regulile fuzzy.

În paralel cu procesarea simbolurilor se prelucrează și gradele de apartenență (*degree*) obținute din memoriile de intrare, și anume se face operația MIN între gradele de apartenență corespunzătoare celor 3 variabile de intrare, realizând astfel operația și logic (AND). Pentru aceasta s-au folosit două rânduri de blocuri MIN cu două intrări, iar la ieșirile acestor blocuri se obțin gradele în care premisele regulilor sunt adevărate (corespunzătoare valorilor a din ecuația (5.3)).

Făcând MIN între aceste valori și funcția de apartenență a termenului variabilei lingvistice de ieșire se obține mulțimea B' corespunzătoare regulii activate. Funcția de apartenență a termenului variabilei lingvistice de ieșire se obține prin utilizarea

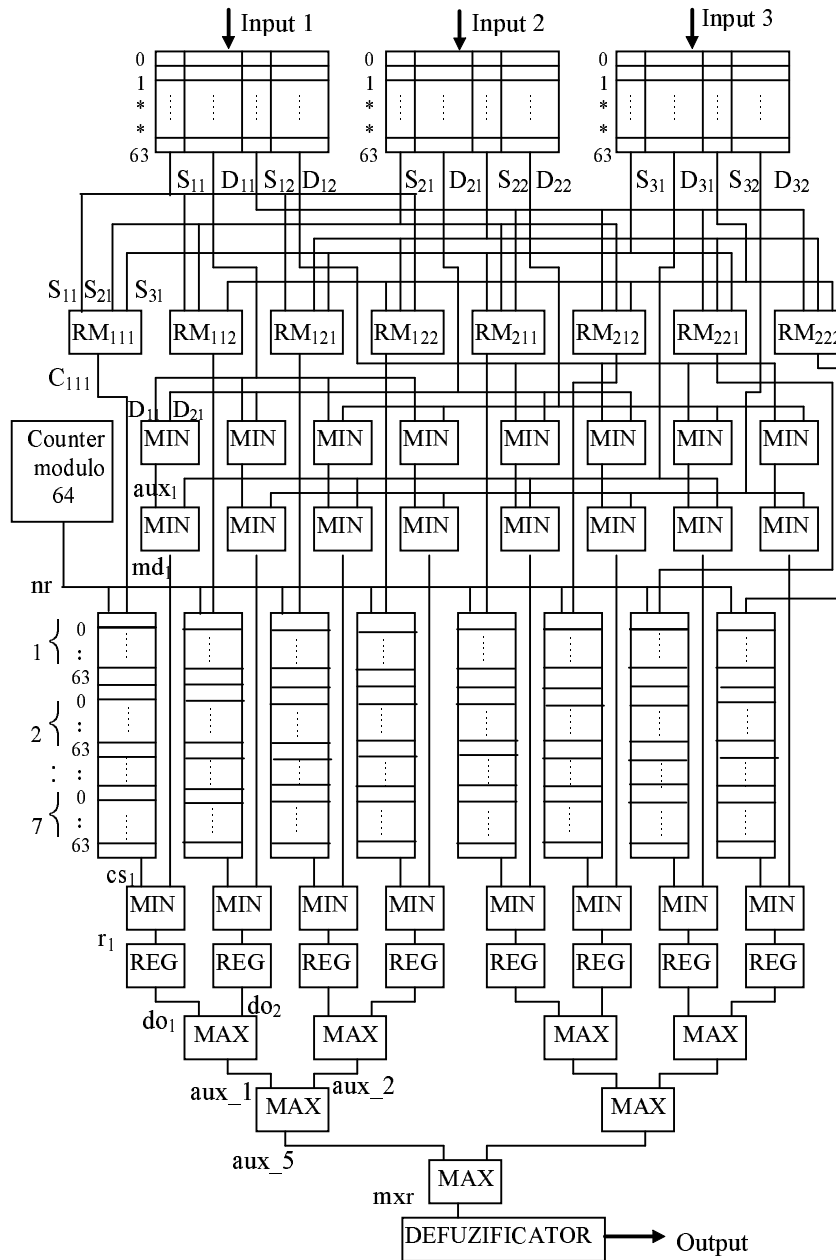


Figura 5.6: FLC simplificat cu 3 intrări

simbolului acestui termen, obținut din memoriile de reguli, ca adresă în memoriile care stochează termenii variabilei lingvistice de ieșire. Fiecare din aceste memorii conține toți termenii ieșirii (7 termeni în figura 5.6), fiecare termen fiind reprezentat prin n valori pe m biți. Cum în cazul din figură $n = 64$, se utilizează un numărător modulo 64 (*Counter modulo 64*) pentru a parcurge întregul domeniu al fiecăruia din termenii din concluzii adresați.

Consecințele regulilor activate se reunesc prin circuitul *MAX* cu 8 intrări, a cărui ieșire intră în defuzzificator.

Registrele REG sunt opționale și au doar rolul de a face pipeline între operația de inferență și cea de defuzzificare.

Timpul de răspuns intrare-ieșire al FLC simplificat a fost evaluat în [9] și este:

$$\Theta_{IN-OUT} = \tau_{MIN} + n\tau_{MIN} + \tau_{MAX} + 3\tau_{MEM} + \tau_{DEFUZZ} \quad (5.7)$$

5.2.3 Implementarea și evaluarea performanțelor controlerelor fuzzy

Pentru un studiu sistematic al performanțelor controlerelor fuzzy nu este suficient să ne raportăm doar la timpul de propagare intrare-ieșire și eventual la alte aspecte legate de implementarea hardware a FLC, cum ar fi costul hardware, exprimat prin dimensiunea memoriilor interne și prin cantitatea de circuite ([74]), ci este foarte important să se determine performanțele FLC în aplicații de reglare. Astfel, există posibilitatea ca prin simplificări ale arhitecturii hardware (de exemplu simplificarea propusă de Chiueh) să se obțină un FLC mai rapid și cu o arhitectură care să permită o implementare hardware mai ieftină, dar dacă în urma acestor simplificări arhitecturale scad performanțele de reglare (de exemplu prin creșterea numărului de pași de reglare necesari pentru a aduce sistemul condus într-o anumită stare), atunci se pierd avantajele obținute prin scăderea timpului de propagare intrare-ieșire.

Ideea propusă în această lucrare (și care a mai fost de asemenea discutată în [104] și [10]) este să se testeze performanțele FLC în aplicații complexe de reglare. Dacă aplicația de reglare este prea simplă, atunci diferențele de performanță (de exemplu numărul de pași de reglare) între diverse FLC pot fi neconcludente, lucru care nu se mai întâmplă într-o aplicație complexă de reglare.

În consecință, pentru a studia performanțele controlerelor fuzzy, nu doar din punct de vedere al timpului de propagare intrare-ieșire, ci și din punct de vedere al comportării FLC în aplicații complexe de reglare, am realizat un model de simulare în limbajul de descriere hardware VHDL ([13]). Am ales limbajul VHDL deoarece acesta

permite atât o modelare apropiată de hardware, pentru descrierea FLC, cât și o modelare la nivel abstract, utilizată pentru procesul reglat.

Zamfirescu și Ussery ([121]) au fost primii care au utilizat limbajul VHDL pentru modelarea sistemelor fuzzy, fiind apoi urmați și de alți cercetători (de exemplu [39]).

Primele modele pe care le-am realizat au fost descrieri la nivelul transferului între registre (*Register Transfer Level – RTL*, a se vedea de exemplu [35] pentru nivelurile la care se poate modela un sistem utilizând VHDL) ale FLC simplificat cu două intrări (prezentat în [107]) și apoi cu trei intrări ([112]). În [105] am descris FLC simplificat cu 3 intrări la nivel mai abstract decât RTL, și anume la nivel comportamental (*behavioural*), realizându-se totodată și o parametrizare a FLC după numărul de intrări, numărul de biți pe care se reprezintă domeniul de valori al intrărilor și după numărul de termeni lingvistici ai variabilelor de intrare.

Modelul VHDL al FLC cu trei intrări a fost apoi validat prin utilizarea sa într-o aplicație complexă de reglare, și anume pentru controlul unui proces cu fază non-minimă, rezultatele fiind publicate în [79].

Am realizat modelarea în VHDL a unui FLC complex și complet parametrizat, precum și studiul performanțelor sale în aplicații complexe de reglare în [104] și [10]. Am implementat 3 variante de FLC, inclusiv FLC clasic și FLC simplificat, iar aplicațiile de reglare folosite au fost: un proces cu o intrare și o ieșire descris în [80], pendulul invers (descriș în [120]) și un proces cu fază non-minimă (descriș în [77] și apoi folosit și în [79]).

Pe lângă parametrii fundamentali ai unui FLC menționați la pagina 117, a mai fost studiată și influența parametrului Δ , reprezentând gradul în care se fuzzifică valorile intrărilor (a se vedea și figura 5.3). Acest parametru este foarte important, deoarece în cazul unei implementări hardware, în cazul $\Delta = 0$ se poate utiliza arhitectura FLC simplificat, care reduce complexitatea hardware a implementării și crește gradul de paralelism al procesării inferențelor fuzzy.

Cele mai interesante rezultate au fost obținute pentru procesul cu fază non-minimă, la care au fost mășurați mai mulți parametri care descriu calitatea reglării atunci când la intrarea sistemului reglat se aplică un impuls treaptă unitate, cei mai importanți parametri fiind *timpul de răspuns* t_r (*settling time*), reprezentând timpul necesar până când ieșirea sistemului atinge valoarea 1, și *deviația* dev a sistemului. Deviația reprezintă diferența dintre valoarea ieșirii sistemului și valoarea de referință (valoarea dorită a ieșirii), măsurată după ce ieșirea sistemului se stabilizează. În mod ideal $dev = 0$, dar se consideră că s-a obținut o calitate acceptabilă a reglării dacă $dev < 2\%$ din valoarea de referință. Parametrul dev caracterizează valoarea maximă a amplitudinii oscilațiilor ieșirii sistemului reglat atunci când el se află în zona de

stabilitate.

Așa cum am arătat și în [104], în urma simulărilor pe care le-am efectuat am constatat că valoarea parametrului m (numărul de biți pe care se reprezintă valorile funcțiilor de apartenență) nu are o influență majoră asupra performanțelor de reglare dacă $m \geq 4$. Numărul de biți pe care se reprezintă domeniile de valori ale funcțiilor de apartenență ale intrărilor, n , are o influență mai mare, obținându-se o calitate complet nesatisfăcătoare a reglării dacă $n < 64$. Pentru $n \geq 64$ timpul de răspuns nu mai este afectat de valorile lui n , dar deviația este mai mică dacă n crește.

Deviația este influențată și de parametrul Δ , obținându-se cele mai bune valori pentru dev atunci când valorile lui Δ sunt aproximativ jumătate din distanța dintre doi termeni fuzzy de intrare consecutivi. În această aplicație termenii fuzzy ai intrărilor au formă triunghiulară, cu excepția termenilor de la capetele domeniului de valori, care au formă trapezoidală, iar distanța dintre centrele triunghiurilor este de $n/8$. Dacă Δ are valori prea mari, comparativ cu distanța dintre termenii fuzzy, atunci calitatea reglării (exprimată prin parametrul dev) scade inacceptabil, deoarece fiecare fapt de intrare declanșează un număr mare de reguli, astfel încât în procesul de reglare sunt prea multe reguli fuzzy active simultan și efectul este că regulile se compensează reciproc. Pentru cazul de interes când $\Delta = 0$ (caz în care poate fi simplificată arhitectura FLC) s-au obținut valori foarte bune pentru dev în cazul $n = 128$ ($dev = 1.7\%$) și valori ceva mai mari de 2% dacă $n = 64$. Este important de precizat că viteza de reglare, exprimată prin timpul de răspuns al sistemului t_r , nu este influențată de valorile lui Δ dacă $n \geq 64$.

Rezultatele studiului cu privire la performanțele controlerelor fuzzy prezentate în această secțiune sunt utilizate în secțiunea 5.3, care prezintă algoritmul fuzzy de AC propus în această teză.

Astfel, controlerul fuzzy utilizat pentru algoritmul de AC propus în această teză și prezentat în secțiunea 5.3, precum și în [103], [102], [106], [108], este identic cu FLC clasic modelat în VHDL și descris în această secțiune. FLC folosit pentru admiterea utilizatorilor a fost implementat în C/C++, pentru a putea fi integrat în modelul de simulare utilizat în această teză (și descris în capitolul 3), dar a fost întâi testat pentru controlul procesului cu fază non-minimă descris anterior și rezultatele obținute au coincis perfect cu rezultatele din [104] ale modelului VHDL.

Mai mult, în urma studiului performanțelor controlerelor fuzzy s-au determinat valorile parametrilor FLC utilizat pentru AC astfel încât să se asigure o calitate cât mai bună a controlerului fuzzy. Astfel pentru n și m s-au stabilit valorile $n = 64$ și $m = 4$, termenii fuzzy folosiți în premise au formă triunghiulară sau trapezoidală,

iar termenii fuzzy folosiți în concluzie sunt de tip *singleton*, la fel ca și în cazul procesului cu fază non-minimă. Pentru gradul de fuzzificare al valorilor intrărilor s-a ales valoarea $\Delta = 0$, care asigură o calitate suficient de bună, în acest stadiu al dezvoltării algoritmului de AC nefiind necesară o mai mare finețe a reglării (ca în cazul aplicațiilor de reglare la care se impune condiția $dev < 2\%$).

5.3 Algoritmul fuzzy de admitere a utilizatorilor

Această secțiune descrie algoritmul de AC bazat pe logică fuzzy pe care îl propune această teză. Algoritmul a mai fost prezentat, în diverse etape ale dezvoltării sale, în [103], [102], [106], [108].

