ROMÂNIA
MINISTERUL EDUCAȚIEI, CERCETĂRII, TINERETULUI ŞI SPORTULUI
UNIVERSITATEA DUNĂREA DE JOS DIN GALAȚI

Strada Domnească nr. 47, cod poştal 800008
Galaţi, România
E-mail: rectorat@ugal.ro

UNIVERSITAS
GALATIENSIS

Tel.: (+4) 0336-130.109; 0336-130.108; 336-130.104
Fax: (+4) 0236 - 461.353
www.ugal.ro

METODĂ DE SCHIMB DE INFORMAȚII PRIVIND ACTUALIZAREA RUTELOR

REZUMAT
TEZĂ DE DOCTORAT

ing. Cosmin Adomnicăi

Conducător ştiinţific: prof. dr. ing. Viorel Mînzu

GALAȚI, 2012

UNIVERSITAS GALATIENSIS

ROMÂNIA
UNIVERSITATEA „DUNĂREA DE JOS"
DIN GALAŢI

MINISTERUL EDUCAŢIEI CERCETĂRII TINERETULUI ŞI SPORTULUI

25366/01.11.2012

C ă t r e

_____

_____

Universitatea " Dunărea de Jos " din Galaţi vă face cunoscut că în data de ***09.11.2012***, ora____, în_____, va avea loc susţinerea publică a tezei de doctorat intitulată: ***"METODĂ DE SCHIMB DE INFORMAŢII PRIVIND ACTUALIZAREA RUTELOR"***, elaborată de domnul/doamna ***ADOMNICĂI COSMIN,*** în vederea conferirii titlului ştiinţific de doctor în Domeniul de doctorat - ***Ingineria sistemelor.***

Comisia de doctorat are următoarea componenţă :

1. Preşedinte:	***Conf.univ.dr.ing. Marian GĂICEANU***
	Universitatea "Dunărea de Jos" din Galaţi
2. Conducător de doctorat:	***Prof.univ.dr.ing. Viorel-Nicolae MÎNZU***
	Universitatea "Dunărea de Jos" din Galaţi
3. Referent oficial:	***Prof.univ.dr.ing. Sergiu ILIESCU***
	Universitatea POLITEHNICA din Bucureşti
4. Referent oficial:	***Prof.univ.dr.ing. Victor-Valeriu PATRICIU***
	Academia Tehnică Militară Bucureşti
5. Referent oficial:	***Prof.univ.dr.ing. Sergiu CARAMAN***
	Universitatea "Dunărea de Jos" din Galaţi

Cu această ocazie vă transmitem rezumatul tezei de doctorat şi vă invităm să participaţi la susţinerea publică. În cazul în care doriţi să faceţi eventuale aprecieri sau observaţii asupra conţinutului lucrării, vă rugăm să le transmiteţi în scris pe adresa Universităţii, str. Domnească nr. 47, 800008 - Galaţi, Fax - 0236 / 461353.

R E C T O R,
Prof.univ.dr.ing. Iulian Gabriel BÎRSAN

ROMÂNIA UNIVERSITATEA DUNĂREA DE JOS DIN GALAŢI RECTORAT

Cuprinsul rezumatului

Cuprinsul tezei

Introducere

În domeniul comunicaţiilor, tendința actuală este migrarea rețelelor de date către IP. Dezvoltarea rapidă a Internetului a dus la un număr din ce în ce mai mare de entități conectate în această rețea globală. Cu toate că, inițial, spațiul de adrese IP era suficient de mare dezvoltarea rapidă a dus la problema epuizării spațiului de adrese. Pentru aceasta a fost dezvoltat protocolul IPv6 care să suporte un număr foarte mare de adrese.

Dirijarea pachetelor este un proces esențial într-o rețea de date. Cu toate că tabelele de dirijare pot fi construite manual, într-o rețea de mari dimensiuni timpul consumat de administrator pentru actualizarea tabelelor de dirijare este foarte mare. De aceea au fost dezvoltate protocoale de dirijare, algoritmi care să construiască și să actualizeze tabelele în mod automat, în funcție de schimbările de structură ale rețelei. Principiile de dirijare existente acoperă o gamă largă de situații. În funcție de structura, tipul, tehnologia, destinația rețelei sunt folosiți anumiți algoritmi. În rețelele de date IP fixe, s-au consacrat două tipuri mari de protocoale: cele care folosesc vectori distanță și cele care folosesc starea legăturilor, însă există și variante hibride care combină cele două metode.

În situația în care apar pierderi de date mari pe legături, ruterele pierd adiacența ceea ce duce la întreruperea sau redirecționarea fluxului de date. Mai mult, dacă apar limitări de debit, în funcție de gradul de limitare, ruterele pot pierde adiacența întrerupând sau redirecționând fluxul de date cu toate că legăturile nu prezintă nicio defecțiune. De aceea, s-a constat nevoia dezvoltării unor metode de trimitere a actualizărilor care să fie rezistentă la pierderi discontinue de pachete de date și la condiții de trafic peste limita de debit alocată.

În acest context, prezenta lucrare își propune să dezvolte o metodă de transmitere a datelor necesare actualizărilor de rute pentru rețelele IPv6.

Pierderea adiacenței are ca efect eliminarea rutelor aferente din tabelele de dirijare, cu toate că legăturile de date nu sunt nefuncționale. Această situație caracterizează o stare de instabilitate a rețelei. De aceea, protocoalele de dirijare intră într-o stare de redobândire a stabilității prin încercarea de restabilire a adiacențelor între rutere. Practic se trece de la o stare în care s-a manifestat disfuncționalitatea rețelei, la o altă stare cu noi adiacențe, proces pe care îl putem caracteriza ca o redobândire a „stabilității”. Trecerea între cele două stări poate fi privită ca o convergență către o stare stabilă. Cu cât durata acestui proces este mai scurtă, cu atât putem vorbi de o convergență mai rapidă.

După prezentarea metodei de transmitere a informațiilor de actualizare propusă în cadrul lucrării, se analizează eficiența acesteia din punctul de vedere al dobândirii stabilității rețelei de comunicații. Această analiză a fost realizată prin simularea unei rețele de date, utilizand mașini virtuale, în condiții de limitare de debit și de trafic intens.

Capitolele 2 și 3 se constituie ca un “state of the art” în domeniul comunicațiilor de date. În **Capitolul 2** sunt prezentate aspecte fundamentale legate de modele logice de referință a transferului de date, protocoalele IP și IPv6, principii de dirijare a traficului de date și bazele dirijării IP.

Principiile de dirijare pentru rețelele IP fixe s-au concretizat în trei protocoale de dirijare consacrate de-a lungul timpului, protocoale prezentate în **Capitolul 3**. Pentru rețelele de mici dimensiuni se folosește protocolul RIP(Routing Information Protocol) și varianta sa RIPng (Next Generation - IPv6), un protocol simplu și rapid bazat pe vectori distanță. Rețelele medii și mari au necesitat un protocol mai complex care să ofere posibilitatea ierarhizării

procesului de dirijare. Acest protocol este OSPF(Open Shortest Path First), protocol standardizat de Internet Engineering Task Force și care folosește starea legăturilor. Un protocol proprietar care combină atât starea legăturilor cât și vectorii distanță este EIGRP (Enhanced Interior Gateway Routing Protocol). Acesta este un protocol rapid și robust, dar care este implementat doar pe echipamentele Cisco Systems.

Există situații în care anumite evenimente din rețea pot induce instabilitate în funcționarea protocoalelor de dirijare și, mai departe, în adăugiri/retrageri repetate de rute. Defectele care pot să apară la legăturile de date sau la echipamentele de rețea pot avea un comportament fluctuant și pot induce instabilitate. De asemenea, limitarea de debit poate avea un efect asemănător defectelor. Pentru a reduce impactul pe care îl au evenimentele din rețea, au fost propuse în secțiunea 3.2. mai multe îmbunătățiri care conduc la o redirijare rapidă a traficului în momentul apariției unor defecte.

Pentru a clarifica impactul pe care instabilitatea, definită ca pierdere a adiacenței, îl poate avea asupra protocoalelor de dirijare, a fost realizat studiul din **Capitolul 4**. Pentru aceasta au fost testate protocoalele RIP, OSPF și EIGRP în scenarii în care au fost introduse pierderi de pachete și limitări de debit în condiții de trafic puternic de date.

Pentru simularea funcționării protocoalelor de dirijare, a fost ales mediul OPNET Modeler. Acest mediu de modelare a rețelelor de date are in componență o suită de modele configurabile de echipamente de rețea cu ajutorul cărora se pot construi rețele complexe. Pentru simulările realizate în prezenta lucrare, a fost folosit modulul DES(Discrete Event Simulation) care permite o estimare apropiată de realitate a funcționării unei rețele la nivel de pachete. Astfel, fiecare pachet de date care apare în rețea este tratat ca într-o rețea reală. Pierderile de pachete și limitarea de debit induc o anumită instabilitate a rețelei. Rezultatele obținute prin simularile realizate au demonstrat că, în situații reale de exploatare ale unei rețele, instabilitatea indusă afectează într-o mare măsură traficul din rețea, ceea ce constituie o problemă importantă ce nu poate fi eludată.

Constatările prezentate mai sus au impus dezvoltarea unei metode prin care să se asigure robustețea procesului de creare și menținere a adiacenței și de transmitere a actualizărilor de rute. Metoda propusă și prezentată în **Capitolul 5** constă în esență în folosirea unui antet extins existent teoretic în structura IPv6, dar nefolosit uzual. În metoda propusă acest antet este completat cu date de dirijare ce vor fi folosite în procesul de adiacență și în distribuirea actualizărilor de informații de dirijare în rețea. În felul acesta se asigură rebustețea procesului de dirijare la pierderi de pachete sau limitări de debit. Datorită principiilor care stau la baza metodei, s-a impus și implementarea unui algoritm de dirijare, transformând, astfel, metoda într-un nou protocol de dirijare.

În lucrare este furnizat și un mod de verificare și analiză a acestei metode prin realizarea unui simulator bazat pe utilizarea unor mașini virtuale Linux. Fiecare mașină virtuală emulează funcționarea unui ruter din rețea. Ținând cont de modul de simulare, metoda propusă a fost implementată în limbajul C++ sub Linux. Pentru a dobândi prioritatea dorită, metoda a fost implementată ca modul de kernel în Linux, ceea ce a ridicat dificultatea implementării simulatorului.

Au fost realizate mai multe scenarii de simulare, cu complexitate din ce în ce mai ridicată, pentru a putea urmări și analiza evenimentele din rețea. Au fost calculați și analizați timpii de stabilizare ai rețelei printr-un procedeu particular descris în secțiunea 4.5.1.

Prin compararea timpilor de stabilizare ai metodei dezvoltate cu cei ai protocoalelor de dirijare consacrate, se constată un timp de stabilizare foarte bun, având în plus și avantajul robustetii.

Evident, se poate pune problema că prin introducerea de date suplimentare în antetul extins, crește gradul de ocupare al legăturii de date și, deci, se micșorează lărgimea de bandă disponibilă. Numai că metoda propusă are calitatea de a asigura o robustețe intrinsecă și micșorarea cu puțin a lărgimii de bandă este un preț mic de plătit pentru asigurarea stabilității rețelei.
În urma implementării și analizei calitative și cantitative, prin simulare, a metodei propuse de asigurare a stabilității unei rețele de comunicații, se poate afirma că metoda este viabilă și eficientă. Analiza poate fi extinsă, ca un subiect al unor cercetări viitoare, prin considerarea influenței altor parametri posibili asupra costului unei rute, cum ar fi: capacitatea legăturilor, latența pe legături, încărcarea legăturilor, stabilitatea legăturilor și stabilitatea vecinilor.

2. Principii ale rețelelor de calculatoare

2.3 Protocolul Internet versiunea 6

IPv6[1] a fost dezvoltat, în principal, pentru a rezolva problema epuizării adreselor IP clasice. Pe lângă aceasta, au mai existat și alte motive care au contribuit la dezvoltarea unei noi versiuni cum ar fi:

- Securitate mai buna;
- Migrarea unei gazde dintr-o rețea în alta fără a-și schimba adresa IPv6;
- Trimiterea multiplă prin specificare domeniului:
- Simplificarea antetului protocolului pentru a reduce timpii de procesare.

Față de IPv4 care are o dimensiune a adresei de 32 e biți, IPv6 are o dimensiune a adresei de 128 de biți, ceea ce permite un număr foarte mare de adrese posibile. Antetul IPv4 este destul de complex având 13 câmpuri. Pentru o procesare mai rapidă și întârzieri mai mici, antetul IPv6 are 7 câmpuri și o lungime fixă de 40 octeți.

O parte din câmpurile definite în IPv4 au devenit opționale în IPv6. Astfel opțiunile au devenit antete suplimentare care pot sau nu să existe. Pentru a scădea din timpul de prelucrare. În cazul în care ruterul este tranzitat de un pachet IPv6 ale cărui antete suplimentare nu îi sunt destinate poate să sară peste ele și să trimită pachetul mai departe.

Câmpul Versiune are valoarea 6 pentru IPv6, iar ruterele vor examina acest câmp pentru a diferenția și trata corespunzător pachetul de date.
Câmpul Tip de Trafic este folosit pentru a distinge între pachetele care au diverse cerințe de livrare în timp real.

Câmpul Eticheta Fluxului este definită în RFC 2460 ca o etichetă atribuită pachetelor care aparțin aceluiași flux logic. Astfel ruterele vor trata pachete unui flux particular în același fel. De exemplu, pachetele de voce vor avea aceeași etichetă de trafic pentru a fi tratate identic de ruterele pe care le traversează.

Câmpul Lungime informație utilă specifică numărul de octeți ocupați de informația care urmează după cei 40 octeți ai antetului pachetului IPv6.

