

Strategii scalabile de dirijare QoS a pachetelor în rețele MPLS folosind mecanisme de cod mobil

autor
Sanda Maria Dragoș

TEZĂ ÎNDRUMATĂ DE: DR. MARTIN COLLIER

Rezumat

Teză susținută în data de 7 September 2006



SCHOOL OF ELECTRONIC ENGINEERING
DUBLIN CITY UNIVERSITY

Abstract

Dirijarea QoS se referă la găsirea, într-o rețea, cel puțin a unei căi care să satisfacă anumite specificări, numite constrângeri QoS.

Într-un Internet în continuă evoluție, este necesară folosirea unor instrumente cum ar fi dirijarea de tip QoS a pachetelor în vederea satisfacerii cerințelor utilizatorilor. La ora actuală este dificil să se dezvolte strategii de dirijare QoS în Internet. Separarea între controlul și trimiterea efectivă a pachetelor introdusă de MPLS (*MultiProtocol Label Switching*) facilitează folosirea strategiilor de dirijare QoS. Chiar și așa, dirijarea QoS ridică probleme de scalabilitate în cazul rețelelor mari. În această teză, eu propun evitarea acestor probleme de scalabilitate prin folosirea metodelor de agregare topologică și dirijare distribuită folosind tehnici moderne ca rețele active și agenți mobili. Metodele de agregare topologică introduc imprecizie în procesul de dirijare. Acest lucru poate avea un efect negativ asupra dirijării QoS. În acest context propun un protocol de dirijare ierarhică, numit *Macro-routing*, ca soluție la aceste probleme. Acest protocol folosește dirijare distribuită, și de aceea permite prelucrarea unor informații mult mai detaliate (de exemplu permite folosirea celei mai precise metode de agregare topologică: *Full-Mesh*). Așadar, acest protocol nu numai că va găsi cea mai bună cale între o sursă și o destinație, dar va oferi mai multe căi alternative.

Dirijarea QoS este o problemă NP-completă. Într-un context ierarhic această problemă devine și mai complexă datorită faptului că procesul de găsire al unei căi poate interfera cu tehnicile de agregare. Pentru a reduce cazurile în care nu se găsește nici o cale, propun o nouă metodă de agregare topologică, care să permită selectarea mai multor căi care să satisfacă constrângerile QoS. Această nouă metodă de agregare se numește *Extended Full-Mesh* și este concepută spre a fi folosită împreună cu protocolul *Macro-routing*. Folosirea a unor astfel de protocoale în Internet va constitui o implementare practică a dirijării cu constrângeri multiple în rețele largi.

Cuprins

1	Introducere	2
1.1	Motivare	2
1.2	Contribuțiile tezei	3
1.3	Structura tezei	3
2	Primitive de dirijarea pachetelor în rețele de calculatoare	4
2.1	Dirijarea pachetelor	4
2.2	Calitatea serviciilor (QoS)	5
2.3	Multiprotocol Label Switching (MPLS)	5
2.4	Rețele active	6
2.5	Agenți mobili	6
3	Probleme și soluții în dirijarea QoS	8
3.1	Dirijare ierarhică	8
3.2	Dirijarea bazată pe constrângeri multiple	10
3.2.1	Soluții propuse pentru dirijarea bazată pe constrângeri multiple	10
3.2.2	Dirijarea ierarhică bazată pe constrângeri multiple	11
3.3	Noi abordări pentru problema dirijării QoS	11
3.3.1	Dirijare bazată pe agenți mobili	11
3.3.2	Dirijare bazată pe rețele active	12
4	Rețele active sau agenți mobili	13
4.1	Integrarea MPLS cu rețele active	13
4.2	Exemple de aplicații	14
4.2.1	MPLS Web Switching	14
4.2.2	Licitări <i>on-line</i>	15
5	Dirijare scalabilă folosind agenți mobili în rețele MPLS	17
5.1	Descrierea protocolului	17
5.2	Rezultatele obținute în urma simulărilor	18
5.2.1	Urmărirea evoluției populațiilor de agenți mobili	18
5.2.2	Limitarea populației de <i>wave-uri</i>	21
6	Dirijare QoS ierarhică folosind constrângeri multiple	23
6.1	Descrierea metodei de agregare <i>extended Full-Mesh</i>	23
6.2	Limitarea vectorului căilor	24
6.2.1	Truncare intervalului EFM	24
6.2.2	Selecție random	25
6.3	Rezultatele testelor	25
7	Concluzii	31

Capitolul 1

Introducere

Comunicarea este un element indispensabil în orice societate. Oamenii au început să comunice încă de la începutul omenirii. Evoluția instrumentelor de comunicare a atins un apogeu odată cu era informațională, când au apărut tehnologii ca și telefonica, radioul, televiziunea, calculatoarele și rețelele de calculatoare.

Toate aceste tehnologii tind acum să convergă. Aceasta convergență a început pe la mijlocul anilor 80, când existau trei rețele de comunicare globale: rețeaua de telefonie care transporta voce, rețeaua de televiziune care transporta video și Internetul care transporta date. Tot în această perioadă, cei care se ocupau de rețeaua de telefonie au decis să creeze o rețea care să transporte toate trei tipurile de trafic. Rezultatul muncii lor a fost protocolul ATM (*Asynchronous Transfer Mode*), care deși a constituit un pas important în această direcție, nu a avut succesul scontat deoarece IP (*Internet Protocol*) s-a dovedit a fi mult mai popular în rețelele de calculatoare. Cu toate acestea, ATM a introdus un concept foarte important, și anume: calitatea serviciilor (QoS).

Pe la mijlocul anilor 90 cercetătorii (atât cei din domeniul telefoniei cât și cei din domeniul rețelelor de calculatoare) au încercat să proiecteze un nou protocol de rețea care să aibă atât simplitatea și flexibilitatea rețelelor IP cât și oferirea de garanții QoS din rețelele ATM. Această competiție a avut ca și rezultat final protocolul MPLS, care oferă separarea dintre procesul de găsimă a căii optime și trimiterea efectivă a pachetelor, precum și un nou mecanism ierarhic de trimitere a pachetelor.

1.1 Motivare

Datorită succesului său, Internetul a cunoscut o evoluție spectaculoasă pe multiple planuri. În primul rând, numărul de calculatoare conectate crește aproape exponențial (dacă în Iulie 1993 Internetul inter-conecta 1.7 milioane de noduri (*hosts*), în Iulie 1998 cuprindea 36 milioane de noduri, iar în Ianuarie 2006 erau peste 394 milioane de noduri). În al doilea rând, Internetul crește în complexitate și importanță deoarece oferă o bază comună de convergență pentru tipurile de comunicație (voce, video, date).

S-au propus numeroase soluții și tehnologii ca și răspuns la provocările datorate creșterii rapide a Internetului în ceea ce privește numărul de utilizatori, și a cerințelor acestora pentru calitate, fiabilitate și eficiență.

Studiul descris în aceasta teză cuprinde cinci tehnologii care constituie domenii importante de cercetare în rețelele de comunicații. Acestea sunt: dirijarea pachetelor în rețelele de calculatoare, QoS, MPLS, rețele active și agenți mobili.

1.2 Contribuțiile tezei

Cercetarea prezentată în această teză a avut ca și scop final găsirea unor soluții optime de implementare eficientă a unor strategii de dirijare QoS în rețele largi de calculatoare. Acest studiu își propune:

- Să diminueze încărcarea/complexitatea introdusă de QoS în procesul de dirijare.
- Să identifice soluții noi pentru problema dirijării bazate pe constrângeri multiple.

Principalele contribuții ale acestei teze sunt:

- Oferirea unei analize comparative între rețelele active și agenții mobili privite din perspectiva implementării strategiilor de dirijare QoS. Prezintă în acest scop avantajele și dezavantajele celor două tehnologii și ajung la concluzia că agenții mobili sunt alegerea preferată. Rețelele active au utilitate limitată pentru procesul de dirijare datorită faptului că codul lor acționează asupra pachetelor care sunt deja în rețea și nu înainte ca acestea să ajungă acolo, așa cum se procedează în majoritatea scenariilor de dirijare QoS.
- Integrarea rețelelor active cu MPLS ca soluție la limitarea MPLS-ului de a realiza comutare de pachete numai la nivelul doi. Am demonstrat că această integrare se poate realiza și am oferit câteva exemple pentru a ilustra utilitatea unei astfel de soluții.
- Un nou protocol de dirijare ierarhică QoS, numit *Macro-routing*, care reduce încărcarea computațională și spațiul de stocare, introduse de dirijarea QoS, prin folosirea unor strategii de dirijare distribuite bazate pe agenți mobili. Acest protocol găsește concomitent mai multe căi fiabile fără nici un cost adițional.
- O nouă metodă de agregare, numită *Extended Full-Mesh*, care dă rezultate mai bune decât *Full-Mesh* în ceea ce privește căutarea/găsirea de căi cu constrângeri multiple, oferind mai multe șanse de găsire a unei astfel de căi. Propun, de asemenea, diferite metode de selectare a căilor folosite de agregarea EFM.

1.3 Structura tezei

Materialul prezentat în această teză este structurat în șapte capitole.

Capitolul 2 introduce cele cinci domenii principale de interes, care constituie baza studiului din această teză: dirijarea pachetelor în rețele de calculatoare, QoS, MPLS, rețele active și agenți mobili.

Capitolul 3 prezintă principalele probleme legate de dirijarea QoS și soluțiile existente.

Capitolul 4 compară rețelele active cu agenții mobili, ca tehnologii care se pot folosi pentru a implementa dirijarea QoS și oferă exemple de aplicații în care rețelele active se dovedesc a fi folosite. Acest capitol se încheie prin a trage concluzia că agenții mobili sunt alegerea preferată pentru a implementa mecanisme eficiente de dirijare QoS.

Capitolul 5 propune un nou protocol ierarhic de dirijare, numit *Macro-routing*, care reduce încărcarea computațională și spațiul de stocare, introduse de dirijarea QoS. În același timp, acest protocol va găsi mai multe căi fiabile. Aceasta performanță vine odată cu costul generării de prea mulți agenți mobili, deci un efort crescut de comunicare. În acest capitol se propun și metode de limitare a unui astfel de efort fără a reduce eficiența algoritmului.

Capitolul 6 propune o metodă nouă de agregare, numită *Extended Full-Mesh*, ca și metodă de agregare folosită de *Macro-routing* pentru a căuta căi care să satisfacă constrângeri multiple. Am propus și analizat, de asemenea, și diferite metode de selectare a căilor folosite pentru agregare.

Capitolul 7 încheie teza prin prezentarea concluziilor și a ideilor de cercetare pentru viitor.

Capitolul 2

Primitive de dirijarea pachetelor în rețele de calculatoare

Cercetarea prezentată în această teză se bazează pe cinci domenii principale de interes: dirijarea pachetelor în rețele de calculatoare, QoS, MPLS, rețele active și agenți mobili, care vor fi prezentate în continuare.

2.1 Dirijarea pachetelor

Dirijarea pachetelor este descrisă în [24, 60] ca fiind procesul de determinare a unei căi de la o sursă la o destinație. În mod tradițional, acest proces este întreținut de protocoale de dirijare care interschimbă informații necesare dirijării (ca de exemplu modul în care nodurile din rețea sunt interconectate și diferite atribute ale legăturilor dintre aceste noduri). Pe baza acestor informații, algoritmi de dirijare determină cea mai bună cale de la o sursă la o destinație. Rezultatele obținute de algoritmi de dirijare sunt folosite pentru a crea tabelele de dirijare care vor fi utilizate ulterior în procesul de transmitere efectivă a pachetelor.

Procesul de dirijare a evoluat odată cu alte tehnologii și cu evoluția rețelelor oferind o diversitate largă de strategii de dirijare:

- **Dirijare statică sau dinamică:** Diferența dintre cele două constă în faptul că prima este o dirijare configurată și întreținută manual care se mai folosește doar în rețele mici, în timp ce dirijarea dinamică totul e tratat automat. Majoritatea rețelelor din ziua de azi folosesc dirijare dinamică.
- **Dirijare neadaptivă sau adaptivă:** Primele protocoale de dirijare nu țineau cont de starea rețelelor în momentul creării tabelor de dirijare (ele sunt considerate neadaptive). Datorită faptului că de obicei există mai multe căi între sursă și destinație, cea care va fi folosită este selectată, în mod tradițional, ca fiind cea mai scurtă. Alte metode de selecție pot de asemenea fi folosite, ca de exemplu lățimea de bandă sau întârzierea. Această modalitate de dirijare, numită dirijare adaptivă a constituit primul pas către dirijarea QoS, principala diferență constând în faptul că prima metodă nu este însoțită de un mecanism de rezervare a resurselor.
- **Dirijarea pas-cu-pas sau dirijarea din nodul sursă:** În dirijarea pas-cu-pas decizia de dirijare se ia în fiecare nod independent, pe baza informațiilor din tabelele de dirijare. Totuși acest model, pe lângă faptul că poate genera căi care conțin cicluri nici nu permite introducerea unor mecanisme QoS. Pentru acest lucru se preferă dirijarea din nodul sursă, când toate informațiile necesare dirijării se colectează în nodul sursă.

- **Dirijarea după vectorul distanțelor sau dirijarea bazată pe starea legăturilor:** În dirijarea după vectorul distanțelor informațiile necesare dirijării se distribuie sub forma unor liste de distanțe către diverse destinații. Această metodă de dirijare se adaptează însă foarte lent la modificările din rețea. Alternativa este distribuirea informațiilor legate de toate legăturile din rețea astfel încât fiecare nod să își poată construi propria sa viziune asupra rețelei.
- **Dirijarea nestructurată sau dirijarea ierarhică:** În rețelele mari nu este scalabilă dirijarea nestructurată (*flat routing*). De aceea, în aceste cazuri, rețeaua se împarte în mai multe domenii între care se distribuie doar o formă agregată a informației de dirijare. Calea finală se va constitui prin concatenarea căilor determinate în cadrul acestor domenii.

2.2 Calitatea serviciilor (QoS)

Calitatea serviciilor (QoS) este un concept familiar telecomunicațiilor datorită faptului că aceste rețelele transportă voce, care este sensibilă la deformarea sunetului, transmitere întârziată, pierderea de pachetelor, etc. Deoarece tendința este ca acum Internetul să transporte pe lângă date, voce și video, trebuie ținute seama și de cerințele speciale ale acestor tipuri de trafic. Acesta se poate face cu ajutorul QoS care este definit în [14] ca fiind mulțimea cerințelor de servicii care trebuie îndeplinite de o rețea în timp ce aceasta transportă un flux.

Cerințele pentru diferite tipuri de trafic de pachete sunt specificate prin intermediul constrângerilor QoS. Ele se pot clasifica în constrângeri care se însumează (aditive), constrângeri care se înmulțesc (multiplicative) sau constrângeri care se selectează în funcție de cea mai mică/mare valoare (concave/convexe). Există trei modalități de implementare a constrângerilor QoS: constrângeri ale legăturilor, constrângeri ale nodurilor, constrângeri ale căilor. Constrângerile de legături și noduri sunt, de obicei, concave, iar constrângerile de căi sunt aditive sau multiplicative. Exemple de constrângeri aditive sunt întârzierea în transmitere (*delay*), fluctuația întârzierilor (*jitter*), costurile, și numărul de salturi. Probabilitatea de pierdere a unui pachet este o constrângere multiplicativă, iar lățimea de bandă este constrângere concavă.