5.3.1 Cerințele algoritmului de admitere a utilizatorilor

În general la un algoritm de AC, pentru ca un nou utilizator să poată fi admis, trebuie ca resursele folosite de utilizatorii deja conectați (RES_{in_use}) plus resursele cerute de noul utilizator (RES_{new}) să nu depășească resursele existente ale rețelei ($RES_{available}$):

$$RES_{in_use} + RES_{new} \leq RES_{available} \quad (5.8)$$

Resursa critică depinde de tipul rețelei. Pentru rețele de tip ATM se consideră că resursa critică este capacitatea bufferelor și dacă algoritmul de AC nu este eficient se poate ajunge la o suprautilizare a bufferelor, ceea ce duce la pierderi de date datorită umplerii bufferelor. În cazul rețelelor radio (cum sunt GSM, GPRS și EGPRS), resursa critică este reprezentată de lățimea de bandă radio, mai exact de canalele radio.

La rețelele radio utilizate pentru voce, cum este GSM, alocarea resurselor radio se face în mod rigid, datorită utilizării comutării de circuite: fiecare conexiune primește o cantitate fixă de resurse, adică de canale radio (în cazul GSM, un singur canal pentru o conexiune), astfel încât o nouă conexiune este acceptată dacă se pot aloca resurse radio pentru ea. Deoarece alocarea resurselor radio se face în fiecare celulă, principala problemă care se pune este de a rezerva resurse pentru utilizatorii care vin din alte celule, astfel încât să se evite întreruperea convorbirilor în curs de desfășurare atunci când utilizatorul se deplasează într-o altă celulă.

Așa cum s-a menționat în secțiunea 2.4, parametrii urmăriți la algoritmi de AC pentru rețelele celulare utilizate pentru voce sunt minimizarea probabilității de blocare a apelurilor telefonice (*call blocking probability*) și mai ales a probabilității de întrerupere a convorbirilor în curs de desfășurare (*call dropping probability*). Este mai

important să se minimizeze probabilitatea de întrerupere a convorbirilor în curs de desfășurare, chiar cu prețul unei ușoare creșteri a probabilității de blocare a convorbirilor noi, deoarece se consideră că este mult mai neplăcut pentru un utilizator să îi fie întreruptă o convorbire sau o sesiune de transfer de date în curs de desfășurare decât să îi fie blocată o încercare de inițiere a unei convorbiri sau sesiuni de date.

La rețelele celulare de date alocarea resurselor se face într-un mod mult mai flexibil decât pentru voce, pe de o parte datorită naturii traficului de date (*bursty*), iar pe de altă parte, datorită faptului că majoritatea aplicațiilor de date pot tolera o anumită fluctuație a resurselor. Desigur, există și aplicații de date care cer o alocare constantă de resurse (clasa *CBR* – *Constant Bit Rate* din ATM sau, în cazul rețelelor radio, clasa de trafic conversațională, plus unele aplicații de tip streaming), dar pentru aceste aplicații se vor rezerva resurse. În această lucrare ne vom concentra asupra aplicațiilor care nu necesită rezervare de resurse. Chiar și dacă se iau în considerare aplicațiile pentru care se rezervă resurse, problema discutată aici rămâne valabilă pentru restul aplicațiilor, pentru care nu se face rezervare de resurse, cu amendamentul că în ecuația (5.8), termenul $RES_{available}$ va conține resursele rețelei care rămân după ce s-au rezervat resurse pentru aplicațiile care necesită rezervare.

Scopul unui algoritm de AC pentru rețele celulare de date, cum sunt GPRS și EGPRS, este să admită doar atâția utilizatori în rețea încât cerințele de QoS ale utilizatorilor să poată fi satisfăcute de resursele existente ale rețelei, iar gradul de utilizare a acestor resurse să fie maximizat. La rețelele celulare de date, în afară de cerința de a minimiza probabilitățile de întrerupere și respectiv de blocare a sesiunilor de date, un algoritm de AC mai trebuie să asigure îndeplinirea cerințelor de calitate a serviciului (QoS) ale utilizatorilor.

5.3.2 Formularea matematică a cerințelor algoritmilor de admitere a utilizatorilor

În cazul utilizatorilor GPRS/EGPRS, cel mai important parametru de QoS pe care trebuie să îl asigure rețeaua este dat de durata transmiterii datelor (*sending delay*), deoarece acest parametru afectează și capacitatea de trecere a rețelei, deci atât aplicațiile sensibile la întârzieri (streaming), cât și cele sensibile la capacitatea de trecere (de exemplu FTP) sunt direct afectate de durata transmiterii datelor. Utilizatorii percep însă întârzierea totală (*total delay*), dar valoarea întârzierii totale este influențată decisiv de valoarea parametrului *sending delay* (întârzierea totală este suma dintre timpul de transfer și timpul de așteptare al fișierelor, dar până la urmă timpul de așteptare este la rândul lui influențat de timpul de transfer). Alți

parametri de QoS, cum ar fi cei legați de fiabilitate, sunt asigurați prin codificare, detecția datelor eronate, retransmisii, etc și nu fac obiectul acestei teze.

Deoarece algoritmi de planificare utilizați la nivel MAC/RLC operează cu blocuri radio, care au o lungime fixă, înseamnă că algoritmul utilizat în această teză pentru planificare la acest nivel, adică WRR (cel mai performant dintre algoritmi de planificare investigați în capitolul 4), este o foarte bună aproximare a algoritmului *Fluid Fair Queueing* (conform [61]), deci timpul de transfer al unui fișier aparținând utilizatorului j , din clasa de QoS i , notat $Sending_delay_j$, se poate aproxima conform ecuației:

$$Sending_delay_j \cong \frac{1}{W_i} \cdot \frac{FL \cdot \sum_i (k_i \cdot W_i)}{B_{avr}} \cdot cntr_cycle \quad (5.9)$$

în care FL reprezintă lungimea fișierului, k_i este numărul de utilizatori din clasa de QoS i , iar W_i este ponderea unui utilizator din clasa de QoS i , iar B_{avr} este numărul mediu de canale utilizate pentru transferul de date (GPRS). Durata unui ciclu de alocare a resurselor s-a notat $cntr_cycle$ și este egală cu 20ms.

Definim încărcarea rețelei ($Network_load$) ca fiind suma ponderilor utilizatorilor activi împărțită la numărul mediu de canale alocate traficului de date:

$$Network_load = \left(\sum_i k_i \cdot W_i \right) / B_{avr} \quad (5.10)$$

Din ecuațiile (5.9) și (5.10) se observă că timpul de transfer este proporțional cu încărcarea rețelei și deci, menținerea timpului de transfer a datelor în anumite limite devine echivalentă cu menținerea încărcării rețelei în anumite limite.

Principiul algoritmului de AC pe care îl propune această teză pornește de la cerințele de QoS ale utilizatorilor și se poate formula astfel:

1. se consideră clasa de QoS cu cerințele cele mai stringente
2. se exprimă cerințele în termeni de întârziere medie pentru transmiterea unui fișier de lungime dată
3. se deduce valoarea dorită pentru $Network_load$ ($Target_network_load$) conform formulelor (5.9) și (5.10)
4. se aplică un algoritm de reglare (fuzzy) având ca scop menținerea valorii încărcării rețelei cât mai apropiată de $Target_network_load$.

Considerând ca și resursă lățimea de bandă, experimentată în algoritmul de AC sub forma încărcării rețelei, cerințele algoritmului de AC din ecuația (5.8) se pot scrie:

$$\left(\sum_i k_i \cdot W_i\right)/B_{avr} + W_{new}/B_{avr} \leq Target_network_load \quad (5.11)$$

în care W_{new} este ponderea utilizatorului care dorește să fie admis.

Dacă inecuația (5.11) este îndeplinită, utilizatorul este admis, în caz contrar este respins.

Din ecuația (5.9) se observă că factorul $1/W_i$ asigură diferențierea între clasele de utilizatori.

Referitor la limitele în care încărcarea rețelei poate varia în jurul valorii $Target_network_load$, ele nu trebuie să fie extrem de stricte, în sensul că se dorește pe cât posibil ca încărcarea rețelei să fie mai mică decât încărcarea țintă, dar, pe perioade scurte de timp, se permite totuși ca încărcarea să depășească încărcarea țintă. Aceasta deoarece majoritatea aplicațiilor tolerează o fluctuație a întârzierilor: aplicațiile care sunt sensibile la capacitatea de trecere nu au nici o problemă în a accepta creșteri ale întârzierii pe anumite perioade de timp, dar chiar și aplicațiile foarte sensibile la întârzieri, cum sunt cele de tip streaming, acceptă o anumită rată de pierderi a pachetelor de date, pierderi care ar putea fi cauzate prin întârzieri mai mari în recepționarea unor pachete de date.

Pentru menținerea valorii încărcării rețelei aproape de valoarea țintă, în această teză se folosește un controler fuzzy (*FLC – Fuzzy Logic Controller*). Intrarea în controler este încărcarea rețelei, inclusiv componenta cu care contribuie utilizatorul care dorește să fie admis, iar ieșirea controlerului este decizia de a admite sau nu utilizatorul respectiv. Scopul controlerului este să mențină mărimea reglată (adică încărcarea rețelei) cât mai aproape de valoarea dorită.

5.3.3 Implementarea algoritmului fuzzy de admitere a utilizatorilor

Structura controlerului fuzzy utilizat a fost prezentată în secțiunea 5.2.2. Se folosește un FLC cu două intrări și o ieșire, intrările fiind încărcarea rețelei (*network load*) și precedența utilizatorilor (*users precedence*), iar ieșirea va fi decizia de acceptare (*admission decision*). Regulile folosite de controllerul fuzzy vor fi de genul “**if** *network load* is high AND *user’s precedence* is low **then** *admission decision* is strong reject”.

Valoarea încărcării rețelei care se introduce în FLC conține atât încărcarea curentă a rețelei, cât și contribuția utilizatorului care dorește să fie admis.

Precedența utilizatorului este o valoare convențională, alocată de operatorul de rețea fiecărui utilizator. Această valoare ține cont atât de clasa de QoS a utilizatorului, cât și de faptul dacă un utilizator vine dintr-o altă celulă, având deci o sesiune în curs

de desfășurare, sau își începe sesiunea în celula curentă. Deoarece se dorește să se evite întreruperea sesiunilor în curs de desfășurare, utilizatorii care vin din alte celule vor primi o valoare mare a precedenței. De asemenea, valoarea precedenței este mai mare pentru utilizatorii din clasa *premium* decât a celor din clasa *standard*, care la rândul său este mai mare decât a utilizatorilor din clasa *economic*, dacă se consideră clasele de QoS în funcție de subscripția utilizatorilor.

Termenii variabilelor lingvistice *network load* și *users precedence* din premize sunt *L* – *Low* (mica), *M* – *Medium* (medie) și *H* – *High* (mare), iar termenii variabilei lingvistice *admission decision* din concluzie sunt *strong reject* (*SR*), *weak reject* (*WR*), *weak admit* (*WA*) și *strong admit* (*SA*). Pentru a simplifica implementarea, se adoptă funcții de apartenență liniare (triunghiuri sau trapeze) pentru termenii din premize, iar termenii din concluzii sunt valori singulare (*singletons*). Figura 5.7 prezintă termenii variabilelor lingvistice din premize, iar figura 5.8 conține termenii variabilei lingvistice din concluzia regulilor.

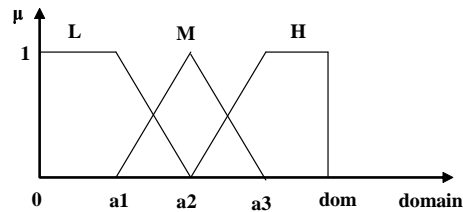


Figura 5.7: Termenii variabilelor lingvistice *network load* și *users precedence* din premisele FLC

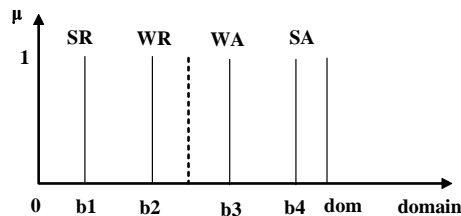


Figura 5.8: Termenii variabilei lingvistice *admission decision* din concluzie

În cele două figuri a_1 , a_2 , a_3 , b_1 , b_2 , b_3 , b_4 sunt parametri, ale căror valori pot fi ajustate. Domeniile de definiție (universul discursului) pentru variabilele lingvistice din premize și din concluzii sunt mapate în cadrul FLC pe intervalul de numere întregi $[0, dom]$ folosind o funcție liniară.

Algoritmul de AC implementat de FLC va mapa valoarea țintă a încărcării rețelei (*network load*) la mijlocul intervalului $[0, dom]$ și va încerca să mențină încărcarea rețelei în jurul acestei valori. Valoarea efectivă a încărcării rețelei este de asemenea mapată în intervalul de numere întregi $[0, dom]$. La cealaltă intrare a FLC va fi valoarea precedenței utilizatorului care, fiind o valoare convențională, atribuită unui utilizator de către operatorul de rețea poate primi valori direct în intervalul dorit $[0, dom]$, astfel încât nu mai este necesar să se apeleze funcția de mapare.

Dacă ieșirea FLC este mai mică decât $dom/2$, utilizatorul va fi respins, iar dacă este mai mare decât $dom/2$, atunci utilizatorul este admis.

Setul de reguli fuzzy utilizat este descris de tabelul 5.1:

		Precedența utilizatorilor		
		L	M	H
Network load	L	WA	SA	SA
	M	WR	WA	SA
	H	SR	WR	WA

Tabelul 5.1: Setul de reguli fuzzy

Dacă încărcarea rețelei este mică (L), atunci sunt admiși toți utilizatorii, indiferent de precedența lor. Dacă încărcarea rețelei este medie (M), atunci sunt respinși utilizatorii cu precedență scăzută (L) și admiși cei cu precedență ridicată (H). În cazul utilizatorilor cu precedență medie (M), decizia ar putea fi atât de admitere la limită (WA), cât și de respingere la limită. Aici s-a optat pentru admiterea la limită a acestei categorii de utilizatori. Dacă încărcarea rețelei este mare (H), atunci ar trebui respinși toți utilizatorii, dar, pentru a asigura condiții mai bune utilizatorilor din clasele de QoS cu cerințe ridicate și mai ales pentru a reduce (chiar până la zero) probabilitatea de întrerupere a sesiunilor în curs de desfășurare, se admit utilizatorii care au precedență ridicată (H), iar ceilalți vor fi respinși.

