



UNIUNEA EUROPEANĂ



GUVERNUL ROMÂNIEI  
MINISTERUL MUNCII, FAMILIEI ȘI  
PROTECȚIEI SOCIALE  
AMFOSDRU



Fondul Social European  
POS DRU 2007-2013



Instrumente Structurale  
2007-2013



MINISTERUL  
EDUCAȚIEI  
CERCETĂRII  
TINERETULUI  
ȘI SPORTULUI  
OIPOSDRU



### Investește în oameni!

FONDUL SOCIAL EUROPEAN

Programul Operațional Sectorial Dezvoltarea Resurselor Umane 2007 – 2013

Axa prioritară: 1 „Educația și formarea profesională în sprijinul creșterii economice și dezvoltării societății bazate pe cunoaștere”

Domeniul major de intervenție: 1.5 „Programe doctorale și postdoctorale în sprijinul cercetării”

Titlul proiectului: Proiect de dezvoltare a studiilor de doctorat în tehnologii avansate- “PRODOC”

Cod Contract: POSDRU 6/1.5/S/5

Beneficiar: Universitatea Tehnică din Cluj-Napoca

## FACULTATEA DE ELECTRONICĂ, TELECOMUNICAȚII ȘI TEHNOLOGIA INFORMAȚIEI

Ing. **Georgeta Lucia Boanea**

# TEZA DE DOCTORAT

## ÎMBUNĂTĂȚIREA RUTĂRII MULTICALE ÎN VIITORUL INTERNET

### -Rezumat-

**Conducător științific**

Prof.dr.ing. **Virgil DOBROTĂ**

### Comisia de evaluare a tezei de doctorat:

Președinte: -Prof.dr.ing. **Monica Borda** – Facultatea de Electronică, Telecomunicații și Tehnologia Informației, Universitatea Tehnică din Cluj-Napoca;

Membri: -Prof.dr.ing. **Virgil Dobrotă** – Conducător științific, Facultatea de Electronică, Telecomunicații și Tehnologia Informației, Universitatea Tehnică din Cluj-Napoca;  
-Prof.dr.mat. **Florian Mircea Boian** – Referent, Universitatea „Babes-Bolyai” din Cluj-Napoca;  
-Prof.dr. ing. **Radu Vasiu** – Referent, Universitatea „Politehnica” din Timișoara;  
-Prof.dr.ing. **Aurel Vlaicu** – Referent, Facultatea de Electronică, Telecomunicații și Tehnologia Informației, Universitatea Tehnică din Cluj-Napoca.

Susținerea publică a tezei de doctorat: 23 septembrie 2011  
Aula „Alexandru Domșa”  
Universitatea Tehnică din Cluj-Napoca  
Strada Constantin Daicoviciu nr. 15

## Mulțumiri

Doresc să mulțumesc tuturor celor care, din punct de vedere profesional, direct sau indirect, prin dezbaterile și sugestiile oferite, au contribuit la elaborarea acestei teze.

Mulțumesc domnului prof.dr.ing Virgil Dobrotă pentru îndrumarea, încurajările și sprijinul continuu oferit de-a lungul întregii perioade de pregătire a doctoratului. De asemenea, le mulțumesc și colegilor din colectivul Unified Communications Laboratories: conf.dr.ing. Daniel Zinca, șl.dr.ing. Tudor Mihai Blaga, as.dr.ing. Cristian Mihai Vancea, dr.ing. Andrei Bogdan Rus, ing. Gabriel Lazăr, ing. Melinda Barabas și ing. Sabin Sărmaș pentru întreaga colaborare.

De asemenea le mulțumesc domnilor prof.dr.ing Aurel Vlaicu, prorector al Universității Tehnice din Cluj-Napoca și conf.dr.ing Daniel Zinca pentru sugestiile primite cu ocazia participării în comisia de evaluare a rapoartelor de doctorat. Mulțumirile se îndreaptă și către colaboratorii cu care am lucrat în diverse proiecte de cercetare: conf.dr.ing Zsolt Polgar și ing. Zsuzsanna Kiss.

Deoarece o parte din munca prezentată în aceasta teză a fost efectuată în cadrul perioadei de mobilitate efectuate la Universitat Politècnica de Catalunya, Barcelona aș dori să îi mulțumesc domnului prof.dr.ing Jordi Domingo-Pascual și colectivului CBA (Broadband Communication Systems) pentru sprijinul acordat.

De asemenea mulțumesc colectivului de la Telecommunications Research Group, Department of Communication Systems, Jozef Stefan Institute, Ljubljana, Slovenia, în special domnului prof.dr. Gorazd Kandus și doamnei drd.ing. Carolina Fortuna, pentru colaborarea pe parcursul perioadei de documentare.

Mulțumesc de asemenea pentru sprijinul acordat coordonatorului proiectului PRODOC, domnul prof.dr.ing Gheorghe Lazea și doamnelor secretare Rodica Brad, Dorina Baraian și Livia Haiduc.

Mulțumiri speciale se adresează surorii mele Maria Boanea și părinților pentru suportul continuu oferit. Nu în ultimul rând aș dori să le mulțumesc prietenilor pentru încurajări și înțelegere.

Ing. Georgeta Boanea

Septembrie 2011



# Cuprins

1	Introducere.....	7
1.1	Motivația tezei.....	8
2	Evaluarea și clasificarea metodelor de rutare multicale.....	10
2.1	Motivația .....	10
2.2	Aspecte generale ale procesului de rutare .....	10
2.3	Rutarea clasică.....	10
2.4	Rutare dinamică bazată pe QoS.....	11
2.5	Aspecte generale privind rutarea multicale .....	11
2.6	Realizarea rutării multicale .....	12
2.7	Soluții de rutare multicale .....	13
2.8	Concluzii .....	14
3	Variantă modificată a algoritmului DFS (Depth-First Search) pentru determinarea căilor multiple într-un graf .....	15
3.1	Motivație .....	15
3.2	Algoritmi de căutare în graf .....	15
3.3	Algoritmul de calcul al căilor multiple propus.....	16
3.3.1	Schema logică a algoritmului .....	16
3.3.2	Exemplu de funcționare al algoritmului .....	17
3.3.3	Influența ordinii nodurilor în listele de adiacență .....	18
3.3.4	Varianta modificată a algoritmului cu determinarea unui număr redus de căi.....	18
3.4	Concluzii .....	18
4	Proiectarea unui nou algoritm de rutare multicale SAMP (Situation Aware Multipath).....	19
4.1	Motivație .....	19
4.2	Criteriile de proiectare.....	19
4.3	Interacțiunea modulelor sistemului de rutare .....	19
4.4	Pașii de proiectare .....	20
4.5	Concluzii .....	23
5	Implementarea algoritmului de rutare multicale SAMP .....	24
5.1	Motivația .....	24
5.2	Modul de funcționare al soluției de rutare multicale SAMP.....	24
5.3	Evaluarea performanțelor algoritmului de rutare SAMP .....	26
5.3.1	Arhitectura rețelei de test .....	26
5.3.2	Rezultatele testelor .....	27
5.4	Concluzii .....	28
6	Proiectarea și implementarea unui modul de identificare a fluxurilor în cadrul simulatorului OPNET .....	29
6.1	Motivația .....	29

6.2	Alegerea simulatorului de rețea.....	29
6.3	Implementarea modului de identificare a fluxurilor .....	29
6.3.1	Definirea contextului .....	29
6.3.2	Comunicarea cu alte module .....	30
6.3.3	Managementul fluxurilor.....	30
6.4	Evaluarea performanțelor .....	31
6.4.1	Topologia rețelei.....	31
6.4.2	Rezultate experimentale .....	31
6.5	Concluzii .....	33
7	Realizarea și evaluarea performanțelor modului SAMP în cadrul simulatorului de rețea OPNET .....	34
7.1	Motivația .....	34
7.2	Implementarea modului de rutare multicale SAMP.....	34
7.2.1	Modul de funcționare .....	34
7.3	Evaluarea performanțelor algoritmului SAMP în cadrul simulatorului OPNET .....	35
7.3.1	Arhitectura rețelei de test .....	35
7.3.2	Împărțirea rețelei în domenii de rutare multicale .....	36
7.3.3	Scenariile de test.....	37
7.3.3.1	Cazul 1: nu există congestie în rețea .....	37
7.3.3.2	Cazul 2: congestie pe una din legăturile din rețea.....	38
7.4	Concluzii .....	39
8	Contribuții .....	40
8.1	Sumarul contribuțiilor .....	40
8.2	Observații finale .....	43
8.3	Premii .....	44
8.4	Lista publicațiilor personale .....	44
	Bibliografie selectivă.....	48

# 1 Introducere

În momentul actual rețeaua Internet joacă un rol important în procesul de comunicare fiind utilizată atât în domeniul relațiilor personale cât și în cel business. Structura curentă are la bază ideea unei arhitecturi simple din punct de vedere al serviciilor, fiind destinată să interconecteze atât sisteme inteligente cât și pe cele cu funcționalități de bază. Inteligența rețelei este împinsă la marginile acesteia. Stratul rețea este capabil să transmită informația de la un punct la altul, însă nu se oferă nici o garanție legată de livrarea pachetelor. Problema ”osificării” devine pregnantă, se prevede că arhitectura curentă își va atinge curând limitele din punct de vedere al domeniului de adrese, accesibilități și cererilor variate de QoS (Quality of Service). Există aspecte care în momentul proiectării rețelei nu au fost luate în calcul, dar care în momentul de față devin o necesitate. Internetul este capabil să livreze pachete dar este inflexibil din punct de al stratului rețea și de asemenea prezintă minusuri din punct de vedere al funcționalităților adiționale. Unele din aspectele care lipsesc în momentul actual sunt [Tse10]: 1) funcționalități de auto-management, 2) facilități de adăugare de noi funcționalități, cum ar fi capabilitatea de activare a unui serviciu la cerere, 3) suport pentru integrarea de nivel superior între servicii și rețele, 4) facilități pentru administrarea securității, îmbunătățirea robusteții și fiabilității, managementul resurselor, 5) facilități de suport QoS, 6) o schemă adecvată de administrare de adrese etc.

Conceptul de Future Internet (FI) cuprinde o serie de idei și tehnologii care urmăresc schimbarea modului de abordare existent în rețeaua Internet curentă. Scopul este de a adapta sau a concepe o nouă rețea în care lipsurile structurii curente să fie acoperite. Unul din domeniile FI prevede realizarea unei transmisii care să poată asigura o anumită calitate a serviciilor fără a fi nevoie de o rezervare de resurse prealabilă.

În cadrul rețelelor de comunicație rutarea reprezintă unul din pionii principali, de modul de realizare al acesteia depinzând în mare măsură performanțele și fiabilitatea rețelei. Proiectarea algoritmilor de rutare reprezintă o provocare deoarece aceștia trebuie să aibă un caracter distribuit, să fie capabili să se adapteze la schimbările condițiilor de trafic și de asemenea să facă față modificărilor care pot să apară în cadrul topologiei rețelei.

Principala sarcină a unui protocol de rutare este de a găsi calea/căile între nodul sursă și nodul destinație și de a dirija traficul de-a lungul rutelor găsite. Alegerea unei anumite căi determină calitatea și stabilitatea transferului de date.

Datorită dinamismului rețelei de comunicații, soluția convențională de rutare, bazată pe găsirea unei singure căi nu poate face față în mod eficient și rapid tuturor provocărilor care apar cum ar fi defectarea unui nod sau a unei legături, apariția congestiei, întârzieri mari sau rate de transfer scăzute. Toate acestea duc la degradarea eficienței de livrare a pachetelor. În acest context existența de căi alternative între un nod sursă și destinație devine o necesitate. O modalitate de îndeplinire a acestor nevoi o reprezintă soluția în care routerele ar putea divide în mod flexibil traficul pe mai multe căi, adică implementarea protoalelor de rutare multicale.

O alternativă la rutarea convențională bazată pe o singură cale este reprezentată de rutarea multicale. Aceasta oferă o mai mare flexibilitate și diversitate în procesul de alegere a rutei. Această abordare vine să rezolve unele din limitările rutării convenționale cum ar fi problema congestiei și utilizarea ineficientă a resurselor existente în rețea. În cazul rutării multicale traficul corespunzător unei anumite destinații este împărțit pe mai multe rute.

Rutarea multicale reprezintă un element important în ingineria traficului, care are ca scop optimizarea performanțelor operaționale ale unei rețele, unele din ținte fiind distribuția traficului, rutarea bazată pe constrângeri și re-rutarea rapidă. Din acest punct de vedere rutarea multicale

oferă soluții pentru calcularea efectivă a căilor multiple și de asemenea modalități de reducere a întârzierilor și creștere a ratei de transfer.

În cadrul viitorului Internet una din preocupări este de a putea face față cerințelor diversificate ale utilizatorilor. Conceptul de QoS are o semnificație largă și cuprinde mai multe aspecte. Una din abordări, utilizată în soluțiile existente, este de a asigura calitatea prin rezervarea de resurse și administrarea de cozi (IntServ, DiffServ). Deși aceste soluții nu sunt noi pe piață, din cauza dezavantajelor pe care le prezintă nu s-a reușit să se ajungă la o extindere în întreaga rețea. În momentul de față în majoritatea cazurilor transmisia de date se realizează în continuare după principiul „Best-Effort”.

În momentul de față protocoalele existente în rețea utilizează din punct de vedere al costurilor legăturilor valori predefinite care pot sau nu să depindă de parametrii legăturilor din rețea. Rutele se calculează pe baza acestor metrici. Protocoalele existente sunt capabile să reacționeze în momentul în care o legătură se defectează, însă în cazul congestiei acestea nu asigură schimbarea traiectoriei pachetelor deoarece nu se ține cont de starea reală a rețelei. Rezolvarea acestor probleme poate fi asigurată de un proces de rutare conștient de starea rețelei (QoS-aware) în care declanșarea procedurilor de schimbare a rutei utilizate este dependentă de parametri reali ai conexiunilor rețelei: rata de transfer disponibilă și întârziere.

Managementul și rutarea, în cadrul actual al rețelei Internet, sunt independente. Partea de management colectează date și realizează statistici iar rutarea are propriile informații pe bază cărora are loc comutarea datelor. Într-un sistem în care entitățile care asigură comunicarea ar colabora transmisia datelor s-ar putea realiza cu o eficiență crescută și probabilitate de blocaj redusă.

## 1.1 Motivația tezei

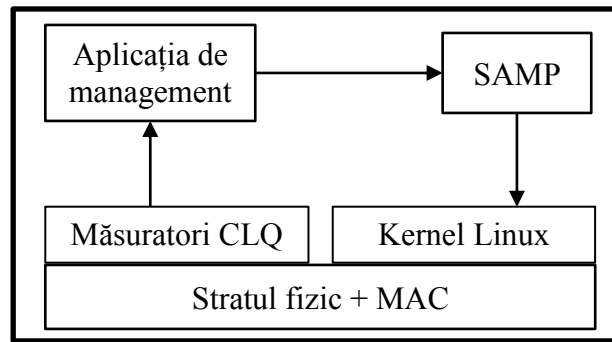
Nevoia de infrastructură și servicii în cadrul Internetului este în continuă ascensiune. O dată cu creșterea numărului de utilizatori s-a dezvoltat și o gama variată de aplicații care să satisfacă necesitățile acestora. Ca urmare a acestui fapt serviciile oferite de rețea trebuie să fie capabile să îndeplinească cerințele utilizatorilor. Un serviciu care a început să predomine este transmisia de informație video (VoIP, VoD, IPTV). Studii arată că în 2014 procentul datelor video va fi de aproximativ 91% din traficul global al consumatorilor [Cis10]. Cerințele din punct de vedere al ratei de transfer și întârzierii sunt destul de stricte în cazul acestui tip de date, ca urmare rețeaua trebuie să asigure anumiți parametri pentru a oferi o transmisie de calitate.

Rutarea convențională utilizată în momentul de față în rețeaua Internet nu este dependentă de starea reală a rețelei, procesul de comutare al pachetelor nu se adaptează, consecința fiind o transmisie de proastă calitate în cazul apariției congestiei. Un alt dezavantaj constă în faptul că spre o destinație tot traficul se transmite pe aceeași cale, ceea ce duce la o utilizare ineficientă a resurselor și la o probabilitate de blocaj mai mare.

Teza de față propune o nouă abordare în care funcțiile de management sunt separate de cele de comutare a pachetelor. În aceasta lucrare se prezintă o soluție de rutare multicale conștientă de starea rețelei care face parte dintr-un sistem de rutare al informației în viitorul Internet. Sistemul este compus din trei entități: aplicația de management [Bar11d], algoritmul de rutare multicale propus, SAMP (Situation Aware Multipath) și aplicația CLQ (Cross-Layer QoS) de măsurare a parametrilor de trafic la strat MAC [Rus11]. Prin această abordare crește eficiența utilizării informației legate de starea rețelei, aceleași date pot fi folosite atât cu scop de management cât și în rutare.

Calitatea transmisiei pachetelor depinde de activitatea tuturor părților componente. Între entități există o strânsă legătură fiecare depinzând de informațiile care se transmit între acestea.





**Figura 1-1: Arhitectura sistemului de transmisie folosind algoritmul SAMP**

Noul algoritm de rutare multicale SAMP, prezentat în această lucrare își propune eliminarea unora din dezavantajele soluțiilor de rutare folosite în prezent în Internet prin : 1) eliminarea problemelor cauzate de congestie, 2) eficientizarea utilizării resurselor existente în rețea, 3) îmbunătățirea fiabilității și robusteții rețelei, 4) diminuarea probabilității de blocaj.

Din punct de vedere al conceptului de QoS, pentru a asigura parametrii necesari diferitelor aplicații utilizator, se realizează o administrare eficientă a resurselor din rețea prin transmisia datelor pe căi multiple. În caz de congestie comutarea pachetelor se realizează astfel încât zona cu probleme să fie evitată. Deoarece există o mare varietate a serviciilor care trebuie asigurate, fiecare cu anumite cerințe, s-a ales împărțirea traficului în două categorii: fluxuri neelastice și fluxuri elastice. În prima categorie intră transmisiile cu necesități mai stricte din punct de vedere al ratei de transfer și întârzierii (cum ar fi traficul video și de voce), iar în a doua categorie fluxurile mai flexibile la variația parametrilor conexiunilor (cum ar fi transferul de fișiere). În procesul de comutare, fluxurile din prima categorie primesc un tratament special fiind primele care sunt re-rutate în caz de congestie. În acest mod se poate asigura o transmisie de calitate chiar și în cazul în care apar probleme în rețea.

## 2 Evaluarea și clasificarea metodelor de rutare multicale

### 2.1 Motivația

În acest capitol pentru început se va prezenta modul de rutare utilizat la momentul actual în rețeaua Internet cu caracteristicile specifice. Se vor pune de asemenea în evidență problemele existente și limitările protocoalelor de rutare și se vor propune unele soluții pentru creșterea fiabilității rețelei. A doua parte a capitolului este destinată aspectelor legate de rutarea multicale, avantajele aduse de această abordare, modurile de realizare ale principalelor funcții, problemele întâmpinate și de asemenea câteva implementări concrete existente în literatura de specialitate.