Dirijarea QoS constă în găsirea unei căi de la o sursă la o destinație care să satisfacă anumite constrângeri specificate. De aceea dirijarea QoS constă în două etape:

1. colectarea și actualizarea/întreținerea informațiilor necesare procesului de dirijare QoS
2. căutarea unei căi posibile, bazate pe informațiile colectate în prima etapă.

Colectarea și stocarea informațiilor necesare dirijării QoS se poate face local, global sau agregat, iar căutarea unei căi care să satisfacă anumite constrângeri QoS se poate realiza central (la nodul sursă), distribuit sau ierarhic. Toate aceste sunt detaliat prezentate atât în teză cât și în următoarele lucrări [9, 8, 28]. Această teză se concentrează în mod special asupra dirijării ierarhice și dirijării distribuite bazate pe agenți mobili și rețele active.

2.3 Multiprotocol Label Switching (MPLS)

Multiprotocol Label Switching (MPLS) [49] este o tehnologie creată pentru a îmbina principiile rețelelor IP care sunt relativ ieftine, robuste și scalabile cu principiile rețelelor ATM care oferă servicii QoS. Principala trăsătură MPLS este că simplifică transmiterea pachetelor prin folosirea unui mecanism bazat pe etichete. Astfel, fiecare pachet poartă o etichetă și este transmis mai departe pe baza aceste etichete. Etichetele au o valabilitate locală (la nivel de legătură) și sunt distribuite folosind protocoale de distribuire. În rețelele MPLS se pot folosi atât CR-LDP [31] cât și RSVP-TE [4] pentru a realiza distribuirea etichetelor pe baza principiilor din ingineria traficului și QoS.

Dirijarea în MPLS se poate realiza în mod pas-cu-pas sau căile se pot stabili în mod explicit [49]. Setarea explicită implică enumerarea nodurilor traversate de o cale. Aceasta înseamnă că MPLS este un instrument deosebit de folositor pentru emularea rețelelor orientate pe conexiuni în rețele de tip datagramă [40]. Așadar, deși rețeaua transportă datagrame, MPLS poate oferi multe dintre avantajele unei rețele orientate pe conexiune, adică servicii QoS.

2.4 Rețele active

În mod tradițional nodurile unei rețele nu efectuează modificări asupra pachetelor ce trec prin ele, cu excepția modificărilor necesare în procesul de dirijare sau în controlul congestiei. Aceste rețele sunt considerate pasive. În 1994 DARPA a inițiat un proiect numit rețele active, prin care li se oferă utilizatorilor posibilitatea de a “programa” rețeaua [62, 61, 47]. Rețelele active pot fi active în două feluri [62]:

1. nodurile rețelei pot efectua modificări asupra pachetelor care trec prin ele
2. utilizatorii pot “programa” rețeaua prin crearea unor programe care să facă modificările menționate anterior.

Implementarea rețelelor active se poate face în trei feluri:

- **Noduri active.** Cu această abordare pachetele conțin doar niște referințe la funcții care se află în nodurile active. Pachetele sunt de asemenea active în sensul că ele sunt cele care decid care funcții să le modifice conținutul, însă codul acestor funcții există în nodurile active. Exemple de astfel de implementări sunt: o arhitectură de rețele active propusă de *Georgia Institute of Technology* [6], *Distributed Code Caching for Active Networks* (DAN) [16] și *Active Network Transport System* (ANTS) [67].
- **Pachete active.** În această abordare codul este transportat în interiorul pachetului. Nodurile sunt și ele active datorită faptului că permit executarea codului transportat de pachete. Exemple de pachete active sunt: *Smart Packets* [57], *Active IP Option* [68] și arhitectura M0 [5].
- **Pachete și noduri active.** A treia abordare este o combinație a primelor două în care pachete active transportă o parte mai mică din cod iar partea mai consistentă-complexă se află în nodurile active. Exemple ale unei astfel de implementări sunt: arhitectura *SwitchWare* [2], *Programming Language for Active Networks* (PLAN) [29] și proiectul *Netscript* [70].

2.5 Agenți mobili

Un agent mobil este un program care acționează în numele altora într-o manieră autonomă, proactivă și reactivă și manifestă abilități de învățare, cooperare și mobilitate [27, 11]. Au fost create numeroase tehnologii având la bază conceptul de agenți mobili. Unele se bazează pe propriile limbaje de programare, pe când altele folosesc limbaje de programare existente ca de exemplu Java sau Tcl-Tk, ambele create de Sun Microsystems. Tabelul 2.1 cuprinde cele mai cunoscute implementări ale conceptului de agenți mobili.

În general fiecare agent se mișcă prin rețea independent de ceilalți agenți, se deplasează de la un nod la altul și poate menține valorile unor variabile în tot acest timp. Pentru a face toate aceste lucruri, un agent mobil trebuie să poarte cu el tot acest fir al execuției (specificarea momentului și locului în care se face deplasarea, modalitatea și decizia de a comunica sau nu cu alți agenți mobili, etc). Așadar, majoritatea tehnologiilor generează agenți mobili “grași” și

Table 2.1: Sisteme care implementează agenți mobili

Sistem	Bazat pe	Organizație
<i>Aglets</i>	Java	IBM Japonia
<i>AgentTcl</i>	Tcl/tk	Dartmouth College
<i>Telescript</i>	custom	General Magic, Inc.
<i>Mole</i>	Java	Universitatea din Stuggart
<i>Ara</i>	Tcl	Universitatea din Kaiserslautern
<i>Tacoma</i>	Tcl	Universitatea Cornell și Universitatea din Tromso

complecși. Pentru a evita acest lucru am folosit în această teză o paradigmă înrudită cu agenții mobili numită WAVE [53, 56, 54, 55]. Această tehnologie diferă de celelalte implementări ale agenților mobili deoarece limbajul de programare folosit de această paradigmă este compus din semantici care descriu mai degrabă “ce trebuie făcut într-o rețea” decât să implementeze cod pentru agenți. Codul scris în acest limbaj este cel care va instrui nodurile din rețea care îl interpretează să genereze entități mici numite *wave*-uri asemănătoare agenților mobili. Această tehnologie are un mecanism de control spațial recursiv care coordonează populațiile micilor “agenți mobili”. Aceste mecanisme de control sunt implementate în nodurile care interpretează codul WAVE. Ele sunt distribuite în rețea. Așadar, dimensiunea și complexitatea *wave*-urilor este mult diminuată în comparație cu implementările tradiționale ale agenților mobili.

O altă proprietate importantă a tehnologiei WAVE este aceea că crează dinamic rețele virtuale distribuite peste rețelele reale. Acesta permite tehnologiei WAVE, spre deosebire de alte tehnologii care implementează agenți mobil, să atribuie cod și informații atât nodurilor cât și legăturilor din aceste rețele virtuale care pot fi navigate, procesate și modificate de către *wave*-uri.

Capitolul 3

Probleme și soluții în dirijarea QoS

Dirijarea QoS este un subiect îndelung dezbătut în comunitatea de cercetători datorită multiplelor probleme care sunt încă nerezolvate. Aceste probleme sunt în general generate de încărcarea pe care o introduce QoS în procesul de dirijare. QoS introduce încărcare la nivel de comunicație datorită creșterii volumului de informație necesar pentru o dirijare QoS. În comparație cu dirijarea tradițională care folosește o singură metrică (numărul de noduri sau un cost administrativ), dirijarea QoS folosește mai multe metrici ca de exemplu întârzierea transmiterii pe fiecare legătură (*delay*), fluctuația întârzierilor (*jitter*), lățimea de bandă (*bandwidth*), etc. Toate aceste informații trebuie să fie făcute cunoscute pentru ca procesul de dirijare QoS să poată calcula o cale. În mod tradițional, determinarea unei căi QoS se realizează în nodul sursă. Calcularea unei astfel de căi, bazate pe mai multe metrici/atribute este o problemă complexă (NP-completă în cazul în care sunt considerate două sau mai multe metrici (independente) aditive sau-și multiplicative [65, 64]). De aceea se consideră că dirijarea QoS introduce și o încărcare computațională. Mai mult decât atât, toate aceste informații trebuiesc stocate în fiecare nod al rețelei, asta însemnând încărcare de stocare.

S-au propus multiple soluții pentru toate aceste probleme. Dintre ele eu m-am concentrat asupra dirijării ierarhice care vine ca o soluție pentru toate cele trei încărcături introduse de QoS, și asupra calculării unei căi bazate pe mai multe constrângeri. De asemenea am prezentat posibile soluții oferite de tehnologii ca agenți mobili și rețele active.

3.1 Dirijare ierarhică

Pentru a face dirijarea în rețele mari scalabilă, acestea se organizează ierarhic. Întreaga rețea este împărțită în mai multe domenii, iar nivelele ierarhice constau în reprezentarea agregată a domeniilor de la nivelul inferior. În acest fel se reduce încărcarea la nivel de comunicare datorită faptului că informația necesară dirijării este distribuită în mod detaliat doar în interiorul unui domeniu, iar în mod agregat între domenii diferite. Procesul de calculare al unei căi este de asemenea redus la găsirea unei căi care să traverseze fiecare domeniu. Aceste căi sunt ulterior concatenate pentru a obține calea finală. Mai mult decât atât, distribuirea între domenii doar a unei forme agregate a informației necesare dirijării are atât avantajul securității cât și a reducerii încărcării de stocare.

Există mai multe metode de agregare [39, 3]. Câteva sunt prezentate în Figura 3.1. Aceste metode de agregare satisfac în proporții diferite compromisul dintre precizie și compactitate. La una dintre extreme se află reprezentarea *Full-Mesh* care oferă cea mai precisă agregare, însă necesită un efort de comunicare prea mare dacă este folosit cu metode de dirijare tradiționale (centralizate în nodul sursă). La cealaltă extremă se află metoda numită *Symmetric Node*, care este cea mai compactă metodă de agregare, însă nu reflectă în mod adecvat topologiile asimetrice.

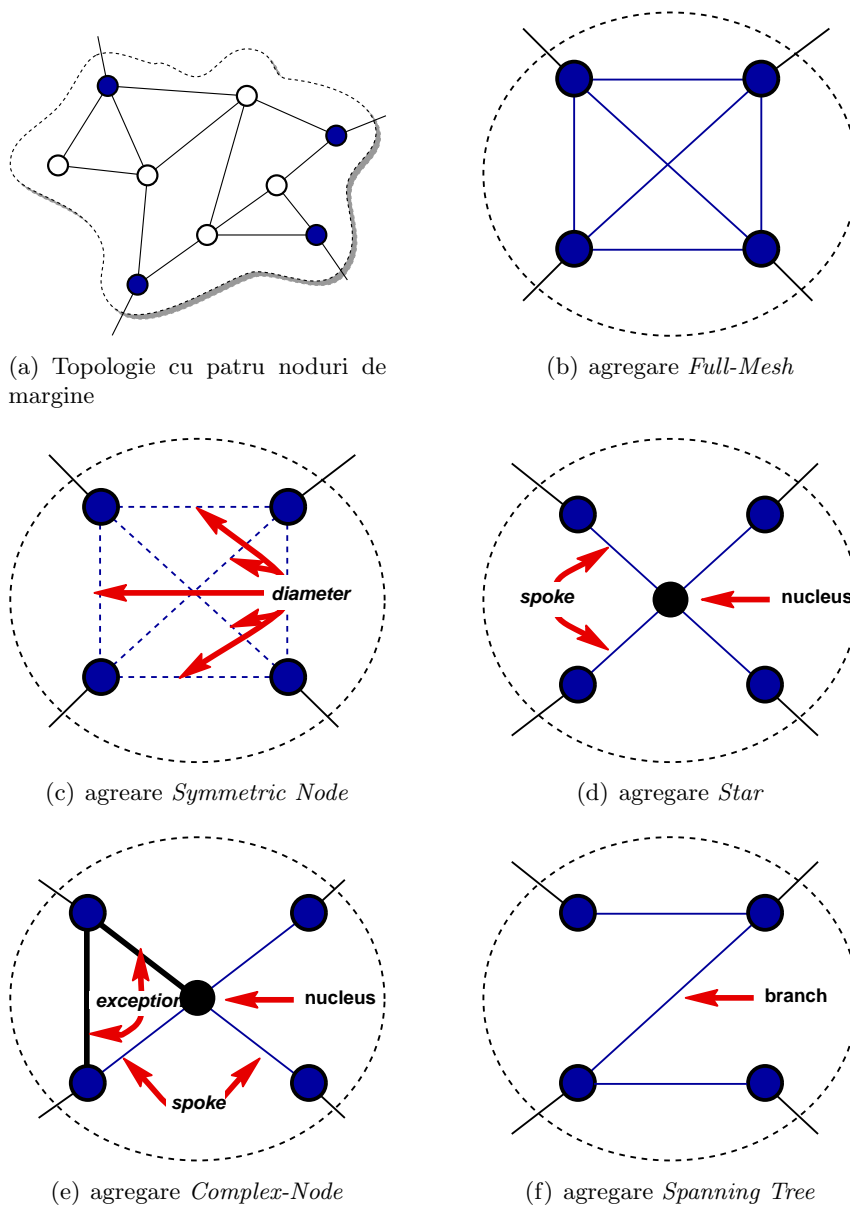


Figure 3.1: Exemple de agregare topologică

Principalele protocoale ierarhice care realizează dirijare QoS sunt:

- ***Private Network-to-Network Protocol (PNNI)*** [3]. care este protocolul folosit în rețelele ATM. Avantajele acestui protocol sunt că este singurul protocol ierarhic cu QoS care a fost standardizat și implementat și oferă scalabilitate prin permiterea a 104 nivele ierarhice. Dezavantajele PNNI constau în faptul că procesul de căutare al unei căi este distribuit neuniform între nodurile rețelei, iar metodele de agregare folosite duc la obținerea unei informații imprecise. O astfel de imprecizie poate conduce la o folosire inefficientă a resurselor din rețea.
- ***Hierarchical Distribution Protocol (HDP)*** [22] este o propunere de protocol de dirijare ierarhic pentru rețele MPLS. Principalul avantaj HDP este că calcularea unei căi se realizează în paralel la fiecare nivel ierarhic. Acest lucru însă generează prea multe mesaje. O altă problemă este faptul că deși în descrierea algoritmului se menționează folosirea unei metode de agregare, este dificil de determinat modalitatea de aplicare a acesteia. Mai mult

decât atât, calcularea căii care începe de la vârful ierarhiei face ca o astfel de metodă de agregare să fie folosită ineficient (chiar dacă ea există) datorită faptului că informația necesară procesului de dirijare devine prea veche în momentul în care protocolul ajunge la nivele inferioare ale ierarhiei, adică la nivelul fizic (la nodurile și legăturile reale).

- **Viewserver** [1] este o altă propunere de protocol ierarhic de dirijare care dorește să înlăture problema introducerii impreciziei de către metodele de agregare. De aceea calcularea căii se realizează de către nodul sursă, care adună toate informațiile necesare dirijării prin traversarea ierarhiei spre vârf pentru a găsi nodul care “vede” atât sursa cât și destinația, urmată de o traversare în jos a ierarhiei pentru a colecta informațiile necesare. Principalul dezavantaj al acestui protocol este că timpul de căutare al unei căi crește considerabil datorită faptului că întreaga cale este calculată de către un singur nod. În același timp, colectarea tuturor informațiilor necesare dirijării într-un singur loc nu e o soluție scalabilă în cazul rețelelor mari.
- **Clearing House** [12] este o propunere de protocol ierarhic de dirijare care rezervă în avans anumite resurse pentru a reduce timpul total necesar procesului de rezervare. Aceste rezervări se bazează pe niște predictorii Gausieni care estimează folosirea viitoare a lățimii de bandă. Performanța unui astfel de protocol depinde în mare măsură de algoritmul folosit pentru precizarea traficului.