Prin maparea valorii țintă a încărcării la mijlocul intervalului $[0, dom]$ și datorită faptului că termenul *M* al variabilei lingvistice *network load* este centrat în jurul valorii de la mijlocul domeniului $[0, dom]$, controlerul fuzzy va încerca să mențină încărcarea rețelei în jurul valorii medii (M).

În funcție de politica pe care dorește să o adopte, operatorul de rețea poate să schimbe în mod flexibil setul de reguli, favorizând mai mult sau mai puțin utilizatorii care vin din alte celule, sau pe cei care au cerințe ridicate de QoS, mai ales în condițiile unei încărcări ridicate a rețelei, sau poate chiar să adopte o politică mai conservatoare de alocare a resurselor, respingând toți utilizatorii dacă încărcarea rețelei este prea

mare.

Prin faptul că se ține cont și de precedența utilizatorilor se complică problema menținerii încărcării dorite a rețelei, deoarece apar situații în care unii utilizatori trebuie să fie admisi chiar dacă încărcarea rețelei este mare. Din acest motiv problema a fost abordată în două etape:

1. la decizia de admitere a utilizatorilor s-a ținut cont doar de încărcarea rețelei (FLC cu o intrare). În acest fel problema admiterii utilizatorilor se transformă într-o problemă obișnuită de reglare, în buclă închisă, fără influența precedenței utilizatorilor, care nu poate fi inclusă în bucla de reglare. Rezultatele acestui caz sunt prezentate în 5.3.4. Setul de reguli fuzzy va fi de asemenea modificat (simplificat).
2. decizia de admitere se face pe baza regulilor din tabelul 5.1 (FLC cu două intrări). Acest caz este prezentat în 5.3.5.

5.3.4 Rezultate în cazul controlerului fuzzy cu o intrare

În condițiile în care admiterea utilizatorilor se face doar în funcție de încărcarea rețelei, regulile fuzzy din tabelul 5.1 se transformă în următorul set de reguli:

1. if network load is H then admission decision is SR.
2. if network load is M then admission decision is WA.
3. if network load is L then admission decision is SA.

Pentru a nu admite un număr prea mare de utilizatori, termenii variabilei lingvistice *admission decision* au fost dispuși în mod asimetric față de mijlocul intervalului $[0, dom]$. Astfel, parametrii din figurile 5.7 și 5.8 au valorile: $dom = 63$, $a1 = dom/4$, $a2 = dom/2$, $a3 = 3 \cdot dom/4$, $b1 = 8$, $b2 = 16$, $b3 = 40$ și $b4 = 48$. În aceste simulări s-a considerat că toți utilizatorii au aceeași schemă de codare (CS1) și că toate cele 8 canale radio sunt alocate traficului de date (adică $B_{avr} = 8$). Fiecare utilizator generează 5 fișiere, fiecare fișier având lungimea de 3620 biți (20 de blocuri de date cu CS1). Utilizatorii aparțin următoarelor trei clase de QoS: QoS 3 (10% din utilizatori), QoS 2 (80% din utilizatori) QoS 1 (restul de 10%). Ponderile utilizatorilor sunt $W = 8$ pentru QoS 3, $W = 4$ pentru QoS 2 și $W = 1$ pentru QoS 1.

Procesul de generare a utilizatorilor este periodic, nu aleator ca în realitate, iar perioada de generare a fost denumită un pas (*step*). S-a făcut această simplificare pentru a permite o mai bună analiză a algoritmului de AC.

În primii 20 de pași, care au loc la timpul de simulare 0, se generează încărcarea oferită inițial rețelei (*initial offered load*) prin admiterea automată a tuturor acestor 20 de utilizatori. FLC va începe practic să funcționeze după timpul de simulare 0. Perioada de generare a utilizatorilor variază între 130ms și 230ms și se consideră de asemenea valori diferite pentru *Target_network_load*.

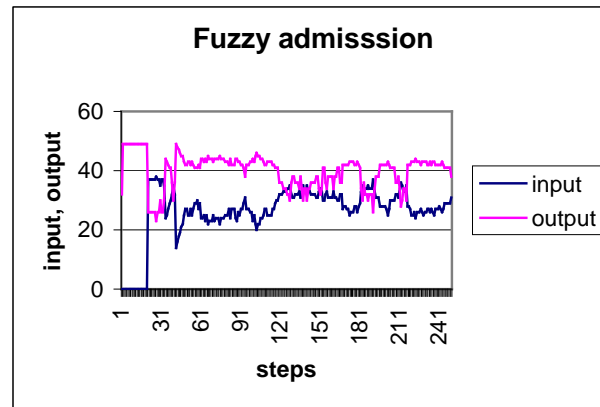


Figura 5.9: Intrarea și ieșirea FLC pentru o încărcare țintă având valoarea 10 și pentru un interval de generare a utilizatorilor de 230ms.

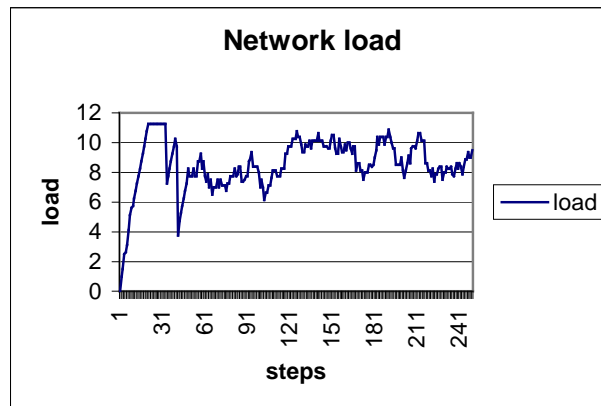


Figura 5.10: Încărcarea rețelei pentru o încărcare țintă având valoarea 10 și pentru un interval de generare a utilizatorilor de 230ms

Figura 5.9 conține reprezentarea ieșirii FLC în funcție de numărul de pași atunci când încărcarea țintă a rețelei are valoarea 10 și când intervalul de generare a utilizatorilor are valoarea de 230ms. După generarea încărcării inițiale (de 20 de utilizatori),

încărcarea rețelei crește și utilizatorii care încearcă să se conecteze în această perioadă vor fi respinși. Atunci când o parte din cei 20 de utilizatori creați la începutul simulării își încheie sesiunile de transfer de date, încărcarea rețelei scade și majoritatea utilizatorilor care vor încerca să se conecteze vor fi admiși.

În figura 5.10 se poate observa că încărcarea rețelei este menținută în majoritatea timpului aproape de valoarea 10 și că valoarea încărcării nu este niciodată mai mare decât 11, cu excepția primilor 20 de pași. Acest lucru înseamnă că algoritmul de AC implementat de FLC funcționează cu succes.

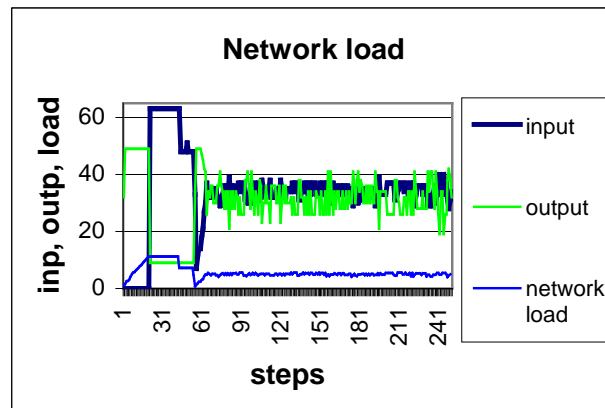


Figura 5.11: Intrarea și ieșirea FLC și încărcarea rețelei pentru o încărcare țintă având valoarea 5 și o perioadă de generare a utilizatorilor de 130ms.

Figura 5.11 prezintă o situație mult mai solicitantă pentru FLC, când încărcarea țintă are valoarea 10 și perioada de generare a utilizatorilor are valoarea de doar 130ms.

După procesarea încărcării inițiale și scăderea încărcării rețelei, ieșirea FLC oscilează în jurul valorii ce desemnează granița dintre acceptarea și respingerea unui utilizator, ceea ce înseamnă că încercările de conectare ale utilizatorilor sunt prea frecvente, comparativ cu durata sesiunilor de transfer de date, iar algoritmul de AC va fi obligat să respingă aproximativ jumătate din încercările de conectare.

Se poate observa în acest caz că încărcarea inițială a rețelei este mai mare de 10, adică mai mult decât dublul valorii țintă, dar FLC reușește să mențină încărcarea rețelei aproape de valoarea dorită chiar și în aceste condiții foarte dificile. De fapt, încărcarea rețelei nu depășește valoarea 5.5 după pasul 55 (când încărcarea inițială a fost redusă ca rezultat al încheierii sesiunilor de date a cel puțin o parte din utilizatorii generați inițial).

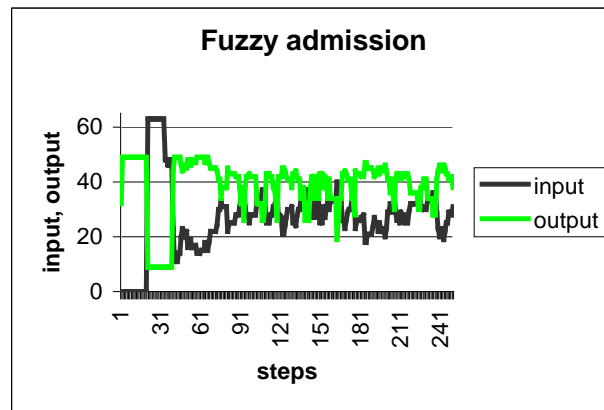


Figura 5.12: Intrarea și ieșirea FLC pentru o valoare țintă a încărcării rețelei egală cu 5 și o perioadă de generare a utilizatorilor de 230ms.

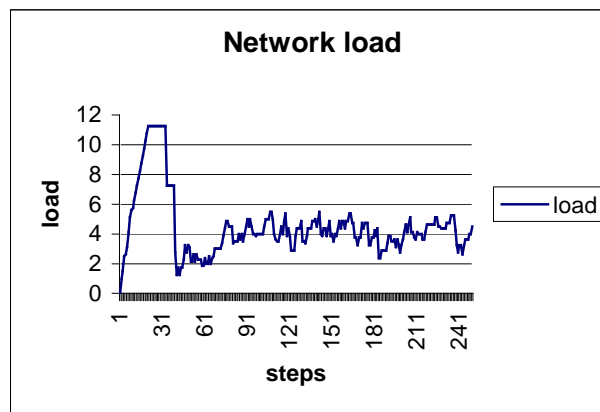


Figura 5.13: Încărcarea rețelei pentru o valoare țintă a încărcării rețelei egală cu 5 și o perioadă de generare a utilizatorilor de 230ms.

Figurile 5.12 și 5.13 prezintă intrarea și ieșirea FLC, respectiv încărcarea rețelei, pentru situația când încărcarea țintă are valoarea 5 și perioada de generare a utilizatorilor are valoarea 230ms. Din nou, după primii 20 de pași, încărcarea rețelei are o valoare mai mare decât 10 dar, după ce se procesează datele utilizatorilor generați inițial, se constată că FLC menține încărcarea rețelei aproape de valoarea dorită, astfel încât după pasul 40 valoarea încărcării nu va depăși 5.5.

Aceste experimente au demonstrat că, în cazul utilizării controlerului fuzzy în buclă de reglare închisă, performanțele sale sunt foarte bune, permițând continuarea experimentelor pentru cazul când se ține cont și de precedența utilizatorilor.

5.3.5 Rezultate în cazul controlerului fuzzy cu două intrari

Rezultatele din această secțiune demonstrează capacitatea algoritmului fuzzy propus de această teză de a menține încărcarea rețelei aproape de o valoare dorită (valoare țintă) și, mai important, de a menține valorile întârzierilor datelor utilizatorilor aproape de valoarea țintă a fiecărei clase de QoS. De asemenea, rezultatele furnizate de algoritmul fuzzy de AC vor fi comparate cu rezultatele unui algoritm de AC care nu utilizează logica fuzzy și care se bazează pe valori de prag fixe. Algoritmii vor fi comparați prin prisma timpilor de transfer ai fișierelor utilizatorilor și a probabilităților de blocare, respectiv de respingere a apelurilor (*call blocking probability* și respectiv *call dropping probability*).

Parametrii simulării

Intervalul de generare a utilizatorilor (nodurilor mobile – MN) este o variabilă aleatoare având distribuție exponențială. Prin modificarea valorii sale medii se schimbă rata de generare a utilizatorilor, deci și *încărcarea oferită* rețelei (*offered load*). Nodurile mobile aparțin la trei clase de QoS, în funcție de subscripția lor: clasa QoS 3 corespunde utilizatorilor premium, clasa QoS 2 celor standard, iar clasa QoS 1 conține utilizatorii de tip economic. Algoritmul utilizat pentru alocarea resurselor la controlul transmiterii datelor (TC) este *Weighted Round Robin (WRR)*, iar ponderile utilizate de către WRR le sunt alocate astfel nodurilor mobile: pondere 1 pentru MN din clasa QoS 1, pondere 2 pentru clasa QoS 2 și pondere 4 pentru MN din clasa premium (QoS 3). Dintre nodurile mobile generate, 10% aparțin clasei QoS 1, 80% clasei QoS 2 și 10% clasei QoS 3.

În aceste simulări, pentru a permite o mai bună comparare a rezultatelor, toate nodurile mobile generează trafic având aceleași caracteristici: fiecare MN generează 5 fișiere, fiecare fișier fiind compus din 5 blocuri de date (adică având o lungime de $5 \cdot 181$ biți cu schema de codare CS1). Se generează un nou fișier doar după ce precedentul fișier a fost transferat cu succes.

Nodul *stat* înregistrează întârzierile la transferul datelor pentru fiecare fișier și apoi determină valoarea medie a întârzierii pentru fiecare MN.