Câmpul Antetul Următor are rolul de a specifica ce antet suplimentar, dacă există, urmează după antetul IPv6. În cazul în care nu există nici un antet suplimentar se specifică protocolul de nivel superior care este încapsulat în pachetul IPv6.

[1] Internet Protocol Versiunea 6

Câmpul Limita Salturilor are același rol ca și în cazul IPv4 și anume împiedică pachetele să fie dirijate prin rețea la infinit.

Ultimele câmpuri, și anume Adresă Sursă și Adresă Destinație conțin adresa IPv6 sursă a celui care trimite pachetul, respectiv adresa IPv6 a destinatarului.

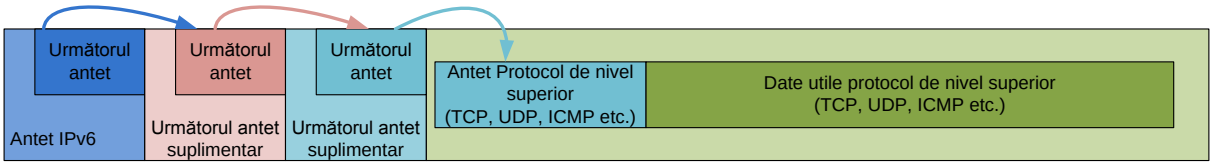

Fig. 2.9 Pachetul IPv6 cu două antete suplimentare

Opțional, în cazul în care este necesară transmiterea de informații suplimentare, se pot folosi antetele suplimentare. Au fost definite 6 antete suplimentare care pot fi folosite, și anume:

- Opțiuni salt – după – salt – »informații pentru rutere;
- Opțiuni pentru destinație –» informații suplimentare care să ajungă la destinație;
- Dirijare –» specifică calea parțială sau exactă de urmat;
- Fragmentare –» pentru tratarea fragmentelor;
- Autentificare –» pentru verificarea autenticității pachetului;
- Criptare –» pentru criptarea datelor utile din pachet.

Antetul suplimentar Opțiuni salt – după – salt este folosit pentru informații care trebuie examinate de fiecare ruter în parte din cale.

Câmpul Următorul Antet are rolul de a defini următorul antet suplimentar care urmează sau, dacă nu mai există, protocolul de nivel superior care este încapsulat în porțiunea de informație utilă.

Lungimea Antetului suplimentar reprezintă lungimea antetului propriu-zis, excluzând primii 8 octeți care sunt obligatorii.

Câmpul Tip opțiune are rolul de a spune ruterului ce să facă cu pachetul în cazul în care nu recunoaște opțiunea.

În acest antet suplimentar pot fi trimise mai multe opțiuni. Delimitarea opțiunilor se realizează cu ajutorul câmpului Lungime opțiune specific fiecărei opțiuni în parte și care reprezintă lungimea în octeți a unei opțiuni particulare.

Antetul suplimentar Opțiuni pentru destinație este identic cu antetul suplimentar Opțiuni salt – după – salt, cu diferența că acesta este examinat doar de către gazda destinație.

Antetul suplimentar Dirijare conține lista de rutere prin care pachetul trebuie să treacă până ajunge la destinație. Adresele din acest antet trebuie vizitate în ordine, însă se acceptă ca pachetul să treacă și prin alte rutere.

Ca și la antetele suplimentare descrise anterior, următorul antet definește următorul antet suplimentar care urmează sau, dacă nu mai există, protocolul de nivel superior care este încapsulat în porțiunea de informație utilă, iar lungime antetului reprezintă lungimea în octeți mai puțin 8 octeți care sunt obligatorii.

Tipul Rutării precizează formatul părții rămase din antet. Deocamdată este definit doar tipul 0 care arată că după primii 32 de biți urmează 32 de biți rezervați și apoi adresele IPv6 ale ruterelor pe care pachetul trebuie să le parcurgă.

Câmpul Segmente Rămase specifică numărul de adrese din listă care nu au fost parcurse de pachet. Acest câmp este decrementat la fiecare trecere printr-un ruter a cărei adresă este specificată în antetul suplimentar.

IPv6 tratează fragmentarea puțin diferit în sensul că nu mai are câmpuri alocate în antetul pachetului ci folosește un antet suplimentar în acest sens. În rest, mecanismul de tratare a fragmentării este asemănător cu cel din IPv4.

Câmpul Antetul următor are același rol ca și la celelalte antete suplimentare. După acesta urmează 8 octeți rezervați și câmpul Deplasamentul fragmentului care indică locul fragmentului în pachetul IPv6 inițial. Următorii 2 octeți sunt rezervați, iar câmpul Mai Multe fragmente specifică dacă mai există fragmentul după cel curent care să aparțină aceluiași pachet IPv6 inițial.

Câmpul Identificare are aceeași valoare pentru toate fragmentele și este folosit la reconstrucția pachetului inițial.

Pachetul IPv6 încapsulează protocoalele de nivel superior cum ar fi: TCP, UDP, ICMPv6[1] (Protocolul pentru mesajele de control în Internet versiunea 6) etc.

Față de IPv4, la IPv6 fragmentarea se poate realiza doar la sursă ceea ce are avantajul unor timpi mai mici de întârziere în prelucrarea pachetelor de către rutere.

3. Protocoale de dirijare şi redirijarea rapidă

3.1 Protocoale și algoritmi de dirijare în rețelele IP fixe

În funcție de modul de luare a deciziilor de dirijare, algoritmii de dirijare pot fi împărțiți în două mari clase, și anume:

- Algoritmi neadaptivi – nu își adaptează deciziile de dirijare în funcție de structura rețelei și estimările curente (dirijarea statică);
- Algoritmi adaptivi – modifică condițiile de dirijare în funcție de structura curentă a rețelei, de modificările de trafic etc.

Cele mai populare protocoale de dirijare de interior au la bază fie dirijarea folosind vectorii distanţă, fie dirijarea folosind starea legăturilor, fie o combinație între acestea. În practică s-au evidenţiat 4 protocoale: protocolul RIP[2] ce foloseşte vectorii distanţă, protocoalele OSPF[3] şi IS-IS[4] ce folosesc starea legăturilor şi protocolul proprietar EIGRP[5] [ACEV94] care este definit ca un protocol hibrid. În rețele de date IP mobile algoritmii de dirijare pot fi împărțiți astfel: algoritmi proactivi, algoritmi reactivi.

3.1.1 Protocolul de dirijare RIPng

RIPng reprezintă evoluția protocolului de dirijare RIP în contextul rețelelor IPv6. Ca și predecesorul său, RIPng este un protocol de dirijare simplu și rapid din clasa protocoalelor vector distanță.

Astfel, în stabilirea costului rutei, se folosește numărul de noduri de rețea(rutere) pe care pachetul trebuie să îl traverseze până la destinație. Pentru a fi evitată buclarea pachetului în rețea în cazul unei situații de instabilitate, costul maxim este limitat la 15. De

[1] Internet Control Message Protocol definit în [RFC 2463]
[2] Routing Information Protocol
[3] Open Shortest Path First
[4] Intermediate System To Intermediate System
[5] Enhanced Interior Gateway Routing Protocol

aceea, RIPNG este adaptat pentru rețele de mici dimensiuni în care distanţa maximă între două rutere nu depășește valoarea 15. Pentru a defini o destinație la care nu se mai poate ajunge se folosește valoarea 16.

Periodic, fiecare ruter distribuie informații legate de rutele sale către vecinii săi, folosind mesaje specifice de actualizare. În momentul în care se primește o actualizare de la un ruter vecin, la fiecare rută din actualizare se adaugă valoare 1 la cost. Dacă noul cost obținut are valoarea 16, ruta respectivă nu mai este adăugată în tabela internă de dirijare.

$$cost_{ruta\ j} = \sum_{\substack{i=1 \\ n<16}}^{n} ruter_i, ruter_i \in ruta_j \qquad (3.1)$$

Pentru procesarea unei noi rute, RIPNG folosește algoritmul Bellman-Ford.

3.1.2 Protocolul de dirijare OSPFv3

OSPFv3, definit în standardul RFC 5340, reprezintă evoluţia protocolului de dirijare OSPFv2, folosit în reţelele IPv4, pentru mediul IPv6. Schimbările care au apărut nu sunt de esenţă, ci, în principal, pentru a face loc mărimii extinse a adreselor IPv6.

OSPF a fost dezvoltat pentru a elimina limitările din protocolul RIP, cum ar fi: diametrul redus, timpul mare de convergenţă şi definirea costului care nu ia în calcul caracteristicile reţelei. Astfel, costul unei legături este definit astfel:

$$cost\ legătură = \frac{10^9}{viteză\ legatura\ [kb/s]} \qquad (3.2)$$

În plus, OSPF poate gestiona eficient o tabelă de dirijare de dimensiuni mai mari faţă de RIP.

OSPF funcționează prin transpunerea rețelei într-un graf orientat în care fiecărui arc îi este atribuit un cost. O conexiune reală între două rutere este reprezentată prin două arce cu sensuri opuse, câte un sens pentru fiecare direcție. Ruta, sau cea mai scurtă cale de la un nod la altul, se construiește luând în considerare costurile arcelor.

Un sistem autonom poate fi organizat în zone numerotate și disjuncte. Pentru ca OSPF să funcționeze trebuie definită cel puțin o zonă, și anume zona 0 denumită și zona de bază. În cazul în care sunt definite mai multe zone, toate zonele trebuie să fie conectate la zona 0 în așa fel încât se existe o cale din orice zonă către orice zonă prin intermediul zonei 0. Astfel, fiecare zonă trebuie să aibă un ruter conectat la zona 0. Acest mod de organizare forțează ca zonele să fie organizate într-o structură de tip stea cu zona 0 în centru.

Toate ruterele dintr-o zonă au aceeași bază de date cu starea legăturilor și folosesc același algoritm pentru calculul celei mai scurte căi. Ruterul care se conectează zona curentă la zona 0 trebuie să aibă bazele de date cu starea legăturilor din ambele zone și trebuie să folosească algoritmul de calcul a căii cele mai scurte separat pentru fiecare zonă.

Fiecare ruter gestionează o bază de date a stărilor legăturilor dintr-o zonă. Această bază de date se construieşte prin schimbul de pachete LSA[1] între rutere. În funcţie de conţinut, LSA poate să fie distribuit către ruterele din aceeaşi zonă, acelaşi sistem autonom sau doar către vecini. Relaţia stabilită între două rutere se numeşte adiacenţă, iar OSPF depinde destul de mult de o adiacenţă stabilă.

În cazul legăturilor punct – la – punct, ruterele vecine formează adiacență. Crearea inițiala a adiacenței și menținerea acesteia se realizează prin trimiterea de pachete HELLO. Într-un LAN în care există mai mult de un ruter, fiecare ruter este vecin cu fiecare. Fiind

[1] Link State Advertisments

ineficient ca toate ruterele să discute între ele, într-un LAN se alege un ruter desemnat care formează adiacențe cu toate celelalte rutere, iar discuțiile între celelalte rutere se realizează prin intermediul ruterului desemnat. Mai mult, pentru a se alege un nou ruter desemnat în situația în care acesta se defectează, se alege și un ruter desemnat de rezervă care devine ruter desemnat imediat ce acesta se defectează, urmând ca mai apoi sa se aleagă un alt ruter desemnat de rezervă.

OSPF poate introduce în tabela de dirijare mai multe căi către aceeaşi destinaţie cu cost identic pe baza cărora traficul poate fi distribuit între acestea.

3.1.3 Protocolul de dirijare EIGRP

Enhanced Interior Gateway Routing Protocol este un protocol de dirijare proprietar dezvoltat de Cisco care reprezintă a îmbunătățire a protocolului inițial IGRP. IGRP a fost dezvoltat pentru a elimina limitările protocolului RIP.

EIGRP nu se bazează pe actualizări periodice complete pentru a reobține convergența, ci creează o tabelă cu structura rețelei. Pentru aceasta se folosesc anunțurile vecinilor și realizează convergența fie prin căutarea unei rute care să nu aibă bucle, fie, dacă nu se găsește nicio rută fără bucle, prin interogarea vecinilor.

Pentru a distribui informațiile de dirijare în rețea, EIGRP folosește actualizări incrementale pe care nu le distribuie periodic. Pentru a propaga informațiile de dirijare în rețea, EIGRP se bazează pe relația de vecinătate între două rutere. Astfel dacă două rutere primesc unul de la celălalt pachete de tip Hello atunci formează relație de vecinătate.

Fața de alte protocoale de dirijare, EIGRP folosește o funcție de cost complexă care ia în calcul mai mulți parametri:

$$cost = \left[(K1 * viteză) + \frac{K2 * viteză}{256 - incarcare} + K3 * intarziere\right] \cdot \frac{K5}{stabilitate + K4} \qquad (3.3)$$

3.2 Redirijarea rapidă IP

Redirijarea rapidă IP constituie o preocupare de noutate prin care se dorește o recuperare cât mai rapidă după apariția unui defect. Prin aceasta se dorește ca volumul de trafic pierdut în urma tranziției de la o cale la alta să fie cât mai redus.

Cercetările recente au dus la mai multe propuneri care folosesc diferite mecanisme prin care recuperarea după apariția unui defect să fie cât mai rapidă IP. Mecanismele, descrise și testate în [GJOK07], sunt următoarele:

- Mecanismul ce folosește căi multiple de cost egal – de obicei implementate în protocoalele de dirijare;
- Mecanismul ce folosește vecinii alternativi [RFC 5286];
- Mecanismul în „U” [ATLA06];
- Mecanismul ce folosește tunele [BRYA07].