3.2 Dirijarea bazată pe constrângeri multiple

Problemele de dirijarea bazată pe constrângeri multiple pot fi împărțite în mai multe clase, în funcție de numărul și tipul constrângerilor. De obicei, constrângerile QoS se împart în două clase principale: aditive (care includ și constrângerile multiplicative datorită faptului că s-a arătat în [46] că acestea pot fi reduse la constrângeri aditive) și neaditive (conțin constrângerile concave și convexe). În cazul constrângerilor aditive, valoarea unei căi se determină prin însumarea atributelor asociate legăturilor care fac parte din calea respectivă. Pentru constrângerile neaditive, valoarea unei căi este minimul sau maximul atributelor asociate legăturilor care fac parte din calea respectivă. De aceea întotdeauna problemele cu constrângeri neaditive se soluționează prin eliminarea din graful de căutare a legăturilor care nu satisfac constrângerile. De aceea majoritatea cercetătorilor consideră doar constrângerile aditive în cadrul problemelor bazate pe constrângeri multiple [10, 37].

3.2.1 Soluții propuse pentru dirijarea bazată pe constrângeri multiple

O clasificare a acestor soluții este prezentată în [46] și cuprinde următoarele clase:

- **compunerea liniară bazată pe Langragian:** Se compune un atribut care să reprezinte liniar toate celelalte atribute, astfel:

$$W^*(P) = \sum_{i=1}^m d_i \cdot w_i(u, v) \quad (3.1)$$

Fiecare “greutate” d_i a unui atribut poate fi determinată și/sau modificată astfel încât să se obțină cea mai bună cale finală. Cu toate acestea, determinarea căii finale se realizează ținând cont de un singur atribut care nu garantează selectarea unei căi optime în relație cu atributele inițiale.

- **abordarea Fallback Routing:** Selectarea căii se face considerând fiecare atribut QoS, pe rând în speranța că calea selectată pe baza unui singur atribut va satisface și celelalte constrângeri.

- **attribute QoS dependente:** Mai multe attribute dependente pot fi redate într-un singur atribut. O astfel de problemă poate fi rezolvată în timp polinomial.

Soluțiile concrete prezentate și descrise în teză sunt:

- aproximarea lui Jaffe [30]
- aproximarea lui Chen [10]
- TAMCRA [15] și SAMCRA [63]
- O soluție euristică pentru MCOP [35]
- Un algoritm randomizat [34]
- Euristica căii limitate [71]
- A*Prune [42]

3.2.2 Dirijarea ierarhică bazată pe constrângeri multiple

Într-un context ierarhic, problema dirijării bazate pe constrângeri multiple devine și mai complexă [38, 33, 43]. Aceasta se datorează dificultății de a desemna căi parțiale pentru a construi reprezentarea agregată.

Metodele convenționale folosite pentru a desemna o cale parțială optimă sunt [39, 38, 33]:

- **Folosirea unui singur atribut.**
- **Folosirea celui mai bun scenariu.** Căile desemnate pentru agregare vor fi căi virtuale care au cele mai bune attribute.
- **Folosirea celui mai rău scenariu.** Căile desemnate pentru agregare vor fi căi virtuale care au cele mai rele attribute.

Majoritatea soluțiilor propun însă folosirea a doar două constrângeri: lățimea de bandă și întârzierea în transmitere [33, 43].

3.3 Noi abordări pentru problema dirijării QoS

Mai multe grupuri de cercetare au propus diferite mecanisme de dirijare bazate pe agenți mobili [19, 45, 13, 25, 32] sau rețele active [44, 66, 23].

3.3.1 Dirijare bazată pe agenți mobili

Majoritatea propunerilor de dirijare bazată pe agenți mobili folosesc conceptul de inteligența grupului (*swarming intelligence*) care este inspirat din comportamentul coloniilor de insecte ca de exemplu furnici, albine, etc. Acesta a fost considerat a fi un model bun pentru o dirijare total descentralizată, și în același timp robustă și adaptivă. Elementul de bază al acestui concept este comportarea/inteligența emergentă în care prin coexistența și interacțiunea mai multor componente autonome care conțin doar primitive simple se poate obține un comportament complex și adeseori chiar inteligent. Principiul care stă la baza unei astfel de interacțiuni se numește *stigmergie*¹ sau comunicare prin intermediul mediului înconjurător. Principalele avantaje ale

¹Acest termen a fost introdus de către Grasse în [26] pentru a descrie comunicarea indirectă dintre anumite entități prin modificarea mediului lor înconjurător.

acestor abordări sunt adaptabilitatea, eficiența, autonomia, robustețea și toleranța la erori, extensibilitatea, distributivitatea și paralelismul. Ele sunt descrise mai detaliat atât în teză cât și în [7, 19, 32, 41].

Prima propunere de dirijare folosind agenți mobili a fost AntNet [19]. Aceasta se inspiră din comportamentul furnicilor care lasă în urma lor o substanță mirositoare (*feromon*) pentru a ghida următoarele furnici care vor trece pe același drum, către calea cea mai scurtă în găsirea hranei. Această nouă modalitate de dirijare a servit ca și model pentru multe alte cercetări [32, 45, 36]. Autorii lucrării [58] au extins acest model pentru a soluționa problema *stagnării* care apare când starea rețelei este neschimbată iar algoritmul de dirijare rămâne blocat într-o stare de optim local fiind incapabil să găsească alte căi mai eficiente.

Au fost de asemenea introduse diverse scheme de cooperare între agenții mobili, în scopul găsirii de căi optime [45, 36, 69]. Alte cercetări au încercat să modeleze matematic comportamentul agenților mobili pe parcursul procesului de dirijare. Din această perspectivă au fost analizate mai ales ritmul de creștere al populației de agenți mobili [59] cât și modul de distribuire al agenților mobili în rețea și probabilitatea de succes al acestora [48]. Aceste rezultate pot constitui puncte de pornire pentru viitoare sisteme de dirijare bazate pe agenți mobili.

Toate aceste soluții de dirijare reprezintă cea mai bună soluție distribuită (și care nu generează cicluri) de dirijare. Dezavantajele acestor sisteme sunt posibile probleme de securitate și dificultatea demonstrării eficienței lor. Testele efectuate până acum sugerează [19] că agenții mobili sunt net superiori atât algoritmilor de dirijare statici cât și celor adaptivi.

3.3.2 Dirijare bazată pe rețele active

Rețelele active au fost de asemenea propuse ca și soluție la problema dirijării QoS [44, 66, 23]. Astfel, autorii lucrării [66] argumentează ca dirijarea *activă* (adică dirijarea folosind rețele active) este un bun candidat pentru oferirea de garanții QoS datorită faptului că permite utilizatorilor să își aleagă singuri constrângerile QoS și metodele de căutare a căilor. O altă propunere descrisă în [44] este de a folosi rețele active pentru a stabili și controla tunele MPLS (LSPs), pentru a efectua operații pe stiva etichetelor MPLS, și pentru a implementa metode eficiente de restaurare. Dezavantajul acestor propuneri este acela că tind să implice utilizatorii (fie ei și doar ISP-uri) în procesul de dirijare. Problema este că nu orice utilizator știe cele mai bune metode de a găsi căi optime sau valorile optime ce trebuie asociate parametrilor QoS, iar unii dintre acești utilizatori nici nu vor să se implice în procesul de trimitere a pachetului către destinație. Chiar și în cazul în care utilizatorul știe și vrea să se implice în acest proces, el nu are o viziune globală a rețelei, astfel încât să se afle în poziția de a lua o decizie bună.

Capitolul 4

Rețele active sau agenți mobili

Cele două tehnologii se aseamănă prin faptul că ambele folosesc conceptul de cod mobil și se deosebesc prin arhitecturi diferite. Principala diferență dintre cele două tehnologii din punctul de vedere al autorului este modalitatea de folosire a codului și efectul executării acestui cod asupra traficului din rețea. Astfel, rețelele active au un efect direct asupra traficului din rețea datorită faptului că în acest caz codul traversează rețeaua deodată cu pachetele sau le așteaptă în diverse noduri pentru a efectua asupra lor operații variind de la redirectarea acestor pachete până la modificarea conținutului lor.

Agenții mobili sunt mult mai pasivi din acest punct de vedere datorită faptului că ei traversează rețeaua independent de traficul din rețea cu scopul principal de descoperire și colectare a informațiilor distribuite în rețea. Influența asupra traficului se produce doar în cazul în care agenții mobili sunt folosiți în procesul de dirijare când aceștia pot modifica modul în care traficul traversează rețeaua.

Din această perspectivă agenții mobili reprezintă un instrument mult mai puternic datorită faptului că pot fi folosiți în găsirea căilor optime înainte ca acestea să fie folosite pentru a dirija traficul curent, la fel cum se întâmplă în majoritatea modelelor de dirijare QoS. De asemenea, agenții mobili pot găsi calea realizând căutarea într-o manieră distribuită și paralelă.

În contextul rețelelor active, traficul existent în rețea poate fi redirectat atât de către codul din pachetele active care traversează rețeaua odată cu celelalte pachete, cât și de către nodurile active prin care trec pachetele. Diferența este ca în rețelele active decizia de dirijare trebuie luată pe loc, în timp ce pachetele trec prin rețea. Aceasta se poate realiza doar într-o manieră pas-cu-pas și prezintă două mari dezavantaje:

1. decizia de dirijare este luată pe baza informațiilor asupra rețelei care sunt accesibile din nodul curent.
2. atât decizia de dirijare cât și întreg procesul de redirectare al pachetelor trebuie să fie realizate la o viteză asemănătoare vitezei de transmitere prin cablu (*wire speed*). Aceasta poate constitui un impediment pentru majoritatea implementărilor de rețele active care folosesc Java datorită portabilității și securității oferite de aceasta.

Există totuși numeroase probleme [17, 18], altele decât cele de dirijare QoS, care pot fi foarte elegant rezolvate folosind rețele active.

4.1 Integrarea MPLS cu rețele active

În continuare propun și eu o astfel de soluție pentru rezolvarea inabilității MPLS de a comuta pachete mai sus de nivelul doi. Acest lucru este dezirabil în contextul în care MPLS, care cunoaște la ora actuală un succes sporit, va pătrunde (există deja MPLS în echipamentele de

rețea, chiar și în cele destinate rețelelor de la marginea Internetului) în rețelele de la marginea Internetului. În acest caz e necesară comutarea pachetelor la nivele superioare nivelului 2 în scopul facilitării unor servicii specifice acestor rețele, ca de exemplu zid de protecție (*firewall*), servere web intermediare (*Web proxies*), rutere pentru trimitere multiplă (*multicast routers*), sau alte servicii similare care necesită accesarea informațiilor din pachet pe baza cărora să se poată decide dacă pachetul se trimite mai departe sau se abandonează. Soluția unor astfel de probleme poate consta în integrarea MPLS cu rețele active după cum este ilustrat în Figura 4.1.

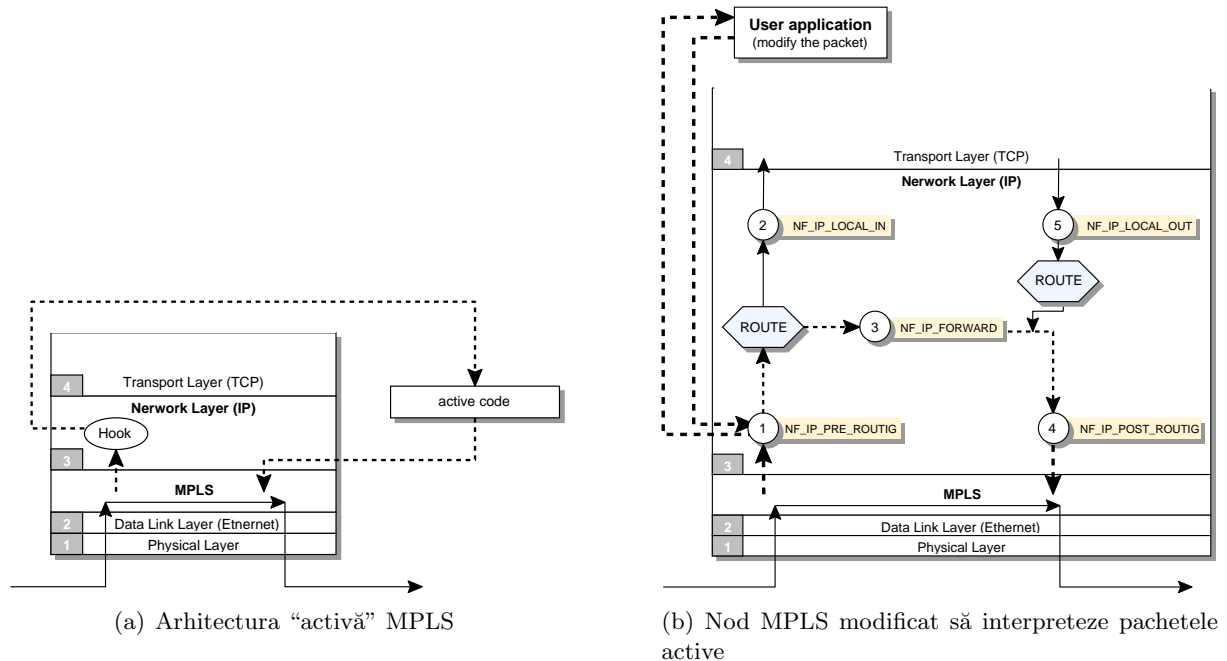


Figure 4.1: Procesarea pachetelor în nodurile dintr-o rețea activă MPLS

Pachetele active sunt preluate din nivelul MPLS și sunt tratate separat. Distincția între pachetele active și celelalte pachete se face pe baza etichetei MPLS. După procesare, pachetele active se reîntorc la mecanismul MPLS de trimitere a pachetelor unde li se vor atribui o noua etichetă MPLS și vor fi trimise mai departe.

Această propunere a fost implementată pe sistemul de operare *Linux*, pe care s-a emulat o rețea de test folosind *User-Mode Linux* [20]. Pentru a configura un nod MPLS "activ" a fost folosit utilitarul *Netfilter* [52, 51, 50]. Pentru trimiterea pachetelor MPLS a fost folosit utilitarul *open-source* de la Sourceforge numit *mpls-linux*¹. Setarea căilor MPLS s-a realizat în mod static.

4.2 Exemple de aplicații

Pentru a demonstra necesitatea unei astfel de integrări dintre MPLS și rețelele active am descris în cadrul tezei două probleme care pot beneficia de o astfel de soluție.

4.2.1 MPLS Web Switching

Această soluție este prezentată în [21] pentru problema încărcării serverelor web. Un grup de servere web e folosit pentru a face față unui număr mare de cereri. Un dispecer este așezat în fața grupului de servere web pentru a atribui cereri pe rând fiecărui server din grup.

Pentru soluționarea problemei continuității TCP (adică toate pachetele dintr-o conexiune TCP trebuie să ajungă la același server web) este folosită o etichetă dedicată, L_{syn} , care

¹<http://sourceforge.net/projects/mpls-linux/>

marchează începutul unei conexiuni TCP. Așadar, pentru fiecare nouă conexiune, dispecerul atribuie a nouă etichetă, L_1 , corespunzătoare cu serverul cu care se va realiza conexiunea. Toate pachetele următoare ale aceleiași conexiuni vor conține această etichetă. Întreg acest scenariu este ilustrat în Figura 4.2.