Toate cele 8 canale radio se alocă pentru traficul de date, ceea ce înseamnă că în ecuațiile (5.9) și (5.10), $B_{avr} = 8$.

Funcția de precedență aleasă pentru aceste simulări asignează o valoare foarte mare nodurilor mobile care vin din alte celule (*handoff MN*), astfel încât aceste noduri mobile să fie admise, chiar dacă încărcarea rețelei este foarte mare. În acest fel la algoritmul de AC pe care îl propune această teză, probabilitatea de întrerupere a

apelurilor (*call dropping probability*) va fi zero pentru toate clasele de QoS.

Valoarea precedenței este mică spre medie la clasa QoS 1, medie la MN din clasa QoS 2 și medie spre mare la utilizatorii din clasa QoS 3.

Procentajul nodurilor mobile care vin din alte celule este un parametru important în aceste simulări deoarece putem crește încărcarea rețelei fie prin creșterea procentajului nodurilor mobile care vin din alte celule (*handoff MN*) menținând constantă rata de generare a MN, fie prin modificarea (creșterea) ratei de generare la un procentaj fix de noduri mobile de tip *handoff*.

În aceste experimente procentajul de noduri mobile *handoff* este menținut constant (25%), iar valoarea medie a intervalului de generare de noduri mobile este modificată între 130ms și 330ms. Dacă valoarea medie a intervalului de generare este de 100ms sau mai mică, atunci rețeaua devine instabilă, iar dacă valoarea medie a intervalului de generare de utilizatori este mai mare de 330ms, atunci probabilitatea de blocare a apelurilor va fi zero. Simularea se oprește după ce s-au generat 4000 de MN.

Parametrii din figura 5.7 au valorile: $dom = 63$, $a1 = dom/4$, $a2 = dom/2$, $a3 = 3dom/4$, iar parametrii $b1$, $b2$, $b3$ și $b4$ din figura 5.8 au valori diferite în funcție de scenariul de simulare.

Rezultate privind capacitatea de reglare a FLC cu două intrări

În acest scenariu de simulare ne propunem să menținem valoarea medie a întârzierii fișierelor utilizatorilor din clasa QoS 3 în jurul valorii de 500ms. Înlocuind în ecuația (5.9) $sending_delay_j = 500ms$, se va obține o valoare țintă pentru încărcarea rețelei egală cu 5, definiția încărcării rețelei (*network load*) fiind dată în ecuația (5.10).

În figurile următoare, pe abscisă este reprezentat timpul de simulare, în milisekunde.

Figura 5.14 prezintă comportarea algoritmului fuzzy de admitere a utilizatorilor pentru cazul când încărcarea oferită rețelei (*offered load*) este foarte mare, adică pentru o perioadă medie de generare a utilizatorilor de 180ms. Valorile pentru mediile întârzierilor fișierelor prezentate în figura 5.14 demonstrează foarte clar diferențierea calității serviciilor între utilizatori, valorile întârzierilor fiind de aproximativ 500ms pentru utilizatorii din clasa premium (QoS 3), de aproximativ 1000ms pentru utilizatorii standard (QoS 2) și în jur de 2000ms pentru utilizatorii din clasa QoS 1. Valoarea medie a întârzierilor pentru utilizatorii din clasa QoS 3 este de 534ms, adică doar cu 7% mai mare decât valoarea țintă, de 500ms.

Pentru aceleași condiții de simulare, figura 5.15 prezintă încărcarea rețelei, care oscilează în jurul valorii țintă 5. Deși amplitudinea oscilațiilor este destul de mare, în final valoarea medie a întârzierilor pentru utilizatorii din clasa QoS 3 este totuși

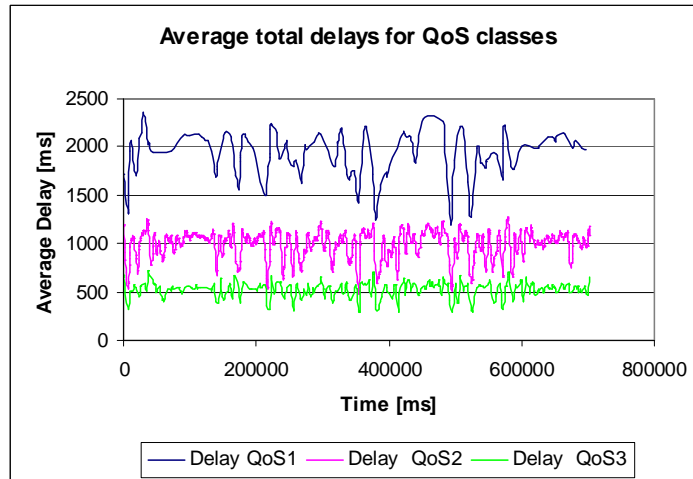


Figura 5.14: Media întârzierilor totale pentru cele 3 clase de utilizatori când valorile ponderilor sunt $W=\{4,2,1\}$ și perioada de generare a utilizatorilor este de 180ms, algoritm fuzzy de AC.

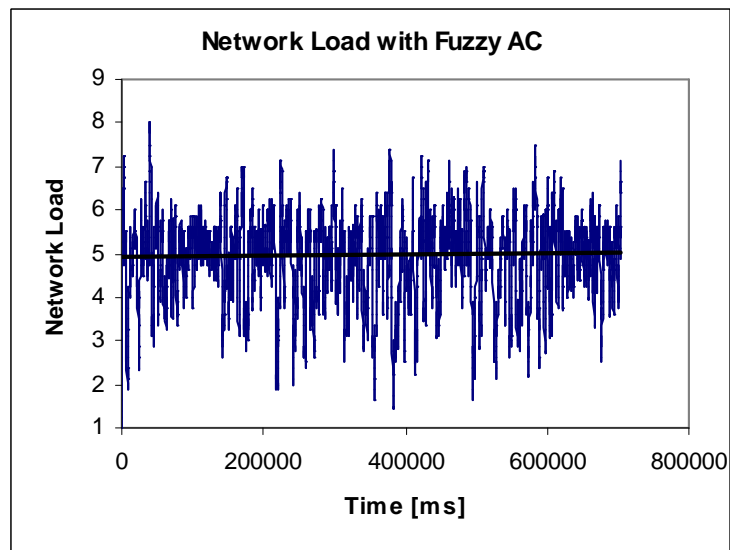


Figura 5.15: Încărcarea rețelei, când valorile ponderilor sunt $W=\{4,2,1\}$ și perioada de generare a utilizatorilor este de 180ms, cu algoritm fuzzy de AC.

destul de apropiată de valoarea dorită.

Aceste rezultate pot fi îmbunătățite dacă se utilizează un algoritm de adaptare a lățimii de bandă (*Bandwidth Adaptation Algorithm – BAA*) similar celui descris în [53], mai ales în perioadele când încărcarea rețelei este mai mare decât cea dorită. Un

astfel de algoritm va mări temporar cantitatea de resurse (lățimea de bandă) alocate utilizatorilor cu prioritate ridicată (celor din clasa QoS 3) și va reduce cantitatea de resurse pentru utilizatorii din clasele QoS 2 și QoS 1. Cantitatea de resurse (lățimea de bandă) este reprezentată în acest caz de ponderea avută de utilizatorii respectivei clase de QoS în algoritmul WRR.

5.3.6 Compararea performanțelor algoritmului fuzzy de admitere a utilizatorilor cu algoritmi bazați pe valori de prag

Am comparat performanțele algoritmului fuzzy de AC cu cele ale unui algoritm non-fuzzy, bazat pe valori de prag (*thresholds*). S-au efectuat simulări pentru diverse scenarii de lucru, iar rezultatele sunt prezentate în cele ce urmează.

Comparare prin prisma întârzierilor utilizatorilor

Figura 5.16 prezintă întârzierile pentru clasa 3 de calitate a serviciului atunci când algoritmul de admitere a utilizatorilor este fuzzy (linia mai închisă la culoare), prin comparație cu întârzierile pentru aceeași clasă de utilizatori când algoritmul de AC este non-fuzzy.

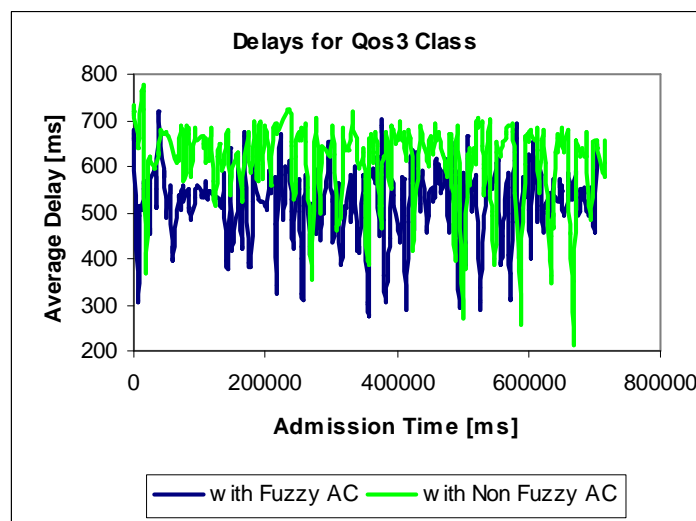


Figura 5.16: Media întârzierilor totale pentru clasa de utilizatori QoS 3, când valorile ponderilor sunt $W=\{4,2,1\}$ și perioada de generare a utilizatorilor este de 180ms, cu algoritm de AC fuzzy și non-fuzzy.

Condițiile în care s-au efectuat simulările sunt: parametrii din figura 5.8 au valorile

$b_1 = 8$, $b_2 = 16$, $b_3 = 40$ și $b_4 = 48$, iar la algoritmul non-fuzzy de AC, pragurile care separă domeniile încărcării rețelei au valorile: 16, ca prag între L (*Low*) și M (*Medium*), și 44 ca prag între M și H (*High*), adică pragurile de separare a termenilor non-fuzzy L , M , și H sunt în punctele a_1 și a_3 (a se vedea figura 5.7).

De asemenea probabilitatea de blocare pentru utilizatorii premium (QoS 3) este mult mai mare pentru algoritmul de AC non-fuzzy (0.25 pentru non-fuzzy față de doar 0.03 pentru fuzzy AC).

Superioritatea algoritmului fuzzy de AC este clar demonstrată prin valorile mai mici ale întârzierilor utilizatorilor premium și prin valorile mai bune ale probabilității de blocare a acestor utilizatori. Probabilitatea de întrerupere a sesiunilor în curs de desfășurare este 0 pentru toate clasele de QoS.

Compararea probabilităților de blocare pentru valori asimetrice ale termenilor ieșirii

O altă posibilă alegere pentru valorile pragurilor la algoritmul non-fuzzy o reprezintă punctele de la intersecția mulțimilor fuzzy *low* și *medium*, respectiv a mulțimilor *medium* și *high* (a se vedea figura 5.7). În acest caz valorile de prag sunt 22 și respectiv 38. După cum era de așteptat, comportarea algoritmului non-fuzzy de AC este în acest caz foarte apropiată de comportarea algoritmului fuzzy de AC, obținându-se valori similare pentru întârzierile fișierelor, dar probabilitatea de blocare pentru utilizatori diferă între cele doi algoritmi (fuzzy și non-fuzzy).

Pentru valori asimetrice ale termenilor variabilei lingvistice de ieșire *admission decision*, adică $b_1 = 8$, $b_2 = 16$, $b_3 = 40$ și $b_4 = 48$, pentru probabilitățile de blocare ale convorbirilor (*call blocking probability*) se obțin valorile din figura 5.17. Cu FZ-1, FZ-2 și FZ-3 s-au notat probabilitățile de blocare pentru utilizatorii din clasele QoS 1, QoS 2 și QoS 3, cu algoritm fuzzy de AC, iar cu NFZ-1, NFZ-2 și NFZ-3 s-au notat probabilitățile de blocare ale apelurilor pentru clasele QoS 1, QoS 2 și respectiv QoS-3 în cazul algoritmului non-fuzzy. Probabilitățile de blocare sunt reprezentate în funcție de valoarea medie a perioadei de generare a utilizatorilor (MN). Din figură se observă că pentru clasele QoS 1 și QoS 2 se obțin valori apropiate pentru cei doi algoritmi, dar algoritmul fuzzy are valori mult mai mici pentru probabilitatea de blocare a convorbirilor la utilizatorii din clasa QoS 3.

Acest lucru se observă mai bine în figura 5.18, care prezintă un detaliu conținând doar valorile probabilității de blocare pentru utilizatorii din clasa QoS 3, în cazul celor doi algoritmi (fuzzy și non-fuzzy).

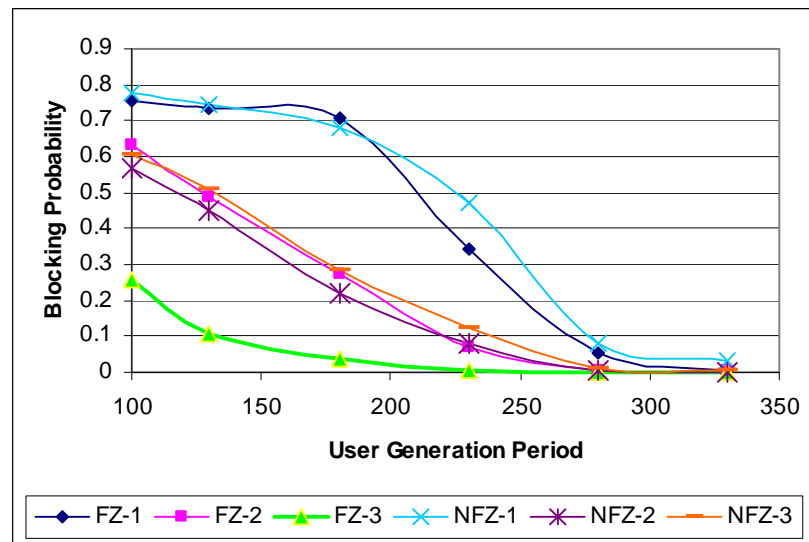


Figura 5.17: Probabilitățile de blocare a apelurilor pentru toate clasele de QoS, pentru valori asimetrice ale ieșirilor.