### 2.2 Aspecte generale ale procesului de rutare

Rutarea reprezintă mecanismul prin care fiecare pachet este transmis prin intermediul unei rețele de la un nod sursă la nodul destinație corespunzător. Principalele atribuții ale unei soluții de rutare sunt: 1) descoperirea topologiei și calcularea rutelor, 2) comutarea pachetelor, 3) stabilirea domeniului de activitate al protocolului de rutare și 4) adaptarea la schimbarea condițiilor din rețea.

### 2.3 Rutarea clasică

Rutarea și comutarea de pachete în actualul Internet este rezultatul unei juxtapuneri complexe de protocoale și mecanisme care au fost dezvoltate și adaptate de-a lungul timpului pentru arhitectura IP. Procesul de rutare funcționează în majoritatea timpului destul de bine dar există unele limitări astfel încât este destul de greu de abordat problema expansiunii fără ca aceasta să aibă ca efect o ruptură din punct de vedere al operării rețelei [Ara09].

Unele din limitările rutării actuale sunt [Boa09]: 1) re-convergența lentă, 2) rutarea se face pe o singură cale, 3) metrica simplă de alegere a rutei, 4) rutare bazată pe QoS limitată, 5) rutarea nu este dependentă de starea reală a rețelei, 6) utilizarea inefficientă a resurselor existente etc.

În momentul de față procesul de rutare din Internet utilizează, în mare parte, protocoale bazate pe găsirea unei singure căi spre o anumită destinație. Există două abordări principale:

- Protocoale bazate pe vector distanță: difuzarea informației se realizează între nodurile adiacente. Cunoașterea rețelei are loc prin intermediul pachetelor vector distanță care cuprind adresa destinație și costul rutei până la acea destinație (numărul de hopuri). Se presupune doar o cunoaștere parțială a rețelei, fiecare nod posedă doar informația primită de la nodurile aflate la un nod distanță.
- Protocoale bazate pe starea legăturii: presupune cunoașterea întregii rețele la nivelul fiecărui nod. Fiecare router construiește o bază de date corespunzătoare topologiei rețelei. Aceasta este obținută pe baza transmisiei de pachete care anunță starea legăturilor LSA (Link State Advertisements), modelul de difuzare a informației este prin „inundarea” rețelei.

## 2.4 Rutare dinamică bazată pe QoS

Calitatea serviciilor este o noțiune generică care poate fi interpretată în mod diferit în domeniul rutării. Sunt definite diferite tehnici prin care se pot garanta o serie de constrângeri în ceea ce privește performanța. În mod tradițional, asigurarea calității serviciilor oferite înseamnă în cele mai multe cazuri rezervarea de bandă pentru anumite fluxuri, adică definirea unor cozi care să trateze în mod preferințial pachetele. Din punct de vedere al abordării tradiționale există două metode mai cunoscute: IntServ [Bra94] și DiffServ [Nic99].

O altă abordare din punct de vedere al calității serviciilor este de a evita procesul de administrare a cozilor, parametrii QoS fiind asigurați prin intermediul procesului de rutare. În lucrarea [Rus10c] s-a realizat rutarea pachetelor bazată pe QoS prin intermediul routerelor virtuale. În acest caz nu are loc rezervarea resurselor ci comutarea pachetelor se face în funcție de parametrii conexiunilor obținuți prin intermediul tehnicilor CLQ.

O metodă alternativă la rutare, care asigură transmisia pachetelor pe baza parametrilor de QoS, este prezentată în [Pol09], [Cor11] și [Rus10a]. Această metodă are la bază tehnici de tip NC (Network Coding). Ideea acestei abordări este de a transmite codat datele între o sursă și o destinație în cazul în care congestia din rețea nu se poate rezolva prin intermediul re rutării. Parametrii legăturilor sunt monitorizați în permanență. Pe baza măsurătorilor făcute, aplicațiile de NC, prezente la nivelul fiecărui nod, vor lua decizia dacă este necesară sau nu codarea datelor. Rezultatele obținute, atât în cadrul mediului simulat cât și în cazul unei rețele reale, demonstrează că această metodă poate îmbunătăți calitatea transmisie în caz de congestie.

## 2.5 Aspecte generale privind rutarea multicale

Majoritatea protocoalelor actuale sunt bazate pe algoritmi de găsim a unei singure căi între un nod sursă și destinație. Rutarea multicale oferă soluții care permit o divizare flexibilă a traficului pe mai multe rute în acest mod obținându-se o utilizare mai eficientă a rețelei.

### A. Avantajele rutării multicale

Implementarea soluțiilor de rutare multicale la scară largă ar aduce multe beneficii, unele din acestea sunt [Boa09]:

- *Adaptarea modului de rutare în funcție de cerințele aplicațiilor.* În funcție de necesitățile unei anumite aplicații există posibilitatea ca aceasta să își aleagă o anumită cale cu parametri care corespund cel mai bine cerințelor serviciului transportat.
- *Îmbunătățirea fiabilității cap-la-cap.* Există posibilitatea de a trece ușor și rapid de la o rută la alta în cazul apariției unei defecțiuni.
- *Creșterea ratei de transfer prin agregarea de rute:* Datele se pot transmite simultan pe mai multe rute.
- *Evitarea căilor congestionate.* Datorită existenței mai multor opțiuni de rutare a unui pachet, se pot evita legăturile care sunt afectate de fenomenul de congestie.
- *Asigurarea cerințelor QoS pentru aplicațiile deservite.* În funcție de aplicație se poate alege una sau mai multe rute din setul disponibil.

### B. Costurile rutării multicale

În cadrul rutării multicale sunt necesare resurse suplimentare atât în planul de control cât și în cel de date [Boa09]:

- *Resurse suplimentare (overhead) în planul de control.* O măsură a acestui proces ar fi mesajele de rutare care trebuie transmise pentru a propaga informația de rutare și timpul necesar CPU de a calcula căile multiple.

- *Supra-antete în planul de date.* Introducerea de date suplimentare în fiecare pachet, cum ar fi un antet extra sau o etichetă și tabelele de rutare mai complexe.
- *Informații suplimentare necesare în procesul de comutare al routerului.* Procesarea suplimentară pentru a transmite fiecare pachet, alegerea interfeței de ieșire corespunzătoare.

## 2.6 Realizarea rutării multicale

Implementarea unei soluții de rutare multicale trebuie să se asigure în primul rând ca procesul de comutare al pachetelor se face în mod corect și pachetele ajung la destinație. Principale acțiuni ale unei soluții de rutare multicale sunt [Boa11c]: 1) calcularea rutelor multiple, 2) decizia modului de divizare a traficului pe mai multe rute, 3) stabilirea mecanismului efectiv de comutare al pachetelor pe mai multe căi, 4) definirea modului de reacție la modificările care au loc în rețea.

În continuare se vor prezenta unele din opțiunile existente pentru implementarea unei soluții de rutare multicale.

**Tabel 2-1: Clasificarea metodelor de realizare a rutării multicale**

	Acțiuni	Opțiuni	Caracteristici
Soluție de rutare multicale	Calcularea rutelor multiple	Algoritmi bazați pe găsirea a primelor k cele mai scurte rute	<ul style="list-style-type: none"> <li>• Numărul de rute este restricționat doar de valoarea k</li> <li>• Probabilitate mare ca rutele să aibă un număr mare de legături comune</li> </ul>
		Algoritmi bazați pe găsirea de rute disjuncte	<ul style="list-style-type: none"> <li>• Numărul de rute este restrâns</li> <li>• Robustețe crescută la defecte</li> </ul>
	Divizarea traficului pe rute multiple	La nivel de pachet	<ul style="list-style-type: none"> <li>• Granularitate fină de divizare a traficului</li> <li>• Problema secvențialității la destinație; nu este indicat în cazul în care rutele au întârzieri diferite</li> </ul>
		La nivel de flux	<ul style="list-style-type: none"> <li>• Granularitatea de divizare depinde de fluxurile din rețea</li> <li>• Este nevoie de identificarea fluxurilor</li> <li>• Nu există problema secvențialității pachetelor la destinație</li> </ul>
		La nivel de flowlet	<ul style="list-style-type: none"> <li>• Granularitate mai fină decât în cazul divizării la nivel de flux</li> <li>• Nu există problema secvențialității pachetelor la destinație</li> </ul>
	Comutarea pachetelor pe căi multiple	Distribuit	<ul style="list-style-type: none"> <li>• Decizia se ia în fiecare nod pe baza informațiilor existente la nivelul fiecărui router</li> <li>• Necesită mai multe informații de control</li> </ul>
Centralizat		<ul style="list-style-type: none"> <li>• Decizia traseului urmat de pachete se ia de nodul sursă.</li> <li>• Are avantajul simplității</li> </ul>	

			<ul style="list-style-type: none"> <li>• Introduce informații suplimentare în planul de date</li> </ul>
Reacția la modificările din rețea	Configurații de rutare multiplă		<ul style="list-style-type: none"> <li>• Se construiesc tabele alternative de rutare care vor fi folosite în cazul în care are loc o defecțiune pe ruta implicită</li> <li>• Resurse suplimentare din punct de vedere al memoriei și al CPU</li> </ul>
	Metode proactice		<ul style="list-style-type: none"> <li>• Deciziile de rerutare se iau la nivelul nodului care a sesizat problema fără a se anunța și restul nodurilor</li> <li>• Pot să apară bucle în rețea</li> </ul>
	Tunelarea		<ul style="list-style-type: none"> <li>• În momentul în care apare o problemă datele sunt transmise tunelat astfel încât să se evite zona congestionată</li> <li>• Este nevoie de determinarea adreselor suplimentarea care vor fi folosite în cazul tunelării</li> </ul>

## 2.7 Soluții de rutare multicale

În literatura de specialitate există numeroase propuneri de soluții de rutare multicale. În această lucrare s-a realizat o clasificare a acestora pe baza mai multor criterii prezentate în tabelul de mai jos:

**Tabel 2-2: Clasificare soluții de rutare multicale**

	Categorie	Implementare concretă
Soluții de rutare multicale	Metode bazate pe variații ale algoritmului lui Dijkstra	ECMP (Equal-Cost Multi-Path) [Hop00]; CRA (Capacity Removal Algorithm) [Che99]; MPA (Multiple Path Algorithm) [Nar99]; MPATH () [Vut00]; QMPDA (Quality Multiple Partial Dissamination Algorithm) [Pal01]
	Metode bazate pe predicția parametrilor de trafic	Soluție bazată pe observarea trendurilor seriilor temporare pe toate conexiunile rețelei [Cai10]; MRATP (Multipath Routing Algorithm Traffic Prediction) [Li09b]
	Scheme de rutare distribuite și centralizate	Soluția de rutare multicale bazată pe CT (Colored Tree) [Ram07], [Jay09]; Rutare pe bază de reguli deflexie [Yua09], [Yan06]
	Metode bazate pe QoS	Rutare multicale cu rezervare de resurse [Rao98], [Zho00]; Rutare bazată pe jetoane [Che01]
	Metode bazate pe modele biologice	Rutare bazată pe modul de deplasare al coloniilor de furnici CACF (Concurrent ACO CMP Forward) [Han09]

## **2.8 Concluzii**

Evoluția protocoalelor de rutare folosite în Internet nu a fost pe măsură cu cea a rețelei (creșterea infrastructurii, a numărului de utilizatori și a tipurilor de aplicații), ca urmare a acestui fapt metodele folosite în majoritatea cazurilor sunt aproape neschimbate față de perioada de început. În acest capitol s-au prezentat câteva aspecte generale ale rutării (ce înseamnă de fapt acest proces). S-a continuat cu rutarea clasică și necesitatea introducerii rutării alternative multicale.

Soluția propusă în această lucrare este un algoritm de rutare multicale distribuit, conștient de starea rețelei. Din perspectiva modurilor de realizare a unei soluții de rutare multicale, algoritmul propus are următoarele caracteristici: 1) modul de calcul al rutelor este bazat pe determinarea celor mai scurte k căi, 2) divizarea traficului se face la nivel de flux, 3) comutarea pachetelor se realizează în mod distribuit și 4) rutarea se face conștient de starea rețelei.

## 3 Variantă modificată a algoritmului DFS (Depth-First Search) pentru determinarea căilor multiple într-un graf

### 3.1 Motivație

Înainte ca un router să fie capabil să comute traficul spre un anumit nod, acesta are nevoie de cunoașterea în prealabil a rutei spre acea destinație. În această lucrare se propune pentru determinarea căilor multiple într-un graf o variantă modificată a algoritmului DFS (Depth First Search). Se profită din plin de diversitatea căilor existente în graf fără a se ține cont de gradul de independență. Algoritmul propus în această lucrare se adresează în special domeniului rutării, scopul lui este de a deservei un algoritm de rutare în procesul de determinare a rutelor multiple spre destinațiile din rețea. În acest caz nodul sursă este considerat routerul la nivelul căruia are loc procesul de determinare a rutelor în rețea.

### 3.2 Algoritmi de căutare în graf

Căutarea rutelor în rețea reprezintă una din sarcinile oricărei soluții de rutare. Principalele metode pentru realizarea acestei sarcini sunt algoritmi de căutare într-un graf. În această reprezentare routerele sunt simbolizate prin intermediul nodurilor din graf iar conexiunile prin intermediul muchiilor care pot fi uni- sau bidirecționale în funcție de proprietățile legăturii fizice.

Există două moduri principale de explorare a unui graf: în adâncime DFS și în lățime BFS (Breadth First Search). În primul caz explorarea se face pe nivele, se avansează până în momentul când s-a ajuns la nodul căutat sau procesul de căutare nu mai poate înainta. În al doilea caz, în primă fază vor fi descoperite toate nodurile care sunt la o distanță  $k$  de sursă după care se trece la cele la o distanță de  $k+1$ . Pe parcursul procesului de căutare există trei stări posibile în care se poate afla un nod: nedescoperit, vizitat și descoperit. În prima fază toate nodurile sunt nedescoperite. Se pleacă de la sursă și se continuă cu nodurile adiacente.

În cazul protocoalelor de rutare bazate pe o singură cale, cum ar fi OSPF (Open Shortest Path First) sau IS-IS (Intermediate System To Intermediate System), se utilizează algoritmul lui Dijkstra. Pentru cele bazate pe vector distanță cum ar fi RIP (Routing Information Protocol), algoritmul folosit este Bellman-Ford iar pentru EIGRP (Enhanced Interior Gateway Protocol) metoda este DUAL (Diffusing Update ALgorithm). Deoarece algoritmi folosiți determină o singură cale între o pereche de noduri, pentru a folosi aceleași principii în cadrul soluțiilor de rutare multicale s-au dezvoltat variante modificate ale algoritmilor de rutare sau combinații ale acestora. De exemplu, pentru varianta multicale a protocolului OSPF, ECMP (Equal-Cost Multi-Path), algoritmul lui Dijkstra a fost modificat astfel încât, dacă între un nod sursă și unul destinație există mai multe rute cu costuri egale, ambele rute să fie folosite pentru transmisia datelor. Alte variante de determinare a căilor multiple sunt: variație a algoritmului lui Dijkstra bazată pe manipularea costurilor legăturilor [Che99], combinație între Dijkstra și DFS [Kau02], combinație DFS și BFS [Xi07] etc.

### 3.3 Algoritm de calcul al căilor multiple propus

În această lucrare se propune o variantă modificată DFS care garantează faptul că toate rutele fără cicluri între o pereche de două noduri din rețea vor fi determinate [Boa10a]. Ideea este de porni de la rădăcină și de a explora pe fiecare ramură până când este posibil, după care prin backtracking repetat se descoperă toate căile posibile între nodul sursă și destinație.

#### 3.3.1 Schema logică a algoritmului

Datele de intrare necesare algoritmului pentru calcularea rutelor între o pereche de noduri sursă-destinație sunt: nodurile rețelei și lista vecinilor direct conectați (listele de adiacență) pentru fiecare nod. Deoarece ideea sistemului propus este de a separa funcțiile de control (management) de cele de comutare al pachetelor, informațiile legate de topologie necesare algoritmului de determinare a rutelor sunt furnizate de aplicația de management prezentată în [Bar11a]. Soluția propusă se adresează rutării intra-domeniu. Ca urmare a acestui fapt este posibilă cunoașterea întregii rețele la nivelul fiecărui nod.

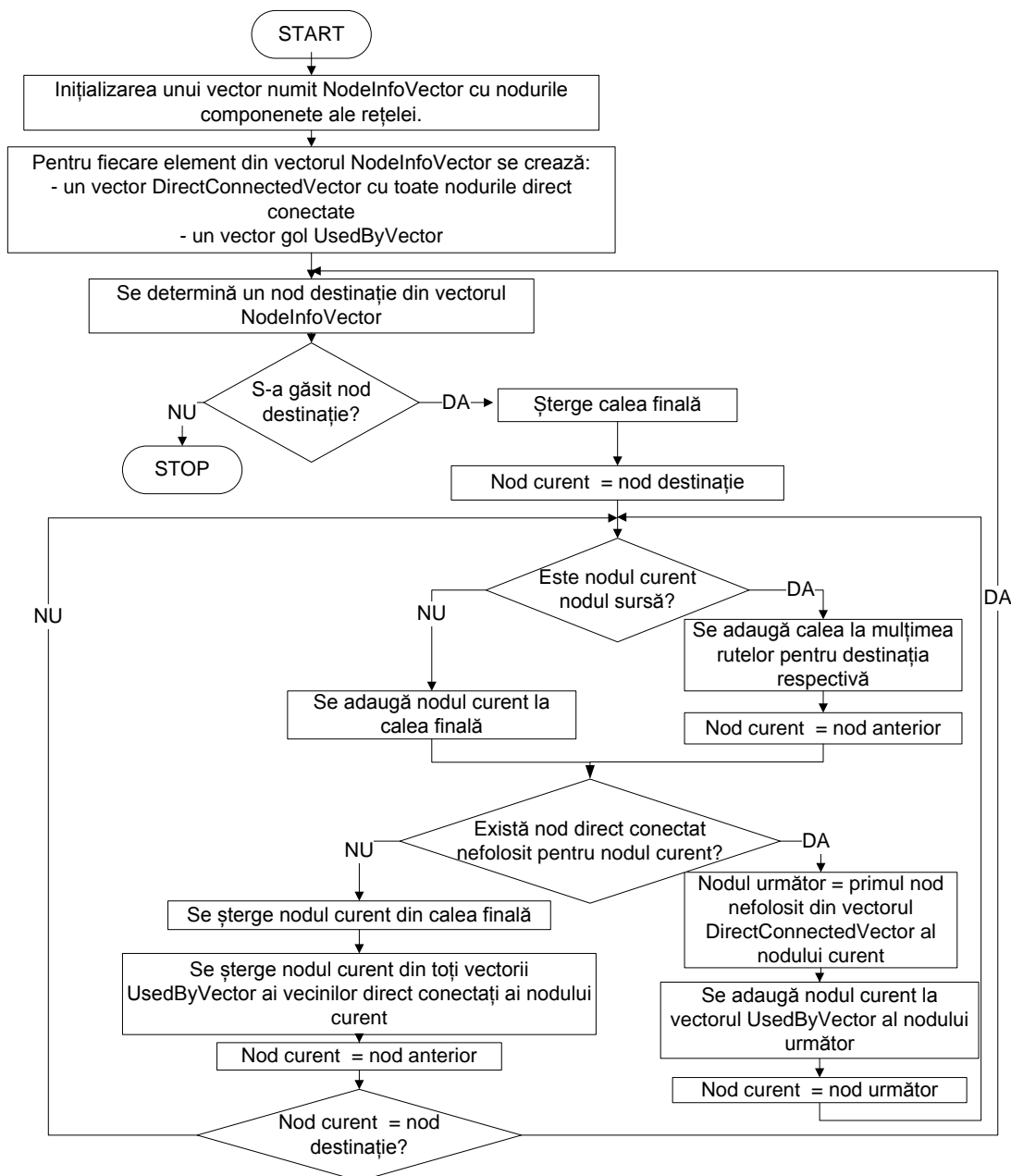


Figura 3-1: Schema logică a algoritmului de determinare a rutelor multiple [Boa10b]



### 3.3.2 Exemplu de funcționare al algoritmului

Pentru a exemplifica modul de funcționare al algoritmului se va prezenta modul de căutare pentru o rețea simplă dar care să ofere destule legături astfel încât să existe căi multiple între un nod sursă și unul destinație. Se urmărește determinarea tuturor rutelor între nodurile *S* și *D*. Căile găsite sunt semnalizate prin culoarea roșie. Se pleacă de la *D* și se determină primul nod direct conectat nevizitat până acum, nodul 2. Din acest punct procesul se repetă, se schimbă doar nodul curent pentru care se caută în listele de adiacență. În momentul în care s-a ajuns la sursă se salvează ruta și algoritmul se întoarce la ultimul nod vizitat. În mod similar se determină toate căile de la *D* la *S* care utilizează nodul direct conectat al destinației 2.