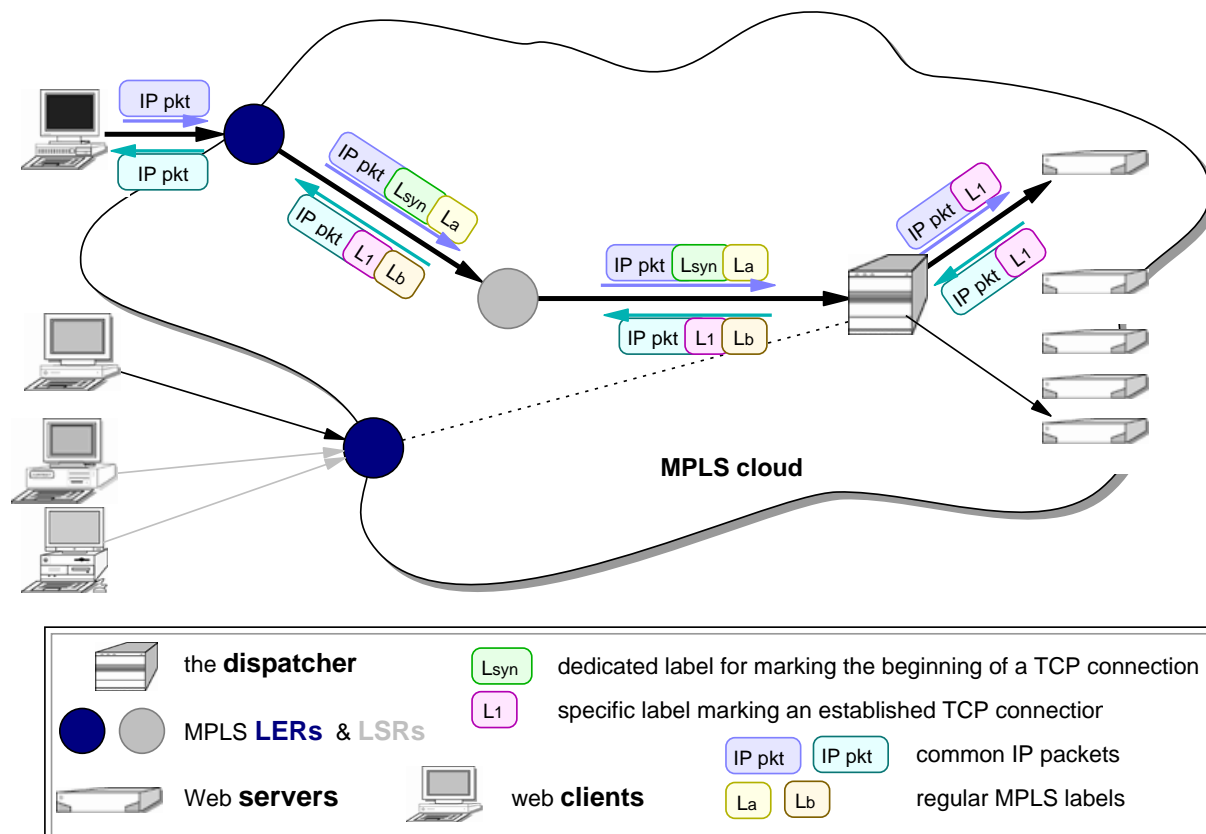


Figure 4.2: Scenariul *MPLS Web Switching*

Prin folosirea rețelelor active, acest mecanism poate fi implementat fără ajutorul etichetelor L_{syn} și L_1 . Ca și prim pas, nodurile prin care se intră în domeniul MPLS trebuie să poată interpreta pachete active. În momentul în care primul pachet dintr-o conexiune TCP, numit *SYN*, intră în domeniul MPLS, nodul de intrare îl redirecționează direct către dispecer care îl identifică ca fiind *SYN* și îl atribuie unui anumit server din grup, după care trimite un pachet activ care “instruiește” nodul de intrare în domeniul MPLS să redirecționeze toate pachetele acestei conexiuni (folosind DNAT (*destination network address translation*)) către serverul ales. Ultimul pachet al conexiunii TCP, numit *FIN* va șterge din înregistrările nodului de intrare în domeniul MPLS redirecționearea DNAT corespunzătoare.

Această nouă soluție oferă o comunicare mai directă între servere și clienți.

4.2.2 Licitări *on-line*

Un alt exemplu se referă la un server web care întreține licitări *on-line*, după cum este prezentat în Figura 4.3. Pentru un număr mare de clienți, serverul poate deveni prea încărcat.

Folosind rețele active, ofertele inferioare prețului de licitare pot fi selectate înainte de a ajunge la server, după cum este ilustrat în Figura 4.4.

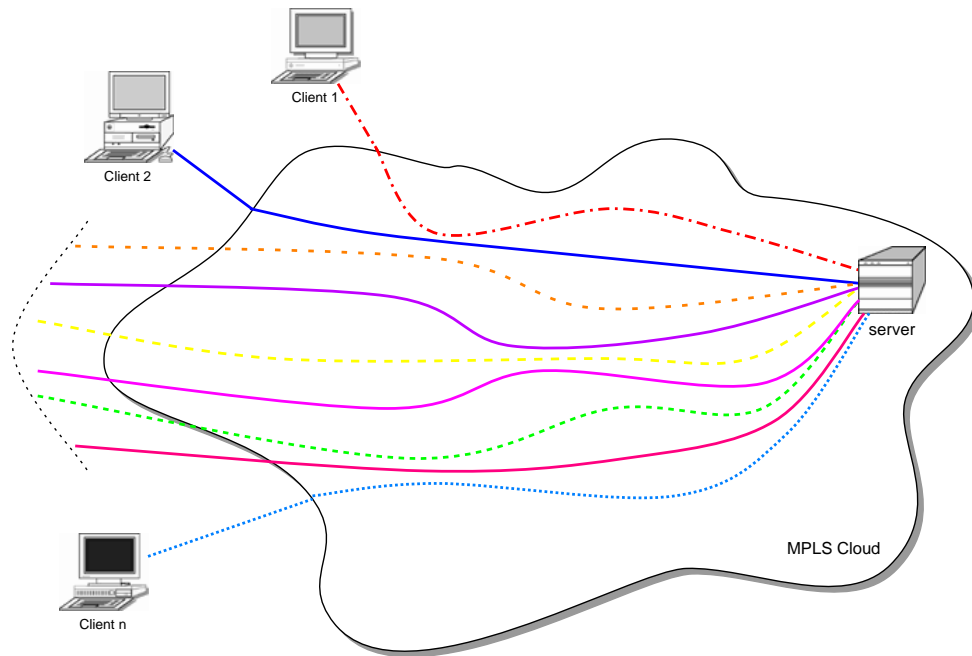


Figure 4.3: Scenariu de licitare *on-line*

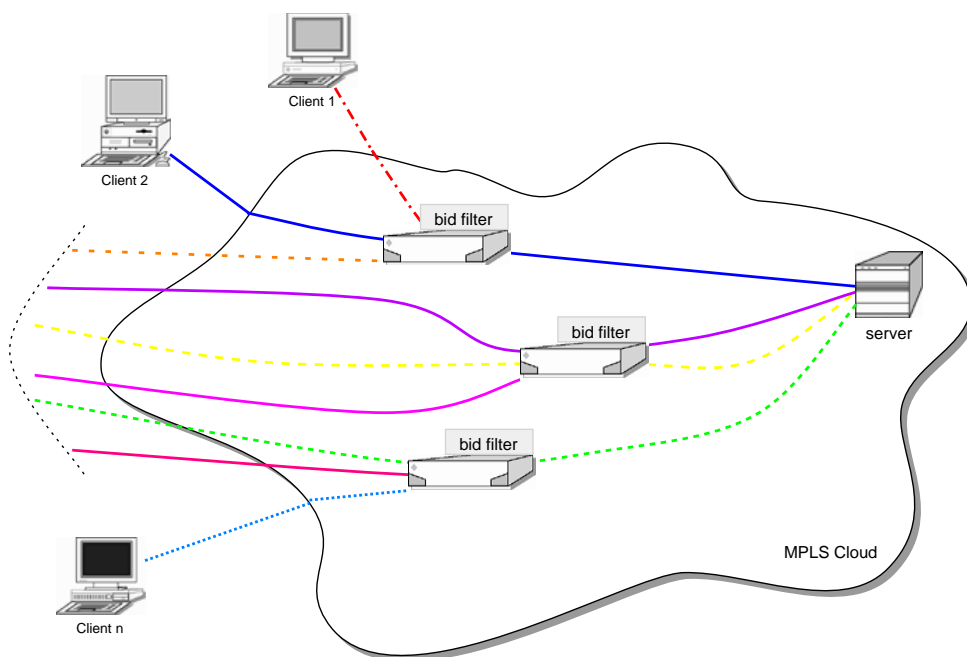


Figure 4.4: Licitare *on-line* folosind filtre de oferte

Capitolul 5

Dirijare scalabilă folosind agenți mobili în rețele MPLS

Am propus în acest capitol un nou protocol de dirijare ierarhic pentru rețelele MPLS numit *Macro-routing*. Acest protocol poate realiza, pe lângă dirijare, atât distribuirea etichetelor MPLS cât și rezervarea resurselor pentru căile determinate în procesul de dirijare.

5.1 Descrierea protocolului

Nodurile de control ale domeniilor din ierarhia *Macro-routing* trebuie să fie capabile să interpreteze agenți mobili. Fiecare nod de control este responsabil cu crearea reprezentării agregate a domeniului pe care îl coordonează. Această reprezentare agregată se construiește în patru pași după cum urmează:

1. Fiecare nod de la marginea unui domeniu (inclusiv nodul sursă și nodul destinație) activează un agent mobil care generează o populație de agenți mobili prin multiplicarea la fiecare nod din rețea cu numărul legăturilor incidente în căutarea tuturor căilor fiabile către toate celelalte noduri de la marginea aceluiași domeniu. Fiecare agent mobil înregistrează calea pe care o urmează, actualizând în același timp valoarea atributelor QoS considerate, odată cu fiecare nod vizitat. Dacă un agent mobil revizitează un nod, sau în cazul în care calea în curs de traversare nu satisface constrângerile QoS, acesta se va termina. Fiecare agent mobil care ajunge la destinație (în cazul nostru la un nod de la marginea domeniului curent diferit de cel care a inițiat căutarea) va transmite nodului de control calea înregistrată și atributele corespunzătoare.
2. Nodul de control alege căile optime dintre fiecare pereche de noduri de la margine domeniului.
3. Nodul de control creează o reprezentare agregată *Full-Mesh* cu ajutorul căilor selectate la pasul anterior. Atributele căilor folosite în agregare vor deveni atribute ale nodurilor când se vor calcula căi la următoarele nivele ale ierarhiei.
4. Celelalte căi, care nu au fost folosite pentru agregare, pot fi folosite ca și căi alternative și/sau pentru cazurile în care calea selectată nu mai este disponibilă.

Există trei faze principale ale protocolului *Macro-routing*. Ele sunt:

1. **Determinarea domeniilor care participă în procesul de dirijare.** Prima fază implică determinarea domeniilor prin care calea poate să treacă. Inițial nodul sursă inițiază o căutare către vârful ierarhiei pentru a găsi nodul părinte de la cel mai jos nivel care

să “vadă” atât sursa cât și destinația. Apoi, nodul părinte inițiază o căutare către baza ierarhiei (către toți copiii) pentru a determina domeniile care participă în procesul de dirijare.

2. **Calcularea efectivă a căii.** Toate nodurile de control ale domeniilor selectate în prima fază vor construi o reprezentare agregată prin metoda de agregare în patru pași descrisă anterior. Începând cu nivelul al doilea al ierarhiei, costurile nodale vor fi de asemenea luate în considerare în procesul de calculare a atributelor căilor traversate de agenții mobili. Acest proces se va repeta la fiecare nivel al ierarhiei. Nodul de control al domeniului din vârful ierarhiei va selecta o cale din mulțimea tuturor căilor posibile între sursă și destinație.
3. **Setarea căii MPLS și rezervarea resurselor.** Pentru ca să se poată folosi calea finală obținută în faza anterioară trebuie ca etichetele MPLS să fie setate, iar resursele necesare să fie rezervate. Acest lucru se poate realiza atât prin folosirea unor instrumente existente sau folosind tot agenți mobili. Avantajele pentru folosirea agenților mobili sunt:
 - Există deja interpretoare de agenți mobili în toate nodurile de control. Așadar nu are rost să se mai instaleze/configureze alte instrumente adiționale.
 - Spre deosebire de *RSVP* care realizează rezervarea resurselor secvențial, folosind agenți mobili, aceeași rezervare se poate realiza în paralel pentru fiecare domeniu din ierarhie.
 - Rezervarea resurselor se poate realiza “ierarhic”, adică primele resurse rezervate vor fi cele de la vârful ierarhiei și continuând cu celelalte nivele până se ajunge la baza ierarhiei. Această strategie are avantajul de a putea substitui orice cale parțială care nu mai deține resursele din momentul dirijării cu una dintre căile alternative deja calculate.

De asemenea setarea unei căi ierarhice într-un domeniu MPLS se potrivește foarte bine cu stiva de etichete din pachetele MPLS. De aceea fiecare cale parțială poate fi tratată independent.

5.2 Rezultatele obținute în urma simulărilor

Pentru realizarea simulărilor s-au folosit topologii generate aleator folosind *Georgia Tech Internet Network Topology Models* (GT-ITM) [72]. Câteva exemple de astfel de topologii sunt prezentate în Figura 5.1.

5.2.1 Urmărirea evoluției populațiilor de agenți mobili

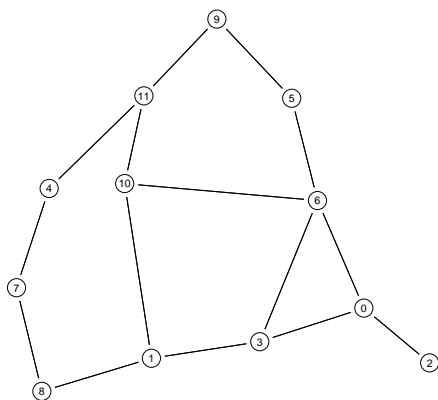
Pentru aceste teste am considerat mai multe topologii cu număr variat de noduri și/sau legături. Astfel, rezultatele prezentate în Figura 5.2(a) pentru topologiile cu 12 noduri și număr variat de legături arată că populația de *wave-uri* crește proporțional cunumărul de legături. Figura 5.2(b) arată că pentru topologiile cu foarte multe legături, numărul *wave-urilor* care se termină în cicluri (și deci nu contribuie la descoperirea de căi noi) este foarte mare.

Populația de agenți mobili (adică *wave-uri* în cazul nostru) a fost împărțită în clasele definite în Definiția 1.