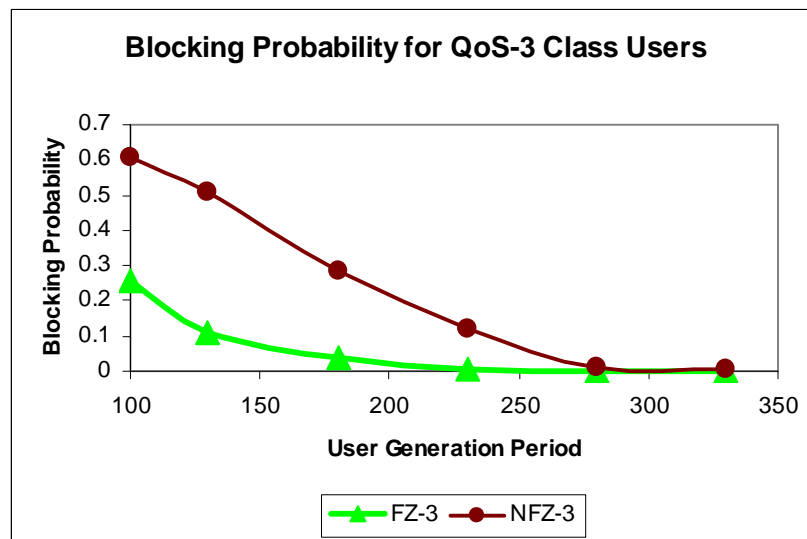


Figura 5.18: Probabilitățile de blocare a apelurilor pentru clasa QoS 3, cu algoritmi de AC fuzzy și non-fuzzy, valori asimetrice pentru termenii variabilei de ieșire.

Compararea probabilităților de blocare pentru valori simetrice ale termenilor ieșirii

O altă situație este atunci când termenii fuzzy din variabila lingvistică de ieșire *admission control* au valori simetrice față de mijlocul intervalului $[0, dom]$, și anume:

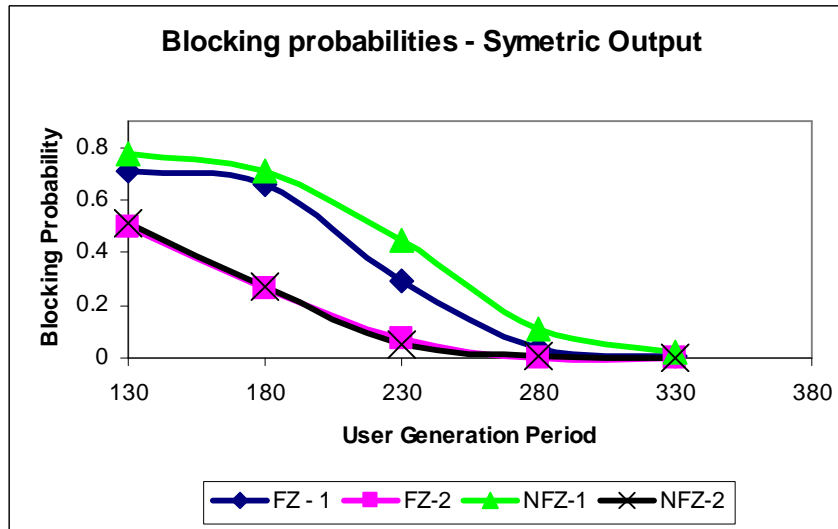


Figura 5.19: Probabilitățile de blocare a apelurilor pentru clasele QoS 1 și QoS 2, cu algoritm de AC fuzzy si non-fuzzy, valori simetrice pentru termenii variabilei de ieșire.

$b_1 = 8$, $b_2 = 24$, $b_3 = 40$ și $b_4 = 56$, valorile de prag ale încărcării rețelei rămânând neschimbate față de scenariul anterior (adică 22 și 38). În acest caz, așa cum se poate observa și din figura 5.19, valorile pentru probabilitățile de blocare la utilizatorii din clasele QoS 1 și QoS 2 sunt foarte apropiate la cei doi algoritmi. E interesant de menționat că în acest caz valorile probabilităților de blocare pentru utilizatorii din clasa QoS 3 sunt 0 la ambii algoritmi.

În toate cazurile prezentate, distincția între algoritmul fuzzy de AC și algoritmul bazat pe valori de prag se face doar pe baza încărcării rețelei, precedența utilizatorilor fiind tratată la fel atât în cazul fuzzy, cât și în cel al algoritmului cu valori de prag.

5.4 Concluzii

Acest capitol prezintă o soluție originală de control a aditerii utilizatorilor într-o rețea GPRS/EGPRS. Această soluție pornește de la identificarea principalelor probleme ale algoritmilor de AC utilizați în rețele celulare de date și apoi formulează cerințele algoritmilor de AC pentru aceste tipuri de rețele. Pentru (E)GPRS este propus un model matematic al cerințelor algoritmilor de AC, care este apoi transpus într-o problemă de reglare. Soluția de reglare propusă se bazează pe logica fuzzy și utilizează un controler fuzzy pentru implementarea algoritmului fuzzy de AC.

De asemenea, se prezintă un studiu al performanțelor controlerelor fuzzy în procese complexe de reglare, studiu care este apoi folosit pentru implementarea controlerului fuzzy folosit la algoritmul de admitere a utilizatorilor.

Am demonstrat prin simulare eficiența algoritmului de AC bazat pe logică fuzzy propus de această teză, chiar și în cazul când rata de generare a nodurilor mobile este foarte ridicată. Algoritmul pe care îl propune această teză este capabil să satisfacă cerințele de QoS (în termeni de întârzieri medii ale fișierelor), prin menținerea acestor întârzieri în jurul unei valori dorite. De asemenea, prin combinarea algoritmului de AC cu algoritmi de planificare utilizați la *transmission control*, se asigură și diferențierea calității serviciului utilizatorilor.

Probabilitatea de întrerupere a convorbirilor (sesiunilor), adică *call dropping probability* este zero, iar probabilitatea de blocare a apelurilor (*call blocking probability*) a algoritmului fuzzy de AC este foarte scăzută (aproximativ 3%) în cazul utilizatorilor din clasa de QoS *premium*.

Algoritmul utilizează logica fuzzy pentru a extinde algoritmi de AC care folosesc politici diferite de admitere a utilizatorilor în diferite regiuni (ale încărcării rețelei în acest caz). Utilizarea logicii fuzzy elimină necesitatea de a determina valori precise ale pragurilor care separă aceste regiuni.

Rezultatele pe care le-am obținut arată că algoritmul de AC bazat pe logică fuzzy se comportă mai bine decât algoritmi non-fuzzy, bazați pe praguri.

Este foarte important de menționat că algoritmul fuzzy de AC propus în această teză asigură o flexibilitate mult mai mare decât a algoritmilor bazați pe valori de prag propuși în literatură. Valorile de prag sunt valabile doar pentru valorile parametrilor rețelei pentru care au fost calculate și vor trebui recalulate la modificarea condițiilor din rețea, pe când algoritmul fuzzy de AC asigură din start un grad ridicat de toleranță la modificări ale condițiilor de lucru. În plus, prin utilizarea variabilelor lingvistice și a inferenței fuzzy, operatorul de rețea are la dispoziție o manieră directă de modificare a regulilor fuzzy, a termenilor variabilelor lingvistice și a valorilor asociate acestor termeni, astfel încât algoritmul fuzzy va putea fi adaptat cu ușurință la schimbări intervenite în sistem sau la dorința operatorului de a schimba politica de admitere a utilizatorilor.

Ca și dezvoltări viitoare, va fi testat algoritmul în scenarii de simulare mai complexe și se va aprofunda studiul influenței precedentei alocate utilizatorilor în cadrul algoritmului de AC.

Contribuțiile științifice ale acestui capitol pot fi rezumate astfel:

- Identificarea problemelor algoritmilor de AC utilizați în rețelele celulare de date.

- Formularea matematică a cerințelor unui algoritm de AC pentru rețele (E)GPRS astfel încât problema admiterii utilizatorilor să poată fi abordată ca o problemă de reglare.
- Propunerea unui algoritm original de AC pentru rețele de tip GPRS și EGPRS, algoritm bazat pe logica fuzzy. Acest algoritm a fost publicat în [102], [106], [108].
- Studiul performanțelor controlerelor fuzzy în aplicații complexe de reglare. Rezultatele acestor studii au fost publicate în [104], [10].
- Utilizarea rezultatelor acestui studiu pentru implementarea unui controler fuzzy performant utilizat la admiterea utilizatorilor și integrarea controlerului fuzzy în modelul de simulare dezvoltat în această teză.

Capitolul 6

Concluzii. Rezumatul contribuțiilor tezei

Concluzia principală a acestei teze este că pentru asigurarea calității diferențiate a serviciului utilizatorilor unei rețele de tip GPRS/EGPRS sunt necesare abordări unitare ale tuturor aspectelor de alocare a resurselor: o strategie de admitere a utilizatorilor corelată cu politicile de planificare a transmiterii datelor.

În această teză s-au investigat ambele aspecte ale alocării resurselor. S-au studiat performanțele algoritmilor de planificare a resurselor la transmiterea datelor, algoritmi utilizați la nivel MAC/RLC (Medium Access Control / Radio Link Control) și s-a propus o soluție originală de control a admiterii utilizatorilor. Prin aceste două mecanisme combinate se poate asigura calitatea dorită a serviciului pentru utilizatorii rețelei: întârzieri diferențiate pe clase de utilizatori, controlul congestiilor, reducerea până la zero a probabilității de întrerupere a convorbirilor (*call dropping probability*) și o probabilitate mică de blocare a apelurilor (*call blocking probability*).

Pentru validarea metodelor propuse de alocare a resurselor în rețele (E)GPRS s-a propus și dezvoltat un model de simulare pornind de la cunoașterea modului cum se face alocarea resurselor în rețele GPRS reale.

Utilizatorii GPRS din model, la fel ca și cei reali, au posibilitatea de a genera date în mod interactiv sau în mod *streaming*. Ei aparțin unor clase diferite de calitate a serviciului (lucru modelat prin ponderea pe care o are fiecare utilizator), aceste clase de QoS putând fi bazate fie pe clasele de trafic, fie pe baza subscripției utilizatorului. De asemenea, în model sunt utilizate schemele de codare din GPRS sau schemele de modulare și codare din EGPRS. Se poate modela de asemenea procesul de retransmitere a blocurilor de date afectate de erori. Utilizatorii GPRS din model au posibilitatea de a se deplasa dintr-o celulă în alta pe parcursul sesiunii de transfer de

date.

Activarea unui context PDP în cazul unui utilizator real este realizată în model cu ajutorul algoritmilor de admitere a utilizatorilor. De asemenea, modelul conține, la fel ca și o celulă GPRS reală, un modul PCU (*Packet Control Unit*), responsabil cu alocarea resurselor rețelei la nivel MAC/RLC.

Nivelul de detaliere al modelului este astfel încât, pe de o parte, permite simularea unor scenarii complexe, în conformitate cu situația dintr-o rețea reală (așa cum s-a demonstrat în [43]), iar pe de altă parte se elimină detaliile inutile, permițând o abordare punctuală a problemelor de alocare a resurselor și o interpretare corectă a rezultatelor, care nu sunt influențate de interacțiuni nedorite cu mecanismele de control ale unor protocoale din stiva de protocoale GPRS.

În ceea ce privește problema controlului transmiterii datelor (*Transmission Control*), în urma unui studiu sistematic realizat prin simulare s-a demonstrat superioritatea algoritmilor care aproximează algoritmul idealizat Fluid Fair Queueing (și anume algoritmi Weighted Round Robin – WRR și Oldest Queue – OQ) față de algoritmi care alocă resurse pe baza cantității de date din cozile utilizatorilor.

Prin investigarea influenței pe care o au ponderile alocate utilizatorilor asupra calității serviciului primit de aceștia (calitatea serviciului fiind exprimată prin întârzierile totale ale fișierelor utilizatorilor) s-a demonstrat o metodă simplă de a asigura atât diferențierea calității serviciului pentru utilizatori, cât și reducerea congestiilor din rețea. Această metodă constă în asocierea ponderii utilizatorilor fie cu clasa lor de calitate a serviciului, în primul caz, fie cu calitatea legăturii radio, în al doilea caz.

Pentru problema admiterii utilizatorilor, în această teză a fost dezvoltată o soluție originală, bazată pe logica fuzzy. Această soluție elimină dezavantajele algoritmilor de admitere a utilizatorilor existenți în literatură, fiind o extensie, prin prisma logicii fuzzy, a algoritmilor de admitere a utilizatorilor bazați pe valori de prag ce separă regiuni în care se aplică politici diferite de admitere a utilizatorilor ([63], [89], [94], [53]). Spre deosebire de algoritmi care utilizează valori de prag, algoritmul propus în această teză oferă o deosebită flexibilitate și adaptabilitate, prin aceea că se elimină necesitatea calculării sau stabilirii unor valori de prag de fiecare dată când se schimbă condițiile din rețea, iar operatorul de rețea poate să modifice cu ușurința politicile de admitere a utilizatorilor.

Soluția de admitere a utilizatorilor propusă în această teză a fost publicată în lucrările [102], [106], [108]. Soluția fuzzy reușește să asigure atât valorile dorite ale întârzierilor fișierelor utilizatorilor, cât și o probabilitate de întrerupere a convorbir-

ilor egală cu zero și o probabilitate redusă de blocare a convorbirilor. La toți acești parametri metoda propusă este superioară algoritmilor non-fuzzy pe care i-am implementat.

O validare indirectă a importanței ideii de a aplica logica fuzzy la admiterea utilizatorilor este și faptul că Dini și Guglielmucci ([32]) au propus, concomitent și independent, o abordare bazată pe logica fuzzy pentru admiterea utilizatorilor în rețele mobile de tip WCDMA.