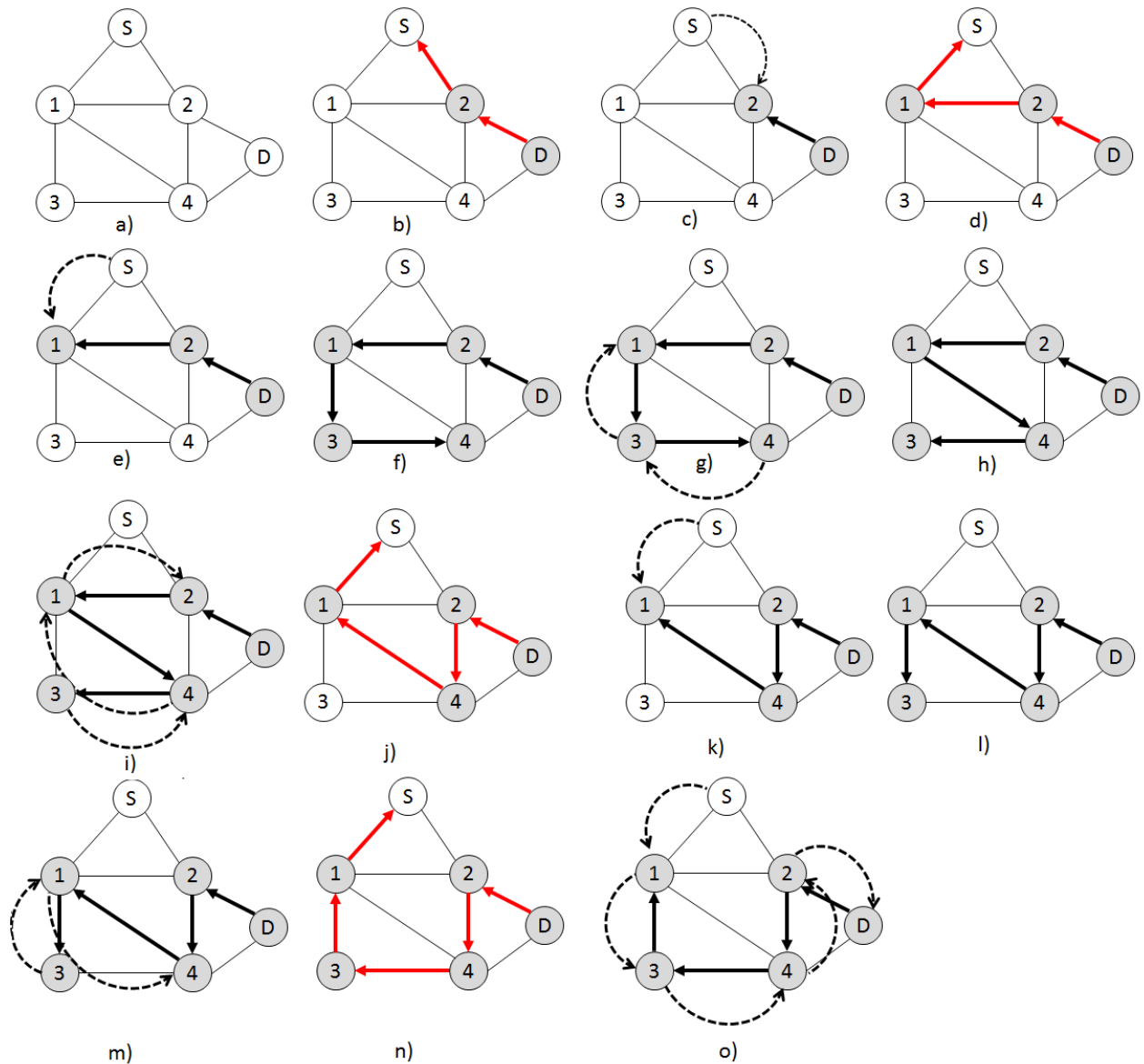


Figura 3-2: Exemplu rețea algoritm căi multiple

Pentru cel de-al doilea nod din lista de adiacență a destinației (nodul 4) procesul de determinare a căilor este similar.

### 3.3.3 Influența ordinii nodurilor în listele de adiacență

Ordinea de găsim a rutelor existente este influențată de ordinea nodurilor direct conectate. În funcție de poziția unui anumit nod în lista de adiacență el va intra, sau nu, în componența primelor rute determinate [Boa11b].

Pentru rețeaua din exemplul dat anterior, dacă parametrii legăturilor sunt similari, atunci ordinea de utilizare a rutelor va depinde doar de numărul de hopuri. Vor fi avantajate rutele care au ca prim nod de la destinație nodul 2 deoarece acesta se determină înaintea celor care au ca nod 4 pe aceeași poziție.

Ordinea în listele de adiacență poate fi folosită pentru a avantaja utilizarea anumitor noduri. Aceasta este o metodă simplă de a favoriza anumite routere fără a fi nevoie de introducerea unor informații de control sau schimbarea algoritmului de căutare al rutelor. În acest scop am definit un parametru al routerelor care reprezintă preferința de utilizare, *factorul de stres* [Boa11b], acesta este direct proporțional cu numărul de interfețe.

### 3.3.4 Varianta modificată a algoritmului cu determinarea unui număr redus de căi

În cazul în care algoritmul se aplică într-o rețea densă cu un număr mare de noduri, calcularea tuturor rutelor poate fi costisitoare din punct de vedere al memoriei și al timpului necesar. Pentru aceste situații am propus o variantă modificată a algoritmului în care se calculează doar căile care au primul hop de la sursă diferit, astfel, numărul de rute calculate depinde de numărul de interfețe ale unui nod. La nivelul fiecărui nod se vor calcula pentru fiecare interfață de ieșire rutele care prezintă cea mai scurtă distanță spre destinație.

## 3.4 Concluzii

Algoritmul de căutare propus este o variantă modificată a metodei DFS de căutare într-un graf. Soluția propusă garantează determinarea tuturor căilor într-un graf, fără a fi condiționată de valoarea costurilor legăturilor sau existența ciclurilor. Se profită de avantajele diversității existente în rețea fără a se ține cont de gradul de independență a căilor determinate. Ideea de bază a algoritmului este de a utiliza DFS pentru a explora graful de la un nod destinație la nodul sursă. Din acest punct se utilizează metoda backtracking în mod repetat pentru determinarea tuturor căilor între cele două noduri. S-a ales utilizarea DFS pentru că această metodă asigură determinarea rapidă a primei căi de la care se pornește căutarea întregului set.

## 4 Proiectarea unui nou algoritm de rutare multicale SAMP (Situation Aware Multipath)

### 4.1 Motivație

În cadrul acestui capitol se vor prezenta pașii de proiectare ai noii soluții de rutare multicale propuse, SAMP (Situation Aware Multipath). Metoda este parte componentă a unui sistem de rutare a datelor în cadrul unei rețele care mai are în componentă o aplicație de management [Bar11d] și o aplicație CLQ [Rus11] de măsurare a parametrilor legăturilor. Proiectarea a plecat de la ideea separării funcțiilor de gestiune de cele de comutare a pachetelor. În acest mod se eficientizează utilizarea informațiilor, datele stocate de aplicația de management pot fi utilizate și în alte scopuri, diferite de rutare. SAMP reprezintă entitatea care îndeplinește funcțiile executive ale sistemului.

### 4.2 Criteriile de proiectare

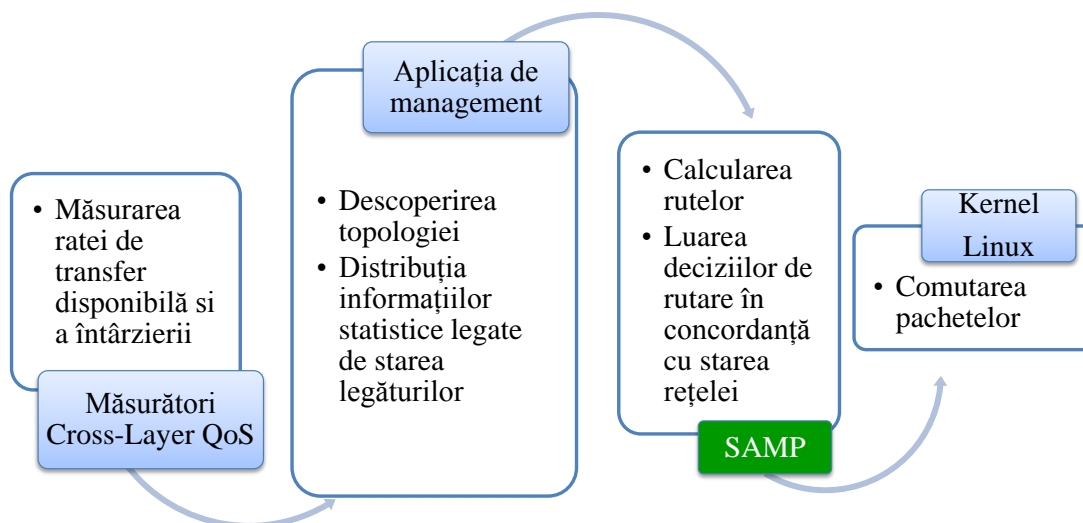
Procesul de proiectare a plecat de la mai multe criterii pe care soluția de rutare trebuie să le îndeplinească cum ar fi:

- Să fie o soluție de rutare multicale distribuită;
- Transmisia pachetelor să se facă simultan pe mai multe rute;
- Soluția să fie transparentă din punct de vedere al utilizatorului final;
- Metoda să permită comunicarea cu aplicația de management;
- Rutarea să se realizeze în funcție de condițiile reale existente în rețea;
- Soluția să fie capabilă să reacționeze în caz de congestie sau defect;

### 4.3 Interacțiunea modulelor sistemului de rutare

Algoritmul de rutare este într-o continuă colaborare cu programul de management care îi furnizează datele statistice legate de starea conexiunilor din rețea. Cea de a doua entitate cu care se comunică direct este reprezentată de kernelul Linux. Se utilizează capacitățile de rutare specifice acestor sisteme pentru comutarea pachetelor. Fată de aplicația de management, algoritmul de rutare are un comportament de subordonare fiind dependent de informațiile primite de la acesta. De asemenea, în procesul de comutare au loc modificări doar în momentul în care se primește informația că există probleme în rețea.

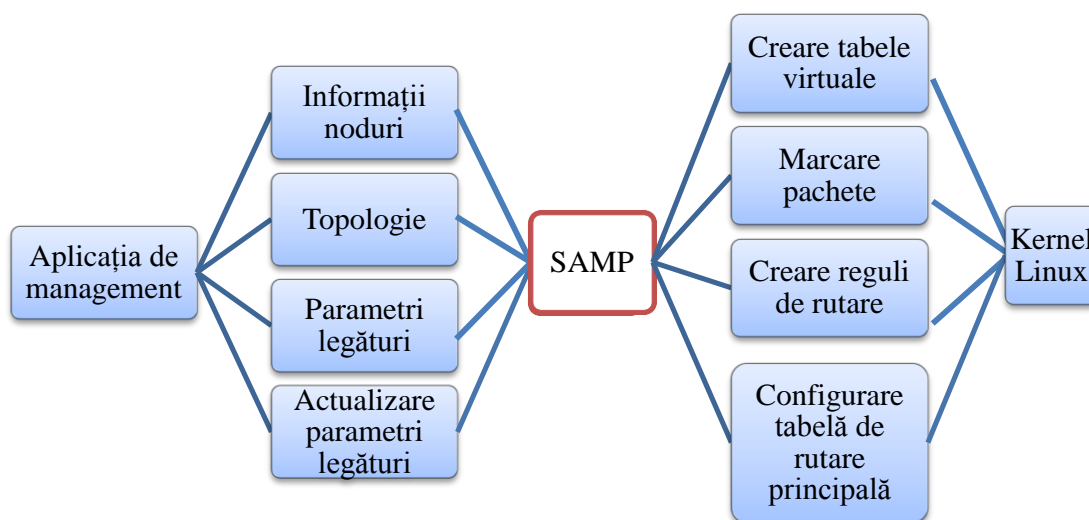
Din punct de vedere al interacțiunii cu modulul de comutare al pachetelor, acesta este comandat de SAMP prin manipularea tabelor de rutare și a fluxurilor care intră în routerul respectiv. Pentru a asigura compatibilitatea cu diferite distribuții ale kernelului Linux s-au folosit doar funcții standard ale acestuia. În figura de mai jos se prezintă fluxul de preluare a datelor, procesare a acestora și comutarea efectivă a pachetelor.



**Figura 4-1: Interacțiunea cu alte aplicații**

Se utilizează două moduri de comunicare[Boa10a]:

- Comunicarea pe bază de fișiere text: aplicația de management→SAMP;
- Comunicare pe bază de comenzi: SAMP→kernelul Linux.



**Figura 4-2: Comunicarea SAMP cu alte module**

## 4.4 Pașii de proiectare

În proiectarea unui sistem dinamic de rutare multicale trebuie avute în vedere mai multe aspecte, cum ar fi [Boa11c]:

### 1) Modul de calcul al rutelor multiple

Procesul de calculare al rutelor necesare este o parte componentă a oricărei soluții de rutare. Alegerea unui anumit algoritm depinde de proprietățile pe care se dorește să le aibă căile determinate. Prin urmare, se poate alege varianta unui număr restrâns de rute dar cu un grad mare de independență sau o mulțime cu mai multe rute dar care nu sunt disjuncte din punct de vedere al nodurilor sau al legăturilor.

În cazul algoritmului de rutare proiectat s-a urmărit să se profite din plin de diversitatea existentă în rețea și ca urmare, pentru a avea o gamă mai largă de rute disponibile nu se ține cont de gradul de independență al acestora. Pentru determinarea căilor multiple s-a utilizat algoritmul prezentat în capitolul precedent. Abordarea propusă presupune fie calcularea tuturor rutelor din rețea, fie determinarea unui număr restrâns de căi. Metoda de calculare a tuturor rutelor din rețea s-a ales pentru a se putea asigura în orice moment, indiferent de momentul și locul apariției congestiei în rețea, că sistemul va fi capabil să reacționeze la problemele apărute. În cazul în care există o rută care nu este afectată de congestie, acea rută va fi folosită.

### 2) Modul de utilizare al rutelor multiple

În funcție de scopul folosirii rutării multicale, există mai multe moduri de utilizare a rutelor multiple calculate. Trei variante de realizare a acestei operații sunt: 1) utilizarea rutelor multiple în mod consecutiv, 2) utilizarea tuturor rutelor multiple în mod simultan, 3) utilizarea unui număr restrâns de rute multiple în mod simultan.

În cazul soluției SAMP dezvoltate s-a ales utilizarea simultană tuturor rutelor disponibile la nodul de control al divizării pachetelor pe mai multe căi. În funcție de numărul de noduri de acest gen în rețea complexitatea soluției crește. Din punct de vedere al congestiei metoda alege acele rute care asigură evitarea zonelor cu probleme. Numărul de căi folosite depinde de topologia rețelei, numărul de conexiuni existente și numărul de noduri cu proprietăți de divizare ale traficului.

### 3) Modul de transmisie al pachetelor pe rute multiple

Din punct de vedere al modului de transmisie al pachetelor pe căile multiple determinate, există mai multe posibilități de a trata pachetele ajunse într-un nod care lucrează în modul multical în funcție de nivelul de granularitatea cu care se face divizarea pachetelor. Pachetele de la o sursă spre o anumită destinație pot fi împărțite la nivel de pachet, nivel de serviciu sau nivel de flux. Deoarece una din cerințele de proiectare era transparența față de utilizatorul final, în cazul soluției de rutare multicale propuse, SAMP, s-a ales divizarea traficului la nivel de flux.

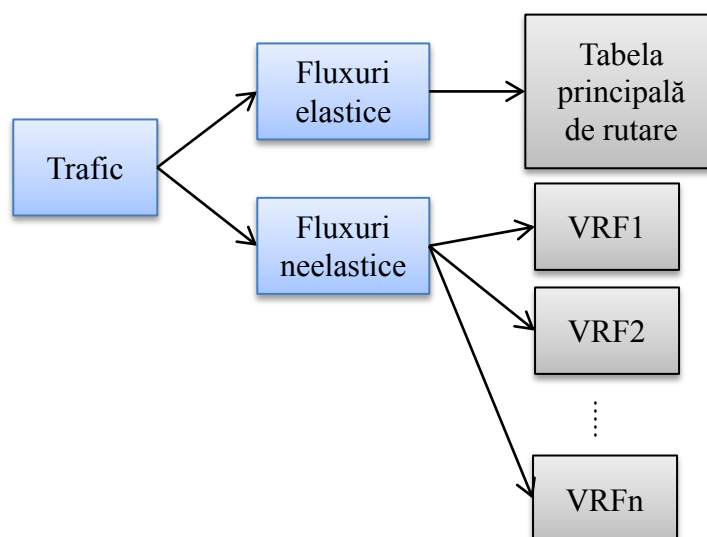
Pentru procesul de clasificare al pachetelor s-a adoptat idea prezentată în [Li09a]. Traficul este împărțit în două tipuri de fluxuri:

- *Fluxuri elastice* - caracterizate de toleranță la variația întârzierii și o cerere de rată de transfer ridicată, un exemplu ar fi datele de tip FTP (File Transfer Protocol).
- *Fluxuri neelastice* – caracterizate de o sensibilitate ridicată la variația întârzierii și o cerere de rată puțin variabilă. În această categorie intră transmisiile video sau de voce cum ar fi VoIP sau IPTV.