4. Stabilitatea în rețelele de date IP

Convergența reprezintă procesul de instalare a rutelor pe rutere. Propagarea informațiilor de dirijare prin rețea se realizează în timp și, de aceea, convergența nu se realizează instantaneu. Convergența poate fi caracterizată prin perioada de timp dintre apariția unui eveniment și momentul în care ultimul ruter își va actualiza tabela de dirijare în mod corespunzător [TEIX06]. Convergența mai are și un alt aspect mai ascuns care ține de implementarea software și hardware a ruterului. Între momentul în care un ruter primește o actualizare și recalculează noile rute și momentul în care se actualizează tabela de dirijare a

ruterului există o diferență de timp. Această diferență de timp poate fi considerată durata procesului de convergență a planului tabelei de dirijare. În funcție de situație, durata de timp necesară poată să fie mai mare sau mai mică. Dacă schimbările de rute sunt reduse, atunci procesul de actualizare a tabelei de dirijare este redus, chiar insesizabil. În situația în care există schimbări majore de rute acest proces poate să fie mai îndelungat și poate duce la pierderi și întârzieri de pachete. O rețea care nu a ajuns la convergență poate avea ca efecte pierderi de pachete sau livrarea pachetelor în altă ordine [LABO08]. Bineînțeles că procesul de convergență a planului tabelelor de dirijare depinde foarte mult de ruter, mai precis de cum a realizat producătorul implementarea software și hardware a rutării și a protocoalelor de dirijare.

În mod ideal, toate informațiile relevante vor fi distribuite către toate ruterele și sistemul va intra într-o stare stabilă. Problemele de instabilitate care apar sunt, în general, cauzate de buclele de dirijare. În cazul în care legăturile dintre rutere sau chiar ruterele sunt instabile în funcționare există posibilitatea ca procesul de convergență să nu se încheie de loc sau ca, după atingerea stabilității, să apară bucle de dirijare care vor introduce sistemul într-o stare de instabilitate (Fig. 4.1).

Deoarece RIP este un protocol adaptat pentru rețele de mici dimensiuni, iar EIGRP este un protocol proprietar Cisco, în rețelele de mari dimensiuni se folosesc protocoale de dirijare bazate pe starea legăturilor. Astfel, mai departe în acest subcapitol, vom aborda această clasă de protocoale de dirijare.

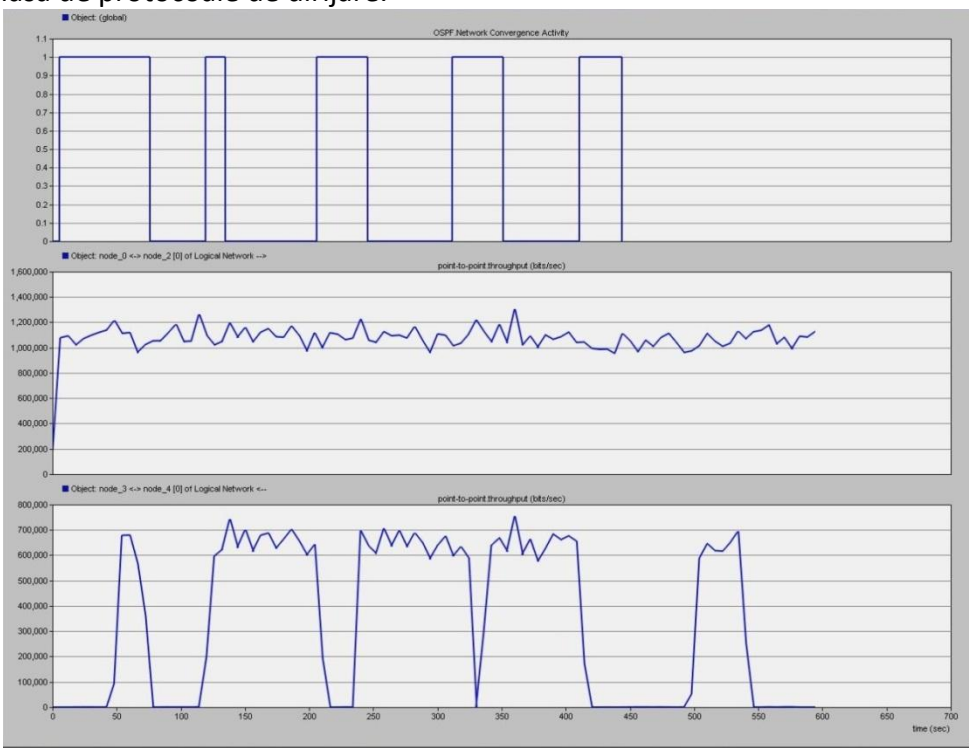

Fig. 4.1 Pierderi de pachete datorate stărilor de instabilitate

Cu toate că protocoalele de dirijare bazate pe starea legăturilor au fost dezvoltate și folosite în rețele IP de mici dimensiuni, în zilele noastre au ajuns să fie folosite și în rețele mari ale companiilor ce oferă servicii de transport de date (Internet, Intranet etc.). Contractul de oferire de servicii de date, în majoritatea cazurilor, specifică un timp de întrerupere sub 0.01%; cu alte cuvinte, serviciul oferit trebuie să fie disponibil cel puțin 99,99% din timp. De aceea, restabilirea convergenței este foarte importantă pentru a nu

afecta disponibilitatea rețelei. Până la urmă convergența se va restabili, dar important este ca acest proces să se realizeze cât mai repede pentru a diminua timpul de nefuncționare.

Cea mai comună cauză a schimbării structurii unei rețele este reprezentată de întreruperea unei legături [MARK04]. Analizele realizate de un mare furnizor de servicii de date au concluzionat că 45% din evenimentele ce duc la instabilitate apar în timpul orelor de mentenanță [IANN04]. Într-un alt studiu se menționează că 20% din evenimentele ce duc la nefuncționarea unei sau mai multe componente ale rețelei [MARK04] sunt datorate mentenanței și, în acest caz, nu ar trebui să apară bucle de dirijare tranziente [DUBO04]. Din restul de 80% de evenimente, 30% sunt datorate unei cauze comune cum ari fi nefuncționarea unor legături datorită defectării unui ruter. Restul de 70% sunt probleme de nefuncționarea a unei legături din cauze diverse [MARK04].

O altă cauză comună care duce la pierderea convergenței este costul asociat interfețelor ruterului. De obicei, costul este modificat manual de către administratorii de rețea pentru a crea rute ocolitoare în cazul unui trafic crescut [TEIX06], deși, în ingineria traficului, au fost propuși algoritmi prin care să fie ajustate automat costurile interfețelor [FORTZ06]. Cu toate acestea furnizorii de servicii de date preferă să ajusteze costurile manual. Mai mult, în practică optimizările la nivelul întregii rețele sunt rare [AGAR05]. Dirijarea optimă este dificil de atins. Natura dinamică a traficului în rețea agravează această problemă [LI05].

În momentul în care o legătură de date devine indisponibilă, strategia optimă de dirijare se poate recalcula integral, rezultând astfel un raport optim al performanței cu costul unor posibile schimbări majore în tiparele de trafic [APPL03]. Mai mult, prin monitorizarea funcționării protocolului OSPF într-o rețea reală, legăturile instabile au fost o sursă predominantă de instabilitate [WATS03].

În afară de problemele generate de legături, ruterele pot să aibă atât erori în funcționare cât și să se defecteze. De obicei erorile de funcționare sunt: supraîncărcare procesorului ruterului, erorile de funcționare datorate diverselor probleme software, erori de implementare și configurare a protocolului de dirijare.

Un alt factor care poate induce stări de instabilitate este derivat din procesul de injecție a rutelor dinspre rețelele utilizatorilor finali. În [WATS03] a fost identificat faptul că instabilitatea din rețelele clienților induce instabilitate în rețeaua nucleu a furnizorului de servicii de date. Mai mult, pentru OSPF, pachetele LSA care indică schimbări de structură au drept cauză schimbările de stare accesibilitate a rețelelor clienților. În plus, clienții sunt conectați folosind linii închiriate a căror accesibilitate este mai volatilă [SHAI02].

Disponibilitatea și stabilitatea sunt elemente importante ale unui ruter și a unui protocol de dirijare.

Stabilitatea unei rute se ia în calcul deoarece, în procesul de tranziție dintre două căi, există un timp în care niciuna din rute nu este disponibilă. De obicei, tranziția se realizează extrem de repede fără a genera pierderi de pachete. Capacitatea de supraviețuire a unei rețele se referă la abilitatea ca o rețea să poată oferi servicii de date indiferent de scara, magnitudinea, durata și tipul de defecte [IANN04].

4.2 Aspecte privind simularea comportamentului protocoalelor de dirijare

Studiul stabilității protocoalelor de dirijare a necesitat identificarea condițiilor care induc instabilitate. Atât în literatura de specialitate cât și în situațiile practice, au fost

evidențiați mai mulți factori cum ar fi defectele din rețea, configurări eronate, rețelele clienților care injectează rute false în rețea etc.

Defectele apărute în infrastructura unei rețele, în funcție de natura acestora, pot duce la perioade de instabilitate, mai lungi sau mai scurte, a tabelelor de dirijare. Cele mai proeminente defecte care induc instabilitate sunt cele care apar pe legăturile de date. De cele mai multe ori, problemele ce apar pe legături, au ca efect pierderea aleatoare de pachete de date. O altă sursă de instabilitate, care nu derivă dintr-un defect, este limitarea de debit.

Pentru simularea funcționării protocoalelor de dirijare a fost ales mediul OPNET Modeler. Acest mediu de modelare a rețelelor de date are in componență o suită de modele configurabile de echipamente de rețea cu ajutorul cărora se pot construi rețele complexe. Pentru simulările realizate a fost folosit modulul DES[1] care permite o estimare apropiată de realitate a funcționării unei rețele la nivel de pachete. Astfel, fiecare pachet de date care apare în rețea este tratat ca într-o rețea reală.

Rețelele de test implementate au fost configurate cât mai apropiat de situații reale pentru a observa comportamentul protocoalelor de dirijare. Pentru acesta, în rețea, au fost introduse stații de lucru care să genereze un anumit volum de trafic. Traficul de date astfel generat are ca destinație tot o stație generatoare de trafic și nu un comutator sau un ruter. Dimensiunea pachetelor se definește folosind anumite distribuții. În simulările realizate a fost folosită o distribuție constantă.

Din punctul de vedere al adreselor IP, OPNET oferă o funcție de generare și atribuire automată de adrese. Această funcție este utilă deoarece elimină timpul necesar atribuirii și configurării adreselor de subrețea și adreselor interfețelor, mai ales în situațiile în rețea sunt introduse sau eliminate des componente.

Ca și pentru adresele IP, OPNET oferă o funcție de configurare automată a protocoalelor de dirijare pe rutere. Prin alegerea protocolului de dirijare sunt configurate interfețe de rețea și activate valorile implicite pentru protocolul ales. Bineînțeles, parametrii specifici pot fi modificați în funcție de necesitățile simulării.

În funcție de ce se dorește a se simula, OPNET conține o multitudine de modele de rutere comutatoare, legături de date și alte echipamente de rețea.

După construirea structurii rețelei și configurării diferiților parametri trebuie alese statisticile care vor fi colectate. Mediul are o paletă largă de date ce pot fi culese. În simulările realizate au fost de interes următorii parametri:

- Activitatea de convergență a rețelei;
- Dimensiunea tabelelor de dirijare de pe rutere;
- Numărul de rute adăugate/retrase la fiecare schimbare de structură a rețelei;
- Volumul de trafic pe legăturile de date, pe fiecare sens, în biți/s.

Mediul de simulare are în componență un generator de numere preudoaleatoare care să introducă un anumit grad de aleatorism în simulările realizate. Setul de valori generate poate fi alterat prin modificare valorii de inițializare a generatorului. Astfel pentru aceeași rețea se pot obține valori diferite de trafic și un comportament diferit al protocoalelor de dirijare.

Odată cu pornirea simulării pachete IP vor fi generate, comutate și dirijare în rețea până la destinație. În acest timp, sunt culese datele care au fost definite anterior.

[1] Discrete Event Simulation

La încheierea simulării, datele pot fi afișate sub formă de grafice. Pentru anumite simulări au fost introduse pe același grafic mai multe seturi de date culese. Mai departe datele din aceste grafice pot fi exportate în Microsoft Excel pentru prelucrare suplimentară.

În esență au fost realizate două tipuri de simulări:

- Simulări pentru evidențierea comportamentului protocoalelor de dirijare în situațiile în care există pierderi de date;
- Simulări pentru evidențierea comportamentului protocoalelor de dirijare în situațiile în care există limitare de debit pe legăturile de date.

Pentru a simula comportamentul protocoalelor de dirijare în condițiile în care există pierderi de date, între două rutere a fost introdus un Nor(Cloud) în care a fost stabilit procentul de pierderi. Această structură a fost aleasă deoarece în OPNET nu se poate introduce un anumit grad de pierderi pe o legătura Ethernet.

Comportamentul protocoalelor de dirijare în condițiile existenței de limitare de debit a fost simulat în două maniere:

- Introducerea a unuia sau mai multe comutatoare între rutere; între comutatoare a fost folosită o legătura Ethernet de 10Mbps. Toate celelalte legături din simulare au capacitate de 100Mbps.
- Folosirea între rutere a legături seriale de capacitate limitată, stațiile generatoare folosind legături de capacitate 100Mbps.

Pentru evidențierea grafică a unor date a fost folosită posibilitatea de export a datelor în Microsoft Excel în care s-au realizat prelucrări suplimentare.