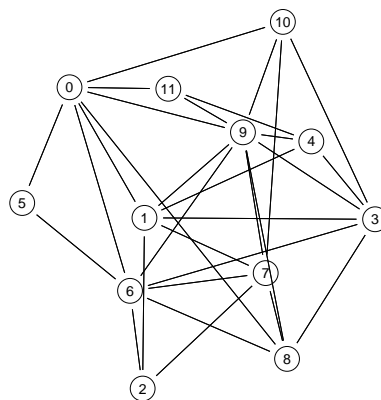
Definiție 1 *Wave-uri terminate cu succes, terminate în cicluri, și neterminate*

Fie n numărul nodurilor vizitate de către populația de wave-uri. Definesc:

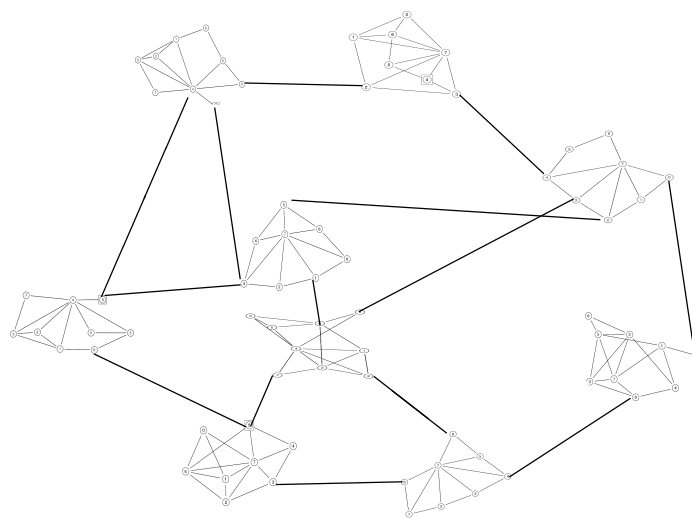
- **wave-uri terminate cu succes** - *wave-uri care ajung într-un nod de la marginea domeniului și deci au găsit o cale de lungime n - $B_w^{(n)}$ este numărul unor astfel de wave-uri;*



(a) Topologie cu conectivitate redusă

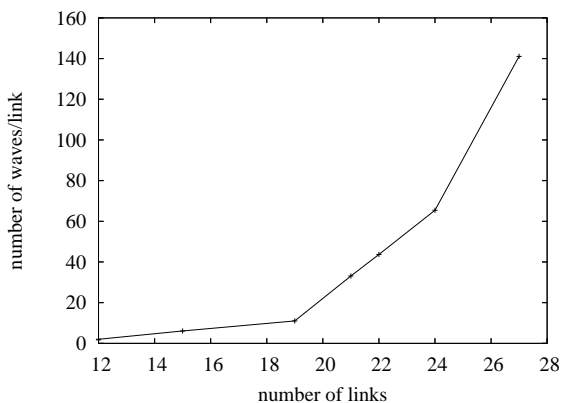


(b) Topologie cu conectivitate mare

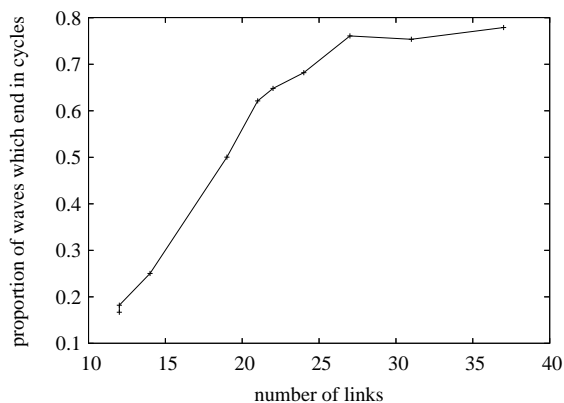


(c) Topologie ierarhică pe două nivele

Figure 5.1: Topologii folosite în simulări



(a) Numărul mediu de *wave*-uri/legătură



(b) Proporția *wave*-urilor care se termină în cicluri în raport cu numărul total de *wave*-uri

Figure 5.2: Evoluția populației de *wave*-uri ca funcție de număr de legături, în topologii cu 12 noduri

- **wave-uri terminate în cicluri** - wave-uri pentru care al n -lea nod vizitat a mai fost vizitat odată, deci wave-uri care au intrat în cicluri - $C_w^{(n)}$ este numărul unor astfel de wave-uri;
- **wave-uri neterminate** - wave-uri care își continuă căutarea deoarece încă nu au ajuns la un nod din marginea rețelei și nici nu au intrat în ciclu - $A_w^{(n)}$ este numărul unor astfel de wave-uri;

Numărul total de wave-uri care au vizitat n noduri este dat de formula:

$$T_w^{(n)} = B_w^{(n)} + C_w^{(n)} + A_w^{(n)} \quad (5.1)$$

Figura 5.3 prezintă numărul de wave-uri dintr-o rețea cu 13 noduri și 20 de legături ca și funcție de numere de noduri vizitate.

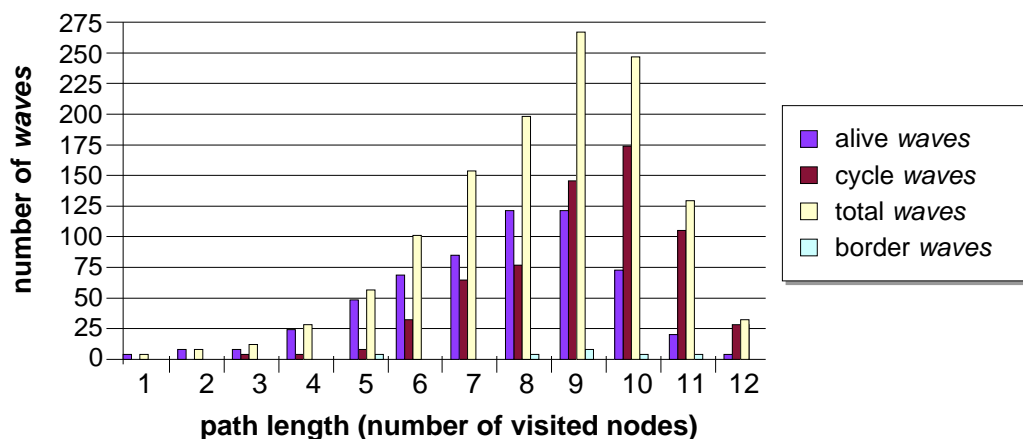


Figure 5.3: Numărul wave-urilor care se propagă dintr-un singur nod de la marginea domeniului

Aceste rezultate permit calcularea probabilității wave-urilor de a se termina în cicluri și efortul calculării unei căi de lungime n . Aceste două concepte sunt definite în ceea ce urmează.

Definiție 2 Probabilitatea wave-urilor de a se termina în cicluri

Probabilitatea wave-urilor de a se termina în cicluri odată cu vizitarea celui de-al n -lea nod este estimată ca fiind:

$$p_{cn} = \frac{C_w^{(n)}}{T_w^{(n)}} \quad (5.2)$$

Definiție 3 Efortul calculării unei căi de lungime n

Efortul calculării unei căi de lungime n este raportul dintre numărul wave-urilor care se termină în cicluri odată cu vizitarea celui de-al n -lea nod și numărul wave-urilor care nu s-au terminat încă:

$$E_n = \frac{C_w^{(n)}}{A_w^{(n)}} \quad (5.3)$$

Rezultatele testelor efectuate pentru determinarea probabilității wave-urilor de a se termina în cicluri sunt prezentate în Figura 5.4. Ele se apropie foarte mult de funcția,

$$F(x) = \begin{cases} 0, & \text{if } x \leq 3 \\ \frac{x-3}{N-3}, & \text{if } x > 3 \end{cases} \quad (5.4)$$

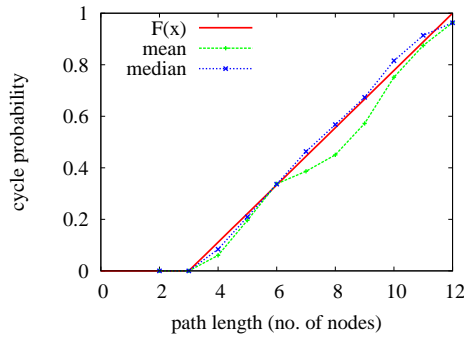


Figure 5.4: Probabilitatea *wave*-urilor de a se termina în cicluri pentru topologii cu 12 noduri și număr de legături variind de la 11 la 66 (a se vedea Definiția 2)

unde x este numărul nodurilor deja vizitate (deci lungimea căii) iar N este numărul total de noduri din domeniu. Valoarea 3 semnifică faptul că sunt necesare cel puțin 3 noduri pentru a exista cicluri.

Efortul calculării unei căi de lungime n indică pragul peste care protocolul devine ineficient. Astfel, Figura 5.5 arată efortul calculării unei căi mai lungi de 9 noduri este prea mare pentru topologiile de 12 noduri. Așadar, e nevoie de o modificare a protocolului care să îi asigure scalabilitatea.

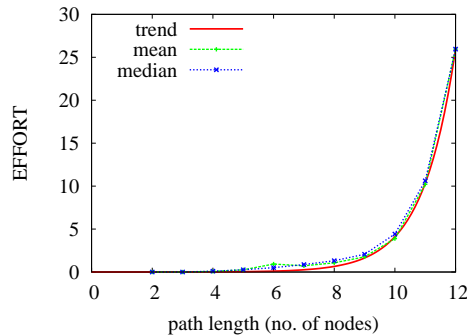


Figure 5.5: Efortul calculării unei căi de lungime n pentru topologii cu 12 noduri și număr de legături variind de la 11 la 66 (a se vedea Definiția 3)

5.2.2 Limitarea populației de *wave*-uri

Am introdus un parametru numit *lifespan* care să limiteze numărul de *wave*-uri generate pe parcursul procesului de dirijare. Astfel, orice *wave* care vizitează *lifespan* noduri se va termina. Algoritmul astfel modificat nu mai garantează obținerea unei căi optime (mai ales când o astfel de cale e mai lungă de *lifespan* noduri).

Următoarele teste verifică influența unui astfel de parametru asupra protocolului *Macro-routing*. Atributele folosite în această evaluare sunt descrise în continuare.

Definiție 4 *Eficiența Macro-routing*

Fie C_{opt} costul căii optimale dintre o anumită sursă și o destinație, iar C_{act} costul căii găsite de algoritmul modificat (prin adăugarea *lifespan*-ului) al protocolului *Macro-routing*. *Eficiența Macro-routing* se definește ca fiind:

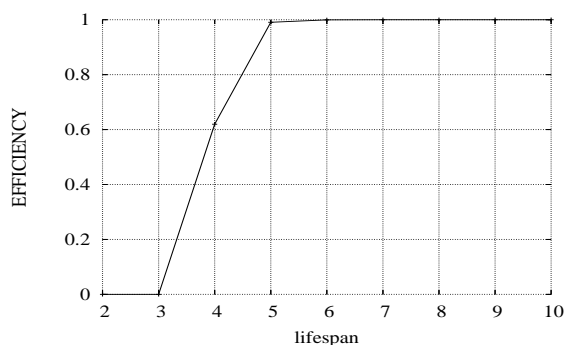
$$E = \frac{C_{opt}}{C_{act}}, \quad (5.5)$$

Definiție 5 *Eșec, succes, optim*

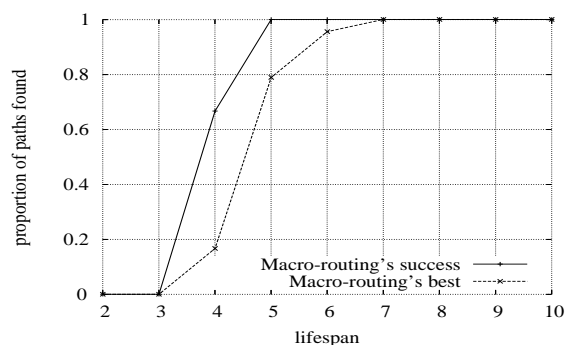
Fie E eficiența Macro-routing definită în Definiția 4. Definesc rezultatele Macro-routing astfel:

- eșec: nu s-a găsit nici o cale deoarece valoarea *lifespan* e prea mică: $E = 0$
- succes: s-au găsit una sau mai multe căi care satisfac constrângerile: $E > 0$
- optim: căile găsite includ calea optimă: $E = 1$.

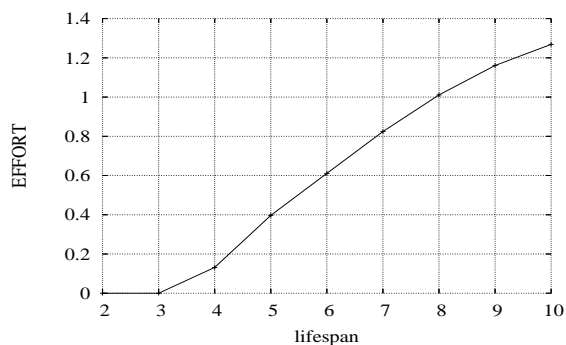
Următoarele teste au fost efectuate pe topologii ierarhice cu două nivele. Rezultatele obținute pe topologii de $12 \times 12 = 144$ noduri sunt prezentate în Figura 5.6.



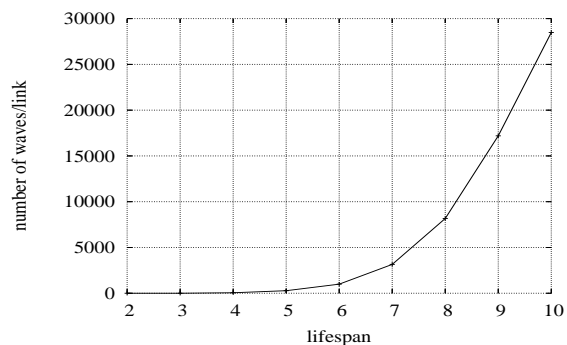
(a) Valuarea medie a eficienței *Macro-routing*



(b) Valuarea medie a proporției de *succes* și *optim*



(c) Valuarea medie a efortului *Macro-routing*



(d) Numărul mediu de *wave*-uri generate pe legătură

Figure 5.6: Performanța *Macro-routing* în cazul aplicării parametrului *lifespan*

Ele arată că există valori ale *lifespan* care reduc în mod considerabil populația de *wave*-uri fără a afecta eficiența protocolului *Macro-routing*.

Capitolul 6

Dirijare QoS ierarhică folosind constrângeri multiple

Acest capitol adresează problema dirijării ierarhice bazate pe constrângeri multiple. De aceea propun o nouă metodă de agregare numită *extended Full-Mesh*.

6.1 Descrierea metodei de agregare *extended Full-Mesh*

Această nouă metodă de agregare extinde reprezentarea *Full-Mesh* prin folosirea mai multor căi în loc de una singură pentru fiecare legătură dintre două noduri ale reprezentării agregate. Astfel, definesc aici câteva concepte folosite în descrierea (și lucrul cu) reprezentarea EFM.

Definiția 1 *intervalul EFM*

Fie

- g un graf descriind topologia domeniului, cu $g = \{V, L, B\}$, unde V este mulțimea nodurilor, L este mulțimea legăturilor, iar B este mulțimea nodurilor de la marginea rețelei, astfel încât $B \subset V$.
- $P_i(u, v)$ a i -a cale dintre nodurile de la marginea domeniului u și v .
- M numărul atributelor asociate cu fiecare legătură și/sau nod din graful g . Aceste atribute sunt folosite în calcularea căilor dintre orice pereche (u, v) de noduri de la marginea domeniului, $u, v \in B$.
- $n_{u,v}$ numărul căilor care leagă orice pereche de noduri de la marginea domeniului $u, v \in B$.
- C matricea costurilor pentru căile dintre u și v , unde $C_{i,j}^{u,v}$ reprezintă valoarea atributului j pentru calea $P_i(u, v)$. Această matrice are dimensiunea $n_{u,v} \times M$. Vectorul coloană $\xi_j^{u,v}$ reprezintă a j -a coloană din $C^{u,v}$ și se numește un vector de atribute deoarece conține valorile atributului j pentru toate căile posibile dintre u și v . Vectorul linie $\zeta_i^{u,v}$ reprezintă a i -a linie din $C^{u,v}$ și se numește un vector al căilor deoarece conține valorile tuturor atributelor pentru calea $P_i(u, v)$.