Tratarea pachetelor se face diferit în funcție de categoria fluxului de care aparțin acestea. Fluxurile din prima categorie nu vor fi identificate și vor fi comutate conform tablei principale de rutare. Aceste pachete au rol de trafic de fundal comparativ cu pachetele din a doua categorie. Pachetele care aparțin fluxurilor neelastice vor fi identificate pe baza a trei câmpuri: adresa IP destinație, adresa IP sursă și portul destinație. Alocarea fluxurilor pe rutele existente se face în ordine descrescătoare a performanțelor oferite de acestea.

Transmisia simultană pe mai multe rute a traficului între o pereche sursă-destinație se asigură prin utilizarea conceptului de tabele virtuale VRF (Virtual Routing Forwarding) [Cis09]. Ideea de bază este existența unor tabele de rutare multiple la nivelul fiecărui router pe lângă tabela de rutare de bază. Fiecare astfel de tabelă va cuprinde următorul hop pentru fiecare pachet care va fi tratat de VRF-ul respectiv. Pentru fiecare interfață se va crea un VRF care cuprinde doar o rută implicită cu nodul direct conectat ca și gateway. O altă abordare presupune introducerea unui nou strat de virtualizare, prin utilizarea de routere virtuale [Rus10c]. În cazul SAMP tratarea

diferită a pachetelor se asigură doar prin manipularea funcțiilor kernelului Linux fără utilizarea unor programe suplimentare.



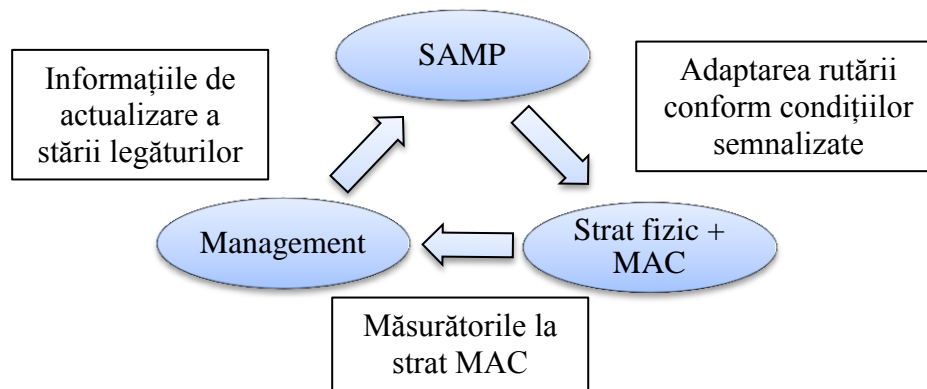
**Figura 4-3: Împărțirea traficului în fluxuri elastice și neelastice**

Divizare pachetelor pe mai multe rute în funcție de flux se va realiza doar de anumite noduri din rețea. SAMP nu introduce încărcare suplimentară în planul de date deoarece nu se transmit informații suplimentare în pachetele care circulă în rețea. S-a ales această abordare în care doar o parte din fluxuri sunt identificate și anume fluxurile neelastice deoarece acest tip de trafic este cel mai sensibil la variațiile parametrilor legăturilor din rețea.

SAMP realizează o rutare bazată pe starea reală a rețelei. Pentru a determina cea mai bună rută spre o anumită destinație se va calcula metrica acesteia pe baza parametrilor legăturilor rețelei primite de la aplicația de management. Parametrii de QoS măsurați sunt: rata de transfer disponibilă ATR (Available Transfer Rate) și întârzierea OWD (One Way Delay).

#### 4) Modul de actualizare al rutelor folosite

Procesul de alegere a rutelor folosite pentru transmisia pachetelor în rețea, se bazează pe starea reală a rețelei (parametrii legăturilor: rata de transfer disponibilă și întârzierea), informație furnizată de aplicația de management care colectează datele pentru fiecare legătură și le distribuie în fiecare nod. În acest mod în cadrul algoritmului SAMP se cunosc în fiecare moment cele mai bune rute existente în rețea. Proiectarea algoritmului de rutare multicale s-a făcut în strânsă legătură cu modul de comunicare cu aplicația de management și informațiile furnizate de aceasta.



**Figura 4-4: Actualizarea rutelor folosite**

În cazul în care nu apar modificări semnificative în cadrul rețelei, rutele alese în perioada inițială nu se modifică chiar dacă se schimbă ierarhia acestora. Principiul folosit este cel corespunzător ideii algoritmului Ford-Fulkerson [Cor09] și anume, după determinarea căii care are capacitate suficientă să transmită fluxul dorit, se va menține acea cale până în momentul în care parametrii legăturilor componente nu mai satisfac cerințele aplicației deservite, moment în care se va derula procesul de realocare a fluxurilor. Acest mod de abordare a fost ales cu scopul de a evita fluctuațiile frecvente în cadrul rețelei care ar putea afecta procesul de livrare al pachetelor la destinație.

Reacția algoritmului de rutare la apariția congestiei în rețea este de a realoca unul sau mai multe fluxuri care folosesc cale/căile afectate pe celelalte rute existente. Mutarea fluxurilor de pe calea afectată are loc în mod succesiv. S-a optat pentru varianta în care eliminarea problemelor cauzate de congestiei sunt rezolvate progresiv. La fiecare pas se mută doar o singur flux de pe ruta afectată. Dacă realocarea ar avea loc într-un singur pas, ar putea apărea situația în care ruta congestionată după acest proces să rămână liberă, situație nedorită deoarece se urmărește realizarea unei încărcări echilibrate a rețelei.

## 4.5 Concluzii

Proiectarea noului algoritm de rutare multicale SAMP a plecat de la ideea separării funcțiilor de management de cele de comutare din cadrul unui sistem de rutare. Această abordare presupune ca procesul de descoperire a topologiei și asigurarea comunicării între routere să fie realizate de aplicația de management. Pornind de la criteriile de proiectare s-au stabilit principalele caracteristici ale algoritmului de rutare multicale: 1) comutarea distribuită a pachetelor pe căi multiple, 2) utilizarea concurentă a rutelor calculate, 3) divizarea traficului la nivel de flux și 4) eliminarea problemelor datorate congestiei prin re-rutarea traficului neelastic.

## 5 Implementarea algoritmului de rutare multicale SAMP

### 5.1 Motivația

Procesul de implementare reprezintă punerea în practică a ideilor prezentate în capitolul anterior, în etapa de proiectare. Pentru realizarea aplicației de rutare multicale distribuite s-a optat pentru limbajul de programare C++ sub sistemul de operare Linux. În acest capitol se vor prezenta detaliile legate de implementarea la nivelul limbajului de programare a algoritmului de rutare multicale SAMP. Abordarea este una dinamică, unde rutele pentru o anumită destinație se schimbă în funcție de starea rețelei astfel încât zonele congestionate să fie evitate. Procesul de selecție al rutelor este influențat în timp real de condițiile legăturilor fizice. Informațiile statistice (sau prezise), bazate pe măsurători CLQ (Cross-Layer QoS) sunt furnizate de aplicația de management. Scopul soluției propuse este de a ameliora sau chiar elimina problemele cauzate de congestie și de a eficientiza utilizarea resurselor existente prin transmisia simultană pe mai multe căi a datelor între o pereche sursă-destinație.

### 5.2 Modul de funcționare al soluției de rutare multicale SAMP

Un sumar al pașilor care se realizează pe parcursul rulării aplicației sunt următorii[Boa10a]:

#### 1) Configurarea rețelei

Primul pas al algoritmului de rutare constă în configurarea nodurilor rețelei. În această lucrare se propune împărțirea rețelei în *domenii multicale*. Un domeniu este reprezentat de o submulțime de noduri. În cadrul unui domeniu routerele au capacități diferite. S-au definit două tipuri de noduri:

- *AMR (Adaptive Multipath Router)*. Acest tip de router va fi plasat la marginea unui domeniu. Capabilitățile acestor noduri cuprind funcții de divizare a traficului care intră în domeniul de care aparțin. În momentul în care fluxurile ajung într-un nou domeniu, acestea vor fi dispersate în acel domeniu.
- *AR (Adaptive Router)*. Rolul acestor noduri este de a asigura dirijarea pachetelor în interiorul domeniului. Pentru fluxurile identificate se vor alege căile stabilite de nodul de intrare în domeniu, iar pentru restul traficului se va utiliza cea mai bună rută disponibilă.

#### 2) Conștientizarea topologiei rețelei

Se preiau informațiile legate de topologia rețelei (nodurile și conexiunile) de la aplicația de management. La finalul acestei etape fiecare nod din rețea va avea o viziune globală a întregii topologii a rețelei, va cunoaște informații despre fiecare nod, vecinii săi și adresele corespunzătoare fiecărei conexiuni.

#### 3) Calcularea rutelor multiple

Determinarea tuturor rutelor din rețea se face pe baza algoritmului de calcul a căilor bazat pe o variantă modificată a algoritmului de căutare într-un graf DFS (capitolul 3), în acest mod se garantează determinarea tuturor rutelor lipsite de cicluri existente între nodul sursă și toate celelalte noduri (nodurile destinație). Se profită la maxim de diversitate din punct de vedere al rutelor existente în rețea și ca urmare nu se ține cont de gradul de independență al acestora.



#### 4) Inițializarea tabelelor de rutare

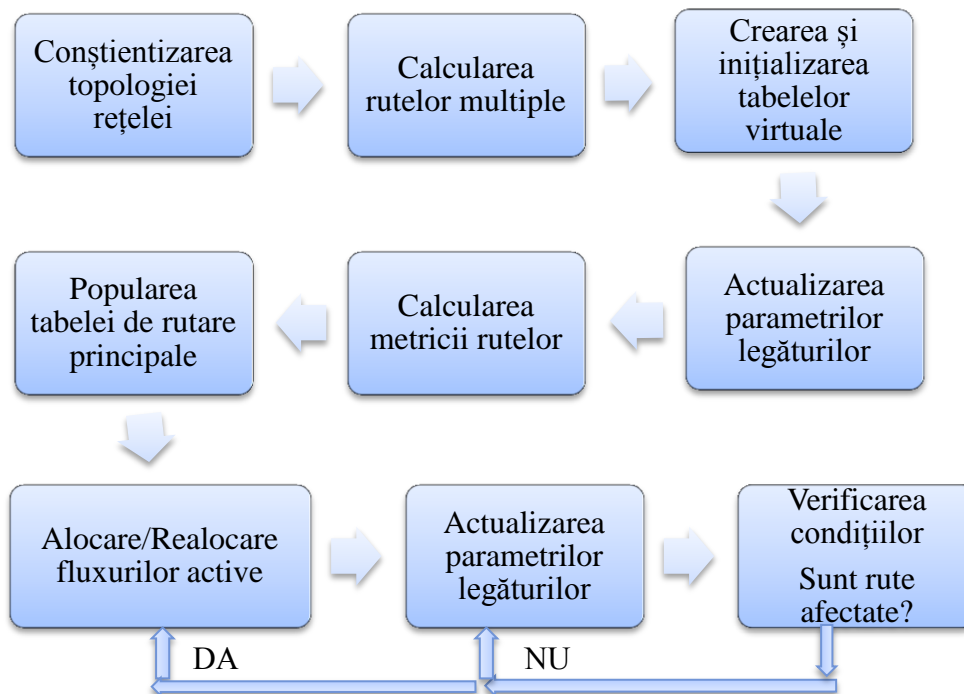
Pentru fiecare interfață a nodului se va crea o tabelă VRF care va avea o singură intrare cu o rută implicită care are ca hop următor nodul direct conectat la acea interfață. Din punct de vedere al tabelii principale de rutare, etapa de inițializare constă în ștergerea tuturor rutelor cu excepția celor direct conectate. Această operație este necesară pentru a evita situația în care dacă pentru o anumită destinație există deja o rută, pachetele să nu urmeze traseul impus de SAMP.

#### 5) Calcularea metricii fiecărei rute

Valoarea metricii compozite a fiecărei rute depinde de informațiile primite de la aplicația de management preluate prin intermediul unui fișier care conține parametrii mășurați ai conexiunilor. Aceste informații sunt date statistice (sau prezise) legate de rata de transfer disponibilă și întârziere. Informațiile conținute în fișier sunt actualizate în timp real, astfel încât SAMP va cunoaște starea reală a rețelei în fiecare moment.

#### 6) Alegerea rutelor cu cea mai bună metrică

În momentul în care algoritmul are o viziune globală asupra stării rețelei acesta poate să decidă care rute vor fi folosite pentru transmisia pachetelor spre o destinație. Tabela principală de rutare va conține pentru toate destinațiile rutele care prezintă cea mai bună metrică. Traficul elastic va fi tratat de această tabelă. Din punct de vedere al traficului neelastice, nodurile AMR identifică fluxurile și stabilesc rutele corespunzătoare. Routerule AR comută pachetele conform modulului impus de routerul AMR. În funcție de ruta aleasă pentru fiecare flux acesta va fi tratat de o anumită tabelă virtuală de rutare.



**Figura 5-1: Modul de funcționare SAMP**

#### 7) Actualizarea metricii rutelor

Acțiunea de actualizare a metricilor este condiționată de evenimentul actualizării parametrilor de rețea de către management. În momentul în care rata de transfer a scăzut sub un anumit prag, aplicația de control anunță algoritmul de rutare că au avut loc modificări semnificative în rețea. Ca urmare a acestui fapt are loc actualizarea parametrilor legăturilor și recalcularea metricilor rutelor folosite de fluxurile neelastice din rețea.

### 8) Verificarea condițiilor impuse

După actualizarea metricii rutelor se verifică dacă se respectă condițiile minime de transmisie. În cazul în care pe una din rutele alese a apărut congestie, are loc realocarea fluxurilor. Se mută progresiv câte un flux de pe ruta afectată pe ruta care oferă cele mai bune condiții cu parametri actuali până în momentul în care rata de transfer disponibilă ajunge să depășească un prag impus sau nu mai există fluxuri care ar putea fi mutate. Din punct de vedere al fluxurilor care nu sunt afectate de congestie, acestea își vor urma în continuare rutele alese inițial.

## 5.3 Evaluarea performanțelor algoritmului de rutare SAMP

Evaluarea performanțelor algoritmului de rutare multicale SAMP s-a realizat în cadrul unei rețele reale de calculatoare. S-a dorit demonstrarea fezabilității asigurării procesului de rutare prin intermediul soluției propuse. Performanțele au fost evaluate din punct de vedere al calității fluxurilor video la destinație pe baza a patru metrici de calitate video: procentul de pachete pierdute, frecvența de apariție a pierderilor, rata de succes și magnitudinea pierderilor [Bar11c].

Rezultatele obținute au fost comparate cu cele obținute în aceleași condiții de protocolul de rutare OSPF și două variante ale protocolului multicale ECMP care diferă din punct de vedere al interpretării noțiunii de flux. ECMP-like consideră aceeași definiție a unui flux ca și în cazul SAMP, adică acesta este identificat prin trei parametri (adresă IP destinație, adresă IP sursă și portul destinație). Varianta tradițională ECMP identifică fluxurile doar prin intermediul primilor doi parametri, această variantă va fi testată în cel de al doilea scenariu. ECMP este singura variantă de rutare multicale suportată de rouerele din actuala rețea Internet [Mer11].

### 5.3.1 Arhitectura rețelei de test

Testarea performanțelor algoritmului de rutare SAMP s-a realizat în cadrul unei rețele formate din șase mașini cu rol de router și două stații. Toate calculatoarele din cadrul testbed-ului rulează sistemul de operare Fedora. Conexiunile între stații sunt legături FastEthernet 100Mbps.

Pe fiecare nod de tip router rulează trei programe:

- Aplicația de rutare multicale SAMP;
- Aplicația de management [Bar11a] [Bar11c];
- Aplicațiile de măsurare a parametrilor de trafic (rata de transfer disponibilă, întârzierea) CLQ [Rus11].

Sistemul propus prezintă o buclă de control dirijată de starea legăturilor din rețea. După etapa de stabilire a rutelor în rețea, modul de rutare al pachetelor se va schimba doar în momentul apariției congestiei în rețea sau a modificării topologiei.

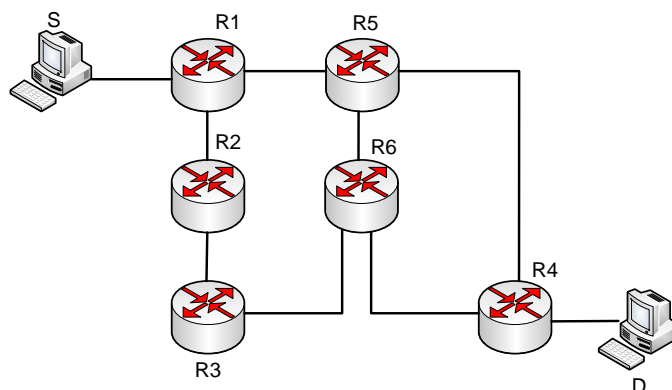


Figura 5-2: Topologia rețelei de test

Scopul testelor efectuate este de a studia comportamentul soluției propuse SAMP în condițiile apariției congestie în rețea și de a compara performanțele obținute cu cele ale rutării tradiționale. Traficul în rețea s-a asigurat prin transmisia mai multor fluxuri video RTP/UDP/IP de la stația sursă S la destinația D.

S-a considerat că întreaga rețea formează un domeniu multicale, cu următoarele noduri:  $AMR = \{R1, R4\}$  și  $AR = \{R2, R3, R5, R6\}$ . Traficul care intră în domeniu prin intermediul routerelor R1 și R4 va fi divizat în întreaga rețea.

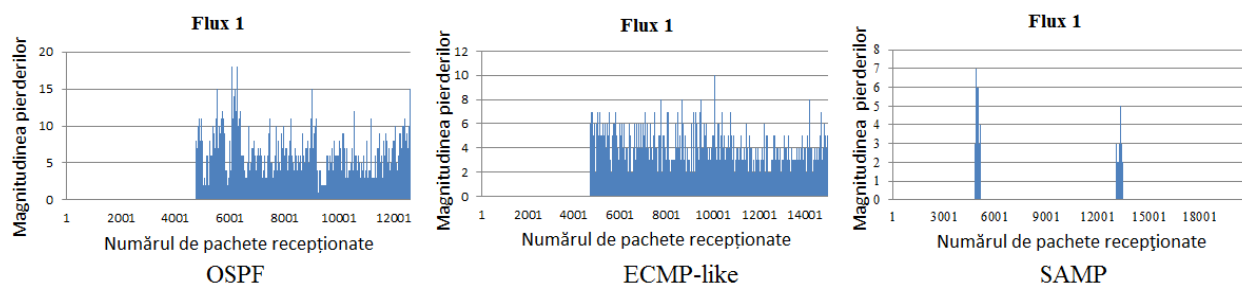
### 5.3.2 Rezultatele testelor

S-au realizat două scenarii de test diferite din punct de vedere al informației primite de la management. În primul caz rutarea se face pe bază de date statistice iar în al doilea pe bază de date prezise și a unui indicator de pierderi de pachete. Comportamentul soluțiilor testate s-a realizat din perspectiva apariției congestiei în rețea pe una sau două legături.

În ambele scenarii sistemul cu SAMP este singura soluție de rutare care reacționează la modificările apărute în rețea și ca urmare rutarea este adaptată la starea reală a parametrilor legăturilor.