4.3 Comportamentul protocoalelor de dirijare în condiții de pierderi de date

4.3.1 Protocolul RIP

Tabelul 4.1 Tabel centralizator cu intervalele de instabilitate pentru protocolul RIP

Protocol de dirijare	Perioadele de timp în care au existat pierderi de date pentru pierderi de				
	0%	10%	20%	30%	40%
RIP	---	---	---	---	---

Tabelul 4.2 Tabel centralizator cu timpul total de instabilitate şi procentul din durata de simulare în care reţeaua a fost instabilă pentru RIP

Protocol de dirijare	Procent pierderi	Timpul total în care au existat pierderi de date	Procent din timpul total în care au existat pierderi de date
RIP	10%	---	---
	20%	---	---
	30%	---	---
	40%	---	---

Deși sunt 10% pierderi de pachete, odată ce ruterele au încheiat procesul de convergență, rețeaua rămâne stabilă, nemaiexistând retrageri/adăugări de rute în tabelele de dirijare.

Dacă procentul de pierderi este crescut la 20%, ca și în cazul anterior, timpul de convergență este mai mare decât în mod normal, însă, odată încheiat procesul de convergență, nu mai au loc schimbări în structura tabelelor de dirijare.

Aceeași situație se regăsește și în cazul în care procentul de pierderi este crescut la 30% cu diferența că momentul în care ruterele converg este mai depărtat în timp, iar, în jur de indexul de timp 160 de secunde, ruterele nu se mai află în convergență, începând din nou procesul de convergență. În această situație, totuși, nu se pierd pachete de date din cauza ruterelor.

Chiar dacă procentul de pierderi este crescut la 40% nu mai apar pierderi de pachete datorită ruterelor după ce acestea au încheiat procesul inițial de convergență. În cazul protocolului RIP, cu cât procentul de pierderi crește, cu atât durata de timp în care ruterele ajung la convergență crește.

4.3.2 Protocolul OSPF

Dacă gradul de pierderi este crescut la 10 % se constată atât un timp mai mare de convergență cât şi pierderi de pachete între momentele de timp 888÷924 secunde. Astfel datorită pierderilor de pachete OSPF devine instabil pentru 35 secunde.

În momentul în care se ajunge la 20% pierderi, pachetele specifice OSPF eliminate din trafic dau instabilitate în funcționare protocolului de dirijare întrerupând traficul de date în trei intervale de timp diferite, totalizând 105 secunde.

Creșterea pierderilor la 30% duce la un număr de cinci perioade de instabilitate. Datorită acestora timp de 163 secunde nu se dirijează trafic datorită pierderii convergenței. Creșterea gradului de pierderi la 40% duce la o instabilitate și mai crescută pentru protocolul OSPF deoarece apar unsprezece perioade de timp în care nu se mai transmit pachete de date datorită eliminării rutelor din tabela de dirijare. Aceste perioade de timp totalizează 911 secunde de întrerupere a traficului.

Tabelul 4.7 Tabel centralizator cu intervalele de instabilitate pentru protocolul OSPF

Protocol de dirijare	Perioadele de timp în care au existat pierderi de date pentru pierderi de:				
	0%	10%	20%	30%	40%
OSPF	---	888÷924	204÷240	60÷84	60÷94
	---	---	336÷372	504÷540	108÷264
	---	---	768÷804	696÷744	264÷300
	---	---	---	804÷840	336÷540
	---	---	---	984÷1008	552÷600
	---	---	---	---	612÷660
	---	---	---	---	672÷708
	---	---	---	---	744÷948
	---	---	---	---	1008÷1032
	---	---	---	---	1044÷1140
	---	---	---	---	1164÷1200

Tabelul 4.8 Tabel centralizator cu timpul total de instabilitate şi procentul din durata de simulare în care reţeaua a fost instabilă pentru OSPF

Protocol de dirijare	Procent pierderi	Numărul de stări de instabilitate	Timpul total în care au existat pierderi de date	Procent din timpul total în care au existat pierderi de date
OSPF	10%	1	35	2,91%
	20%	3	105	8,75%
	30%	5	163	13,58%
	40%	11	911	75,91%

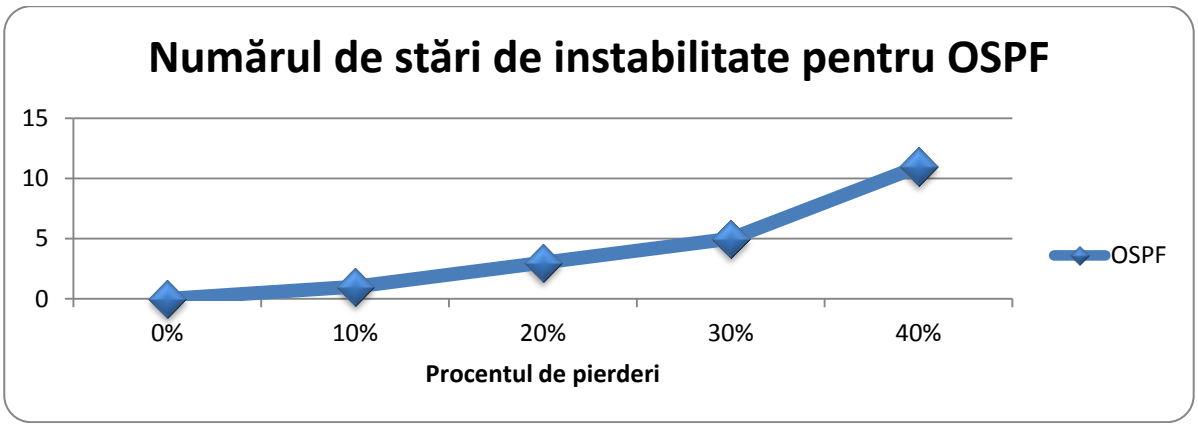

Fig. 4.13 Numărul de stări de instabilitate pentru OSPF în funcţie de simulare

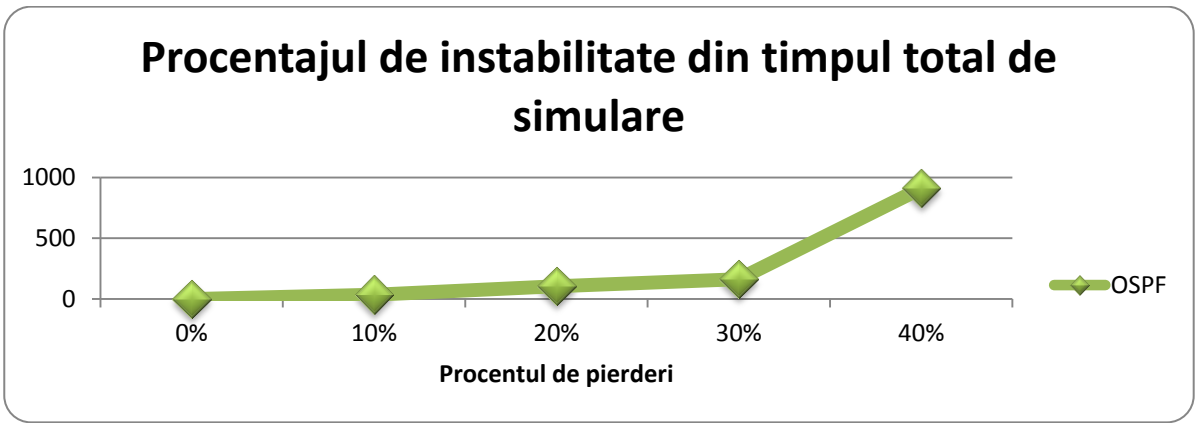

Fig. 4.14 Procentajul de instabilitate din timpul total de simulare pentru OSPF în funcţie de simulare

Analizând datele obținute în urma simulărilor (Fig. 4.13, Fig. 4.14), odată cu creștere gradului de pierderi de pachete de date, se observă o creștere foarte abruptă a timpului total de instabilitate (Fig. 4.14). Dacă pentru pierderi între 10% și 30% durata totală de instabilitate este sub 15%, la o valoare a pierderilor de 40% durata totală de instabilitate ajunge la 75,91%.

Numărul de stări de instabilitate pentru pierderi de sub 30% este maxim 5. Odată cu creșterea pierderilor la 40%, numărul de stări de instabilitate crește foarte mult atingând valoarea de 11.

Dacă pentru pierderi sub 30% comunicația se mai poate realiza în condiții acceptabile, la 40% pierderi de date instabilitatea din rețea face dificilă menținerea fluxului de date.

Pentru pierderi de 20% - 40% durata inițiale de stabilizare este mai mare decât pentru situațiile cu 0%-10% pierderi de pachete.

4.3.3 Protocolul EIGRP

Prin creșterea procentului de pierderi de pachete de date la 10%, funcționarea protocolului de dirijare EIGRP este puternic afectată prin apariția a două perioade lungi de timp în care convergența este pierdută. Durata totală de instabilitate ajunge la 357 secunde.

La 30% pierderi de date, apar 3 stări de instabilitate din care doar una de lungă durată. În total, timp de 358 secunde, traficul nu este dirijat. Trebuie precizat că simularea se încheie pe durata ultimei perioade de instabilitate. Astfel nu se poate aprecia durata acesteia și este luată în considerare doar durata pe perioada simulării.

Creșterea pierderilor la nivelul de 30% aduce schimbări majore în sensul că, după activitatea inițială de convergență, apar 8 perioade de instabilitate ce totalizează 837 de secunde. Creșterea procentului de pierderi la 40% duce la o scădere a numărului de stări de instabilitate, dar durata totală a acestora rămâne în continuare mare. Astfel, în acest caz avem 5 stări de instabilitate ce totalizează 842 secunde.

Pentru situațiile 0%-20% pierderi stabilizarea inițială se realizează foarte repede. În schimb, pentru 30% și 40% pierderi stabilizarea inițială se realizează mai lent.

Tabelul 4.13 Tabel centralizator cu intervalele de instabilitate pentru protocolul EIGRP

Protocol de dirijare	Perioadele de timp în care au existat pierderi de date pentru pierderi de:				
	0%	10%	20%	30%	40%
EIGRP	---	720÷1044	108÷396	5÷165	5÷84
	---	1165÷1200	1104÷1140	240÷300	132÷324
	---	---	1164÷1200	372÷456	348÷564
	---	---	---	516÷660	660÷768
	---	---	---	696÷768	864÷1116
	---	---	---	792÷1032	---
	---	---	---	1068÷1116	---
	---	---	---	1164÷1200	---

Tabelul 4.14 Tabel centralizator cu timpul total de instabilitate şi procentul din durata de simulare în care reţeaua a fost instabilă pentru EIGRP

Protocol de dirijare	Procent pierderi	Numărul de stări de instabilitate	Timpul total în care au existat pierderi de date	Procent din timpul total în care au existat pierderi de date
EIGRP	10%	2	357	29,75%
	20%	3	358	29,84%
	30%	8	837	69,75%
	40%	5	842	70,17%

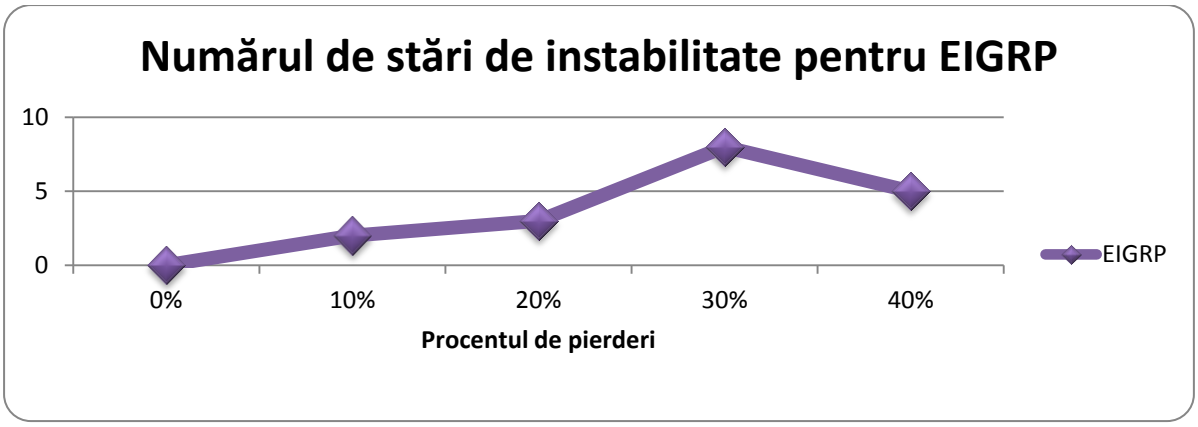

Fig. 4.20 Numărul de stări de instabilitate pentru EIGRP în funcţie de simulare

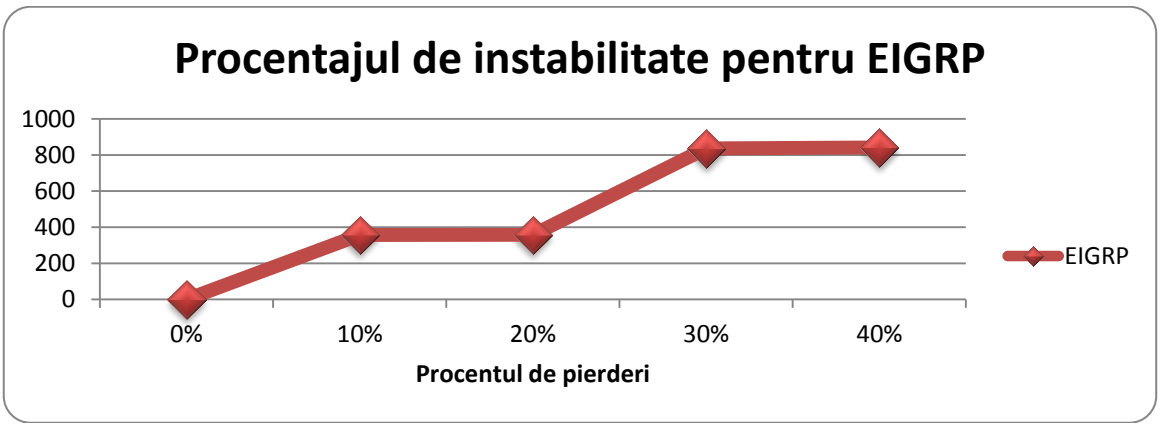

Fig. 4.21 Procentajul de instabilitate din timpul total de simulare pentru EIGRP în funcţie de simulare

Pentru EIGRP se remarcă o sensibilitate crescută la cantitatea de pierderi. Doar 10% pierderi pe legătura de date dintre rutere induce 29,75% durată de instabilitate din timpul total de simulare. Și la 20% pierderi, durata totala de pierderi este aproximativă cu simularea cu 10% pierderi.