Definiția 2 *Căi posibile*

Fie $C^{u,v}$ o matrice a costurilor și $\Delta^{u,v}$ un vector de constrângeri de dimensiune M care conține valorile tuturor constrângerilor pentru toate atributele. Dacă pentru o anumită valoare a lui i și pentru oricare $0 < j \leq M$, se îndeplinește condiția:

$$\begin{aligned} & C_{i,j}^{u,v} \leq \Delta_j^{u,v}, \quad \text{unde al } j\text{-lea atribut este aditiv, multiplicativ sau convex} \\ \text{or } & C_{i,j}^{u,v} \geq \Delta_j^{u,v}, \quad \text{unde al } j\text{-lea atribut este concav} \end{aligned} \quad (6.1)$$

atunci calea $P_i(u, v)$ este una posibilă.

6.2 Limitarea vectorului căilor

Există anumite circumstanțe, ca de exemplu resurse suficiente pentru satisfacerea constrângerilor, care pot duce la o creștere a vectorului căilor în așa măsură încât ar putea influența negativ întregul proces de căutare/calcularea a unei căi. De aceea se propun două clase principale de selectare a căilor care vor fi folosite pentru reprezentarea agregată EFM.

În descrierea acestor metode s-au considerat următoarele cazuri speciale¹:

- Metodele se aplică unui interval EFM bi-dimensional (aceste metode se pot aplica pe orice interval n -dimensional, unde $n \geq 2$)
- Toate atributele sunt normalizate să fie în intervalul 0-1, unde 0 reprezintă valoarea optimă a atributului, iar 1 cel mai rea/defavorabilă valoare a acestui atribut.

6.2.1 Truncare intervalului EFM

Aceste metode de selecție a căilor încep eliminarea celor mai “rele” căi atâta timp până când mai rămân doar T căi. Dificultatea acestei probleme constă în determinarea celei mai “rele” căi bazate pe constrângeri multiple. S-au considerat trei metode posibile. Ele sunt reprezentate schematic în Figura 6.1.

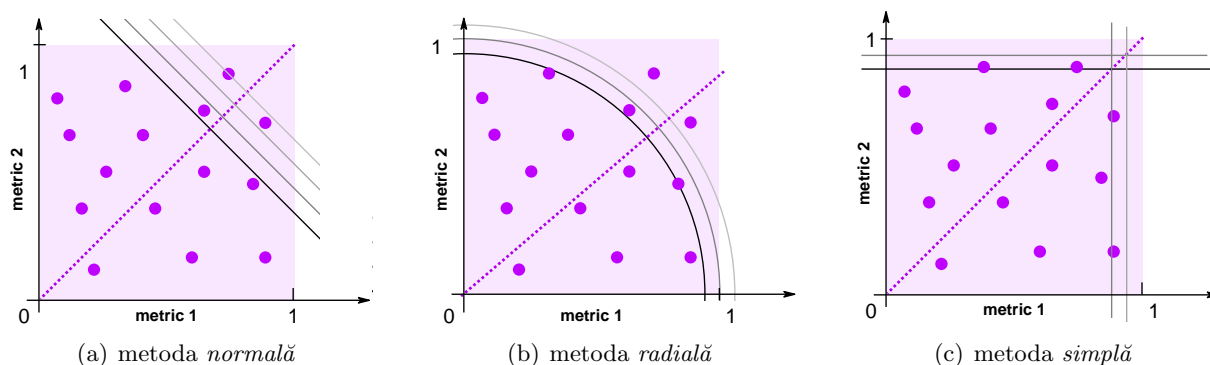


Figure 6.1: Metode de truncare a unui interval EFM bi-dimensional

Prima metodă, reprezentată în Figura 6.1(a) elimină pe rând toți vectorii căilor care se intersectează cu linia $f(x, t) = 2t - x$, care stă perpendicular pe diagonala dusă din colțul (1,1) în colțul (0,0), în timp ce parametrul t descrește de la 1 înspre 0, până la o anumită valoare $t = t_0$ la care mai rămân doar T vectori de căi. Vectorii căilor care satisfac inegalitatea $f(x, t_0) > y$ sunt cei selectați².

O altă metodă, prezentată în Figura 6.1(b), este similară metodei normale cu diferența că delimitarea dintre căile selectate și cele eliminate nu este făcută de către o linie, ci de un arc. Acest arc face parte din cercul cu centrul în $x = 0, y = 0$. Raza cercului, r , descrește începând de la $\sqrt{2}$ înspre 0 până când rămân doar T vectori de căi. Presupunem că în acel moment raza arcului este $r = r_0$, iar toți vectorii de căi din interiorul arcului sunt selectate. De aceea, un vector de cale este selectat dacă satisface inegalitatea $\sqrt{x^2 + y^2} < r_0$.

Figura 6.1(c) ilustrează a treia metodă de truncare, unde este eliminată cea mai rea cale determinată pe baza unui singur atribut. Atributul pe baza căruia se face selectarea căilor poate fi desemnat în mod aleator, sau folosind o metodă *round-robin* pe baza unei priorități. Alt atribut poate fi considerat pentru fiecare eliminare. În cazul în care se folosește un singur atribut în procesul de selecție, metoda se va numi *truncare singulară*.

¹Restrângerea problemei la aceste cazuri s-a făcut doar în scopul unei prezentări mai clare.

² x și y sunt două atribute în acest exemplu bi-dimensional.

6.2.2 Selecție random

O altă metodă de limitare a numărului de căi EFM este de a reduce densitatea vectorilor de căi din intervalul EFM folosind o selecție aleatoare. Astfel, T căi EFM sunt selectate în mod aleator dintre căile existente.

6.3 Rezultatele testelor

Metodele de selectare a căilor EFM au fost implementate și testate în cadrul reprezentării EFM. Această reprezentare agregată a fost folosită de către protocolul *Macro-routing*. Testele efectuate au ca și scop final atât determinarea celei mai optime dintre metodele descrise în secțiunea anterioară cât și compararea performanței *Macro-routing* cu alte protocoale. Pentru prezentarea acestor rezultate se vor folosi următoarele prescurtări:

- $TS[j]$ - pentru metoda TRUNCARE SINGULARĂ folosind atributul j ;
- TN - pentru metoda TRUNCARE NORMALĂ
- TR - pentru metoda TRUNCARE RADIALĂ
- QR - pentru metoda aleatoare (*quasi random*)

Au fost efectuate 30 de teste pe diferite topologii cu două nivele ierarhice. Pentru aceste teste s-au folosit două metrici aditive ($administrative\ cost \in [1, 15]$ și $delay \in [2ms, 45ms]$). Numărul căilor EFM a fost limitat la $T = 5$. Rezultatele obținute pentru o topologie de 20×20 noduri sunt prezentate în Figura 6.2.

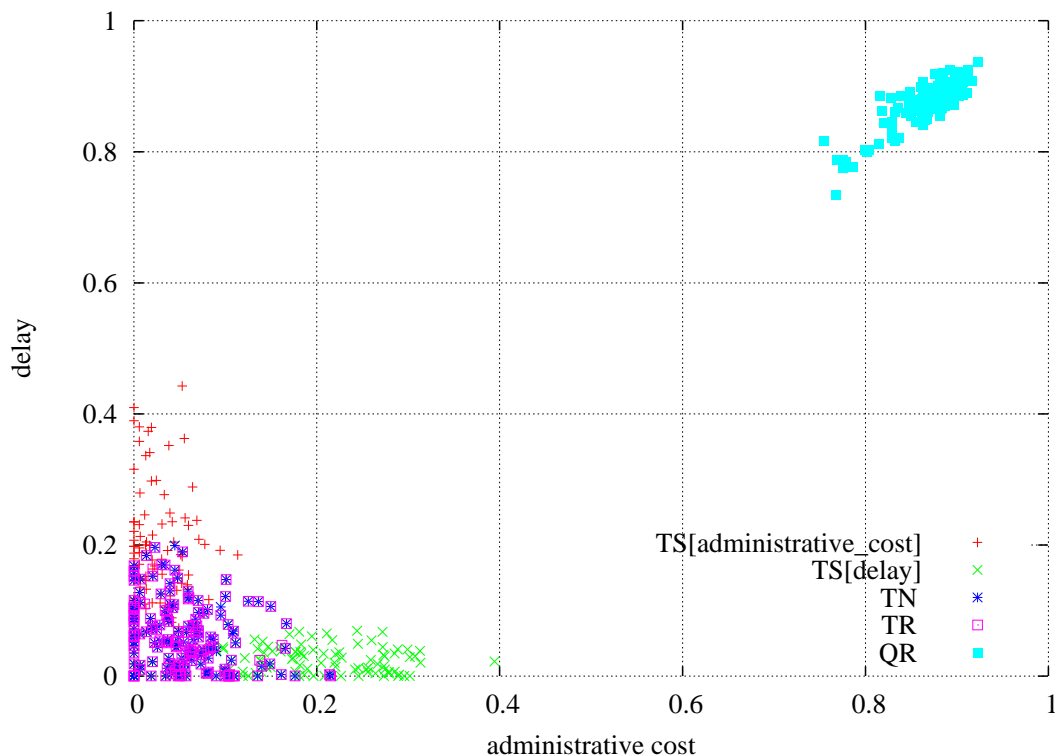


Figure 6.2: Compararea rezultatelor celor patru metode de selectare pentru topologii de 20×20 noduri

O prezentare separată asupra celor trei metode, și anume $TS[administrative_cost]$, $TS[delay]$, și QR , se poate observa în Figura 6.3.

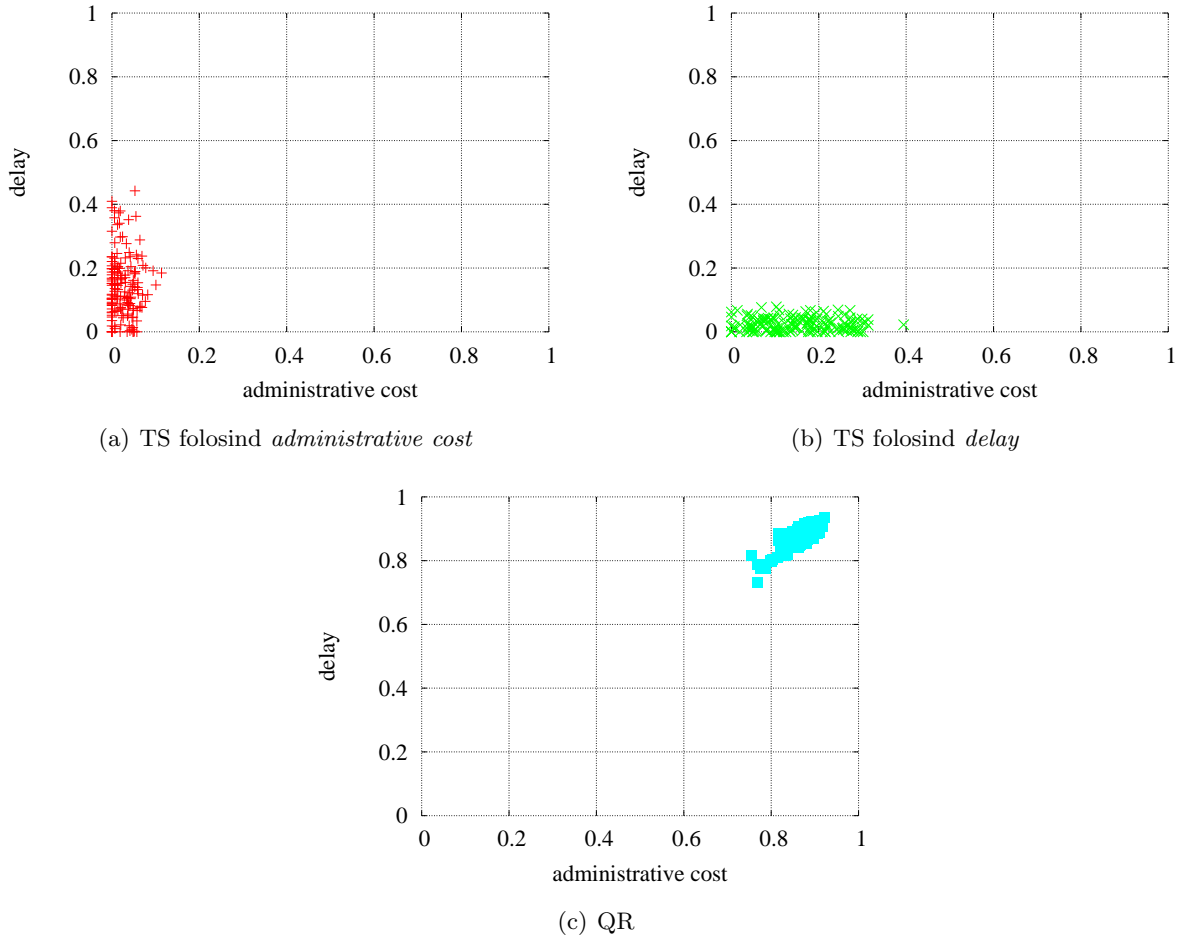


Figure 6.3: Comparare detaliată a rezultatelor obținute de $TS[administrativecost]$, $TS[delay]$ și QR pentru topologii de 20×20 noduri

Aceste rezultate arată că:

- $TS[j]$ obține rezultate bune doar pentru atributul j considerat, însă poate produce rezultate mai rele decât TN sau TR ;
- Există câteva cazuri când QR găsește căi cu cost optim, însă asemenea situații sunt foarte rare și imposibil de prevăzut.

Diferența dintre TN și TR este nesemnificativă, după cum se poate observa în imaginile detaliate din Figura 6.4. Aceasta sugerează ca orice metodă de troncarea care nu favorizează un anumit atribut oferă performanțe similare. Pentru a determina dacă acest lucru e valabil în general, s-au realizat teste pe topologii de $12 \times 12 = 144$ noduri (N), dar cu număr diferit de legături (L). Topologiile cu care s-a lucrat s-au împărțit în două clase, în funcție de gradul de conectare $c_d = 2L/N$, astfel încât $c_d \in [4, 4.5]$ și $c_d \in [5, 5.3]$. Pentru ambele clase s-au determinat proporția căilor care satisfac constrângerile (x, y) , unde $min_{adm_cost} \leq x \leq MAX_{adm_cost}$ și $min_{delay} \leq y \leq MAX_{delay}$ (min și MAX sunt valorile minime, respectiv maxime corespunzătoare fiecărui atribut pentru căile existente). Figura 6.5 ilustrează faptul că diferențele dintre cele două metode de limitare a vectorului de căi (TRUNCARE NORMALĂ și TRUNCARE RADIALĂ) apar doar în cazul topologiilor puternic conectate (adică pentru $c_d \in [5, 5.3]$). În aceste situații vectorul căilor găsit de metoda NORMALĂ satisfac constrângeri mai stricte decât căile găsite de către metoda RADIALĂ.