#### 1) *Scenariul 1*: rutare bazată pe date statistice

În rețea se transmit două fluxuri video de la S la D. Se introduce congestie pe legăturile R5-R4, respectiv R5-R6 la un interval de un minut distanță. OSPF transmite tot traficul pe aceeași cale iar ECMP-like, la fel ca SAMP, împarte cele două fluxuri pe două rute diferite.



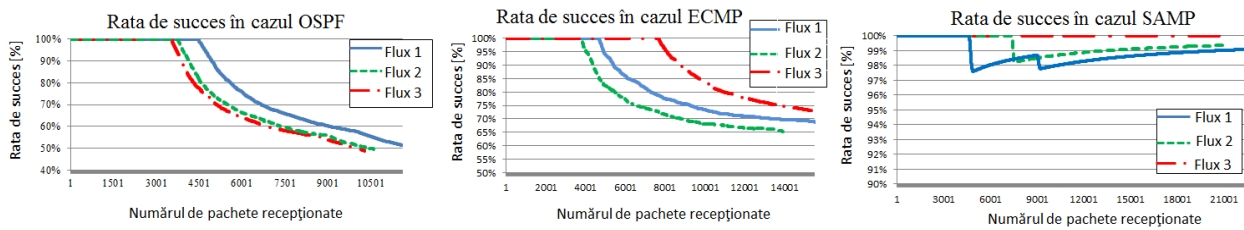
**Figura 5-3: Scenariul 1 – Magnitudinea pierderilor**

Metodele tradiționale de rutare OSPF, ECMP-like nu reacționează la modificările apărute în rețea. Pierderile de pachete debutează în momentul apariției congestiei și se mențin din acest punct pe tot parcursul transmisiei. Astfel, din punct de vedere al magnitudinii pierderilor se pot observa pierderi continue de blocuri de pachete. În cazul variantei multicale, ECMP-like, deoarece traficul este divizat pe mai multe rute, congestia afectează în mod diferit fluxurile transmise în funcție de ruta alocată, un flux este protejat.

SAMP este singura soluție testată conștientă de starea reală a rețelei. Se înregistrează pierderi de pachete doar pentru intervale scurte de timp între momentul apariției congestiei și conștientizarea problemei. Din punct de vedere al calității transmisiei video, imaginea va fi afectată doar pentru câteva momente. Se obțin următoarele rezultate în ceea ce privește procentul de pachete pierdute: 1% SAMP, 46% OSPF și 15% ECMP-like

#### 2) *Scenariul 2*: rutare bazate pe date prezise și indicator de pierderi

În acest caz în rețea se transmit trei fluxuri spre două destinații din aceeași rețea. OSPF are un comportament similar cu cazul anterior, iar ECMP împarte fluxurile pe rute diferite în funcție de nodul destinație. În cazul algoritmului SAMP fiecare flux este transmis pe o cale diferită. Se păstrează același procedeu de introducere a congestiei.



**Figura 5-4: Scenariul 2 - Rata de succes**

La fel ca și în scenariul 1, doar SAMP reacționează la congestie. Pierderile de pachete în cazul celorlalte soluții sunt continue din momentul în care fluxurile sunt afectate de congestie. Rata de succes scade în cazul OSPF în jur de 50% , iar pentru ECMP în jur de 65%. Deoarece SAMP modifică rutele utilizate astfel încât zona cu probleme să fie ocolită, rata de succes scade doar în momentul apariției congestiei. La finalul transmisiei valoarea acesteia este în jur de 99%. Transmisia este afectată pentru un interval mai scurt de timp, comparat cu primul scenariu, deoarece reacția la congestie este mai promptă datorită datelor precise și a indicatorului de pierderi de pachete. Situația pachetelor pierdute este următoarea:

**Tabel 5-1: Scenariul 2 Procentul global de pachete pierdute**

	Procentul global de pachete pierdute
OSPF	50%
ECMP	30%
SAMP	0,53%

## 5.4 Concluzii

În acest capitol sunt prezentate detaliile de implementare ale soluției de rutare multicale propuse în limbajul de programare C++ sub sistemul de operare Linux. SAMP reprezintă partea executivă a sistemului de rutare, fiind modulul care reacționează la modificările apărute în rețea pe baza datelor colectate de celelalte entități componente. Din punct de vedere al rutării pe bază de QoS, s-a ales varianta în care nu au loc rezervări de resurse ci se încearcă obținerea unei transmisii de calitate prin utilizarea în mod eficient a resurselor existente.

Evaluarea performanțelor s-a realizat pe baza a patru metrici video obiective: procentul de pachete pierdute, magnitudinea pierderilor, frecvența pierderilor și rata de succes. Rezultatele obținute au fost comparate cu cele obținute de protocolul de rutare OSPF și două variante ECMP, diferite din punct de vedere al interpretării noțiunii de flux. Testele realizate demonstrează superioritatea soluției propuse aceasta fiind singura care ține cont de starea reală a rețelei. Din punct de vedere al procentului de pachete pierdute, s-au obținut următoarele rezultate: primul scenariu 1% SAMP, 46% OSPF și 15% ECMP-like, al doilea scenariu: 0.5% SAMP, 50% OSPF și 30% ECMP. Se poate observa că din punct de vedere al algoritmului SAMP în al doilea caz transmisia de date este mai puțin afectată de congestie deoarece prin intermediul predicției și al indicatorului de pierderi de pachete detectarea congestiei este mai rapidă și ca urmare reacția algoritmului de rutare este mai promptă.

## 6 Proiectarea și implementarea unui modul de identificare a fluxurilor în cadrul simulatorului OPNET

### 6.1 Motivația

Modelul de rutare bazat pe tratarea diferită a pachetelor în funcție de fluxul de care aparțin necesită o etapă premergătoare și anume identificarea datelor. În acest capitol se prezintă pașii de proiectare și implementare a unui modul de identificare a fluxurilor din rețea, în cadrul simulatorului OPNET (Optimized Network Engineering Tools). Procesul de recunoaștere a fluxurilor se bazează pe trei câmpuri din pachetul IP: adresa IP destinație, adresa IP sursă și portul destinație. S-a ales această abordare deoarece se dorește diferențierea fluxurilor care au aceeași adresă sursă și destinație. În acest mod se oferă suport pentru soluția de rutare care împarte traficul între aceeași pereche de noduri pe mai multe căi.

### 6.2 Alegerea simulatorului de rețea

În momentul de față există pe piață un număr considerabil de simulatoare de rețea. Principalele caracteristici care diferențiază aceste aplicații sunt: acuratețea, viteza, ușurința de utilizare și costurile bănești [Flo03]. Câteva din cele mai cunoscute programe de simulare, utilizate atât în domeniul academic cât și cel industrial, sunt următoarele [Boa11a]: OPNET [OPN08], ns-2 [Fal10], ns-3 [Ns311], OMNet++ [Var08], JiST [Bar05].

Una din proprietățile importante ale unui simulator este fidelitatea față de sistemul real. Cu cât rezultatele obținute în cadrul aplicației sunt mai apropiate de cele obținute în testarea reală cu atât simulatorul este mai bun. În urma analizei realizate s-a optat pentru simulatorul OPNET din următoarele motive: 1) este un mediu orientat pe obiecte ceea ce face mai facilă implementarea și integrarea unor componente, 2) oferă o diversitate mare de componente implementate, multe dintre ele corespund unor echipamente reale, 3) stabilitatea crescută, 4) oferă modalități diverse de analiză a rezultatelor obținute, 5) fidelitatea față de sistemul real, 6) interfață grafică ușor de folosit și 7) crearea facilă a noilor topologii de rețea.

### 6.3 Implementarea modului de identificare a fluxurilor

Scopul acestui modul este de a menține o listă a fluxurilor active în rețea. Principalele sale acțiuni vor fi: prelucrarea pachetelor din rețea (extragerea câmpurilor de importanță), identificarea fluxurilor noi, actualizarea fluxurilor existente și ștergerea fluxurilor inactive. Rezultatele obținute de acest modul vor fi utilizate de algoritmul de rutare multicale SAMP în procesul de divizare al pachetelor de date pe rute diferite în funcție de fluxul de care aparțin.

#### 6.3.1 Definirea contextului

Modelele de procese descriu comportamentul unei componente în cadrul unui model de sistem mai complex. În acest caz sistemul este reprezentat de router. Un router este compus dintr-un grup de procese. În structura de bază a acestuia modulele sunt deja conectate și funcționează ca un întreg. Pentru realizarea acțiunii de identificare a fluxurilor la nivelul fiecărui nod, în cadrul

simulatorului OPNET, s-a implementat un modul special care va fi integrat în cadrul tuturor routerelor componente ale rețelei de test. Noul modul este compus dintr-un singur proces. Din punct de vedere al modului de creare, această acțiune se va efectua în mod static la începutul simulării. Prima etapă constă în identificarea modulelor independente ale sistemului. În această fază ne interesează doar elementele cu care noul modul va intra în contact și a căror evoluție va fi influențată de acesta. Pentru a evidenția relațiile ce se stabilesc între modulele independente s-a realizat o diagrama cu blocurile implicate și relațiile dintre acestea.

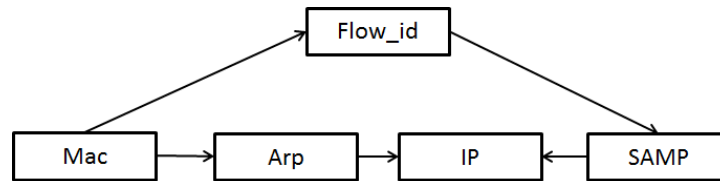


Figura 6-1: Diagrama blocurilor sistemului

### 6.3.2 Comunicarea cu alte module

Modulul de identificare a pachetelor comunică cu două module componente al routerului:

- *Modulul mac*: comunicare prin fluxuri de pachete;
- *Modulul de rutare multicale SAMP*: comunicare prin intermediul fișierelor binare și a întreruperilor la distanță.

În primul caz relația este de obținere a informației de la modulul *mac*, iar în al doilea caz informația prelucrată este livrată modulului de rutare. S-a ales varianta de oferire a informațiilor prin intermediul fișierelor binare deoarece în acest fel informația nu este specifică unui anumit modul și poate fi utilizată și în alte scopuri, diferite de rutare.

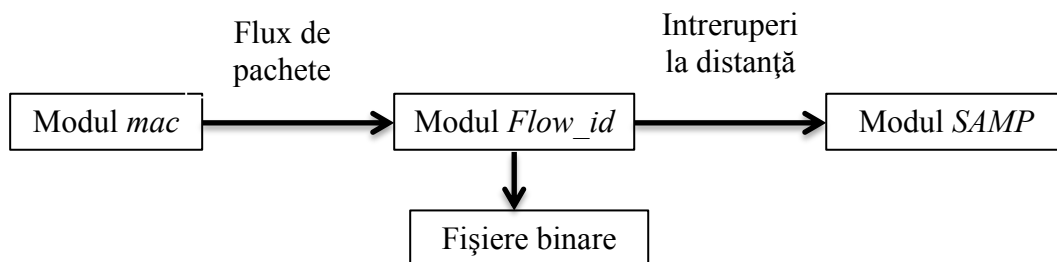


Figura 6-2: Comunicarea cu alte module

### 6.3.3 Managementul fluxurilor

Identificarea fluxurilor se realizează pe baza unui triplet format din: *adresa IP sursă*, *adresa IP destinație*, *portul destinație*. Astfel, dacă între un nod sursă și un nod destinație există mai multe fluxuri active, în funcție de portul destinație pachetele vor urma rute diferite. Procesul de extragere al pachetelor are loc la nivelul modulului *mac*.

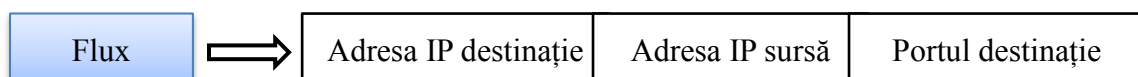


Figura 6-3: Definirea fluxului la nivelul modulului Flow\_id

Modul de funcționare al modulului presupune:

1. Crearea listei de fluxuri prin intermediul căreia se va ține evidentă acestora.
2. Recepționarea pachetului de la modulul *mac*.
3. Prelucrarea pachetului: descompunerea pachetului în antetele componente și preluarea datelor legate de flux (adresă IP sursă, adresă IP destinație, port destinație).

4. Identificarea pachetului, se verifică dacă acesta aparține unui flux deja identificat sau aparține unui nou flux.
5. Actualizarea listei de fluxuri:
  - a. Introducerea de fluxuri noi.
  - b. Actualizarea timpului ultimului pachet sosit.
  - c. Ștergerea fluxurilor care nu mai sunt active.

În funcție de tipul pachetului recepționat (aparține de flux nou sau deja existent) modulul va trimite entității de rutare multicale o întrerupere cu un anumit cod care îi indică acesteia modul de tratare a pachetului

## 6.4 Evaluarea performanțelor

### 6.4.1 Topologia rețelei

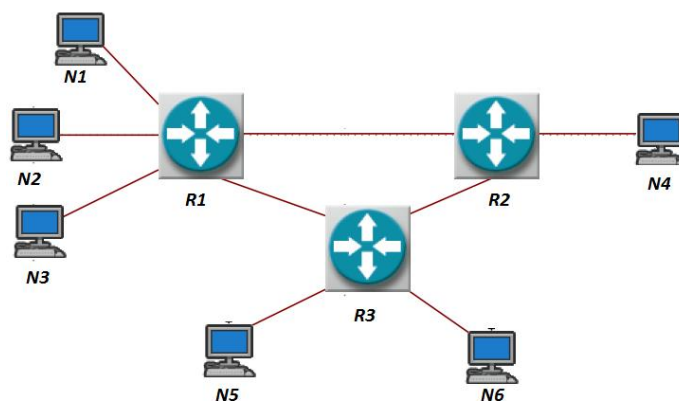
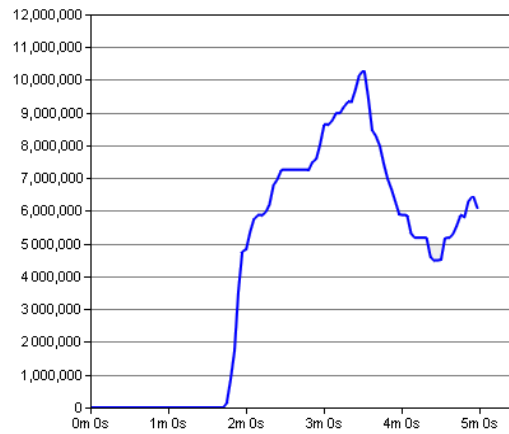


Figura 6-4: Topologia rețelei

Rețeaua de test are o topologie simplă în care nu s-a pus accentul pe procesul de rutare ci pe activitatea modulului de identificare a fluxurilor transmise de mașinile utilizatorilor. Elementele componente ale rețelei sunt: 3 noduri cu rol de router și 6 mașini utilizator. Pe lângă elementele fizice ale rețelei s-au mai definit un *modul de aplicații* și un *modul de profil* care definește comportamentul utilizatorilor.

### 6.4.2 Rezultate experimentale

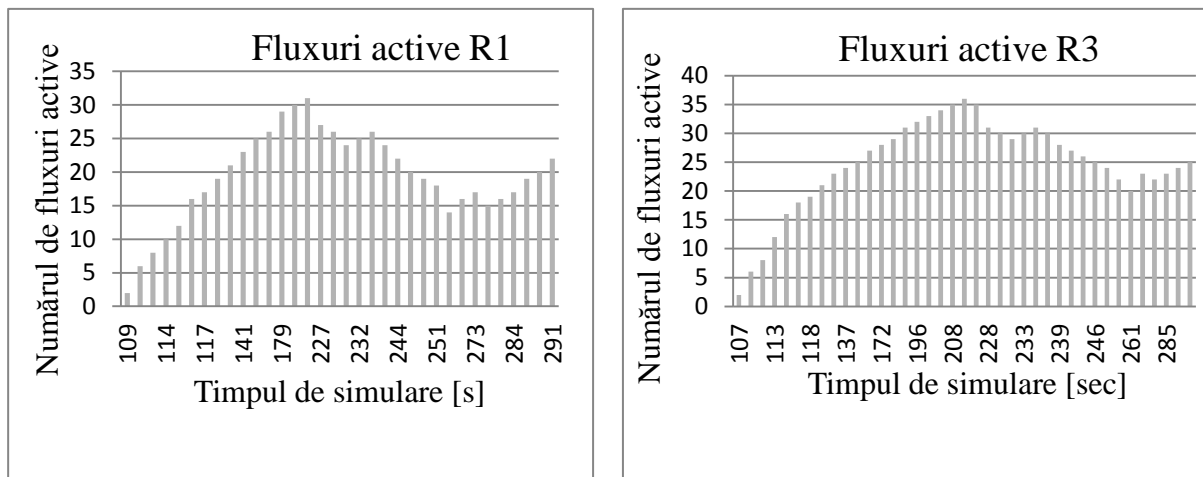
Testele realizate au constatat în transmisia unor fluxuri multiple de tip video-conferință între stațiile utilizator. Scopul simulărilor realizate este de a urmări fluxurile din rețea identificate de noul modul de identificare a fluxurilor. În figura de mai jos este prezentat graficul global în ceea ce privește rata de transfer a aplicației de tip video conferință.



Traficul video conferință global [Bps]

**Figura 6-5: Traficul de tip video conferință global**

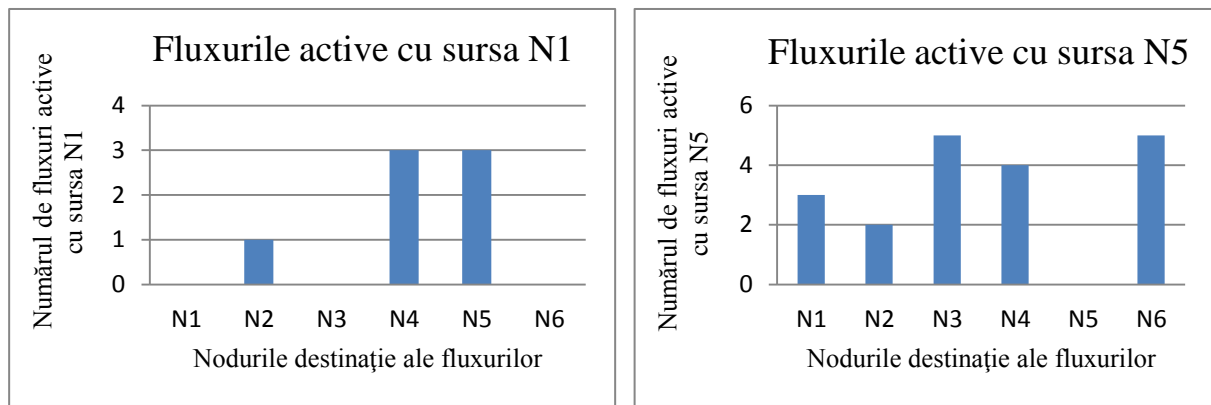
Deoarece în setările profilului utilizator s-a impus începerea și terminarea repetată a sesiunilor, se poate observa că după un interval de timp în jur de 1.7 minute de la începerea transmisiei (3.5 minute din timpul total de simulare) traficul începe să scadă urmând ca rata transmisiei să aibă din nou o pantă ascendentă spre finalul simulării. Pentru o simulare cu durata de 5 minute, din punct de vedere al traficului transmis în rețea, s-au obținut următoarele rezultate:

**Figura 6-6: Situația fluxurilor active din rețea**

Graficele din figurile de mai sus prezintă situația numărului de fluxuri active din rețea pe tot parcursul simulării. Intervalul maxim între două pachete consecutive ale unui flux s-a considerat de 15 secunde. Se poate observa că numărul de fluxuri active identificate de modulul flow\_id este în concordanță cu forma traficului transmis în rețea. Punctul de minim, după începerea transmisiei, este după 4 minute de simulare (255-260 secunde).