Pentru 30% și 40% pierderi durata totală de instabilitate reprezintă aproximativ 70% din timpul de simulare. În aceste situații, este foarte dificilă folosirea serviciilor rețelei din punctul de vedere al utilizatorilor.

4.4 Comportamentul protocoalelor de dirijare în condiţii de limitare de debit

4.4.1 Protocolul RIP

Analizând comportamentul protocolului RIP în cele două situaţii (o singură legătură între rutere şi două legături între rutere) observăm că nu există pierderi de stabilitate. Cu toate că în procesul de limitare de debit sunt eliminate şi pachete specifice protocoalelor de dirijare, protocolul RIP nu elimină rute din tabela de dirijare internă. Acest comportament se datorează faptului că acesta implementează temporizatori de timp destul de mari.

Implicit, protocolul RIP trimite odată la 30 secunde o actualizare periodică. Până la eliminarea unei rute din tabela de dirijare, implicit, protocolul RIP aşteaptă 240 secunde. În acest timp este foarte puţin probabil ca să nu apară o altă actualizare periodică. Cu alte

cuvinte, utilizând valorile implicite de temporizare, este nevoie ca să se piardă 8 pachete consecutive ce conţin actualizări periodice.

Cu toate că acest protocol conferă o stabilitate crescută, reacţia la schimbări de stare a legăturii este redusă. Cu alte cuvinte, RIP nu reacţionează la o deteriorare a debitului legăturii sau la pierderi de pachete.

4.4.2 Protocolul OSPF

Protocolul OSPF are implementată posibilitatea de a distribui traficul pe mai multe căi către o anume destinaţie. Prin această facilitate traficul este distribuit în mod egal pe mai mute legături. Faţă de situaţia în care între rutere există o singură legătură, în cazul în care între rutere există două legături, limitarea de trafic se va situa la valoarea de 20Mbps, deoarece traficul este distribuit pe cele două legături limitate la valoare de 10Mbps.

Sintetizând, pentru protocolul OSPF cu un volum de trafic generat în valoare de 40Mbps, în cazul unei singure legături între rutere, din 1200 secunde de funcţionare a reţelei 288 de secunde traficul nu a fost dirijat, adică 24% din timpul de simulare. Pentru situaţia cu două legături, timp de 684 secunde (57%) traficul nu a fost dirijat.

Tabelul 4.17 OSPF – Perioadele de timp în care au existat pierderi de date pentru trafic generat de 40Mbps

Protocol de dirijare	Numărul de legături	Perioadele de timp în care au existat pierderi de date pentru trafic generat de 40Mbps
OSPF	2 legături	372÷672
		804÷840
		1056÷1116

Tabelul 4.18 OSPF –Perioadele de timp în care au existat pierderi de date pentru trafic generat de 10Mbps,15Mbps, 20Mbps, 30Mbps şi 40Mbps

Protocol de dirijare	Numărul de legături	Perioadele de timp în care au existat pierderi de date pentru trafic generat de				
		10Mbps	15Mbps	20Mbps	30Mbps	40Mbps
OSPF	1 legătură	---	---	---	---	324÷528
		---	---	---	---	696÷732
		---	---	---	---	972÷1020
	2 legături	---	---	---	---	372÷672
		---	---	---	---	804÷840
		---	---	---	---	1056÷1116

Tabelul 4.19 OSPF – Procent din timpul total în care au existat pierderi de date pentru 40Mbps trafic generat

Protocol de dirijare	Numărul de legături între rutere	Timpul total în care au existat pierderi de date	Procent din timpul total în care au existat pierderi de date pentru 40Mbps trafic generat
OSPF	1	288 secunde	24%
	2	684 secunde	57%

4.4.3 Protocolul EIGRP

Ca şi OSPF protocolul EIGRP are implementată posibilitatea de a distribui traficul simultan pe mai multe căi către o anume destinaţie. Prin această facilitate traficul este distribuit pe mai mute legături. Faţă de situaţia în care între rutele R1 şi R2 există o singură legătură, în cazul în care între ruterele R1 şi R2 există două legături, limitarea de trafic se va situa la valoarea de 20Mbps deoarece traficul este distribuit pe cele două legături limitate la valoare de 10Mbps.

În cazul unei singure legături între rutere şi 30Mbps trafic generat, din 1200 secunde de funcţionare a reţelei 960 de secunde traficul nu a fost dirijat pe ambele legături simultan, adică 80% din timpul de simulare.

Daca volumul de trafic generat este crescut la 40Mbps timp de 912 secunde (76% din timpul de simulare) traficul nu este dirijat simultan pe ambele legături.

Pentru situaţia cu două legături şi 30Mbps trafic generat, timp de 984 secunde (82% din timpul de simulare) traficul nu a fost dirijat simultan pe ambele legături.

Odată cu creşterea volumului de trafic generat la 40Mbps, timp de 624 secunde (52% din timpul de simulare) traficul nu este dirijat simultan pe ambele legături.

Tabelul 4.22 EIGRP –Perioadele de timp în care au existat pierderi de date pentru trafic generat de 10Mbps,15Mbps, 20Mbps, 30Mbps şi 40Mbps

Protocol de dirijare	Numărul de legături	Perioadele de timp în care au existat pierderi de date pentru trafic generat de				
		10 Mbps	15 Mbps	20 Mbps	30 Mbps	40 Mbps
EIGRP	1 legătură	---	---	---	36÷60	48÷72
		---	---	---	120÷192	144÷204
		---	---	---	228÷408	288÷456
		---	---	---	456÷936	492÷1128
		---	---	---	996÷1200	1176÷1200
	2 legături	---	---	---	48÷84	204÷240
		---	---	---	108÷180	456÷528
		---	---	---	204÷432	684÷1200
		---	---	---	552÷1200	---
		---	---	---	---	---

Tabelul 4.23 EIGRP – Procent din timpul total în care au existat pierderi de date pentru 30Mbps şi 40Mbps trafic generat

Protocol de dirijare	Numărul de legături între rutere	Trafic generat	Timpul total în care au existat pierderi de date	Procent din timpul total în care au existat pierderi de date
EIGRP	1 legătură	30Mbps	960	80%
		40Mbps	912	76%
	EIGRP	30Mbps	984	82%
	2 legături	40Mbps	624	52%

4.5 Timpul inițial de stabilizare și instabilitatea totală

Pentru a realiza un studiu asupra timpului de stabilizare au fost create patru scenarii de test. Aceste scenarii cuprind atât situația trivială cu două rutere și un singur link între ele, cât și situații mai complicate cum ar fi ultimul scenariu în care sunt implementate cinci rutere și opt legături de date.

Legăturile de date folosite în aceste scenarii sunt legături ideale, fără pierderi și cu timp de propagare instantaneu. Pentru a măsura timpul inițial de stabilizare al rețelei și timpul cumulat de instabilitate pe durata simulării au fost implementate stațiile S1÷S5 care generează trafic de date la un volum de 100Mbps indiferent de limitarea de debit aplicată pe legăturile de date dintre rutere. Legăturile de date dintre stații și rutere au rămas limitate la valoare de 100Mbps indiferent de ce alte limitări sunt aplicate.

În aceste scenarii au fost simulate două protocoale de dirijare, și anume OSPF și EIGRP. Protocolul RIP nu a fost luat în considerare deoarece este prea puțin sensibil la volumul de trafic de date dintre rutere.

4.5.1 Timpul de stabilizare

În aceste scenarii generatoarele de trafic (stațiile S1÷S5) pornesc la momentul de timp 0 secunde. Ruterele de la R_1 până la R_{n-1} pornesc la momentul de timp 10 secunde, iar ultimul ruter din rețea (R_n) va porni la momentul de timp 20 de secunde.

Pentru a calcula timpul inițial de stabilizare al rețelei se folosește următoarea ecuație [ADO12b]:

$$S_t = T_S - T_{Start}, \text{ unde} \qquad (4.1)$$

T_s – momentul de timp la care tabelele de dirijare s-au stabilizat;
S_t – diferența de timp dintre T_s și momentul de timp la care a pornit protocolul de dirijare pe ultimul ruter din rețea.

În aceste simulări T_{Start}=20. Astfel, protocolul de dirijare de pe ultimul ruter din rețea va porni după ce vor porni ruterele de la R_1 până la R_{n-1}.

Creşterea numărului de rutere, a numărului de legături şi a limitării de debit duce la o creştere a timpului de stabilizare (Fig. 4.61). Trebuie avut în vedere faptul că simulatorul foloseşte un generator de numere pseudoaleatoare în funcţionarea sa, ceea ce face ca rezultatele obţinute în urma simulărilor să aibă o oarecare variaţie.

Prin acest studiu am evidenţiat trei factori care au influenţă negativă asupra timpului de stabilizare în situaţia în care există trafic de date pe legături, şi anume:

- numărul de rutere;
- numărul de legături;
- limitarea de debit a legăturilor.

Creşterea valorii timpului de stabilizare este mare pentru limitări mari de debit şi un număr nu prea mare de rutere. Dacă în reţea nu ar fi trafic de date de valoare ridicată, timpul de stabilizare nu ar avea o valoare aşa de mare chiar în condiţiile unui număr însemnat de rutere. Astfel, limitarea de debit şi traficul de valoare ridicată au o influenţă negativă asupra stabilităţii reţelei.

Tabelul 4.24 Timpul de stabilizare pentru fiecare scenariu în funcție de limitarea de debit

Scenariul	Limitarea de debit	Protocol de dirijare	
		OSPF	EIGRP
2 rutere	100Mbps	55,3284186	0,0000181
	10Mbps	85,2377326	26,1991942
	4Mbps	85,2377326	55,0003318
	2Mbps	85,2377326	120,0732936
	1Mbps	97,7174924	123,0832458
	512Kbps	65,2377326	15,0025391
	256Kbps	65,2377326	15,0050703
	128Kbps	65,2377326	15,0101328
3 rutere	100Mbps	65,0124368	10,0000264
	10Mbps	75,0674960	95,0002940
	4Mbps	157,4686344	67,9981038
	2Mbps	251,9731680	66,0091604
	1Mbps	247,6650884	79,0413892
	512Kbps	253,9055734	255,7586042
	256Kbps	65,0674960	30,0147272
	128Kbps	65,0674960	30,0294459
4 rutere	100Mbps	65,0123888	10,0000341
	10Mbps	75,0076834	76,2005021
	4Mbps	75,0130148	210,0021702
	2Mbps	75,0136668	135,0892806
	1Mbps	92,0300147	258,5975114
	512Kbps	253,6913950	76,9513348
	256Kbps	65,0225503	30,0670387
	128Kbps	65,0327378	30,1140700
5 rutere	100Mbps	55,2378101	10,0000385
	10Mbps	242,4952815	41,2030643
	4Mbps	75,2411877	33,0029499
	2Mbps	75,2446037	30,0066762
	1Mbps	252,9252765	177,1386093
	512Kbps	251,0513176	110,3639136
	256Kbps	264,2858395	40,0398508
	128Kbps	108,6048951	35,0006335

Tabelul 4.25 Timpul de stabilizare pentru fiecare valoare a limitării de debit în funcție de scenariu

Limitarea de debit	Scenariul	Protocol de dirijare	
		OSPF	EIGRP
100Mbps	2 rutere	55,3284186	0,0000181
	3 rutere	65,0124368	10,0000264
	4 rutere	65,0123888	10,0000341
	5 rutere	55,2378101	10,0000385
10Mbps	2 rutere	85,2377326	26,1991942
	3 rutere	75,0674960	95,0002940
	4 rutere	75,0076834	76,2005021
	5 rutere	242,4952815	41,2030643
4Mbps	2 rutere	85,2377326	55,0003318
	3 rutere	157,4686344	67,9981038
	4 rutere	75,0130148	210,0021702
	5 rutere	75,2411877	33,0029499
2Mbps	2 rutere	85,2377326	120,0732936
	3 rutere	251,9731680	66,0091604
	4 rutere	75,0136668	135,0892806
	5 rutere	75,2446037	30,0066762
1Mbps	2 rutere	97,7174924	123,0832458
	3 rutere	247,6650884	79,0413892
	4 rutere	92,0300147	258,5975114
	5 rutere	252,9252765	177,1386093
512Kbps	2 rutere	65,2377326	15,0025391
	3 rutere	253,9055734	255,7586042
	4 rutere	253,6913950	76,9513348
	5 rutere	251,0513176	110,3639136
256Kbps	2 rutere	65,2377326	15,0050703
	3 rutere	65,0674960	30,0147272
	4 rutere	65,0225503	30,0670387
	5 rutere	264,2858395	40,0398508
128Kbps	2 rutere	65,2377326	15,0101328
	3 rutere	65,0674960	30,0294459
	4 rutere	65,0327378	30,1140700
	5 rutere	108,6048951	35,0006335