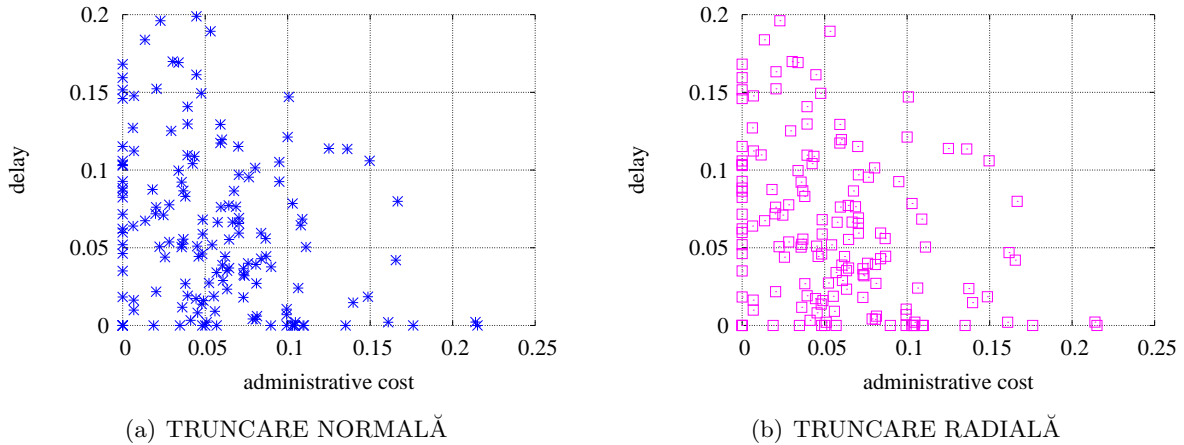


Figure 6.4: Comparare detaliată a rezultatelor obținute de TN și TR pe topologii de 20×20 noduri

În continuare s-a urmărit compararea protocolului *Macro-routing* cu un protocol de dirijare ierarhică care ar putea obține căi optime cu instrumentele existente. Algoritmul folosit pentru *Macro-routing* a fost cel bazat pe agregarea EFM (cu $T = 4$) și mecanismul de selecție prin TRUNCARE NORMALĂ. Acesta a fost comparat cu o agregare *Full-Mesh* și o selecție a căilor pentru construirea agregării bazată pe compoziția *Lagrangiană*:

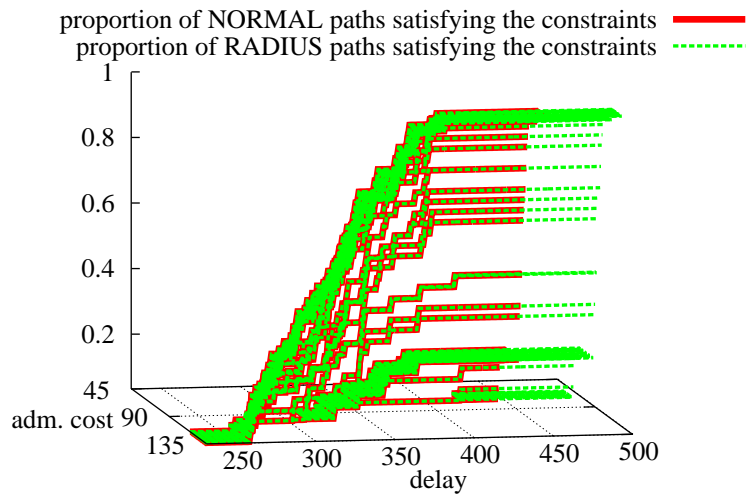
$$W_i = \frac{\sum_{k=1}^n c_i^k}{n} \quad (6.2)$$

Mecanismul de selecție bazat pe (6.2) conduce la aceleași rezultate ca și *Multi-constraint Macro-routing* când se folosește selectarea prin TRUNCARE NORMALĂ cu $T = 1$. De aceea, cele două implementări diferite protocol vor fi numite/notate $T = 4$ și $T = 1$.

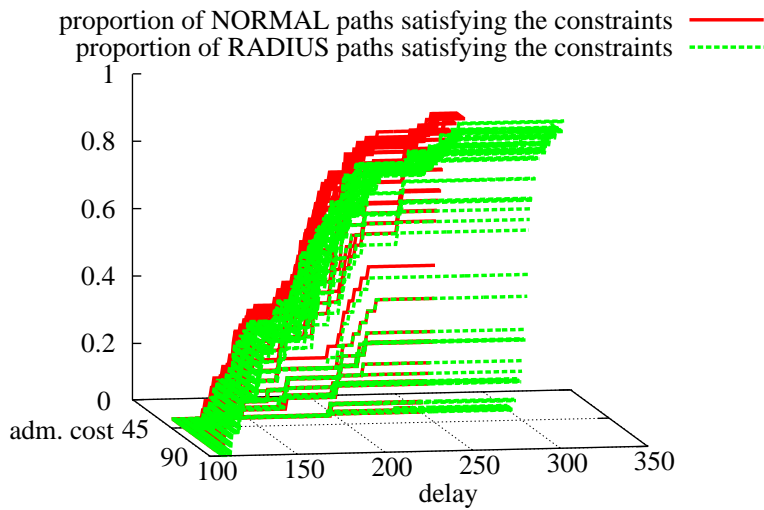
Rezultatele prezentate în Figura 6.6 arată că $T = 4$ găsește căi mai bune decât $T = 1$.

Următorul set de rezultate ilustrează distribuția lungimii (sau numărului de noduri vizitate) tuturor căilor găsite în fiecare domeniu al ierarhiei. Rezultatele prezentate în Figura 6.7, prezintă proporția căilor a căror lungime nu depășește l , ca o funcție de l . Se observă că mulțimea căilor determinate de una dintre metodele de selecție propuse sunt mai scurte decât toate căile găsite/existente. Spre exemplu, Figura 6.7(d) arată că într-o topologie de 20 de noduri nici una dintre căile selectate de către una dintre metodele de truncare nu e mai lungă decât 12, în timp ce metoda aleatoare găsește căi cu lungimea de 18 noduri.

Aceste rezultate demonstrează din nou necesitatea introducerii parametrului *lifespan*, prezentat în Secțiunea 5.2.2.

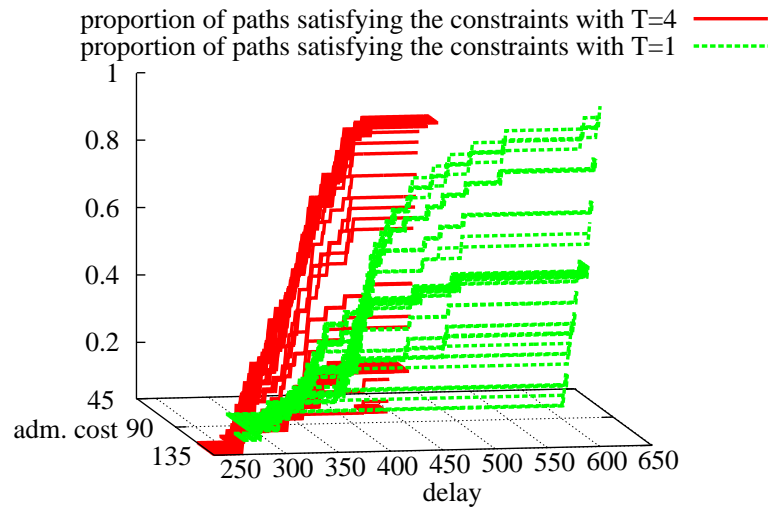


(a) $c_d \in [4, 4.5]$

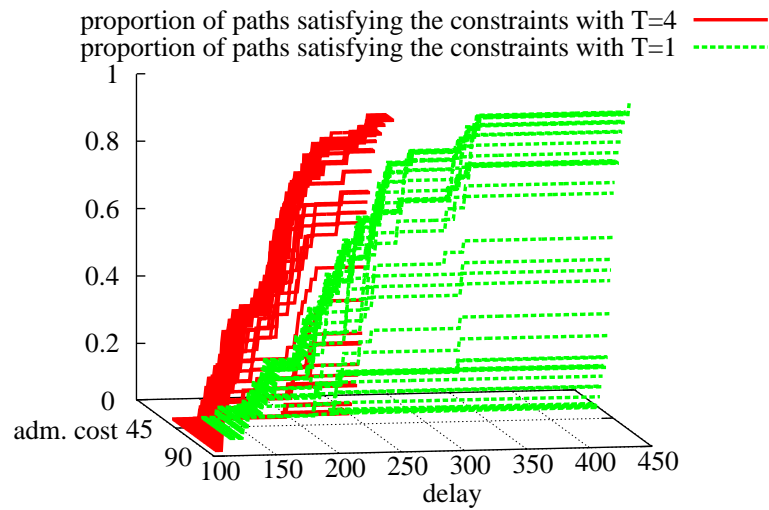


(b) $c_d \in [5, 5.3]$

Figure 6.5: Compararea dintre metodele TN și TR pe topologii de 12×12 noduri

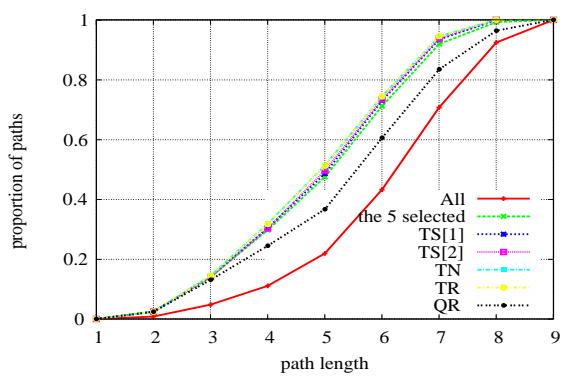


(a) $c_d \in [4, 4.5]$

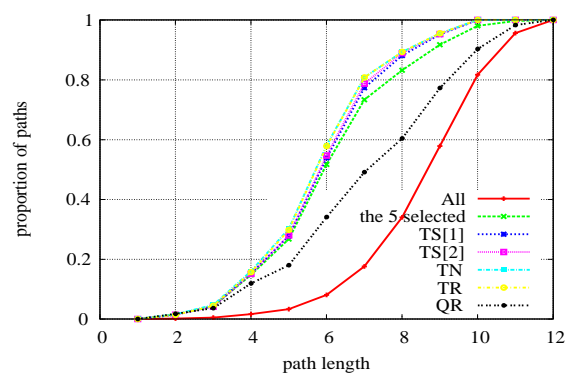


(b) $c_d \in [5, 5.3]$

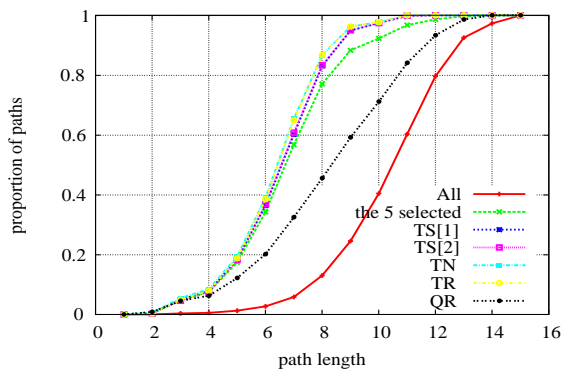
Figure 6.6: Compararea TN cu $T = 4$ și $T = 1$ pe topologii de 12×12 noduri



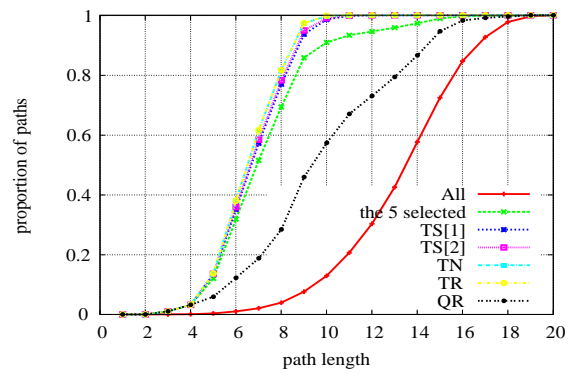
(a) într-o topologie de 9×9 noduri



(b) într-o topologie de 12×12 noduri



(c) într-o topologie de 15×15 noduri



(d) într-o topologie de 20×20 noduri

Figure 6.7: Convergența lungimilor căilor găsite

Capitolul 7

Concluzii

Tehnicile tradiționale de dirijare a pachetelor în Internet nu mai sunt adecvate pentru noile aplicații care apar. Deși dirijarea QoS poate face față cerințelor noilor aplicații, nu este încă o soluție matură, necesitând cizelarea unor probleme care nu au încă o soluție optimă.

În această teză am prezentat principalele probleme cu care se confruntă QoS, precum și soluțiile existente pentru aceste probleme. Am propus mecanisme noi și mai eficiente de dirijare QoS, de rezervare de resurse și setare a căilor MPLS. Toate aceste soluții folosesc tehnici moderne precum rețele active și agenți mobili. Aceste soluții, implementate într-o rețea activă, ar putea contribui la migrarea de la serviciile nediferențiate la servicii QoS.

Ca și direcții de cercetare pentru viitor se diferențiază două ramuri principale. Pe de o parte, continuarea studiului asupra protocolului *Macro-routing* în sensul implementării sale în rețele reale în vederea comparării performanțelor sale cu cele ale protoalelor existente, ca de exemplu PNNI. Pe de altă parte este investigarea posibilității extinderii acestui protocol pentru alte contexte decât rețele MPLS, și anume, rețele *wireless* sau rețele optice.

Bibliografie selectivă

- [1] Cengiz Alaettinoglu and A. Udaya Shankar. The Viewserver Hierarchy for Interdomain Routing: Protocols and Evaluation. *IEEE Journal of Selected Areas in Communications*, 13(8):1396–1410, 1995.
- [2] D. Scott Alexander, William A. Arbaugh, Michael W. Hicks, Pankaj Kakkar, Angelos D. Keromytis, Jonathan T. Moore, Carl A. Gunter, Scott M. Nettles, and Jonathan M. Smith. The SwitchWare Active Network Architecture. *IEEE Network Magazine Special issue on Active and Controllable Networks*, 12(3):29–36, May/June 1998.
- [3] ATM Forum. Private network-network interface specification, version 1.0. Technical Report af-pnni-0055.000”, ATM Forum, March 1996.
- [4] Daniel O. Awduche, Lou Berger, Der-Hwa Gan, Tony Li, Vijay Srinivasan, and George Swallow. RSVP-TE: Extensions to RSVP for LSP Tunnels. RFC 3209, IETF, December 2001. Status: STANDARDS TRACK.
- [5] Albert Banchs, Wolfgang Effelsberg, Christian Tschudin, and Volker Turau. Multicasting Multimedia Streams with Active Networks. In *IEEE Local Computer Network Conference (LCN)*, pages 150–159, Boston, Massachusetts, USA, October 1998.
- [6] Samrat Bhattacharjee, Kenneth L. Calvert, and Ellen W. Zegura. Active Networking and the End-to-End Argument. In *5th IEEE International Conference on Network Protocols (ICNP)*, Atlanta, Georgia, USA, October 1997. IEEE Computer Society.
- [7] Andrzej Bieszczad, Bernard Pagurek, and Tony White. Mobile Agents for Network Management. *IEEE Communications Surveys*, 1(1), 1998.
- [8] Shigang Chen. *Routing Support for Providing Guaranteed end-to-end Quality-of-Service*. PhD thesis, Engineering College of the University of Illinois at Urbana-Champaign, Urbana, Illinois, 1999.
- [9] Shigang Chen and Klara Nahrstedt. An Overview of Quality-of-Service Routing for the Next Generation High-Speed Networks: Problems and Solutions. *IEEE Network Magazine*, 12(6):64–79, December 1998.
- [10] Shigang Chen and Klara Nahrstedt. On Finding Multi-constrained Paths. In *IEEE International Conference on Communications (ICC)*, pages 874–879, Atlanta, Georgia, USA, June 1998.
- [11] David Chess, Colin Harrison, and Aaron Kershenbaum. Mobile Agents: Are They a Good Idea? RC 19887, IBM, Yorktown Heights, New York, USA, 1994. (December 21, 1994 - Declassified March 16, 1995).