Rezumatul contribuțiilor tezei

Contribuțiile aceste teze pot fi grupate pe trei categorii:

1. Dezvoltarea unui model de simulare pentru alocarea resurselor în GPRS și EGPRS.
2. Studiul performanțelor algoritmilor utilizați la transferul datelor
3. Dezvoltarea unei soluții originale pentru problema admiterii utilizatorilor în rețele (E)GPRS, soluție bazată pe logica fuzzy.

Contribuțiile aduse prin dezvoltarea unui model de simulare pentru alocarea resurselor în GPRS și EGPRS pot fi detaliate astfel:

- Realizarea unui model de simulare care să permită studiul tehnicilor de alocare a resurselor în rețele de tip GPRS/EGPRS.
- Caracteristicile acestui model, comparativ cu modele similare prezentate în literatură sunt:
 - gradul de abstractizare al modelului, care permite pe de o parte simularea unor situații complexe, de tip real, iar pe de altă parte evită detaliile de implementare care nu se referă la problemele investigate.
 - permite studiul algoritmilor de alocare a resurselor la transferul datelor (TC), fără ca rezultatele privind performanțele algoritmilor să fie influențate de factori cum ar fi mecanismele de control ale protocoalelor TCP sau LLC, sau ale interferențelor complexe dintre protocoalele din stiva de protocoale (E)GPRS
 - permite de asemenea studiul performanțelor algoritmilor de admitere a utilizatorilor (AC).

- Modelul a fost validat prin compararea rezultatelor simulărilor cu rezultatele analitice obținute pentru algoritmul idealizat *Fluid Fair Queueing*, atunci când utilizatorii generează fluxuri continue de date.
- Modelul a fost proiectat și implementat în întregime de către autorul tezei.
- Descreri ale acestui model au fost publicate în lucrările [43] și [100].

Contribuțiile la studiul performanțelor algoritmilor utilizați la transferul datelor sunt:

- Un studiu sistematic prin simulare al performanței a două categorii de algoritmi de planificare a resurselor radio în rețele GPRS și EGPRS: o categorie de algoritmi ce aproximează algoritmul idealizat Fluid Fair Queueing, categorie reprezentată de algoritmi WRR și OQ și o categorie de algoritmi la care resursele se alocă pe baza cantității de date din cozile utilizatorilor (cu reprezentanții LQ, TQL și TFL). În urma simulărilor efectuate s-a arătat că algoritmi WRR și OQ sunt mai eficienți decât cei bazați pe cantitatea de date din cozile utilizatorilor. Rezultate legate de aceste aspecte au fost publicate în [110], [109], [111]. Aceste articole au fost citate în [71].
- Interpretarea și explicarea originală, pe un caz general, a rezultatului obținut prin simulare, adică a faptului că algoritmi bazați pe cantitatea de date din cozile utilizatorilor nu sunt eficienți pentru implementarea transferului datelor în rețele (E)GPRS. Această concluzie este originală și importantă în domeniul transferului datelor în rețele celulare, deoarece există autori care au utilizat algoritmi bazați pe cantitatea de date din cozile utilizatorilor. Acest rezultat a fost publicat în [110].
- Studiul influenței factorilor de pondere ai utilizatorilor, atunci când acești factori de pondere sunt asociați clasei de calitate a serviciului (QoS) a utilizatorilor ([110]).
- Studiul efectului factorilor de pondere, atunci când aceștia sunt asociați cu calitatea legăturii radio a utilizatorilor. Prin favorizarea utilizatorilor cu o legătură radio mai bună se pot reduce congestiile din rețea. Investigații pe această temă au fost prezentate în [109] și [111].
- Utilizarea unor combinații de metode statistice pentru a asigura un grad ridicat de încredere al rezultatelor obținute prin simulare.

Contribuțiile referitoare la dezvoltarea unei soluții originale pentru problema admiterii utilizatorilor în rețele (E)GPRS, soluție bazată pe logica fuzzy pot fi detaliate astfel:

- Identificarea problemelor algoritmilor de AC utilizați în rețelele celulare de date.
- Formularea matematică a cerințelor unui algoritm de AC pentru rețele (E)GPRS astfel încât problema admiterii utilizatorilor să poată fi abordată ca o problemă de reglare.
- Propunerea unui algoritm original de AC pentru rețele de tip GPRS și EGPRS, algoritm bazat pe logica fuzzy. Acest algoritm a fost publicat în [102], [106], [108].
- Studiul performanțelor controlerelor fuzzy în aplicații complexe de reglare. Rezultatele acestor studii au fost publicate în [104], [10].
- Utilizarea rezultatelor acestui studiu pentru implementarea unui controler fuzzy performant utilizat la admiterea utilizatorilor și integrarea controlerului fuzzy în modelul de simulare dezvoltat în această teză.

Bibliografie

- [1] Wessam Ajib and Philippe Godlewski. Service disciplines performance for best-effort policies in packet-switching wireless cellular networks. In *Proceedings of IEEE Spring Vehicular Technology Conference VTC00*, Tokyo, Japan, May 2000.
- [2] Wessam Ajib and Philippe Godlewski. Service disciplines performance for WWW traffic in GPRS systems. In *Proceedings of IEE 3G 2000 Mobile Communications Technologies*, 2000.
- [3] Wessam Ajib, Philippe Godlewski, and Loic Le Bris. Service disciplines performance for best-effort policies in WWW traffic over packet-switching wireless cellular networks. In *Proceedings of IEEE 3G*, San Francisco, USA, June 2000.
- [4] AlcatelUniversity. *GPRS QoS and Parameters*. Alcatel University, 2003.
- [5] AlcatelUniversity. *Mobile Networks. Introduction to the GPRS*. Alcatel University, 2003.
- [6] Hugo Araújo, José Costa, and Luis M. Correia. Analysis of a traffic model for GSM/GPRS. In *Proceedings of 3rd Conference on Telecommunications, ConfTele2001*, Figueira da Foz, Portugal, April 2001.
- [7] Giuseppe Ascia, Vicezno Catania, Giuseppe Ficili, Sergio Palazzo, and Daniela Panno. A VLSI fuzzy expert system for real time traffic control in ATM networks. *IEEE Transactions on Fuzzy Systems*, 5(1):20–31, February 1997.
- [8] Antonios F. Atlasis, Nikolaos H. Loukas, and Athanasios V. Vasilakos. The use of learning algorithms in ATM networks call admission control: a methodology. *Computer Networks*, 34:341–353, 2000.
- [9] Ana-Maria Bădulescu. Studiul performanțelor circuitelor fuzzy utilizând limbajul VHDL, Lucrare de diplomă, conducător științific Doru Todinca, Universitatea Politehnica Timișoara, Dep. Calculatoare, Facultatea de Automatică și Calculatoare, 1999.
- [10] Ana Maria Bădulescu, **Doru Todinca**, and Radu Emil Precup. Performance considerations of fuzzy logic controllers using VHDL. *Periodica Politehnica, Transactions on Automatic Control and Computer Science*, 44(58):179–186, 1999.

- [11] Andrea Baiocchi, Francesca Cuomo, and Cristina Martello. Optimizing the radio resource utilization of multiaccess systems with a traffic-transmission quality adaptive packet scheduling. *Computer Networks*, 38:225–246, 2002.
- [12] Christian Bettstetter, Hans-Jörg Vogel, and Jörg Ebersacher. GSM phase 2+ general packet radio service GPRS: Architecture, protocols, and air interface. *IEEE Communications Surveys*, 2(3), 1999.
- [13] J. Bhasker. *A VHDL Primer. Revised edition*. Prentice Hall PTR, 1995.
- [14] M. Bourgois, E. Vivier, M. Terré, N. Bret, and B. Ridard. Necessity of an emulator to validate new resources allocation protocols. In *Proceedings of IEEE Spring Vehicular Technology Conference VTC 2001*, Rhodos, Greece, 2001.
- [15] Götz Brasche and Bernhard Walke. Concepts, services, and protocols of new GSM phase 2+ general packet radio service. *IEEE Communications Magazine*, 35(8):94–104, August 1997.
- [16] Jain Cai, Li Fung Chang, Kapil Chawla, and Xioxin Qiu. Providing differentiated services in EGPRS through packet scheduling. In *Proceedings of IEEE Globecom'00*, November 2000.
- [17] Jian Cai and David J. Goodman. General packet radio service in GSM. *IEEE Communications Magazine*, 35(10):122–131, October 1997.
- [18] Vincenzo Catania, Giuseppe Ficili, and Daniela Panno. A fuzzy logic based approach to multipriority control in ATM networks. *Computer Standards & Interfaces*, 21:19–32, 1999.
- [19] Vincenzo Catania, Giuseppe Ficili, Sergio Palazzo, and Daniela Panno. A comparative analysis of fuzzy versus conventional policing mechanisms for ATM networks. *IEEE/ACM Transactions on Networking*, 4(3):449–459, June 1996.
- [20] Vincenzo Catania, Giuseppe Ficili, and Daniela Panno. Using fuzzy logic in ATM source traffic control: lessons and perspectives. *IEEE Communications Magazine*, pages 70–81, November 1996.
- [21] Vincenzo Catania, Giuseppe Ficili, and Daniela Panno. On the impact of traffic control algorithms on resource management in ATM networks. *Computer Communications*, 22:258–265, 1999.
- [22] P.M.L. Chan, R.E. Sheriff, Y.F. Hu, P. Conforto, and C. Tocci. Mobility management incorporating fuzzy logic for a heterogeneous IP environment. *IEEE Communication Magazine*, pages 42–51, December 2001.
- [23] Rupenaguntla Naga Satish Chandra and Dilip Sarkar. Call admission control in mobile cellular CDMA systems using fuzzy associative memory. In *Proceedings of IEEE International Conference on Communications ICC 2004*, Paris, France, 2004.

- [24] Li Fung Chang, Xioxin Qiu, Kapil Chawla, and Jain Cai. Providing differentiated services in EGPRS through radio resource management. In *Proceedings of IEEE ICC'01*, June 2001.
- [25] Bor-Sen Chen, Sen-Chueh Peng, and Ku-Chen Wang. Traffic modeling, prediction, and congestion control for high-speed networks: A fuzzy AR approach. *IEEE Transactions on Fuzzy Systems*, 8(4):491–508, October 2000.
- [26] J.-L. Chen, J.-H. Gong, and H.-C. Chao. Resource reservation with mobile hosts using fuzzy matrices. *Computer Communications*, 24:434–444, 2001.
- [27] R.-G. Cheng and C.-J. Chang. Design of a fuzzy traffic controller for ATM networks. *IEEE/ACM Transactions on Networking*, 4(3):460–469, June 1996.
- [28] R.-G. Cheng, C.-J. Chang, and L.-F. Lin. A QoS-provisioning neural fuzzy connection admission controller for multimedia high-speed networks. *IEEE/ACM Transactions on Networking*, 7(1):111–121, February 1999.
- [29] T. Chiueh. Optimization of fuzzy logic inference architecture. *IEEE Computer*, pages 67–71, May 1992.
- [30] Renaud Cuny, Gerardo Gómez, and Héctor Montes. GERAN QoS evolution towards UMTS. In Timo Halonen, Javier Romero, and Juan Molero, editors, *GSM, GPRS and EDGE Performance*, pages 95–118. John Wiley & Sons, LTD, 2002.
- [31] M. de Prycker. *Asynchronous Transfer Mode. Solutions for Broadband ISDN. Third Edition*. Prentice Hall, 1995.
- [32] Paolo Dini and Silvia Guglielmucci. A call admission control strategy based on fuzzy logic for WCDMA systems. In *Proceedings of IEEE International Conference on Communications ICC 2004*, Paris, France, 2004.
- [33] George Edwards, Abraham Kandel, and Ravi Sankar. Fuzzy handoff algorithms for wireless communications. *Fuzzy Sets and Systems*, 110:379–388, 2000.
- [34] George Edwards and Ravi Sankar. Microcellular handoff using fuzzy techniques. *Wireless Networks*, 4:401–409, 1998.
- [35] Petru Eles, Krzysztof Kuchinski, and Zebo Peng. *System Synthesis with VHDL*. Kluwer Academic Publishers, 1998.
- [36] Marcial P. Fernandez, Aloysio C.P. Pedroza, and José F. Rezende. Converting QoS policy into fuzzy logic parameters. In *Proceedings of ITC 2003*, Berlin, Germany, 2003.
- [37] Giuseppe Ficili and Daniela Panno. A fuzzy algorithm control of traffic parameters: assesment and key issues. *Computer Communication*, 22:199–210, 1999.

- [38] Chuan Heng Foh, Beatrice Meini, Bartek Wydrowski, and Moshe Zuckerman. Modeling and performance evaluation of GPRS. In *Proc. IEEE VTC 2001*, pages 2108–2112, Rhodes, Greece, 2001.
- [39] D. Galan, C.J. Jimenez, A. Barriga, and S. Sanchez-Solano. VHDL package for description of fuzzy logic controllers. In *Proceedings of EURO-DAC'95*, pages 528–534, 1995.
- [40] Sumit Ghosh, Qutaiba Razouqi, H.Jerry Schumacher, and Aivars Celmins. A survey of recent advances in fuzzy logic in telecommunications networks and new challenges. *IEEE Transactions on Fuzzy Systems*, 6(3):443–447, August 1998.
- [41] Brahim Ghribi and Luigi Logrippo. Understanding GPRS: the GSM packet radio service. *Computer Networks*, 34 (2000):763–779, 2000.
- [42] David J. Goodman. *Wireless Personal Communications Systems*. Addison-Wesley, 1997.
- [43] Hubert Graja, Philip Perry, **Doru Todinca**, and John Murphy. Novel GPRS simulator for testing MAC protocols. In *Proceedings of IEE 4th International Conference on 3G Mobile Communications Technologies*, pages 409–412, June 2003.
- [44] Hakon Gudding. Capacity analysis of GPRS. White Paper. Revised Edition of Master Thesis, Department of Telematics, Faculty of Electrical Engineering and Telecommunications, Norwegian Institute of Science and Technology, March 2000.
- [45] Zygmunt Haas, Joseph Y. Halpern, Li Li, and Stephen B. Wicker. A decision theoretic approach to resource allocation in wireless multimedia networks. In *Proc. 4th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (dialM)*, 2000.
- [46] Markus Hakaste, Eero Nikula, and Shkumbin Hamiti. GSM/EDGE standards evolution (up to Rel). In Timo Halonen, Javier Romero, and Juan Molero, editors, *GSM, GPRS and EDGE Performance*, pages 3–63. John Wiley & Sons, LTD, 2002.
- [47] Christian Hoymann and Peter Stuckmann. Traffic engineering for the evolution of GPRS/EDGE networks. In *Proceedings of the IEEE VTC'02*, 2002.
- [48] Tim Irnich and Peter Stuckmann. Analytical performance evaluation of Internet access over GPRS and its comparison with simulation results. In *Proceedings of PIMRC*, 2002.
- [49] Zhimei Jiang, Li Fung Chang, and N.K. Shankaranarayanan. Providing multiple service classes for bursty data traffic in cellular networks. In *Proc. of IEEE INFOCOM 2000*, volume 3, pages 1087–1096, Tel-Aviv, Israel, March 2000.