Traficul între două noduri utilizator va fi compus din mai multe fluxuri. De exemplu în cazul nodurilor N1, N5 și restul nodurilor din rețea, situația este următoarea:





**Figura 6-7: Situația fluxurilor active pentru nodurile N1 și N5**

## 6.5 Concluzii

Acest capitol prezintă proiectarea și implementarea unui nou modul de identificare a fluxurilor din rețea în cadrul simulatorului OPNET. Principalii pași în realizarea acestei entități au fost: 1) identificarea modulelor exterioare cu care se va comunica, 2) implementarea efectivă a acțiunilor specifice noului modul și 3) integrarea acestuia în cadrul structurii existente a routerului. Procesul de identificare are loc pe baza a trei câmpuri: adresa IP destinație, adresa IP sursă și portul destinație. Rolul noului modul este de a menține o listă a fluxurilor active în rețea.

Evaluarea performanțelor noului modul s-a realizat în cadrul unei rețele cu o topologie simplă în care se transmit multiple fluxuri video. Testele realizate demonstrează corespondența între rata de transfer a datelor din rețea și numărul fluxurilor identificate. În momentul în care transmisia totală își micșorează rata, se poate observa și scăderea numărului de fluxuri identificate la nivelul routerelor. Această asociere poate fi făcută deoarece în rețea se transmit fluxuri multiple de date de tip video-conferință (cu parametri apropiați) care necesită o rată aproximativ constantă pe parcursul unei sesiuni.

## 7 Realizarea și evaluarea performanțelor modulului SAMP în cadrul simulatorului de rețea OPNET

### 7.1 Motivația

În cadrul rețelelor de comunicare moderne unul din parametrii de bază îl reprezintă performanța, acest lucru fiind valabil atât în cazul rețelelor locale cât și a celor cu o întindere de nivel global. Utilizarea unui sistem de simulare reprezintă o soluție fiabilă cu costuri scăzute pentru testarea de noi scenarii cum ar fi adăugarea de noi elemente în rețea sau schimbarea protocoalelor utilizate. În acest capitol se va prezenta procesul de implementare și testare a algoritmului de rutare multicale SAMP în cadrul simulatorului de rețea OPNET. Se dorește demonstrarea capacității algoritmului de a îmbunătăți performanțele rețelei prin utilizarea eficientă a resurselor și atenuarea efectelor provocate de congestie prin evitarea zonelor problemă.

### 7.2 Implementarea modulului de rutare multicale SAMP

Principalele obiective ale algoritmului de rutare multicale propus sunt [Boa11a]: 1) distribuirea încărcării rețelei prin transmisia traficului între un nod sursă și un nod destinație pe mai multe căi și 2) rutarea pachetelor astfel încât să se evite zonele congestionate, obținându-se în acest mod o transmisie eficientă și de înaltă calitate.

După analiza structurii de bază a unui router în cadrul simulatorului OPNET s-a observat că protocoalele tradiționale, ca OSPF sau RIP, sunt reprezentate prin intermediul unui modul care comunică prin obiecte de tip flux de pachete cu modulul *ip*. Algoritmul de rutare multicale va fi reprezentat prin intermediul unui nou modul care va fi integrat în cadrul routerului. Nu există comunicare între modulul *ip* și modulul *samp*, deoarece *samp* nu transmite date în rețea ci doar manipulează tabelele de rutare.

Topologia rețelei la nivelul acestui modul se presupune cunoscută, această funcție fiind asigurată de partea de management. În momentul declanșării procesului de determinare a rutelor în rețea, informația de configurare se consideră că este disponibilă.

Primul pas în proiectarea acestui modul a fost găsirea unei posibilități de interacțiune între structura de bază și noul modul. Integrarea entității s-a realizat prin modificarea elementului *ip*, astfel în momentul în care configurarea din punct de vedere a adreselor IP este gata, modulul *ip* va transmite o întrerupere elementului de rutare pentru a-l anunța că poate începe procesul de creare a tabelor de rutare. Acest tip de comunicare este specific colaborării între *ip* și modulele de rutare.

Ideea principală în procesul realizării rutării multicale este utilizarea conceptului de tabele de rutare virtuale, VRF. Acest concept presupune existența mai multor tabele de rutare independente în cadrul aceluiași router. În acest mod traficul poate fi tratat diferit în funcție de flux.

#### 7.2.1 Modul de funcționare

1. *Inițializarea*: acțiuni specifice OPNET cum ar fi: obținerea identicatorului modulului și a nodului de care aparține, înregistrarea în registrul global și proceduri de inițializare a topologiei și configurare a routerelor.

2. *Calcularea rutelor multiple și a metricilor corespunzătoare*: se utilizează varianta DFS modificată propusă de determinare a căilor multiple într-un graf cu constrângerea de utilizare a nodurilor mai puțin stresate.
3. *Pregătirea tabelelor de rutare*: pentru fiecare interfață se crează o tabelă de rutare virtuală care va conține o singură intrare cu rută implicită. Gatewayul în cazul fiecărei tabele va fi reprezentat de nodul direct conectat la interfața respectivă. Tabela principală de rutare va conține cele mai bune rute spre toate destinațiile disponibile în acel moment.
4. *Divizarea traficului pe rute multiple*: pe baza informațiilor obținute de la modulul de identificare a traficului fiecărui flux  $i$  se va alocă o anumită rută, ca urmare va fi tratat de un anumit tabel de rutare virtual.

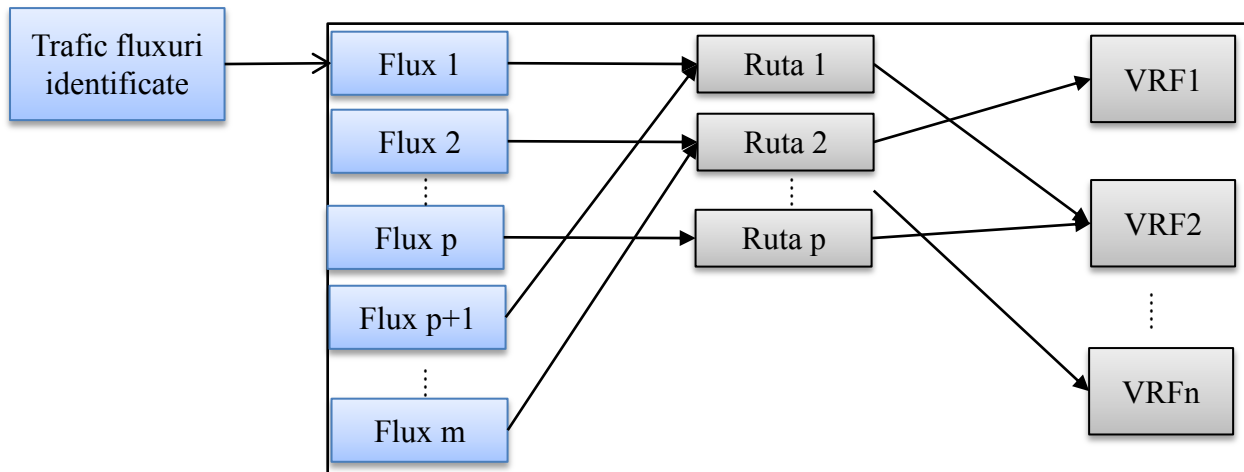


Figura 7-1: Divizarea traficului pe rute multiple

5. *Actualizarea rutelor alocate în caz de congestie*: are loc re-rutarea rapidă a traficului afectat prin realocarea aleatorie a câte unui flux de pe ruta cu probleme.

## 7.3 Evaluarea performanțelor algoritmului SAMP în cadrul simulatorului OPNET

Evaluarea performanțelor algoritmului de rutare SAMP se vor demonstra în cadrul unei rețele implementate în cadrul simulatorului OPNET. Rezultatele obținute se vor compara cu cele ale unuia din cele mai folosite protocoale de rutare, OSPF, varianta sa multicale ECMP și EIGRP [Alb94], [Rie09].

### 7.3.1 Arhitectura rețelei de test

SAMP nu este dependent de o anumită topologie de rețea. Cu toate acestea, pentru a putea demonstra capabilitățile de distribuție a traficului și utilizare eficientă a resurselor rețelei, arhitectura rețelei de test trebuie să ofere cel puțin două rute între o pereche de noduri sursă-destinație.

Rețeaua de test este compusă din 14 noduri cu rol de router și două mașini utilizator. Pe lângă elementele fizice ale rețelei s-au mai definit un *modul de aplicații* și un *modul de profil* care definește comportamentul utilizatorilor.

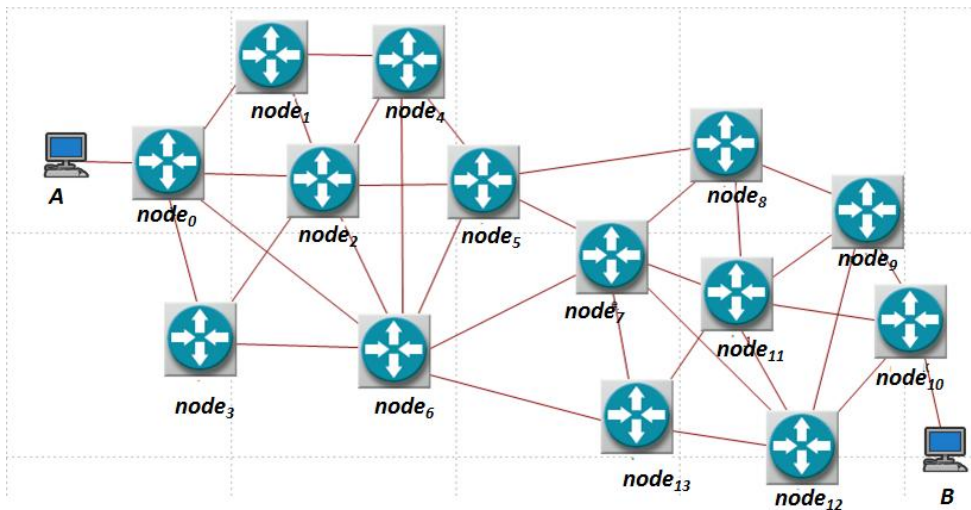


Figura 7-2: Topologia rețelei de test

### 7.3.2 Împărțirea rețelei în domenii de rutare multicale

Pentru asigurarea unei divizări echilibrate a traficului în rețea, topologia de test a fost divizată în două domenii de rutare multicale. S-a ales o divizare în care numărul de noduri din fiecare domeniu este egal. Acesta este doar un exemplu de împărțire și nu reprezintă o regulă sau o constrângere. Pe lângă cele două domenii formate din routere s-a mai considerat și un al treilea compus din mașinile utilizator.

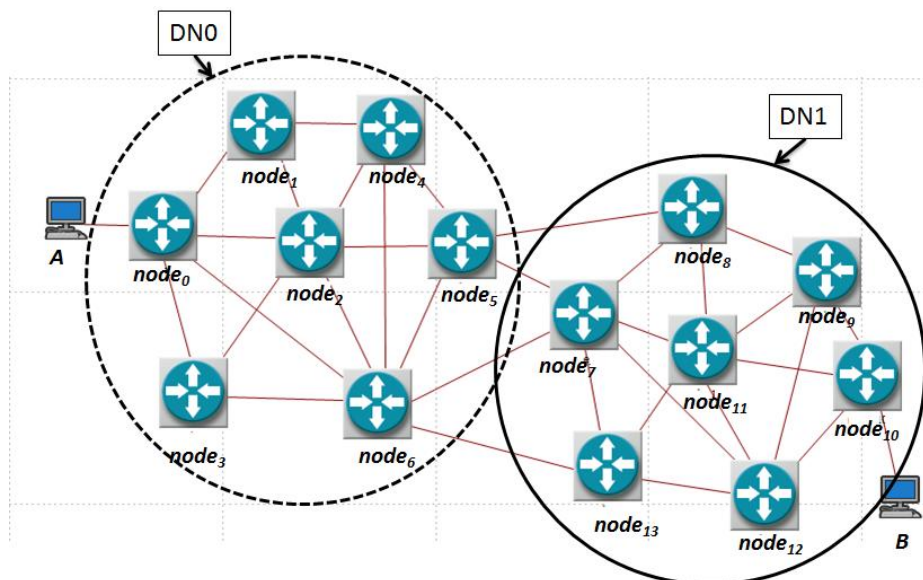


Figura 7-3: Domeniile rețelei

Pentru fiecare domeniu se stabilesc nodurile AR și AMR

$$AR_{DN0} = \{node_1, node_2, node_3, node_4, \}$$

$$AMR_{DN0} = \{node_0, node_5, node_6, \}$$

$$AR_{DN1} = \{node_9, node_11, node_12\}$$

$$AMR_{DN1} = \{node_7, node_8, node_10, node_13\}$$

Nu există o limită în ceea ce privește numărul de domenii în care poate fi împărțită rețeaua, numărul maxim este egal cu numărul nodurilor din rețea. Prin această divizare se stabilește

pentru care interfața de intrare se va diviza traficul pe mai multe rute. De asemenea numărul de noduri dintr-un domeniu nu este impus, acesta poate varia de la de unu la toate nodurile din rețea.

### 7.3.3 Scenariile de test

Testarea performanțelor algoritmului de rutare multicale SAMP s-a realizat prin intermediul a două scenarii. În ambele cazuri rezultatele obținute se vor compara cu performanțele obținute de protocoalele de rutare OSPF, ECMP și EIGRP. Traficul transmis, de la *A* la *B* și viceversa, este generat de aplicații standard oferite de simulatorul OPNET de tip FTP și video-conferință. Parametrii urmăriți de-a lungul simulărilor sunt *rata de transfer* și *întârzierea cap-la-cap*. Se va analiza evoluția acestora în diferite condiții de simulare. S-a ales o durată a simulării de 5 minute, deoarece s-a considerat că acesta reprezintă un interval de timp suficient pentru demonstrarea punctelor esențiale ale noului algoritm de rutare multicale implementat.

#### 7.3.3.1 Cazul 1: nu există congestie în rețea

În acest prim caz se consideră că legăturile rețelei sunt libere și în rețea circulă doar traficul între nodurile *A* și *B*, ambele având atât rol de sursă cât și rol de destinație.

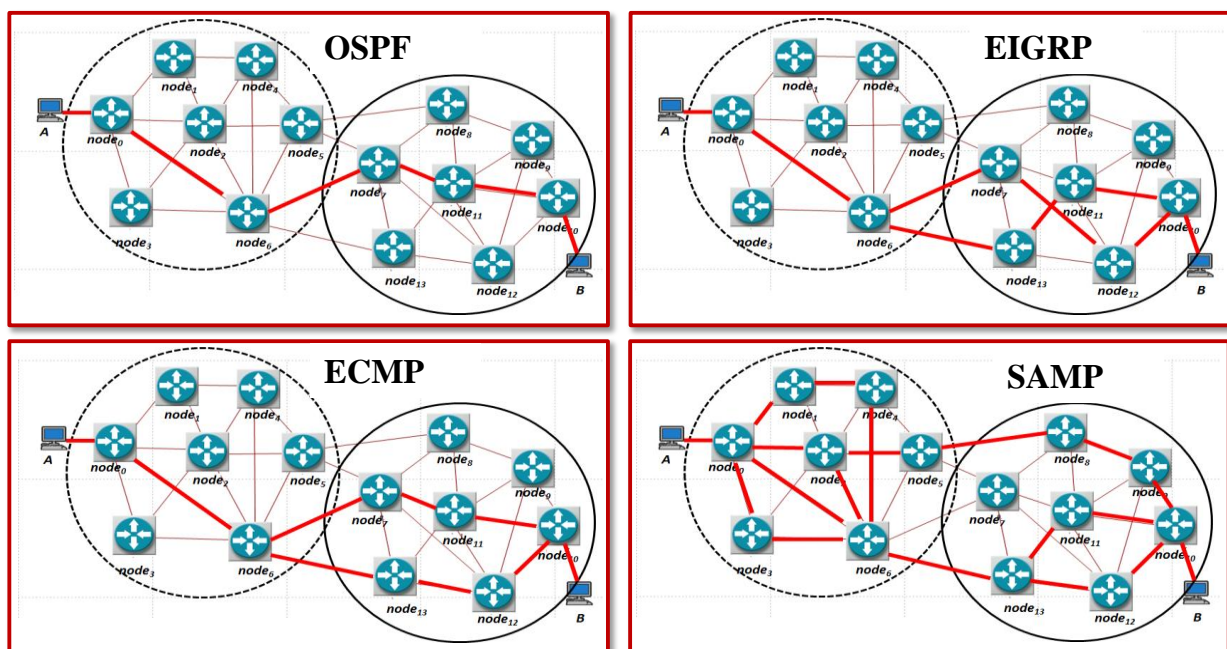


Figura 7-4: Utilizarea legăturilor rețelei

În figura de mai sus este prezentat modul de utilizare a legăturilor pentru asigurarea transmisiei datelor de la *A* la *B* și viceversa în cazul celor patru soluții de rutare testate. OSPF și EIGRP utilizează o singură cale pentru rutarea pachetelor. Diferența între cele două constă în faptul că EIGRP transmite traficul pe căi diferite în funcție de direcția acestuia. ECMP împarte traficul, cu o granularitate la nivel de pachet, în cadrul nodului 6, pe două rute cu costuri egale. În cazul algoritmului de rutare propus, SAMP, se poate observa că numărul legăturilor utilizate este mai mare față de oricare din celelalte soluții testate. Divizarea traficului are loc în două puncte deoarece datele traversează două domenii de rutare multicale. În cazul traficului de la *A* la *B* nodurile de divizare sunt reprezentate de nodul 0 și nodul 13, iar pentru direcția opusă nodul 10 și nodul 6. Din punct de vedere al eficienței utilizării resurselor conexiunilor rețelei, SAMP s-a

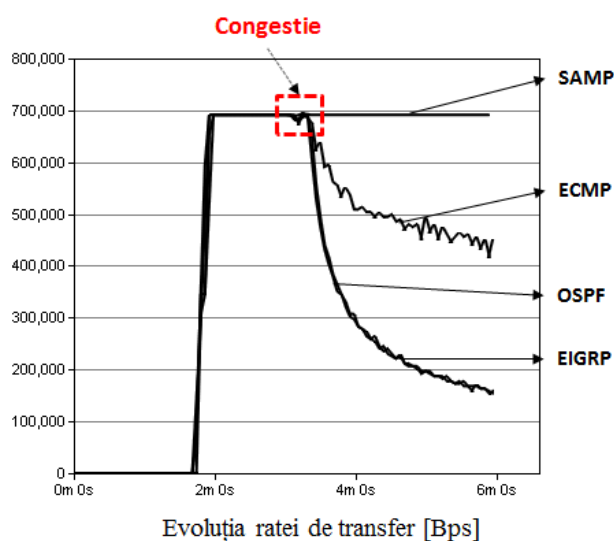
dovedit a fi superior soluțiilor bazate pe o singură cale: OSPF și EIGRP, dar și comparativ cu soluția multicală ECMP.