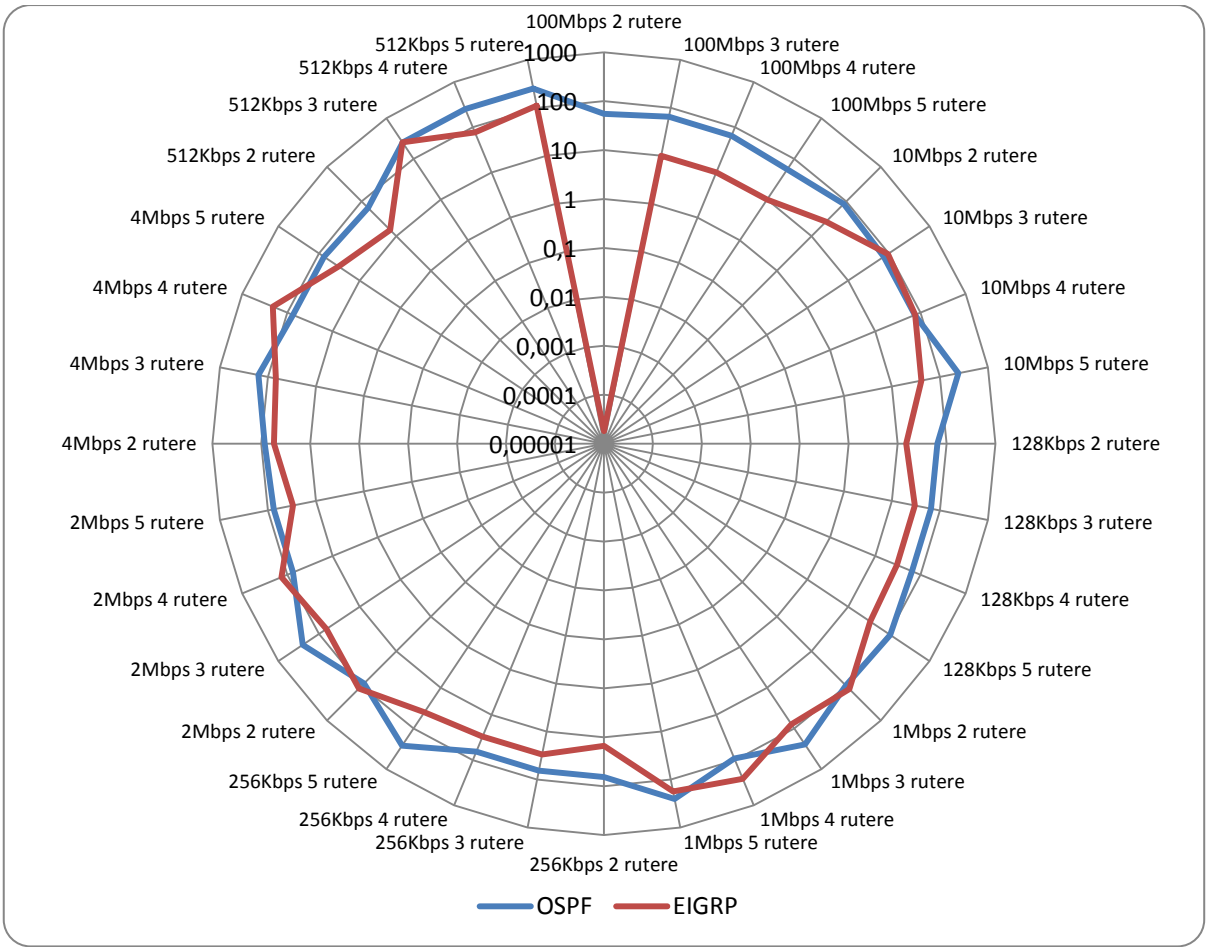

Fig. 4.61 Diagrama timpului de stabilizare obținut pentru OSPF și EIGRP în funcție de valoarea limitării de debit și simulare

4.5.2 Timpul cumulat de instabilitate

Se observă că, atât pentru OSPF cât și pentru EIGRP, numărul de rutere nu are o influență asupra numărului de stări de instabilitate. Scenariile cu 2,3,4 și 5 rutere generează stări de instabilitate în condițiile limitărilor de debit. Pe de altă parte, nu reiese o regulă care să determine creșterea sau scăderea numărului stărilor de instabilitate în funcție de limitarea de debit. Variația valorilor este determinată de generatorul de numere pseudoaleatoare implementat în mediul de simulare care are influență asupra pachetelor care se pierd datorită limitării de debit. Cu toate acestea, reiese clar faptul că limitarea de debit are un rol important în destabilizarea funcționării protocoalelor de dirijare.

Fiecare stare de instabilitate este concretizată prin pierderea convergenţei tabelelor de dirijare de pe rutere. Aceste tabele de dirijare sunt construite de protocoalele de dirijare. În momentul în care apare o modificare în structura reţelei (reală sau indusă de alţi factori cum ar fi limitarea de debit), protocoalele de dirijare trebuie să comunice între ele pentru a distribui noile modificări către fiecare ruter în parte. Cu alte cuvinte, tabelele de dirijare trebuie să conveargă din nou spre o stare stabilă. Inițial, protocolul de dirijare se află într-o stare stabilă. În momentul în care primeşte o actualizare de rute, acesta trece într-o stare instabilă de durată foarte mică.

Tabelul 4.26 Valorile pentru instabilitate în cazul OSPF şi EIGRP

		Activități de converge		Stări de instabilita		Timp cumulat de instabilitate		Procent al timpului cumulat de instabilitate pe durata simulării	
Simulare	Limitare	OSPF	EIGRP	OSPF	EIGRP	OSPF	EIGRP	OSPF	EIGRP
2 rutere	100Mbps	1	1	0	0	55,3284186	0,0000181	4.61	0.000002
	10Mbps	3	1	2	0	162,2837702	26,1991942	13.52	2.18
	4Mbps	3	2	2	1	149,8276837	55,0003318	12.49	4.58
	2Mbps	3	1	2	0	138,1045329	120,0732936	11.51	10.01
	1Mbps	3	1	2	0	174,2475025	123,0832458	14.52	10.26
	512Kbps	5	7	4	6	108,7739480	105,2116568	9.06	8.77
	256Kbps	4	4	3	3	102,8392711	60,2413148	8.57	5.02
	128Kbps	3	3	2	2	86,9845950	45,2536022	7.25	3.77
3 rutere	100Mbps	1	1	0	0	65,0124368	10,0000264	5.42	0.83
	10Mbps	3	4	2	3	180,3954207	95,0002940	15.03	7.92
	4Mbps	2	2	1	1	220,4725456	67,9988200	18.37	5.67
	2Mbps	1	2	0	1	251,9731680	81,0091604	21.00	6.75
	1Mbps	1	3	0	2	247,6650884	79,0413892	20.64	6.59
	512Kbps	1	1	0	0	253,9055734	255,7586042	21.16	21.31
	256Kbps	4	4	3	3	145,8958922	114,8876463	12.16	9.57
	128Kbps	5	3	4	2	130,6692777	85,0307456	10.89	7.09
4 rutere	100Mbps	1	1	0	0	65,0123888	10,0000341	5.42	0.83
	10Mbps	2	4	1	3	112,6627713	96,2017225	9.39	8.02
	4Mbps	4	1	3	0	156,0286235	210,0021702	13.00	17.50
	2Mbps	4	1	3	0	160,6338403	135,0892806	13.39	11.26
	1Mbps	3	1	2	0	206,0404999	258,5975114	17.17	21.55
	512Kbps	1	2	0	1	253,6913950	76,9513348	21.14	6.41
	256Kbps	3	4	2	3	201,2511640	99,1281287	16.77	8.26
	128Kbps	5	3	4	2	163,2420637	85,7698209	13.60	7.15
5 rutere	100Mbps	1	6	0	5	55,2378101	10,0000385	4.60	0.83
	10Mbps	1	7	0	6	242,4952815	106,2006065	20.21	8.85
	4Mbps	3	2	2	1	198,6715149	53,0013564	16.56	4.42
	2Mbps	1	2	0	1	75,2446037	30,0066762	6.27	2.50
	1Mbps	1	1	0	0	252,9252765	177,1386093	21.08	14.76
	512Kbps	1	1	0	0	251,0513176	110,3639136	20.92	9.20
	256Kbps	1	6	0	5	264,2858395	135,2135805	22.02	11.27
	128Kbps	2	3	1	2	191,5855152	80,1531335	15.97	6.68

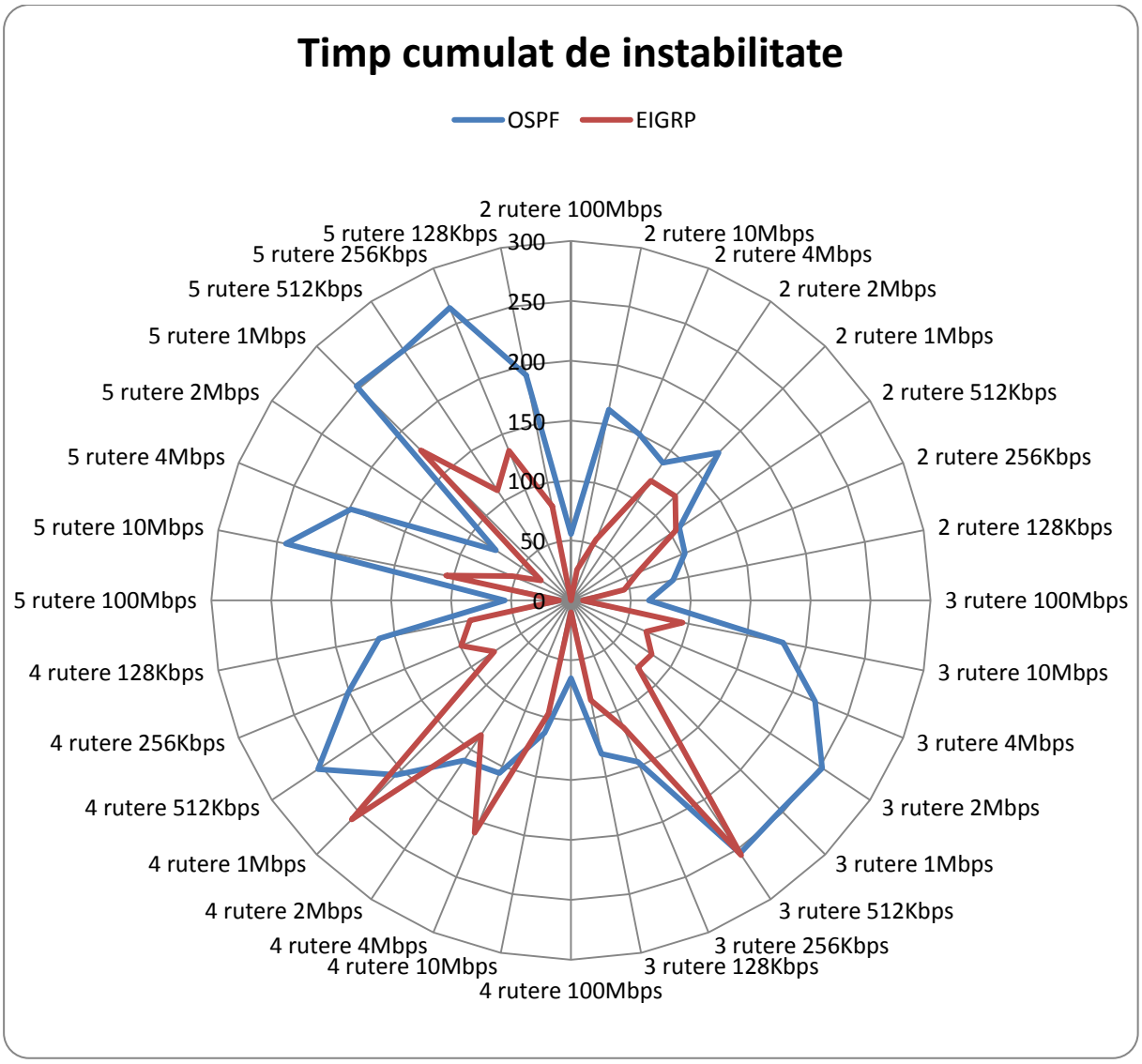

Fig. 4.63 Diagrama timpului cumulat de instabilitate obținut pentru OSPF și EIGRP în funcție de simulare și valoarea limitării de debit

5. Metodă de schimbare de informaţii privind actualizarea rutelor

5.1 Folosirea antetului extins Opțiuni pentru destinație

Antetul extins Opțiuni pentru destinație al IPv6 poate fi folosit pentru a trimite orice fel de date adiționale între sursa care a generat pachetul și destinația din antetul IPv6.

Având în vedere faptul că în mod implicit valoarea MTU[9] pentru Ethernet este de 1500 octeți, pentru a face loc antetului adițional este nevoie de o reduce a MTU cu 40 de octeți. Astfel pentru fiecare pachet IPv6 care este trimis către următorul ruter, din maximul de 1500 de octeți, 40 de octeți sunt alocați către datele aferente protocolului de dirijare.

Antetul extins Opțiuni pentru destinație poate conține mai multe opțiuni, pentru fiecare opțiune fiind două câmpuri obligatorii, si anume: tip opțiune și lungimea opțiunii. În plus, antetul extins Opțiuni pentru destinație mai necesită două câmpuri obligatorii care sunt necesare pentru a crea înlănțuirea de antete suplimentare. Astfel, trebuie definit următorul antet extins care urmează sau, dacă nu mai există alt antet extins, tipul de protocol care este

[9] Lungimea maximă a unei unităţi pentru transitere (Maximum Transmission Unit)

folosit pentru încapsularea segmentului din partea utilă a pachetului IPv6. De asemenea, pentru delimitarea antetelor suplimentare și/sau a părții utile din pachetul de date este nevoie de completarea câmpului lungime a antetului extins Opțiuni pentru destinație.