- [50] Zhimei Jiang and N.K. Shankaranarayanan. Channel quality dependent scheduling for flexible wireless resource control. In *Proceedings of IEEE Globecom 2001*, Texas, USA, November 2001.
- [51] Abraham Kandel, Orly Manor, Yaron Klein, and Sharon Fluss. ATM traffic management and congestion control using fuzzy logic. *IEEE Transactions on Systems, Man, and Cybernetics – Part C: Applications and Reviews*, 29(3):474–480, August 1999.
- [52] Giorgios Karagiannis and Geert Heijenk. QoS in GPRS. TR-CTIT-00-29. Technical report, University of Twente, 2000.
- [53] S. Kim, T. Kwon, and Y. Choi. Call admission control for prioritized adaptive multimedia services in wireless/mobile networks. In *Proceedings of IEEE Vehicular Technology Conference (VTC'00 Spring)*, Tokyo, Japan, 2000.
- [54] Y.-C. Ko, S.-C. Park, C.-Y. Chun, H.-W. Lee, and C.-H. Cho. An adaptive QoS provisioning distributed call admission control using fuzzy logic control. In *Proceedings of IEEE International Conference on Communications ICC 2001*, volume 2, pages 356–360, 2001.
- [55] Shie-Jue Lee and Chun-Liang Hou. A neural-fuzzy system for congestion control in ATM networks. *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics*, 30(1):2–9, February 2000.
- [56] H.H. Lim and B. Qiu. A predictive measurement-based fuzzy logic connection admission control. In *Proceedings of IEEE International Conference on Communications ICC 2001*, volume 3, pages 920–924, 2001.
- [57] Meng-Ju Lin, Hui Luo, and Li Fung Chang. A Linux-based EGPRS real-time test bed software for wireless QoS and differentiated service studies. In *Proceedings of IEEE ICC'02*, New York, NY, USA, May 2002.
- [58] Phone Lin and Yi-Bing Lin. Channel allocation for GPRS. *IEEE Transactions on Vehicular Technology*, 50(2):375–387, March 2001.
- [59] Christoph Lindemann and Axel Thümmler. Evaluating the GPRS radio interfaces for different quality of service profiles. In *Proceedings of the IEEE International Conference on Networking ICN'01*, Colmar, France, 2001. Springer LNCS.
- [60] Christoph Lindemann and Axel Thümmler. Performance analysis of the general packet radio service. In *Proceedings of the 21st International Conference on Distributed Computing Systems ICDCS*, pages 673–680, Phoenix, Arizona, April 2001.
- [61] S. Lu, V. Bharghavan, and R. Srikant. Fair scheduling in wireless packet networks. *IEEE/ACM Transactions on Networking*, 7(4):473–489, 1999.

- [62] Giridhar D. Mandyam and M.D. Srinath. Applications of fuzzy system theory to telecommunications. In Cornelius T. Leondes, editor, *Fuzzy Theory Systems: Techniques and Applications*, volume 2, pages 559–575. Academic Press, 1999.
- [63] Jay R. Moorman, John W. Lockwood, and Sung-Mo Kang. Real-time prioritized call admission control in a base station scheduler. In *Proceedings of ACM Womom*, pages 28–37, 2000.
- [64] T.J. Mousley. Performance of UMTS high speed downlink packet access for data streaming applications. In *Proceedings of IEE 3rd International Conference on 3G Mobile Communications Technologies*, pages 302–307, London, UK, May 2002.
- [65] Eugen Mârza and Călin Simu. *Comunicații mobile. Principii și standarde*. Editura de Vest, Timișoara, 2003.
- [66] Ion Năftănăilă and Ionel Năftănăilă. *Dictionar de Internet și telecomunicații englez francez român*. Editura Tehnică, București, 2000.
- [67] C.V. Negoită and D. A. Ralescu. *Mulțimi vagi și aplicațiile lor*. Editura Tehnică, 1974.
- [68] Nelson O.L. Ng and C.K. Tham. Connection admission control of ATM network using integrated MLP and fuzzy controllers. *Computer Networks*, 31:61–79, 2000.
- [69] T.S Eugene Ng, D.C. Stevens, Ion Stoica, and Hui Zhang. Supporting best-effort traffic with fair service curve. In *Proceedings of IEEE Globecom*, Rio de Janeiro, Brazil, 1999.
- [70] T.S. Eugene Ng, Ion Stoica, and Hui Zhang. Packet fair queueing algorithms for wireless networks with location-dependent errors. In *Proceedings of IEEE INFOCOM'98*, pages 1103–1111, San Francisco, CA, USA, 1998.
- [71] Edward Nowicki. Resource allocation for multimedia messaging services over EGPRS. Master Thesis, School of Electronic Engineering, Dublin City University, 2003.
- [72] Qixiang Pang, Amir Bigloo, Victor C.M. Leung, and Chris Scholefield. Service scheduling for general packet radio service classes. In *Proc. of IEEE Wireless Communications and Networking Conference WCNC'99*, New Orleans, LA, USA, 1999.
- [73] A.K. Parekh and R.G. Gallager. A generalized processor sharing approach to flow control in integrated services networks: The single node case. *IEEE/ACM Transactions on Networking*, 1(3):344–357, 1993.

- [74] Marek J. Patyra. Design considerations of fuzzy logic controllers. In Marek J. Patyra and D.M. Mlynek, editors, *Fuzzy Logic: Implementations and Applications*, pages 143–175. John Wiley & Sons Ltd. and B.G. Teubner, 1996.
- [75] H.G. Perros and K.M. Elsayed. Call admission control schemes: A review. *IEEE Communications Magazine*, 34(11):82–91, November 1996.
- [76] Emilia Petrișor. *Probabilități și statistică matematică. Aplicații în economie și inginerie*. Editura Politehnica, Timișoara, 2001.
- [77] Radu Emil Precup. On the parameter adaptation of a fuzzy controller meant for the speed control of hydrogenerators. In *Proceedings of Third EUFIT'95 European Congress*, volume 2, pages 1105–1109, Aachen, Germany, 1995.
- [78] Radu-Emil Precup and Ștefan Preitl. *Fuzzy Controllers*. Editura Orizonturi Univrsitare, Timișoara, 1999.
- [79] Radu Emil Precup, **Doru Todinca**, and Ștefan Preitl. VHDL implementation of PID fuzzy controller applied to control of non minimum phase systems. In *Proceedings of the VI International Conference on Systems, Automatic Control and Measurements SAUM'98*, pages 369–374, Nis, Yugoslavia, 1998.
- [80] Ștefan Preitl and Radu-Emil Precup. *Introducere în conducerea fuzzy a proce-selor*. Editura Tehnică, București, 1997.
- [81] Qutaiba Razouqi, Seong-Soon Joo, and Sumit Ghosh. Performance analysis of fuzzy thresholding-based buffer management for a large-scale cell-switching network. *IEEE Transactions on Fuzzy Systems*, 8(4):425–441, August 2000.
- [82] Qutaiba Razouqi, Tony Lee, and Sumit Ghosh. A guaranteed-no-cells-dropped buffer management scheme with selective blocking for cell-switching networks. *Computer Communications*, 21:930–946, 1998.
- [83] J. Rendón, F. Casadevall, L. García, and R. Jimenez. Characterization of the GPRS radio interface by means of a statistical model. In *Proceedings of IEEE Vehicular Technologies VTC 2001*, pages 2392–2396, 2001.
- [84] J. Rendón, F. Casadevall, L. García, and R. Jimenez. Simulation model for performance evaluation of internet applications using GPRS radio interface. *IEE Electronics Letters*, 37(12):786–787, June 2001.
- [85] V. Rexhepi. Wireless internet QoS, Master Thesis, University of Twente, Department of Computer Science and Electrical Engineering, 2000.
- [86] Jonathan Sau and Chris Scholefield. Scheduling and quality of service in the general packet radio service. In *Proceedings of IEEE ICUPC*, pages 1067–1071, 1998.

- [87] N.K. Shankaranarayanan, Anupam Rastogi, and Zhimei Jiang. Performance of a wireless data network with mixed interactive user workloads. In *Proceedings of ICC 2002*, 2002.
- [88] X. Shen and J.M. Mark. Mobility information for resource management in wireless ATM networks. *Computer Networks*, 31:1049–1062, 1999.
- [89] Peter Stuckmann. Quality of Service management in GPRS-based radio access networks. *Telecommunication Systems*, 19:3,4:515–546, 2002.
- [90] Peter Stuckmann, Niels Ehlers, and Bianca Wouters. GPRS traffic performance measurements. In *Proceedings of the IEEE VTC'02*, 2002.
- [91] Peter Stuckmann, Henrik Finck, and Thomas Bahls. A WAP traffic model and its appliance for the performance analysis of WAP over GPRS. In *Proceedings of the IEEE International Conference on Third Generation Wireless and Beyond (3G Wireless'01)*, San Francisco, USA, 2001.
- [92] Peter Stuckmann and Jörg Franke. The capacity and performance gain reachable with link quality control in EGPRS networks. In *Proceedings of the IEEE International Conference on Third Generation Wireless and Beyond (3G Wireless'01)*, San Francisco, USA, 2001.
- [93] Peter Stuckmann and Christian Hoymann. Traffic performance analysis of multimedia applications in evolved GSM networks. In *Multiaccess, Mobility and Teletraffic for Wireless Communications*, volume 6, pages 279–293. Kluwer Academic Publishers, June 2002.
- [94] Peter Stuckmann and Oliver Möller. Advanced scheduling and admission control techniques for cellular packet radio networks. In *Proceedings of European Personal Mobile Communications Conference EPMCC*, Glasgow, UK, 2003.
- [95] Peter Stuckmann and Frank Müller. GPRS radio network capacity and quality of service using fixed and on-demand channel allocation techniques. In *Proceedings of IEEE Conference on Vehicular Technology, VTC2000-Spring*, Tokyo, Japan, May 2000. Miller Freeman, New York, 2000).
- [96] Peter Stuckmann and Frank Müller. Quality of service management in GPRS networks. In *Proceedings of the IEEE International Conference on Networking ICN'01*, Colmar, France, 2001. Springer LNCS.
- [97] Peter Stuckmann and Oliver Paul. Dimensioning GSM/GPRS networks for circuit- and packet-switched services. In *Proceedings of the 10th Symposium on Wireless Personal Multimedia Communications*, pages 597–602, Aalborg, Denmark, September 2001.
- [98] Andrew S. Tanenbaum. *Rețele de Calculatoare. Ediția a treia*. Computer Press Agora S.R.L., 1997.

- [99] **Doru Todinca**. Data transfer over GPRS/EGPRS networks. Mecanisme de transmitere a datelor in sistemul GPRS/EGPRS. Referat numarul 1. Technical report, University “Politehnica” Timisoara, Department of Computers, April-May 2003.
- [100] **Doru Todinca**. Modeling resource allocation techniques for GPRS/EGPRS networks. *Periodica Politehnica, Transactions on Automatic Control and Computer Science*, 48(62):56–61, June 2003.
- [101] **Doru Todinca**. Rezultate experimentale privind performantele algoritmilor pentru transmiterea datelor in GPRS/EGPRS. Experimental results concerning the performance of algorithms for data transfer over GPRS/EGPRS networks. Referat numarul 2. Technical report, University “Politehnica” Timisoara, Department of Computers, June 2003.
- [102] **Doru Todinca**. Applying fuzzy logic to admission control in GPRS/EGPRS networks. In *Proceedings of 1st Romanian - Hungarian Joint Symposium on Applied Computational Intelligence SACI 2004*, Timisoara, Romania, May 2004.
- [103] **Doru Todinca**. Tehnici de alocare a resurselor pentru cresterea performanțelor transiterii datelor in rețelele de telefonie mobilă de tip GPRS/EGPRS. Resource allocation techniques for improving the performance of data transfer over GPRS/EGPRS networks. Referat numarul 3. Technical report, University “Politehnica” Timisoara, Department of Computers, April 2004.
- [104] **Doru Todinca**, Ana Maria Badulescu, and Radu Emil Precup. VHDL approach to performance analysis of fuzzy logic controllers. In *Proceedings of 8th Information Processing and Management of Uncertainty in Knowledge-Based Systems Conference (IPMU 2000)*, pages 896–901, Madrid, Spain, July 2000.
- [105] **Doru Todinca** and Virgil Cimpu. VHDL behavioural description of a fuzzy inference processor. *Periodica Politehnica, Transactions on Automatic Control and Computer Science*, 42 (56):229–234, 1997.
- [106] **Doru Todinca**, Ștefan Holban, Philip Perry, and John Murphy. Fuzzy logic based admission control for GPRS/EGPRS networks. In *Proceedings of the 6th International Conference on Technical Informatics CONTI 2004*, Timișoara, Romania, May 2004.
- [107] **Doru Todinca**, Ciprian Ghișe, Mihai Moisa, and Ioan Brînzăș. VHDL simulation of a fuzzy processor. *Periodica Politehnica, Transactions on Automatic Control and Computer Science*, 41 (55):207–214, 1996.
- [108] **Doru Todinca**, Hubert Graja, Philip Perry, and John Murphy. Novel admission control algorithm for GPRS/EGPRS based on fuzzy logic. In *Proceedings of IEE 5th International Conference on 3G Mobile Communications Technologies 3G2004*, London, UK, October 2004.