**Tabel 7-1: Gradul de utilizare al resurselor existente**

Metoda de rutare	Gradul de utilizare al resurselor existente
SAMP	38%
OSPF	13%
ECMP	22%
EIGRP	13%

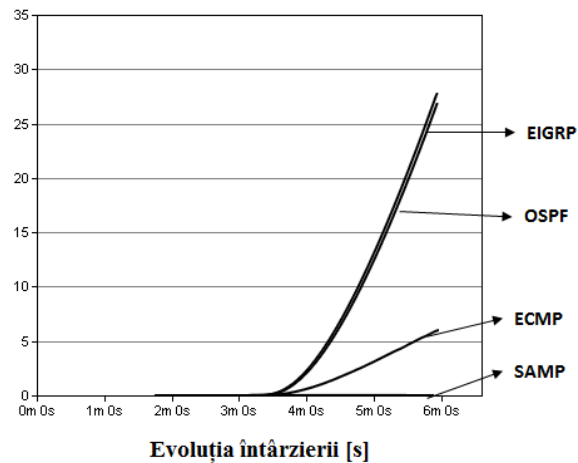
### 7.3.3.2 Cazul 2: congestie pe una din legăturile din rețea

În acest caz se introduce trafic de fundal cu o rată de 1Gbps pe legătura dintre nodurile 11 și 10, consecința fiind congestionarea conexiunii respective. În cazul utilizării algoritmului SAMP fluxurile de date sunt afectate doar pentru o perioadă scurtă de timp, intervalul între momentul în care apare fenomenul de congestie și momentul în care această situație este conștientizată de aplicația de rutare.



**Figura 7-5: Evoluția ratei de transfer**

Din figura de mai sus se poate observa că restul soluțiilor testate nu reacționează la congestie și ca urmare a acestui fapt rata de transfer scade cu o pantă accentuată. Din punct de vedere al evoluției întârzierii, la fel ca și în cazul ratei de transfer doar SAMP reușește să mențină valoarea acestui parametru la valori care asigură o transmisie de calitate.



**Figura 7-6: Evoluția întârzierii**

Valorile foarte ridicate în cazul întârzierii s-au obținut din cauza faptului că s-a considerat o situație extremă în care conexiunea cu probleme este congestionată aproape în totalitate. Prin această abordare s-a dorit să se demonstreze că soluțiile existente de rutare nici în aceste situații nu reacționează la probleme. Scopul acestui test a fost de a demonstra capabilitatea protocolului de rutare multicale implementat, SAMP, de a atenua efectele congestiei prin realizarea unei rutării conștiente de starea rețelei

## 7.4 Concluzii

Simularea este una din etapele importante în dezvoltarea unui algoritm de rutare multicale. Acest pas permite testarea în medii diferite și de asemenea se poate observa comportamentul algoritmului în cadrul unei rețele extinse. În capitolul de față se prezintă implementarea, integrarea și testarea în cadrul simulatorului de rețea OPNET a unui modul de rutare multicale SAMP. Principalul obiectiv în procesul de simulare a fost demonstrarea scalabilității algoritmului și a avantajelor utilizării acestuia în cadrul unei rețele.

Evaluarea performanțelor s-a realizat din punct de vedere al procentului de resurse din rețea (numărul de conexiuni) utilizat, al ratei de transfer globale asigurate și a întârzierii. Rezultatele au fost comparate cu cele obținute de protocoalele: OSPF, ECMP și EIGRP. În urma testelor realizate s-au demonstrat capabilitățile de distribuție a traficului și evitarea congestiei ale noului algoritm de rutare multicale SAMP. Astfel, din punct de vedere al utilizării resurselor rețelei, în cadrul topologiei testate s-a observat o îmbunătățire semnificativă în cazul algoritmului SAMP, soluția propusă oferă un procent de utilizare a resurselor de aproape trei ori mai mare decât OSPF și EIGRP (38% comparat cu 13%) și comparat cu metoda de rutare multicale, s-a adus o îmbunătățire de peste 50%. Din punct de vedere al ratei de transfer și al întârzierii, SAMP este singura soluție testată care ține cont de starea rețelei. Ca urmare a acestui fapt rata de transfer este afectată doar pentru o perioadă scurtă de timp față de celelalte metode care mai pot asigura o rata sub un sfert din cea necesară.

## 8 Contribuții

### 8.1 Sumarul contribuțiilor

În continuare se vor prezenta contribuțiile prezentate în această teză.

#### 1. Evaluarea și clasificarea metodelor de rutare multicale

În cadrul acestei contribuții s-a realizat o imagine de ansamblu a aspectelor legate de rutarea multicale. În urma identificării dezavantajelor rutării tradiționale s-a realizat o clasificare a soluțiilor de rutare multicale pe baza mai multor criterii: 1) algoritmul de determinare a rutelor multiple (găsirea primelor  $k$  cele mai scurte rute sau calcularea de rute disjuncte), 2) tehnica de divizarea traficului (la nivel de pachet, flux sau flowlet), 3) modul de comutarea al pachetelor (centralizat sau distribuit), 4) reacția la modificările din rețea (configurații de rutare multiplă, metode proactive sau tunelare). S-au identificat de asemenea avantajele și dezavantajele metodelor propuse. Pentru a exemplifica modurile de implementare a unei soluții de rutare multicale, s-a realizat o trecere în revistă a câtorva soluții concrete existente în literatură. Metodele prezentate au fost împărțite în mai multe categorii cum ar fi: 1) soluții bazate pe variante modificate ale algoritmului lui Dijkstra, 2) soluții bazate pe predicția parametrilor legăturilor, 3) metode bazate pe QoS, 4) soluții distribuite și centralizate și 5) metode bazate pe modele biologice.

Publicații: [Boa09], [Pol 09], [Boa11c], [Cor11]

#### 2. Variantă modificată a algoritmului DFS pentru determinarea căilor multiple într-un graf

O componentă a oricărei soluții de rutare este algoritmul de determinare a rutelor din rețea. Pentru acest proces topologia rețelei este reprezentată sub forma unui graf în care nodurile sunt routerele iar muchiile conexiunile între acestea. S-au analizat doi dintre algoritmi de bază de calculare a căilor într-un graf: BFS (Breadth First Search) și DFS (Depth First Search). Aceste soluții sunt specializate în determinarea unei singure căi între o sursă și o destinație, ca urmare nu pot fi folosite în cadrul unei metode de rutare multicale. Pornind de la algoritmi de bază s-au analizat variante modificate sau combinații ale acestora, capabile să determine un set de căi multiple. Câteva dintre acestea sunt: DT (Dijkstra Transversal), Dijkstra+DFS, DFS+BFS și alte variații ale algoritmului lui Dijkstra. S-a propus o variantă modificată a algoritmului DFS prin care se determină toate căile între un nod sursă și un nod destinație. Ideea este de a pleca de la un nod destinație și de a explora graful conform DFS până la sursă. Din acest punct prin backtracking repetat se determină toate rutele între acea sursă și destinație. S-a ales utilizarea DFS pentru că această metodă asigură determinarea rapidă a primei căi de la care se pornește căutarea întregului set. S-a propus și o versiune modificată a algoritmului, potrivită pentru operatori, care determină un set restrâns de rute multiple. În acest caz se preferă căile care vor reprezenta rute cu gateway diferit față de căile care asigură un cost minim. Nu se abordează problema multi-homing. Dacă pentru un anumit gateway există mai multe rute cu cost minim, toate vor fi luate în considerare.

Publicații: [Boa10a], [Boa10b], [Boa11c], [Rus10b]



### 3. Proiectarea unui nou algoritm de rutare multicale SAMP (Situation Aware Multipath)

Ideea de proiectare de la care s-a pornit a fost separarea funcțiilor de rutare de cele de management. S-a ales această abordare deoarece deși s-a considerat că soluțiile de rutare sunt capabile să se ocupe și de atribuțiile de management, acest lucru nu se face în mod eficient. Protocoalele de rutare nu reușesc să facă față tuturor problemelor din rețea, una din acestea fiind congestia. Plecând de la această idee s-a proiectat un algoritm de rutare multicale conștient de starea rețelei (SAMP) care nu cuprinde printre funcționalități: descoperirea topologiei și colectarea informațiilor legate de nodurile sau legăturile din rețea. Toate aceste informații sunt asigurate de aplicația de management. Principalele atribuții ale algoritmului SAMP sunt utilizarea eficientă a resurselor legăturilor rețelei prin transmisia datelor pe căi multiple și rezolvarea problemelor cauzate de congestie prin re-rutarea traficului astfel încât zonele problemă să fie evitate. Principalele criterii de proiectare ale algoritmului SAMP au fost: 1) să fie o soluție de rutare multicale distribuită, 2) transmisia pachetelor să se facă simultan pe mai multe rute, 3) să se asigure transparența din punct de vedere al utilizatorului, 4) metoda să permită comunicarea cu aplicația de management, 5) rutarea să se realizeze în funcție de condițiile reale existente în rețea și 6) soluția să fie capabilă să reacționeze în caz de congestie sau defect. Principalele caracteristici ale algoritmului de rutare stabilite în procesul de proiectare sunt: utilizarea simultană a rutelor multiple calculate, divizarea la nivel de flux a datelor, comutarea pachetelor în funcție de starea rețelei (se evită zonele problemă în caz de congestie). Rutarea este condiționată de primirea informației statistice (sau pe baza de predicție) legate de starea conexiunilor rețelei (rata de transfer disponibilă și latența) de la management care se bazează pe măsurători în timp real oferite de CLQ (Cross Layer QoS).

Publicații: [Boa10a], [Boa10b], [Boa11c], [Rus10a], [Rus10c], [Bar11a], [Bar09]

### 4. Implementarea algoritmului de rutare multicale SAMP

Algoritm de rutare proiectat în contribuția anterioară a fost implementat în limbajul de programare C++ sub Linux. Comutarea traficului între o pereche sursă-destinație pe mai multe rute se realizează fără a interveni în structura pachetului. Informațiile necesare rutării (topologia, parametrii legăturilor) sunt furnizate de aplicația de management, rutarea fiind dependentă de această comunicare. Rutarea se face conștient de starea rețelei, astfel se asigură utilizarea celor mai bune rute și de asemenea evitarea zonelor congestionate. Pentru transmisia diferită a datelor spre o destinație la nivelul unui nod, s-a utilizat conceptul de tabele virtuale de rutare VRF (Virtual Routing Forwarding). Testarea algoritmului SAMP s-a realizat, în primă fază, în cadrul unei rețele reale compuse din 6 noduri cu rol de router și 2 stații cu rol de sursă-destinație. Evaluarea performanțelor sistemului format din SAMP+DFS modificat (capitolul 3), aplicația de management [Bar11a], [Bar11c] și modulul CLQ [Rus10a] s-a realizat prin compararea rezultatelor din punct de vedere al unor metrici de calitate video (procentul de pachete pierdute, magnitudinea pierderilor, frecvența pierderilor și rata de succes) cu protocolul OSPF și două variante ale protocolului ECMP. S-au efectuat două scenarii de test care diferă din punct de vedere al informației primite de la management. În primul caz rutarea se face pe bază de informații statistice legate de parametrii legăturilor iar în al doilea caz pe bază de date precise și se mai introduce un nou parametru, pierderile la nivelul fiecărui nod. În ambele scenarii sistemul propus s-a dovedit a fi net superior protocoalelor testate, fiind singurul care ia în considerare starea reală a rețelei. Din punct de vedere al procentului de pachete pierdute, s-au obținut următoarele rezultate: primul scenariu 1% SAMP, 46% OSPF și 15% ECMP-like, al doilea scenariu: 0.5% SAMP, 50% OSPF și 30% ECMP. În ceea ce privește cele două variante ale

sistemului, varianta cu predicție s-a dovedit a fi mai performantă deoarece asigură o reacție mai promptă la congestie.

Publicații: [Boa10a], [Boa10b], [Bar11a], [Bar11b], [Bar11c]

## 5. Proiectarea și implementarea unui modul de identificare a fluxurilor în cadrul simulatorului OPNET

Utilizarea unui simulator permite verificarea scalabilității soluției de rutare multicale propuse, în cadrul unei rețele mai extinse față de topologia reală utilizată (capitolul 5). Întrucât SAMP este un algoritm care realizează divizarea traficului pe mai multe rute cu o granularitate la nivel de flux, are nevoie de informațiile legate de pachetele care circulă în rețea. Pentru a asigura furnizarea acestor date s-a proiectat și implementat în cadrul simulatorului OPNET un modul de identificare a fluxurilor. Alegerea acestui simulator s-a realizat pe baza facilităților oferite, cum ar fi: 1) fidelitatea față de mediul real, 2) faptul că este un mediu orientat pe obiecte ceea ce face mai facilă implementarea și integrarea unor componente, 3) oferă modalități diverse de analiză a rezultatelor obținute, 4) dispune de o diversitate mare de componente implementate. Modulul *flow\_id* proiectat în ProtoC (limbaj specific OPNET) identifică fluxurile din rețea pe baza a trei câmpuri: adresa IP destinație, adresa IP sursă, portul destinație. Rolul entității este de a realiza o listă a fluxurilor active din rețea și de a furniza aceste informații modului de rutare SAMP. Integrarea noului element s-a realizat prin modificarea modului *mac* existent astfel încât acesta să comunice cu noua entitate prin fluxuri de pachete. Interacțiunea cu SAMP se are loc prin intermediul întreruperilor la distanță și a fișierelor binare. Testarea componentei *flow\_id* s-a realizat în cadrul unei rețele cu o topologie simplă în care s-au transmis mai multe fluxuri video (cu aceeași parametri). Din rezultatele obținute s-a putut observa corelația între rata de transmisie din rețea și numărul de fluxuri identificate.

Publicații: [Boa11a], [Boa11b]

## 6. Realizarea și evaluarea performanțelor unui modul SAMP în cadrul simulatorului OPNET

Etapa de simulare reprezintă o etapă importantă în dezvoltarea unui algoritm de rutare deoarece oferă posibilitatea determinării situațiilor neprevăzute, ajustarea de parametrii și de asemenea o varietate din punct de vedere al testelor care pot fi efectuate. Deși s-a realizat și o implementare reală a algoritmului de rutare multicale propus, pentru testarea comportamentului într-o rețea extinsă s-a ales implementarea unui modul SAMP în cadrul simulatorului OPNET. Rolul acestui modul este de a asigura rutarea pachetelor pe căi multiple în funcție de starea rețelei. Informațiile care în mod normal erau obținute de la aplicația de management, în acest caz se consideră cunoscute. S-a păstrat aceeași idee de a nu modifica conținutul pachetelor, comutarea acestora fiind asigurată prin controlul tabelelor de rutare. Și în acest caz transmisia spre aceeași destinație pe căi diferite a fluxurilor se face folosind principiul VRF. Integrarea modului s-a realizat prin modificarea comportamentului modului *ip* prezent în structura routerului. Acțiunile noii entități sunt de a crea tabela de rutare principală și tabellele virtuale. Pe baza informațiilor primite de la modulul de identificare a fluxurilor (capitolul 6) va activa o anumită tabelă de rutare care va fi utilizată de modulul *ip* în procesul de comutare. Evaluarea performanțelor s-a realizat din punct de vedere al procentului de resurse din rețea (numărul de conexiuni) utilizat, al ratei de transfer globale asigurate și a întârzierii. Rezultatele au fost comparate cu cele obținute de protocoalele:

OSPF, ECMP și EIGRP. SAMP oferă un procent de utilizare a resurselor de aproape trei ori mai mare decât OSPF și EIGRP (38% comparat cu 13%) iar față cu metoda de rutare multicale, s-a adus o îmbunătățire de peste 50%. Este de asemenea singura soluție testată care ține cont de starea legăturilor și ca urmare reacționează la congestia din rețea prin evitarea zonei problemă. Ca urmare a acestui fapt, în cazul SAMP, rata de transfer este afectată doar pentru o perioadă scurtă de timp față de celelalte metode care mai pot asigura o rata sub un sfert din cea necesară.

Publicații: [Boa11a], [Boa11b]

## 8.2 Observații finale

Teza de față prezintă o soluție de rutare multicale conștientă de starea rețelei, SAMP, care face parte dintr-un sistem de rutare al informației în viitorul Internet. S-a propus o nouă abordare, în care funcțiile de management sunt separate de cele de rutare ale pachetelor. În prima etapă, pentru a identifica neajunsurile modului actual de comutare al datelor, s-a realizat o clasificare a metodelor de rutare multicale, identificându-se avantajele și dezavantajele acestora. Calcularea rutelor către toate destinațiile din rețea reprezintă o operație prezentă în cadrul oricărei soluții de rutare, în acest scop s-a realizat o variantă modificată a algoritmului DFS care determină toate căile într-un graf.

Pe baza studiului realizat asupra soluțiilor de rutare multicale existente, în a doua etapă s-a proiectat o soluție de rutare multicale distribuită conștientă de starea rețelei, SAMP. În această fază s-au ales concret modurile în care soluția va reacționa în diferite situații.

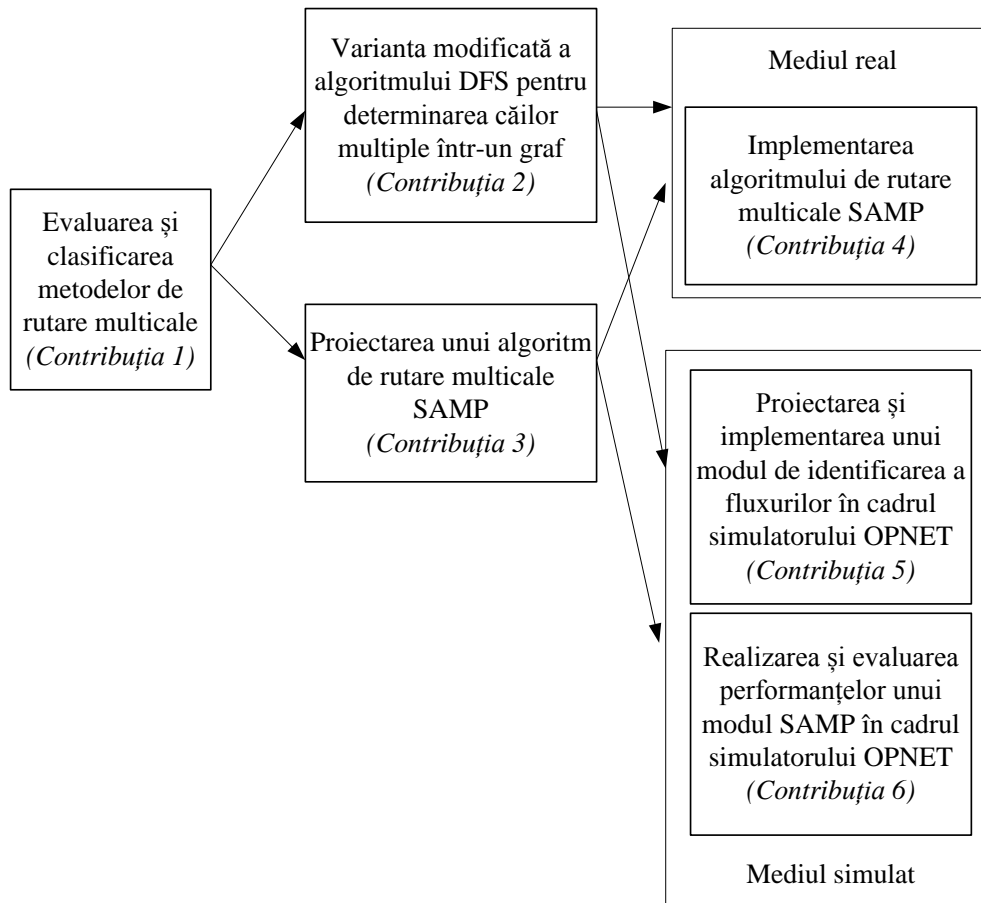


Figura 8-1: Structura contribuțiilor tezei

În următoarea etapă, în primă fază, s-a realizat implementarea și evaluarea performanțelor soluției propuse în mediul real, urmând că în ultima etapă, pentru demonstrarea scalabilității, evaluarea performanțelor să se realizeze în mediul simulat.