- [12] Chen-Nee Chuah, Lakshminarayanan Subramanian, Randy H. Katz, and Anthony D. Joseph. QoS provisioning using a clearing house architecture. In *8th International Workshop on Quality of Service (IWQoS)*, pages 115–126, Pittsburgh, Pennsylvania, USA, June 2000.
- [13] Martin Collier. Netlets: the future of networking? In *1st IEEE Conference on Open Architectures and Network Programming*, San Francisco, California, USA, April 1998.
- [14] Eric S. Crawley, Raj Nair, Bala Rajagopalan, and Hal Sandick. A Framework for QoS-based Routing in the Internet. RFC 2386, IETF, August 1998. Status: INFORMATIONAL.
- [15] Hans De Nerve and Piet Van Mieghem. TAMCRA: A Tunable Accuracy Multiple Constraints Routing Algorithm. *Computer Communications*, 23:667–679, 2000.
- [16] Dan Decasper and Bernhard Plattner. DAN: Distributed Code Caching for Active Networks. In *IEEE Conference on Computer Communications (INFOCOM)*, volume 2, pages 609–616, San Francisco, California, USA, April 1998.
- [17] Kalaiarul Dharmalingam and Martin Collier. An Active Network Solution to RSVP Reservation Gaps. In *IEE London Communication Symposium*, London, England, September 2002.
- [18] Kalaiarul Dharmalingam, Karol Kowalik, and Martin Collier. RSVP Reservation Gaps: Problems and Solutions. In *IEEE International Conference on Communications (ICC)*, volume 3, pages 1590–1595, Anchorage, Alaska, USA, May 2003.
- [19] Gianni Di Caro and Marco Dorigo. Mobile Agents for Adaptive Routing. In *31st Hawaii International Conference on System Science (HICSS)*, Kohala Coast, Hawaii, USA, January 1998.
- [20] Jeff Dike. User Mode Linux. Running Linux on Linux. *Linux Magazine*, April 2001. http://www.linux-mag.com/2001-04/user_mode.01.html.
- [21] Radu Dragoş, Sanda Dragoş, and Martin Collier. Design and implementation of an MPLS based load balancing architecture for Web switching. In *15th ITC Specialist Seminar*, pages 24–32, Würzburg, Germany, July 2002.
- [22] Mohamed El-Darieby, Dorina C. Petriu, and Jerry Rolia. A Hierarchical Distributed Protocol for MPLS path creation. In *7th IEEE International Symposium on Computers and Communications (ISCC)*, pages 920–926, Taormina, Italy, July 2002.
- [23] Giuseppe Di Fatta, Salvatore Gaglio, Giuseppe Lo Re, and Marco Ortolani. Adaptive Routing in Active Networks. In *3rd IEEE Conference on Open Architectures and Network Programming (OpenArch)*, Tel-Aviv, Israel, March 2000.
- [24] Werner Feibel. *Encyclopedia of Networking & Telecommunications*. The Network Press, 3rd edition, November 1999. ISBN: 0-7821-2255-8.
- [25] Sergio González-Valenzuela. QoS-Routing for MPLS Networks through Mobile Processing. Master’s thesis, Electrical and Computer Engineering, Faculty of Graduate Studies, University of British Columbia, January 2002.
- [26] Pierre-Paul Grassé. La reconstruction du nid et les coordinations interindividuelles chez *Bellicositermes natalensis* et *Cubitermes* sp. La theorie de la stigmergie: Essai d’interpretation des termites constructeurs. *Insectes Sociaux*, 6(1):41–81, 1959.

- [27] Shaw Green, Leon Hurst, Brenda Nangle, Pdraig Cunningham, Fergal Somers, and Richard Evans. Software Agents: A Review. Technical Report TCS-CS-1997-06, Trinity College, Dublin, Ireland, 1997.
- [28] Roch Guerin and Ariel Orda. QoS-based Routing in Networks with Inaccurate Information: Theory and Algorithms. In *IEEE Conference on Computer Communications (INFOCOM)*, pages 75–83, Kobe, Japan, April 1997.
- [29] Michael Hicks, Pankaj Kakkar, Jonathan T. Moore, Carl A. Gunter, and Scott Nettles. PLAN: A Programming Language for Active Networks. *3rd ACM SIGPLAN International Conference on Functional Programming Languages*, 34(1):86–93, 1999.
- [30] J. M. Jaffe. Algorithms for Finding Paths with Multiple constraints. *Networks Magazine*, 14:95–116, 1984.
- [31] Bilel Jamoussi, Loa Andersson, Ross Callon, Ram Dantu, Liwen Wu, Paul Doolan, Tom Worster, Nancy Feldman, Andre Fredette, Muckai K. Girish, Eric Gray, Juha Heinanen, Timothy E. Kilty, and Andrew G. Malis. Constraint-Based LSP Setup using LDP. RFC 3212, IETF, January 2002. Status: STANDARDS TRACK.
- [32] Ioannis N. Kassabalidis, Mohamed A. El-Sharkawi, Robert J. Marks II, Payman Arabshahi, and Andrew A. Gray. Swarm Intelligence for Routing in Communication Networks. In *IEEE Global Telecommunications Conference (GLOBECOM)*, San Antonio, Texas, November 2001.
- [33] Turgay Korkmaz and Marwan Krunz. Source-Oriented Topology Aggregation with Multiple QoS Parameters in Hierarchical ATM Networks. In *7th International Workshop on Quality of Service (IWQoS)*, pages 137–146, London, UK, January 1999.
- [34] Turgay Korkmaz and Marwan Krunz. A randomised algorithm for finding a path subject to multiple QoS. *Computer Networks*, 36(2-3):251–268, 2001.
- [35] Turgay Korkmaz and Marwan Krunz. Multi-Constrained Optimal Path Selection. In *22st IEEE Conference on Computer Communications (INFOCOM)*, volume 2, pages 834–843, Anchorage, Alaska, April 2001. ISBN:0-8186-8061-X.
- [36] Kwindla Hultman Kramer, Nelson Minar, and Pattie Maes. Tutorial: Mobile Software Agents for Dynamic Routing. *Mobile Computing and Communications Review*, 3(2):12–16, 1999.
- [37] Fernando A. Kuipers, Turgay Korkmaz, Marwan Krunz, and Piet Van Mieghem. An Overview of Constraint-Based Path Selection Algorithms for QoS Routing. *IEEE Communications Magazine*, 40(12):50–55, December 2002. special issue on IP-Oriented Quality of Service.
- [38] Whay C. Lee. Spanning Tree Method for Link State Aggregation in Large Communication Networks. In *IEEE Conference on Computer Communications (INFOCOM)*, pages 297–302, Boston, Massachusetts, USA, April 1995. ISBN: 0-8186-6990-X.
- [39] Whay C. Lee. Topology Aggregation for Hierarchical Routing in ATM Networks. *ACM SIGCOMM Computer Communication Review*, 25(2):82–92, April 1995.
- [40] Tony Li. MPLS and the Evolving Internet Architecture. *IEEE Communication Magazine*, 37(12):38–41, December 1999.

- [41] Steffen Lipperts and Birgit Kreller. Mobile Agents in Telecommunications Networks - A Simulative Approach to Load Balancing. In *5th International Conference on Information Systems, Analysis and Synthesis (ISAS)*, pages 231–238, Orlando, Florida, USA, 1999.
- [42] Gang Liu and K. G. Ramakrishnan. A*Prune: an algorithm for finding k shortest paths subject to multiple constraints. In *20th IEEE Conference on Computer Communications (INFOCOM)*, pages 743–749, Anchorage, Alaska, April 2001.
- [43] King-Shan Lui, Klara Nahrstedt, and Shigang Chen. Hierarchical QoS Routing in Delay-Bandwidth Sensitive Networks. In *25th Annual IEEE Conference on Local Computer Networks (LCN)*, pages 579–588, Tampa, Florida, USA, November 2000.
- [44] Nicholas F. Maxemchuk and Steven H. Low. Active Routing. *IEEE Journal on Selected Areas in Communications (JSAC)*, 19(3):552–565, March 2001.
- [45] Nelson Minar, Kwindla Hultman Kramer, and Pattie Maes. *Cooperating Mobile Agents for Dynamic Network Routing*, chapter 12. Springer-Verlag, 1999. ISBN: 3-540-65578-6.
- [46] Pragyansmita Paul and S. V. Raghavan. Survey of QoS routing. In *15th IEEE International Conference on Computer Communication (ICC'02)*, volume 1, pages 50–75, Mumbai, Maharashtra, India, 2002.
- [47] Konstantinos Psounis. Active Networks: Applications, Security, Safety, and Architectures. *IEEE Communications Surveys*, 2(1):1–16, 1999.
- [48] Wenyu Qu and Hong Shen. Some Analysis on Mobile-Agent Based Network Routing. In *International Symposium on Parallel Architectures, Algorithms and Networks (ISPAN)*, pages 12–17, Hong Kong, China, May 2004.
- [49] Eric C. Rosen, Arun Viswanathan, and Ross Callon. Multiprotocol Label Switching Architecture. RFC 3031, IETF, January 2001. Status: STANDARDS TRACK.
- [50] Paul “Rusty” Russell. Netfilter: Packet Mangling in 2.4. In *6th International Linux Kongress*, Augsburg, Germany, September 1999.
- [51] Paul “Rusty” Russell. Writing a Module for netfilter. *Linux Magazine*, June 2000. http://www.linux-mag.com/2000-06/gear_01.html.
- [52] Paul “Rusty” Russell and Harald Welte. Linux netfilter Hacking HOWTO. <http://www.netfilter.org/documentation/HOWTO//netfilter-hacking-HOWTO.html>.
- [53] Peter S. Sapaty. The WAVE paradigm. Technical Report 17/92, Dept. of Informatics, Univ. of Karlsruhe, Karlsruhe, Germany, July 1992. Also published in Proc. Post-Conference Joint Workshop on Distributed and Parallel Implementations of Logic Programming Systems, JICSLP'92, pages 106-148, Washington, D. C., Nov. 13-14, 1992.
- [54] Peter S. Sapaty. Mobile processing in open systems. In *5th IEEE International Symposium on High Performance Distributed Computing (HPDC)*, Syracuse, New York, USA, August 1996.
- [55] Peter S. Sapaty. Organisation and Management of Distributed Dynamic Systems in WAVE. In *Symposium on Unmanned Systems*, Baltimore, MD, USA, July 2001.
- [56] Peter S. Sapaty and P. M. Borst. An overview of the WAVE language and system for distributed processing in open networks. Technical report, Dept. of Electronic and Electrical Eng., Univ. of Surrey, Surrey, UK, June 1994.

- [57] Beverly Schwartz, Wenyi Zhou, Alden W. Jackson, W. Timothy Strayer, Dennis Rockwell, and Craig Partridge. Smart packets for active networks. In *IEEE Open Architectures and Network Programming (OPENARCH)*, pages 90–97, New York, USA, March 1999.
- [58] Kwang Mong Sim and Weng Hong Sun. Multiple Ant-Colony Optimisation for Network Routing. In *1st IEEE International Symposium on Cyber Worlds (CW)*, pages 277–281, Tokyo, Japan, November 2002. ISBN: 0-7695-1862-1.
- [59] John Sum, Hong Shen, Chi sing Leung, and Gilbert H. Young. Analysis on a Mobile Agent-Based Algorithm for Network Routing and Management. *IEEE Transactions on Parallel and Distributed Systems*, 14(3):193–202, March 2003.
- [60] Andrew S. Tanenbaum. *Computer Networks*. Prentice Hall, 4th edition, 2003. ISBN: 0-13-066102-3.
- [61] David L. Tennenhouse, Jonathan M. Smith, W. David Sincoskie, David J. Wetherall, and Gary J. Minden. A Survey of Active Network Research. *IEEE Communications Magazine*, 35(1):80–86, January 1997.
- [62] David L. Tennenhouse and David J. Wetherall. Towards an Active Network Architecture. *Computer Communication Review*, 26(2):5–18, April 1996.
- [63] Piet Van Mieghem and Hans De Nerve. Hop-by-hop quality of service routing. *Computer Networks*, 37(3-4):407–423, October 2001.
- [64] Zheng Wang and Jon Crowcroft. QoS Routing for Supporting Resource Reservation. *IEEE Journal on Selected Areas in Communications (JSAC)*, 14(7):1288–1294, September 1996.
- [65] Zheng Wang and Jon Crowcroft. Quality of Service Routing for Supporting Multimedia Applications. *IEEE Journals on Selected Areas in Communications (JSAC)*, 14(7):1228–1234, September 1996.
- [66] Michael Welzl, Alfred Cihal, and Max Mühlhäuser. An Approach to Flexible QoS Routing with Active Networks. In *4th IEEE Annual International Workshop on Active Middleware Services (AMS)*, pages 75–82, Edinburgh, UK, July 2002.
- [67] David J. Wetherall, John V. Guttag, and David L. Tennenhouse. ANTS: A Toolkit for Building and Dynamically Deploying Network Protocols. In *IEEE Open Architectures and Network Programming (OPENARCH)*, San Francisco, California, USA, April 1998.
- [68] David J. Wetherall and David L. Tennenhouse. The ACTIVE IP Option. In *7th ACM SIGOPS European Workshop*, Connemara, Ireland, September 1996.
- [69] Johnny S.K. Wong and Armin R. Mikler. Intelligent mobile agents in large distributed autonomous cooperative systems. *Journal of Systems and Software*, 47:75–87, 1999.
- [70] Yechiam Yemini and Sushil Da Silva. Towards programmable networks. In *IFIP/IEEE International Workshop on Distributed Systems: Operations and Management*, L'Aquila, Italy, October 1996.
- [71] Xin Yuan. Heuristics algorithms for multi-constrained quality-of-service routing. *IEEE/ACM Transactions on Networking*, 10(2):244–256, 2002.
- [72] Ellen W. Zegura, Kenneth L. Calvert, and Samrat Bhattacharjee. How to Model an Internetwork. In *IEEE Conference on Computer Communications (INFOCOM)*, volume 2, pages 594–602, San Francisco, California, USA, March 1996.

Traducerile termenilor de specialitate

A

aleator - **random**
algoritm de dirijare - **routing algorithm**
agenți mobili - **mobile agents**
agregarea topologiei - **topology aggregation**
atribute - **weights, metrics**

C

calitatea serviciilor - **Quality of Service (QoS)**
comunicare prin intermediul mediului înconjurător - **stigmergy**
comutare - **switching**

D

datagramă - **datagram**
dirijare - **routing**
dirijarea pas-cu-pas - **hop-by-hop routing**
dirijarea din nodul sursa - **source routing**
dirijarea după vectorul distanțelor - **distance-vector routing**
dirijarea bazată pe starea legăturilor - **link-state routing**
dirijarea ierarhică - **hierarchical routing**
dirijarea QoS - **QoS routing**
dirijarea bazată pe constrângeri multiple - **multi-constraint routing**

F

fluctuație - **jitter**
flux - **flow**

I

inteligența grupului - **swarming intelligence**
inteligența emergentă - **emergent behaviour**

Î

încărcare - **overhead**

L

lățime de bandă - **bandwidth**
legătură - **link**

N

nod (într-o rețea de calculatoare) - **router or switch**
nod de control - **managing node**
nod de la marginea rețelei/domeniului - **border node**

R

rețea de calculatoare - **computer network**
rețele active - **active networks**
rețele de la marginea Internetului - **access networks, access areas**
rezervare de resurse - **resource reservation**

S

salt (hop) - **hop**
servere web intermediare - **Web proxies**
serviciu nediferențiat - **best-effort**

T

trimiterea pachetelor - **packet forwarding**
tunel/cale MPLS - **Label Switching Path (LSP)**

V

viteză de transmitere pe cablu - **wire speed**