În abordarea propusă, pentru a nu aloca octeți suplimentari în mod inutil se folosește doar o singură opțiune din antetul suplimentar mai sus menționat. De asemenea se definesc câmpurile necesare trimiterii unei actualizări între două rutere (Fig. 5.3):

- Următorul antet
- Lungimea antetului
- Tip opțiune
- Lungime opțiuni date
- Rețea
- Mască de rețea
- Operație ruta
- Cost
- Confirmare Rețea și mască asociată

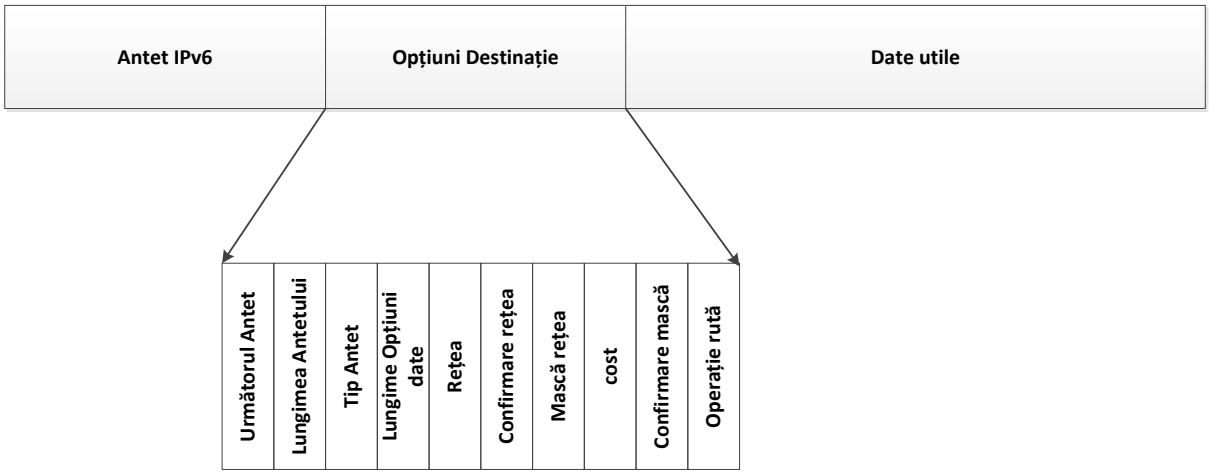

Fig. 5.3 Pachetul IPv6 împreună cu antetul suplimentar Opțiuni pentru destinație

Pe baza rețelelor configurate pe interfețe, va fi populată o tabelă de dirijare tampon internă. Folosind această tabelă se actualizează tabela de dirijare a maşinii.
Se definește operație rută astfel:

$$operatie\ ruta = \begin{cases} 1, daca\ ruta\ contine\ o\ retea\ care\ trebuie \\ \quad adaugata\ in\ tabela\ ruterelor\ partenere \\ -1, daca\ ruta\ contine\ o\ retea\ care\ trebuie \\ \quad stearsa\ din\ tabela\ ruterelor\ partenere \end{cases} \tag{5.1}$$

Tabela internă tampon are următoarele câmpuri:

- Rețea
- Mască rețea
- Cost
- Interfața pe care trebuie transmisă ruta
- Confirmare
- Operație rută
- Rută internă sau externă

Astfel, rutele generate inițial, rute ce conțin rețelele direct conectate la ruter, vor avea operație rută=1. Ulterior, dacă o rețea nu mai este validă va fi trimisă o actualizare ce conține operație rută=-1. Dacă un ruter primește o actualizare de rută cu câmpul operație rută=1, ruta respectivă va fi adăugată la tabela de dirijare tampon și mai departe la tabela de dirijare a ruterului. În caz contrar, dacă se primește o actualizare cu câmpul operație rută=-1 ruta respectivă va fi ștearsă la tabela de dirijare tampon și mai departe la tabela de dirijare a ruterului.

5.2 Transmiterea actualizărilor şi a confirmărilor

Pentru a elimina probleme ce apar datorită pierderilor de pachete, este folosită o strategie de tip ARQ persistent[10]. Fiecare actualizare este trimisă până când se primește o confirmare din partea ruterului partener. De asemenea fiecare confirmare este trimisă până în momentul în care o confirmarea este la rândul ei confirmată. În acest moment se opreşte şi transmiterea confirmărilor.

În cazul în care nu se primește confirmare pentru ruta trimisă într-un interval predefinit de timp, respectivul ruter vecin va fi declarat ca indisponibil și nu va mai fi luat în considerare. De asemenea, rutele care îl iau în considere ca vecin vor fi trimise ca actualizări către ceilalți vecini cu valoarea operație rută=-1 și, apoi, vor fi eliminate din tabela tampon internă și din tabela de dirijare a ruterului. În cazul în care se primește confirmare de la ruterul vecin pentru actualizarea de rută adăugată în antetul extins Opțiuni pentru destinație, ruterul nu mai transmite actualizare cu ruta respectivă și va trece la trimiterea următoarei actualizări, dacă aceasta există.

Această abordare este asemănătoare cu sistemul folosit în protocoalele de dirijare vector distanță cum ar fi RIP și parțial EIGRP. La aceste protocoale sunt distribuite pas cu pas, de la ruter la ruter. În urma primirii unei actualizări, fiecare ruter își va modifica corespunzător tabela internă și pe baza acesteia va transmite mai departe actualizările.
Deși în cazul protocoalelor de dirijare vector distanță “veștile rele” traversează rețeaua încet, în acest caz, actualizările traversează rețeaua rapid datorită menținerii constante a adiacenței între rutere.

În perioadele în care nu există trafic de date, nu se mai pot transmite actualizări. De aceea, dacă nu există trafic de date pe interfețe, ruterele încep să genereze trafic pentru a menține adiacența cu vecinii acestora și pentru a trimite actualizări de rute. În momentul în care apare trafic, vor fi folosite pachetele IPv6 utile pentru trimiterea datelor specifice.

Datorită antetului suplimentar care este introdus în fiecare pachet IPv6, aproximativ 2,66% din capacitatea de transport este folosită de către protocolul de dirijare. Într-adevăr, față de alte protocoale de dirijare, acest procent este ridicat, însă se obține avantajul robusteţii chiar în situațiile în care există pierderi mari de pachete.

Momentan, pentru calculul costului unei rute se ia în considerare numărul de salturi dintre ruterul actual şi destinaţie. În dezvoltarea protocolului, pentru calculul costului unei rute, se pot folosi mai mulți parametri cum ar fi:

- Lungimea rutei constând în numărul de rutere între sursă și destinație;
- Viteza legăturilor dintre rutere;
- Pierderile care apar pe legături;
- Stabilitatea legăturilor.

[10] Informaţii legate de strategiile ARQ se pot găsi în [FAIR02] şi [PETE12]

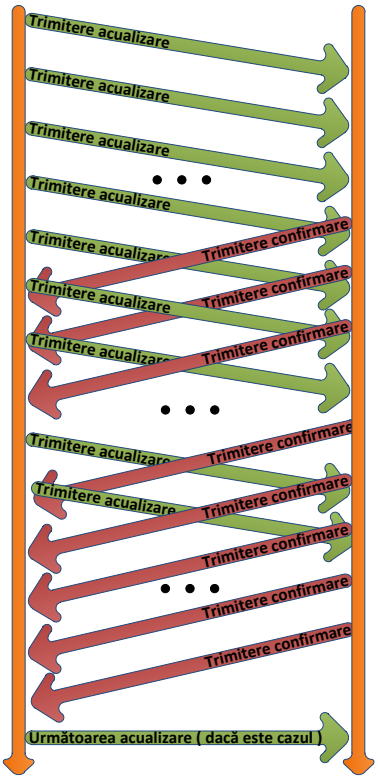

Fig. 5.4 Mecanismul de transmitere și confirmare a actualizărilor

5.3 Aspecte legate de implementarea metodei de transmitere a informaţiilor de dirijare

Implementarea modulului de transmitere a actualizărilor de rute a fost realizată folosind sistemul de operare Linux. Implementarea software s-a realizat sub forma unui modul de kernel. Prin aceasta se prioritizează funcţionarea modulului faţă de procesele care rulează în spaţiul de memorie alocat utilizatorului, iar întârzierile apărute în urma prelucrării pachetului sunt minime.

Implementarea unui modul de kernel presupune o provocare deoarece, pentru preluarea datelor legate de pachetele IPv6, este necesară folosirea funcțiilor cu care este construit kernelul. Aceste funcții sunt puțin documentate, înțelegerea acestora presupunând o activitate laborioasă. Dezvoltarea software-ului realizat s-a făcut într-o mașină virtuală datorită dificultăților apărute în identificarea erorilor de programare.

Metoda propusă fiind implementată sub forma unui modul de kernel, nu s-a putut realiza rularea pas cu pas pentru a fi urmărită evoluția variabilelor folosite sau oprirea execuției la erori. Majoritatea erorilor în rularea programului aveau ca rezultat blocarea sistemului de operare. Evoluția programului a fost dificilă și de lungă durată. Fiecare modificare de esență în cod a urmat în ordine pașii:

1.Salvarea stării curente a mașinii virtuale;

2.Operarea modificării;

3.Adăugarea de mesaje către consola virtuală pentru a fi identificate eventualele erori;

4. Compilarea programului;

5. Inserarea în lista de module și rularea acestuia.

În situația în care apăreau erori, se depanau folosind mesajele date de program și erorile pe care le trimitea kernelul către consola virtuală. Apoi se restaura mașina virtuală la starea anterioară și se remediau erorile identificate.

Deoarece se intervenea asupra pachetului IPv6, a fost nevoie ca modificările realizate să fie urmărite în pachetul de date care părăsea interfața de rețea. Pentru aceasta a fost folosit software-ul Wireshark care captura și interpreta pachetele de date.

Deşi se puteau folosi şi alte antete extinse, s-a ales antetul Opţiuni pentru destinaţie datorită modului cum este tratat de către rutere. Antetele Opţiuni salt-după-salt şi Opţiuni dirijare sunt destinate pentru transportul de informaţii specifice între două rutere adiacente şi sunt eliminate din pachet la destinaţie sau de primul ruter în cazul în care destinaţia se află la mai multe salturi distanţă. În procesul de dirijare, antetul Opţiuni pentru destinaţie este păstrat până când pachetul ajunge la destinaţie chiar dacă sunt parcurse mai multe rutere.

Pentru a captura pachetele de date în vederea modificării lor se foloseşte platforma NETFILTER[11] care pune la dispoziţie metode prin care să se obţină acces la pachetele de date care intră sau ies pe interfeţele fizice sau logice de reţea. Cu ajutorul a două constante se alege momentul din lanţul de procesare în se capturează pachetul de date. Astfel, în cazul `NF_IP6_PRE_ROUTING` se capturează pachetele IPv6 care sunt primite, iar în cazul `NF_IP6_POST_ROUTING` se capturează pachetele care urmează a fi trimise. Manipularea rutelor din tabela de dirijare a kernelului, în sensul adăugării sau ştergerii acestora, foloseşte platforma NETLINK[12].

5.3.1 Tratarea pachetelor primite

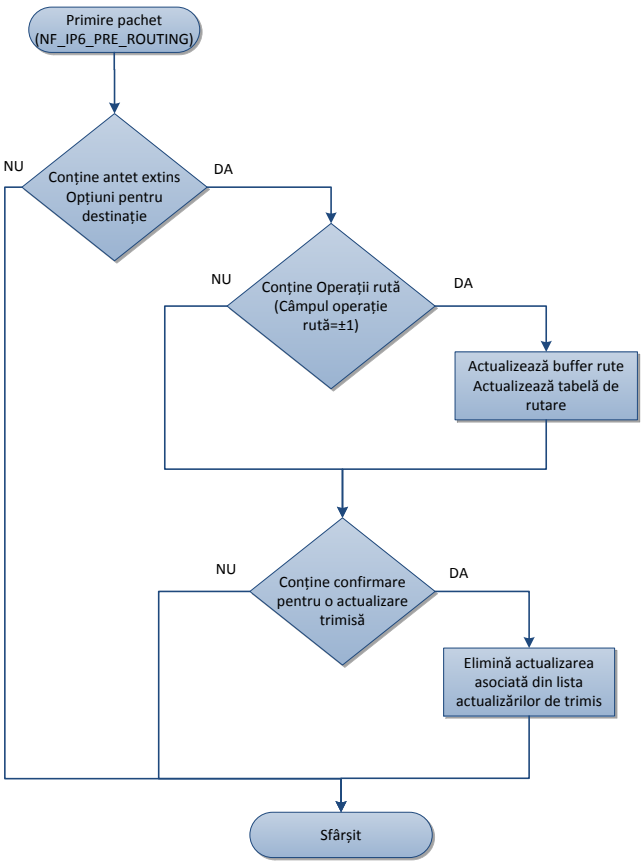

Fig. 5.5 Schema logică primire pachet

[11] Informaţii legate de NETFILTER se pot găsi în [BEN05]
[12] Aspecte legate de NETLINK se pot găsi în [BEN05]

5.3.2 Tratarea pachetelor trimise

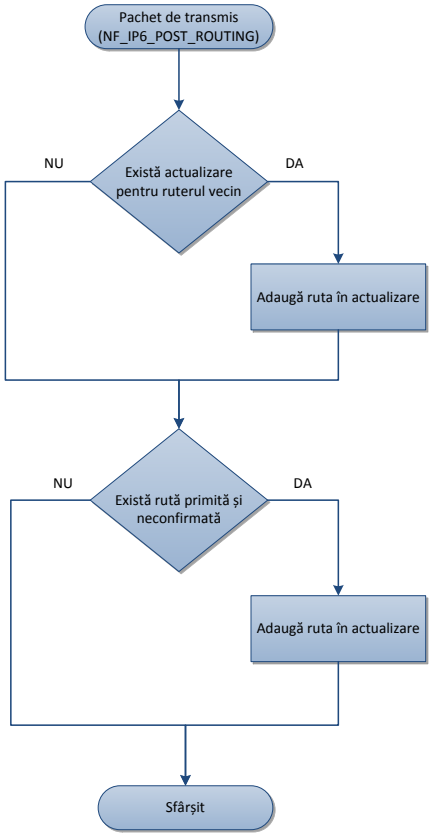

Fig. 5.6 Schema logică trimitere pachet

5.3.3 Simularea rețelelor de test

Pentru a analiza software-ul realizat a fost nevoie de realizarea unui simulator adaptat pentru necesitățile analizei. Astfel, s-a folosit mediul VMware ce permite crearea și rularea de mașini virtuale.