### 8.3 Premii

- **Mențiune** la Simpozionul Studentesc de Electronică și Telecomunicații - SSET 2010, organizat de Facultatea de Electronică, Telecomunicații și Tehnologia Informație, Universitatea Tehnică Cluj-Napoca, Mai 2010.

### 8.4 Lista publicațiilor personale

#### Cărți

- [Cor11] L. M. Correia, H. Abramowicz, M. Johnsson & K. Wünnel (editors), V. Dobrota, **G. Boanea** (inclusă în lista de autori) s.a., "Chapter 12 - Prototypes", *Architecture and Design for the Future Internet*, 4WARD Project, Series: Signals and Communication Technology, 1st Edition., 2011, XXIX, Springer, ISBN: 978-90-481-9345-5

#### Articole indexate BDI

- [Bar09] M. Barabas, **G. Boanea**, K. Steenhaut, V. Dobrota, „*Evaluating the Performances of the CastGate Tunnel Server over TCP and UDP Links in Multi-Client Configuration*,” *ACTA TECHNICA NAPOCENSIS*, Electronics and Telecommunications, ISSN 1221-6542, Vol.50, No.4, 2009
- [Bar11a] M. Barabas, **G. Boanea**, A. B. Rus, V. Dobrota, „*Routing Management Based on Statistical Cross-Layer QoS Information Regarding Link Status*,” 11th International Conference on Knowledge in Telecommunication Technologies and Optics KTTO 2011, pp. 12–17, ISBN 978-80-248-2399-7, Szczyrk, Poland, June 2011
- [Bar11b] M. Barabas, **G. Boanea**, A. B. Rus, V. Dobrota, J. Domingo-Pascual, „*Evaluation of Network Traffic Prediction Based on Neural Networks with Multi-task Learning and Multiresolution Decomposition*,” IEEE International Conference on Intelligent Computer Communication and Processing ICCP 2011, Cluj-Napoca, Romania, August 2011
- [Bar11c] M. Barabas, **G. Boanea**, V. Dobrota, „*Multipath Routing Management using Neural Networks-based Traffic Prediction*,” The Third International Conference on Emerging Network Intelligence, Lisbon, Portugal, November 2011 (accepted)
- [Boa10b] **G. Boanea**, M. Barabas, A. B. Rus, V. Dobrota, „*Design Principles and Practical Implementation of a Situation Aware Multipath Routing Algorithm*,” 17th Int. Conf. on Software, Telecommunications & Computer Networks IEEE SOFTCOM 2010, Split-Bol (Island of Brac), Croatia, Print ISBN: 978-1-4244-8663-2, INSPEC Accession Number: 11637618, pp.321-325, September 2010

- [Boa11b] **G. Boanea**, M. Barabas, A. B. Rus, V. Dobrota, J. Domingo-Pascual, „*Performance Evaluation of a Situation Aware Multipath Routing Solution*,” RoEduNet IEEE International Conference “Networking in Education and Research”, pp. 51–56, ISSN 2247-5443. Iași, Romania, June 2011
- [Boa11c] **G. Boanea**, M. Barabas, V. Dobrota, „*An Overview of Today’s Multipath Routing*,” *ACTA TECHNICA NAPOCENSIS*, Electronics and Telecommunications, ISSN 1221-6542, Vol.52, No.3, 2011 (submitted)
- [Pol09] Z. Polgar, Z. Kiss, A. B. Rus, **G. Boanea**, M. Barabas, V. Dobrota, „*Preliminary Implementation of Point-to-Multi-Point Multicast Transmission Based on Cross-Layer QoS and Network Coding*,” 17th Int.Conf. on Software, Telecommunications & Computer Networks *IEEE SOFTCOM 2009*, Split-Hvar, Croatia, Print ISBN: 978-1-4244-4973-6, INSPEC Accession Number: 10951348, pp.131-135, September 2009
- [Rus10a] A. B. Rus, M. Barabas, **G. Boanea**, Z. Kiss, Z. Polgar, V. Dobrota, „*Cross-Layer QoS and Its Application in Congestion Control*,” IEEE Workshop on Local and Metropolitan Area Networks LANMAN 2010, Long Branch, NJ, USA, ISSN: 1944-0367, pp.1-6, May 2010
- [Rus10b] A. B. Rus, V. Dobrota, A. Vedinas, **G. Boanea**, M. Barabas, „*Modified Dijkstra’s Algorithm with Cross-Layer QoS*,” *ACTA TECHNICA NAPOCENSIS*, Electronics and Telecommunications, ISSN 1221-6542, Vol.51, No.3, 2010
- [Rus10c] A. B. Rus, M. Barabas, **G. Boanea**, V. Dobrota, „*Implementation of QoS-Aware Virtual Routers*,” International Symposium on Electronics and Telecommunications, ISETC 2010, Timisoara, Romania, ISBN: 978-1-4244-8460-7, pp. 161-164, November 2010

## Alte articole

- [Boa10c] **G. Boanea**, M. Barabas, A. B. Rus, V. Dobrota, „*Preliminary Implementation of a Situation Aware Multipath Routing Algorithm*,” Novice Insights in Electronics, Communications and Information Technology, Issue 9, ISSN 1842-6085, pp 58-63, 2010

## Rapoarte de cercetare

- [Boa09] **G. Boanea** „*Stadiul actual al rutării virtuale multicale*” Raport de cercetare 1, Universitatea Tehnică Cluj-Napoca, Decembrie 2009
- [Boa10a] **G. Boanea** „*Evaluarea performanțelor rutării virtuale multicale*” Raport de cercetare 2, Universitatea Tehnică Cluj-Napoca, Iulie 2010
- [Boa11a] **G. Boanea** „*Optimizarea rutării virtuale multicale*” Raport de cercetare 3, Universitatea Tehnică Cluj-Napoca, Martie 2011

## **Proiecte în care am fost implicată**

- FP7-ICT-2007-1 No. 216041 „4WARD – Architecture and Design for the Future Internet”, 2008-2010
- POSDRU/6/1.5/S/5 ID 7676 „PRODOC - Proiect de dezvoltare a studiilor de doctorat in tehnologii avansate”, 2008-2011



## Bibliografie selectivă

- [Alb94] R. Albrightson, J.J. Garcia-Luna-Aceves, J. Boyle, „*EIGRP - a fast routing protocol based on distance vectors*,” Proceedings of Networld/Interop 94, 1994
- [Ara09] P. Aranda, T. Biermann, D. Bursztynowski, s.a. „*Description of Generic Path Mechanism*,” FP7-ICT-2007-1-216041-4WARD/D-5.2.0, 2009.
- [Bar05] R. Barr, Z. J. Haas, R. van Renesse, „*JiST: an efficient approach to simulation using virtual machines*,” *Softw, Pract. Exper*, 35(6):539–576, 2005
- [Bar09] M. Barabas, **G. Boanea**, K. Steenhaut, V. Dobrota, „*Evaluating the Performances of the CastGate Tunnel Server over TCP and UDP Links in Multi-Client Configuration*,” *ACTA TECHNICA NAPOCENSIS*, Electronics and Telecommunications, ISSN 1221-6542, Vol.50, No.4, 2009
- [Bar11a] M. Barabas, **G. Boanea**, A. B. Rus, V. Dobrota, „*Routing Management Based on Statistical Cross-Layer QoS Information Regarding Link Status*,” 11th International Conference on Knowledge in Telecommunication Technologies and Optics KTTO 2011, pp. 12–17, ISBN 978-80-248-2399-7, Szczyrk, Poland, June 2011
- [Bar11b] M. Barabas, **G. Boanea**, A. B. Rus, V. Dobrota, J. Domingo-Pascual, „*Evaluation of Network Traffic Prediction Based on Neural Networks with Multi-task Learning and Multiresolution Decomposition*,” IEEE International Conference on Intelligent Computer Communication and Processing ICCP 2011, Cluj-Napoca, Romania, August 2011
- [Bar11c] M. Barabas, **G. Boanea**, V. Dobrota, „*Multipath Routing Management using Neural Networks-based Traffic Prediction*,” The Third International Conference on Emerging Network Intelligence, Lisbon, Portugal, November 2011 (accepted)
- [Bar11d] M. Barabas, „*Managementul rutării în viitorul Internet*” Teză de doctorat, Universitatea Tehnică Cluj-Napoca, Romania, 2011
- [Boa09] **G. Boanea** „*Stadiul actual al rutării virtuale multicale*” Raport de cercetare 1, Universitatea Tehnică Cluj Napoca, Romania Decembrie 2009
- [Boa10a] **G. Boanea** „*Evaluarea performanțelor rutării virtuale multicale*” Raport de cercetare 2, Universitatea Tehnică Cluj Napoca, Romania Iulie 2010
- [Boa10b] **G.Boanea**, M.Barabas, A.B.Rus, V.Dobrota, „*Design Principles and Practical Implementation of a Situation Aware Multipath Routing Algorithm*,” 17th Int.Conf. on Software, Telecommunications & Computer Networks IEEE SOFTCOM 2010, Split-Bol (Island of Brac), Croatia, Print ISBN: 978-1-4244-8663-2, INSPEC Accession Number: 11637618, pp.321-325, September 2010
- [Boa11a] **G. Boanea** „*Optimizarea rutării virtuale multicale*” Raport de cercetare 3, Universitatea Tehnică Cluj Napoca, Romania Martie 2011



- [Boa11b] G. Boanea, M. Barabas, A. B. Rus, V. Dobrota, J. Domingo-Pascual, „Performance Evaluation of a Situation Aware Multipath Routing Solution,” RoEduNet IEEE International Conference “Networking in Education and Research”, pp. 51–56, ISSN 2247-5443. Iași, Romania, June 2011
- [Boa11c] G. Boanea, M. Barabas, V. Dobrota, „An Overview of Today’s Multipath Routing,” *ACTA TECHNICA NAPOCENSIS*, Electronics and Telecommunications, ISSN 1221-6542, Vol.52, No.3, 2011 (submitted)
- [Bra94] R. Braden, D. Clark, S. Shenker, „Integrated Services in the Internet Architecture: an Overview”, RFC1633, June 1994
- [Cai10] L. Cai, J. Wang, C. Wang, L. Han, „A Novel Forwarding Algorithm over Multipath Network,” International Conference on Computer Design and Applications, pp. V5-353–V5-357. Qinhuangdao, China, 2010
- [Che01] Y. Chen, R. Hwang, Y. Lin, „Multipath QoS routing with bandwidth guarantee,” GLOBECOM '01, vol. 4, 2001
- [Che99] J. Chen, „New Approaches to Routing for Large-Scale Data Networks,” Houston, Texas, June, 1999
- [Cis09] Cisco Systems, Inc., „Cisco Active Network Abstraction 3.6.7 Technology Support and Information Model Reference Manual,” Chapter 4: Virtual Routing and Forwarding, 2009
- [Cis10] Cisco Systems, Inc., „Cisco Visual Networking Index: Forecast and Methodology, 2009–2014,”, June, 2010
- [Cor09] T. Cormen, C. Leiserson, R. Rivest, C. Stein, „Introduction to Algorithms,” (Third Edition), MIT Press/McGraw-Hil, ISBN-13: 978-0262033848, 2009
- [Cor11] L. M. Correia, H. Abramowicz, M. Johnsson & K. Wünnstel (editors), V. Dobrota, G. Boanea (inclusă în lista de autori) s.a., ”Chapter 12 - Prototypes”, *Architecture and Design for the Future Internet*, 4WARD Project, Series: Signals and Communication Technology, 1st Edition., 2011, XXIX, Springer, ISBN: 978-90-481-9345-5
- [Fal10] K. Fall, K. Varadhan, „The ns Manual,” The VINT Project, May, 2010, [http://www.isi.edu/nsnam/ns/doc/ns\\_doc.pdf](http://www.isi.edu/nsnam/ns/doc/ns_doc.pdf)
- [Flo03] G. Flores Lucio, M. Paredes-Farrera, E. Jammeh, M. Fleury, and M J Reed, „Opnet modeler and ns-2 - comparing the accuracy of network simulators for packet-level analysis using a network testbed,” WSEAS Transactions on Computers, 700—707, July 2003
- [Han09] L. Han, J. Wang, C. Wang, „A Concurrent Ant Colony Optimization Multipath Forwarding Algorithm in IP Networks,” Progress In Electromagnetics Research Symposium, Beijing, China, March 23–27, 2009
- [Hop00] C. Hopps, „Analysis of an Equal-Cost Multi-Path Algorithm,” RFC2992, November 2000

- [Jay09] G. Jayavelu, S. Ramasubramenian, O. Younis, „*Maintaining Colored Trees for Disjoint Multipath Routing Under Node Failures*,” IEEE/ACM Transactions on Networking, Volume 17, February 2009
- [Kau02] H. Kaur, S. Kalyanaram, „*A connectionless approach to intra- and inter-domain traffic engineering*,” New York Area Metro Area Networking Workshop, (Columbia University), September 2002
- [Li09a] R. Li, A. Eryilmaz, L. Ying, and N. B. Shroff, „*A unified approach to optimizing performance in networks serving heterogeneous flows*,” International Conference on Computer Communications, IEEE INFOCOM, pp. 253–261, 2009
- [Li09b] Z. Li, R. Wang, J. Bi, „*A Multipath Routing Algorithm Based on Traffic Prediction in Wireless Mesh Networks*,” International Conference on Natural Computation, Volum 6, pp. 115–119. Tianjin, China, 2009
- [Mer11] P. Mérindol, J. Pansiot, S. Cateloin, „*An efficient algorithm to enable path diversity in link state routing networks*,” Computer Networks: The International Journal of Computer and Telecommunications Networking, Volume 55, Issue 5, April 2011
- [Nar99] P. Narvaez, K. Siu, „*Efficient Algorithms for Multi-Path Link State Routing*,” ISCOM'99, Kaohsiung, Taiwan, 1999
- [Nic99] K. Nichols, V. Jacobson, L. Zhang, „*A Two-bit Differentiated Services Architecture for the Internet*,” RFC2638, 1999
- [Ns311] Ns-3 Project, „*Ns-3 Manual, Release ns-3.10*,” Ianuarie, 2011, <http://www.nsnam.org/docs/release/3.10/manual/ns-3.pdf>
- [OPN08] OPNET Technologies, Inc. Bethesda MD, „*OPNET Modeler Documentation Set*,” Versiunea 15.0, 2008
- [Pal01] H. Palakurthi, „*Study of Multipath Routing for QoS Provisioning*,” EECS 803 - Introduction to Research, October, 2001
- [Pol09] Z. Polgar, Z. Kiss, A. B. Rus, G. Boanea, M. Barabas, V. Dobrota, „*Preliminary Implementation of Point-to-Multi-Point Multicast Transmission Based on Cross-Layer QoS and Network Coding*,” 17th Int.Conf. on Software, Telecommunications & Computer Networks *IEEE SOFTCOM 2009*, Split-Hvar, Croatia, Print ISBN: 978-1-4244-4973-6, INSPEC Accession Number: 10951348, pp.131-135, September 2009
- [Ram07] S. Ramasubramanian, H. Krishnamoorthy, M. Krunz, „*Disjoint multipath routing using colored trees*,” Elsevier COMNET, vol. 51, no. 8, pp. 2163–2180, June 2007
- [Ric09] A. Riesco, A. Verdejo, „*Implementing and analyzing in Maude the Enhanced Interior Gateway Routing Protocol*,” Electr. Notes Theor. Comput. Sci. 238(3): 249-266, 2009

- [Rus10a] A. B. Rus, M. Barabas, **G. Boanea**, Z. Kiss, Z. Polgar, V. Dobrota, „*Cross-Layer QoS and Its Application in Congestion Control*,” IEEE Workshop on Local and Metropolitan Area Networks LANMAN 2010, Long Branch, NJ, USA, ISSN: 1944-0367, pp.1-6, May 2010
- [Rus10b] A. B. Rus, V. Dobrota, A. Vedinas, **G. Boanea**, M. Barabas, „*Modified Dijkstra’s Algorithm with Cross-Layer QoS*,” ACTA TECHNICA NAPOCENSIS, Electronics and Telecommunications, ISSN 1221-6542, Vol.51, No.3, 2010
- [Rus10c] A. B. Rus, M. Barabas, **G. Boanea**, V. Dobrota, „*Implementation of QoS-Aware Virtual Routers*,” International Symposium on Electronics and Telecommunications, ISETC 2010, Timisoara, Romania, ISBN: 978-1-4244-8460-7, pp. 161-164, November 2010
- [Rus11] A. B. Rus, „*Quality of services through cross- layer techniques for the future Internet*,” Teză de doctorat, Universitatea Tehnică Cluj-Napoca, 2011
- [Tse10] G. Tselentis, A. Galis, A. Gavras, S. Krco, V. Lotz, E. Simperl, B. Stiller, T. Zahariadis, „*Towards the Future Internet*,” IOS Press ISBN-13: 978-1607505389, May 2010
- [Var08] A. Varga, R. Hornig, „*An overview of the OMNeT++ simulation environment*,” Proceedings of the First International Conference on Simulation Tools and Techniques for Communications, Networks and Systems (SIMUTools 2008’), March, 2008
- [Vut00] S. Vutukury, J. Garcia-Luna-Aceves, „*MPATH: A Loop-free Multipath Routing Algorithm*,” Microprocessors and Microsystems, Volume 24, Number 6, October 2000
- [Xi07] K. Xi, H. Chao, „*ESCAP: Efficient scan for alternate paths to achieve ip fast rerouting*,” IEEE Globecom, 2007
- [Yan06] X. Yang, D. Wetherall, „*Source selectable path diversity via routing deflections*,” : ACM SIGCOMM, 2006
- [Zho00] Y. Zhong, X. Yuan, „*Impact of resource reservation on the distributed multi-path quality of service routing scheme*,” IWQOS. 2000