- [109] **Doru Todinca**, Philip Perry, and John Murphy. Resource allocation algorithms for GPRS network performance improvement during congested periods. In *IEI/IEE Postgraduate Symposium Telecommunications Systems Research*, Dublin, Ireland, November 2001.
- [110] **Doru Todinca**, Philip Perry, and John Murphy. Algorithms for resource allocation in data transfer over EGPRS networks. In *Proceedings of the 2nd European Conference on Universal Multiservice Networks (ECUMN'2002)*, pages 246–250, Colmar, France, April 2002.
- [111] **Doru Todinca**, Philip Perry, and John Murphy. Novel prioritised EGPRS medium access regime for reduced file transfer delay during congested periods. In *Proceedings of IEE 3rd International Conference on 3G Mobile Communications Technologies*, pages 550–554, London, UK, May 2002.
- [112] **Doru Todinca** and Radu Emil Precup. VHDL simulation of a general-purpose fuzzy processor. In *Proceedings of Second International Conference CONTI'96*, volume I, pages 112–118, Timisoara, Romania, 1996.
- [113] Masaki Togai and Hiroyuki Watanabe. Expert system on a chip: An engine for real-time approximate reasoning. *IEEE Expert*, I(3):55–62, 1986.
- [114] András Varga. OMNeT++ Discrete event simulator system. User manual. Technical report, Technical University of Budapest, Faculty of Electrical Engineering and Informatics, Department of Telecommunications, 2001.
- [115] E. Vivier, M. Terré, N. Bret, and M. Bourgois. GPRS network emulator. In *Proceedings of EPMCC 2001*, Vienna, Austria, February 2001.
- [116] Jean Warland and Pravin Varayia. *High Performance Communications Networks. Second edition*. Morgan Kaufmann publishers, 2000.
- [117] Hiroyuki Watanabe, James R. Symon, Wayne D. Detloff, and Kathy E. Yount. VLSI fuzzy chip and inference accelerator board systems. In Lotfi A. Zadeh and Janusz Kacprzyk, editors, *Fuzzy Logic for the Management of Uncertainty*, pages 211–243. John Wiley & Sons, Inc., 1993.
- [118] Lars Wischhof and John W. Lockwood. Packet scheduling for link-sharing and Quality of Service support in wireless local area networks WUCS-01-35. Technical report, Washington University, Department of Computer Science Applied Research Lab, Saint Louis, MO, November 2001.
- [119] Mohamad Hossien Yaghmaee, Mostafa Safavi, and Mohamad Bagher Menhaj. An intelligent usage parameter controller based on dynamic rate leaky bucket for ATM networks. *Computer Networks*, 32:17–34, 2000.
- [120] T. Yamakawa. Stabilization of an inverted pendulum by a high speed fuzzy logic controller hardware system. *Fuzzy Sets and Systems*, 32:161–180, 1989.

- [121] A. Zamfirescu and C. Ussery. VHDL and fuzzy logic if-then rules. In *Proceedings of EURO-DAC'92*, pages 636–641, 1992.
- [122] Hui Zhang. Service disciplines for guaranteed performance service in packet switching networks. *Proceedings of the IEEE*, 83:1374–1396, October 1995.
- [123] H.-J. Zimmermann. *Fuzzy set theory - and its applications*. Kluwer Academic Publishers, 1985.

Lista figurilor

2.1	Arhitectura sistemului GSM	9
2.2	Arhitectura sistemului GPRS	10
2.3	Axa de planificare a resurselor [50]	22
3.1	Arhitectura de protocoale GPRS. Planul transmisiei datelor	47
3.2	Fluxul de date în GPRS	49
3.3	Stările managementului mobilității pentru o stație mobilă	51
3.4	Procedura de activare a contextului PDP	53
3.5	Modelul de simulare	58
3.6	Modulul utilizator	60
3.7	Captură de ecran pentru modulul utilizator în timpul simulării	65
3.8	Captură de ecran conținând evenimentele de la timpul de simulare 5280ms	66
3.9	Modulul PCU	67
3.10	Reprezentarea OMNeT++ a modulului PCU	67
3.11	Reprezentarea OMNeT++ a modulului generator de utilizatori	70
3.12	Reprezentarea OMNeT++ a întregului sistem	72
4.1	Descrierea în pseudocod a algoritmului WRR (IRR)	80
4.2	Descrierea în pseudocod a algoritmilor OQ, LQ, TQL și TFL	83
4.3	Valorile medii pentru <i>sending delay</i> , distribuție triunghiulară a inter- valelor de generare a datelor, 3 fișiere pe pagină	88
4.4	Valorile medii pentru <i>total delay</i> , distribuție triunghiulară a intervalelor de generare a datelor, 3 fișiere pe pagină	89
4.5	Valorile medii ale <i>total delay</i> , distribuție exponențială a intervalelor de generare a datelor, 3 fișiere pe pagină	90
4.6	Evoluția în timp a numărului de fișiere din bufferul de fișiere al uti- lizatorului <i>user₁</i> , distribuție exponențială, 3 fișiere, încărcare 95%, al- goritm OQ	90
4.7	Evoluția în timp a numărului de fișiere din bufferul de fișiere al uti- lizaotrului <i>user₉</i> , distribuție exponențială, 3 fișiere, încărcare 95%, al- goritm LQ	91
4.8	Valorile medii ale <i>total delay</i> , pentru o distribuție triunghiulară a in- tervalelor de generare a datelor, 1 fișier pe pagină	92

4.9	Valorile medii ale <i>total delay</i> , pentru o distribuție exponențială a intervalelor de generare a datelor, 1 fișier pe pagină	93
4.10	Valorile medii pentru <i>total delay</i> , distribuție exponențială, 1 fișier pe pagină, $W_{HP}=2$, $W_{LP}=1$, setul 2 de simulări	94
4.11	Valorile medii pentru <i>total delay</i> , distribuție exponențială, 1 fișier pe pagină, $W_{HP}=4$, $W_{LP}=1$, setul 2 de simulări	94
4.12	Valorile medii pentru <i>total delay</i> , distribuție exponențială, 1 fișier pe pagină, $W_{HP}=8$, $W_{LP}=1$, setul 2 de simulări	95
4.13	Valorile medii pentru <i>total delay</i> , distribuție exponențială, 1 fișier pe pagină, $W_{HP}=10$, $W_{LP}=1$, setul 2 de simulări	95
4.14	Scenariu de simulare, cu utilizatorii generând date la momentele t_0, t_1, \dots, t_{N-1}	97
4.15	Comparația valorilor medii ale întârzierilor totale pentru clase diferite de utilizatori, algoritm WRR și ponderi 3, 2, 1	99
4.16	Comparația valorilor medii ale întârzierilor totale pentru clase diferite de utilizatori, algoritm WRR și ponderi 8, 4, 1	100
4.17	Comparația valorilor medii ale întârzierilor totale pentru clase diferite de utilizatori, algoritm WRR și ponderi 8, 2, 1	100
4.18	Influența valorilor ponderilor asupra întârzierii totale pentru WRR, distribuție triunghiulară, setul 1 B de simulări	101
4.19	Influența valorilor ponderilor asupra întârzierii totale a utilizatorilor HP pentru WRR, distribuție exponențială, setul 2 de simulări	102
4.20	Influența valorilor ponderilor asupra întârzierii totale a utilizatorilor LP pentru WRR, distribuție exponențială	103
5.1	Mecanismul de inferență fuzzy	115
5.2	Schema generală a unui proces de reglare fuzzy	116
5.3	Operația de fuzzificare	117
5.4	FLC clasic cu 2 intrări	119
5.5	Termenii variabilei de intrare, împărțiți în pari și impari	121
5.6	FLC simplificat cu 3 intrări	122
5.7	Termenii variabilelor lingvistice <i>network load</i> și <i>users precedence</i> din premisele FLC	130
5.8	Termenii variabilei lingvistice <i>admission decision</i> din concluzie	130
5.9	Intrarea și ieșirea FLC pentru o încărcare țintă având valoarea 10 și pentru un interval de generare a utilizatorilor de 230ms.	133
5.10	Încărcarea rețelei pentru o încărcare țintă având valoarea 10 și pentru un interval de generare a utilizatorilor de 230ms	133
5.11	Intrarea și ieșirea FLC și încărcarea rețelei pentru o încărcare țintă având valoarea 5 și o perioadă de generare a utilizatorilor de 130ms.	134
5.12	Intrarea și ieșirea FLC pentru o valoare țintă a încărcării rețelei egală cu 5 și o perioadă de generare a utilizatorilor de 230ms.	135
5.13	Încărcarea rețelei pentru o valoare țintă a încărcării rețelei egală cu 5 și o perioadă de generare a utilizatorilor de 230ms.	135

5.14	Media întârzierilor totale pentru cele 3 clase de utilizatori când valorile ponderilor sunt $W=\{4,2,1\}$ și perioada de generare a utilizatorilor este de 180ms, algoritm fuzzy de AC.	138
5.15	Încărcarea rețelei, când valorile ponderilor sunt $W=\{4,2,1\}$ și perioada de generare a utilizatorilor este de 180ms, cu algoritm fuzzy de AC.	138
5.16	Media întârzierilor totale pentru clasa de utilizatori QoS 3, când valorile ponderilor sunt $W=\{4,2,1\}$ și perioada de generare a utilizatorilor este de 180ms, cu algoritm de AC fuzzy și non-fuzzy.	139
5.17	Probabilitățile de blocare a apelurilor pentru toate clasele de QoS, pentru valori asimetrice ale ieșirilor.	141
5.18	Probabilitățile de blocare a apelurilor pentru clasa QoS 3, cu algoritm de AC fuzzy și non-fuzzy, valori asimetrice pentru termenii variabilei de ieșire.	141
5.19	Probabilitățile de blocare a apelurilor pentru clasele QoS 1 și QoS 2, cu algoritmi de AC fuzzy și non-fuzzy, valori simetrice pentru termenii variabilei de ieșire.	142

Lista tabelelor

2.1	Caracteristicile principale ale GSM	7
2.2	Exemplu cu GPS și PGPS [73]	14
3.1	Clasele de QoS în GPRS release 1999	56
3.2	Comparație între valorile obținute prin simulare și valorile calculate pentru <i>sending_delay</i>	74
5.1	Setul de reguli fuzzy	131

Anexa A

Glosar

3G	Third Generation
AC	Admission Control
ARQ	Automatic Repeat reQuest
ATM	Asynchronous Transfer Mode
AUC	Authentication Center
BAA	Bandwidth Adaptation Algorithm
BCS	Block Check Sequence bits
BLER	Block Error Rate
BSC	Base Station Controller
BSS	Base Station Subsystem
BSSGP	BSS GPRS Application Protocol
BTS	Base Transceiver Station
C/I	Channel to Interference Ratio
cdf	cummulative distribution function
CDMA	Code Division Multiple Access
CS	Circuit Switching
CS	Coding Scheme
EDGE	Enhanced Data Rates for Global Evolution
EGPRS	Enhanced GPRS
EIR	Equipment Identity Register
ETSI	European Telecommunications Standard Institute
FCS	Frame Check Sequence bits
FDMA	Frequency Division Multiple Access
FFQ	Fluid Fair Queuing
FH	Frame Header
FLC	Fuzzy Logic Controller
FTP	File Transfer Protocol
GGSN	Gateway GPRS Support Node
GMSC	Gateway Mobile Switching Center
GPRS	General Packet Radio Service
GPS	Generalized Processor Sharing

GSM	Global System for Mobile communications
GTP	GPRS Tunneling Protocol
HLR	Home Location Register
IP	Internet Protocol
IRR	Iterative Round Robin
ISDN	Integrated Services Data Network
LA	Location Area
LA	Link Adaptation
LLC	Logical Link Control
LQ	Longest Queue
MAC	Medium Access Control
MCS	Modulation and Coding Scheme
MDP	Markov Decision Process
MMPP	Markov Modulated Poisson Process
MN	Mobile Node
MNSAR	Maximum Number of Simultaneously Active Rules
MOF	Maximum Overlap Factor
MS	Mobile Station
MSC	Mobile Switching Center
NED	Network Description
OMNeT++	Objective Modular Network Testbed in C++
OQ	Oldest Queue
PCU	Packet Control Unit
PDA	Personal Data Assistant
PDCH	Packet Data Channel
pdf	probability density function
PDN	Packet Data Network
PDP	Packet Data Protocol
PDTCH	Packet Data Traffic Channel
PDU	Protocol Data Unit
PDU	Payload Data Unit
PER	Packet Error Rate
PFQ	Packet Fair Queueing
PGPS	Packetized GPS
PH	Packet Header
PS	Packet Switching
PSTN	Public Switched Telephone Network
QoS	Quality of Service
RA	Routing Area
RLC	Radio Link Control
RNC	Radio Network Controller
RR	Round Robin

SGSN	Serving GPRS Support Node
SMS	Short Message Service
SNDCP	Subnetwork Dependent Convergence Protocol
TBF	Temporary Block Flow
TC	Transmission Control
TCP	Transmission Control Protocol
TDMA	Time Division Multiple Access
TFL	Total File Length
TFT	Traffic Flow Template
TLLI	Temporary Logic Link Identity
TQL	Total Queue Length
UDP	User Datagram Protocol
UMTS	Universal Mobile Telecommunications Systems
VHDL	VHSIC Hardware Description Language
VHSIC	Very High Speed Integrated Circuit
VLR	Visited Location Register
WAP	Wireless Application Protocol
WCDMA	Wideband CDMA
WFQ	Weighted Fair Queueing
WRR	Weighted Round Robin