Numărul de mașini virtuale care se poate crea depinde strict de performanțele sistemului gazdă. În cel mai complex scenariu de test (Fig. 5.23) a fost nevoie de un număr de cinci mașini virtuale. Pentru a satisface cerințele unei astfel de simulări, a fost folosit un sistem gazdă cu următoarele caracteristici:

- Procesor: QuadCore AMD Athlon II X4 620, 2600 MHz (13 x 200);
- Memorie RAM: 8 GB;
- Disc stocare rapid: WDC WD7501AALS-00E8B0 (750 GB, 7200 RPM, SATA-II).

Pentru fiecare ruter a fost creată câte o mașină virtuală la care au fost alocați 1024 MB memorie RAM. Pe fiecare mașină în parte a fost instalat sistemul de operare openSUSE versiune 12.1 ce are la baza kernelul de Linux versiunea 3.1.0-1.2.

Software-ul VMware permite crearea de interfețe de rețea virtuale și a unui număr nelimitat de segmente LAN. La fiecare segment LAN se pot atribui interfețele de rețea virtuale create în cadrul fiecărei mașini. În acest mod se poate crea orice structură de rețea.

Pentru a realiza limitarea de debit, inițial, au fost luate în considerare 2 variante, și anume:

- Folosire posibilității de a realiza limitare pe fiecare interfață de rețea virtuală creată pe fiecare mașină în parte;

- Realizarea limitării din sistemul de operare gazdă prin folosirea opțiunii de „modelare a traficului”.

În urma încercărilor realizate, a fost aleasă ultima variantă datorită faptului ca traficul era limitat fără a apărea variații mari de la limită.

După crearea mașinilor virtuale, crearea interfețelor de rețea și atribuirea acestora la segmentele LAN corespunzătoare a urmat instalarea sistemului de operare, copierea codului sursă și compilarea acestuia. Pentru dezvoltarea modulului a fost folosit mediul de dezvoltare KDevelop versiunea 4.0. Deoarece valorile variabilelor nu puteau fi urmărite a fost nevoie de generarea de mesaje către consola virtuală (Fig. 5.7).

Cu ajutorul acestor mesaje s-a realizat depanarea programului și verificarea funcționării blocurilor de cod ale programului:

- Transmitere a actualizărilor;
- Interpretare a actualizărilor primite;
- Monitorizarea listelor cu actualizările primite și de trimis;
- Funcționarea mecanismului de confirmare;
- Monitorizarea tabelei de dirijare;
- Verificarea operațiilor de adaugare/retragere rute;
- Verificarea funcționării funcției de tratare a evenimentelor interfețelor de rețea;
- Duratele de timp pentru operațiile realizate.

Fiecare mesaj care apare pe consola virtuală are alocat de către kernel un index de timp. Acest index de timp a fost foarte util în determinarea duratei diverselor operații care aveau loc. Măsurarea timpilor de stabilizare din secțiunea 5.5 a fost realizată prin măsurarea diferențelor de timp dintre momentul în care se inițializa modulul și momentul în care tabela de dirijare era actualizată complet.

Resursele de procesare ale sistemului gazdă erau distribuite către toate mașinile virtuale și, mai ales în scenariul de test cu cinci rutere, microprocesorul sistemului era încărcat în proporție de 100%. Realizarea unor teste cu mașini dedicate oferă posibilitatea ca timpii de rulare să scadă, ceea ce ar duce la stabilizări mult mai rapide.

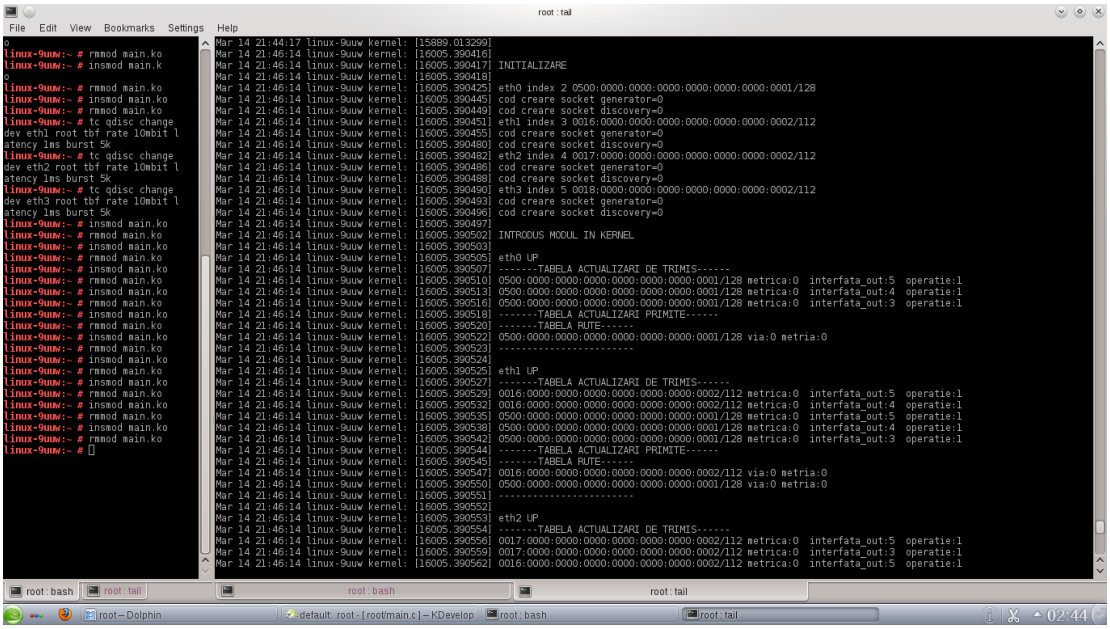

Fig. 5.7 Exemplu de rulare a unei mașini virtuale

5.4 Rezultate experimentale și comparative

5.4.1 Comparație cu privire la timpul de stabilizare în diverse scenarii de test

Pentru a compara timpul de stabilizare a rețelei se va măsura diferența de timp dintre momentul de pornire a protocolului de dirijare pe ultimul ruter din rețea și momentul de timp în care este încheiat procesul de adiacență dintre ultimul ruter și vecinii acestuia.

Astfel, în realizarea simulărilor, ruterele R_i ($1 \le i \le n-1$))vor porni la indexul de timp 10 secunde, iar ultimul ruter, R_n, va porni la indexul de timp 20 secunde. Timpul de stabilizare T_s se va calcula astfel:

$$T_s = M_s - 20, \qquad (5.2)$$

unde M_s reprezintă indexul de timp la care s-a realizat adiacența dintre ruterul R_n și vecinii acestuia.

Rularea simulărilor s-a realizat în absența traficului de date pentru a nu crește timpul de stabilizare în cazul protocoalelor RIP, OSPF, EIGRP. Pentru metoda propusă, traficul nu va afecta timpul de stabilizare, având în vedere faptul că simulările au fost realizate în prezența traficului de date la valoarea de 100 Mbps.

5.4.6 Rezultate comparative

Se evidențiază faptul că protocoalele OSPF și EIGRP au timpul de stabilizare neafectat de scenariul în care au fost simulate. Pentru protocolul RIP se remarcă o creștere semnificativă a timpului de stabilizare al rețelei.

Metoda propusă este puțin afectată de creșterea numărului de rutere din rețea. Față de protocolul EIGRP, timpul de stabilizare este mai crescut, însă nu cu mult. Acest fapt era de așteptat datorită modului în care am propus transmiterea actualizărilor, și anume secvențial. Celelalte protocoale de dirijare au posibilitatea de a trimite mai multe actualizări simultan, fapt evidențiat în stabilizare mai rapidă a protocolului EIGRP.

Tabelul 5.5 Tabel cu valorile timpului de stabilizare pentru protocoalele studiate și metoda propusă în scenariile simulate

	Scenariul	RIP	OSPF	EIGRP	Metoda propusă
Timp de stabilizare T_s (secunde)	2 rutere	0, 00001	55,37253	0,00004	0,000048
	3 rutere	0,00001	55,43788	0,00004	0,004102
	4 rutere	20,00001484	55,149528	0,00004	0,013720
	5 rutere	20,000013	55,41409	0,00005	0,009929

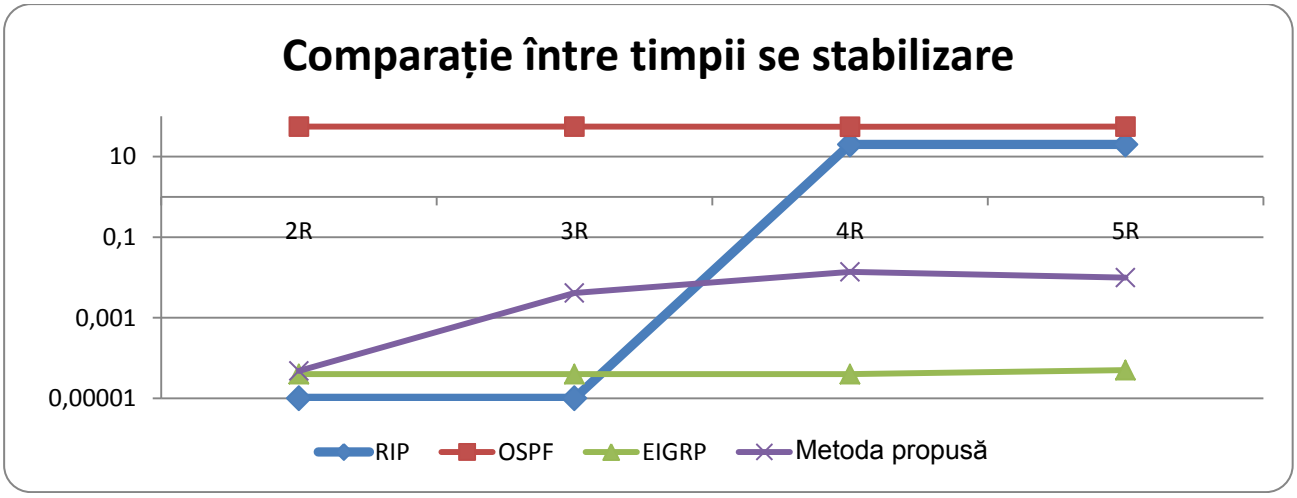

Fig. 5.28 Comparație între timpii de stabilizare dintre protocoalelor RIP, OSPF, EIGRP și metoda propusă

5.5 Efectele limitării de debit

Tabelul 5.6 Valorile timpului de stabilizare pentru metoda propusă în diferite scenarii de test

Scenariul / Limitare		2 rutere	3 rutere	4 rutere	5 rutere
100 Mbps	Timp start	31001,997576	28788,317715	11926,815049	13837,508440
	Timp stop	31001,997528	28788,313613	11926,801329	13837,498511
	Diferență timp	**0,000048**	**0,004102**	**0,013720**	**0,009929**
10 Mbps	Timp start	31155,481891	29094,356575	12314,424491	14178,498433
	Timp stop	31155,481873	29094,347532	12314,402031	14178,477451
	Diferență timp	**0,000018**	**0,009043**	**0,022460**	**0,020982**
4 Mbps	Timp start	33188,723677	29257,769866	12929,903372	14825,112311
	Timp stop	33188,723659	29257,755170	12929,843311	14825,057310
	Diferență timp	**0,000018**	**0,014696**	**0,060061**	**0,055001**
2Mbps	Timp start	33356,067426	29406,570290	13167,020429	15119,451417
	Timp stop	33356,067409	29406,546211	13166,956222	15119,349759
	Diferență timp	**0,000017**	**0,024079**	**0,064207**	**0,101658**
1 Mbps	Timp start	33476,939271	29623,198137	13537,304050	14414,593306
	Timp stop	33476,939254	29623,108455	13537,250022	14414,386643
	Diferență timp	**0,000017**	**0,089682**	**0,054028**	**0,206663**
512 Kbps	Timp start	33600,262387	29807,536966	13785,425205	15369,731147
	Timp stop	33600,262370	29807,467402	13785,325527	15369,310020
	Diferență timp	**0,000017**	**0,069564**	**0,099678**	**0,421127**
256 Kbps	Timp start	33731,183921	29981,795321	14158,257457	15751,669560
	Timp stop	33731,183903	29981,609076	14157,927953	15750,687921
	Diferență timp	**0,000018**	**0,186245**	**0,329504**	**0,981639**
128 Kbps	Timp start	33859,891683	30286,959644	14448,531096	16007,407206
	Timp stop	33859,891664	30286,305022	14447,821113	16005,443787
	Diferență timp	**0,000019**	**0,654622**	**0,709983**	**1,963419**

Pentru metoda propusă de transmitere de actualizărilor de rute, limitarea de debit nu are un efect major în ceea ce privește creșterea timpului de stabilizare a rețelei. Acest fapt demonstrează clar utilitatea asigurării unui așa denumit ”canal” alocat special pentru protocolul de dirijare.

În cel mai defavorabil caz utilizat în scenariile de test, valoarea maximă a timpului de transmitere a tuturor actualizărilor de rute este de 1,96 secunde în situația unei rețele cu 5 rutere și limitare de debit la valoarea de 128Kbps. Scenariul 5 rutere cu limitare de banda la valoarea de 128Kbps este unul destul de nefavorabil pentru funcționarea stabilă a unei rețele ce folosește protocoale de dirijare.

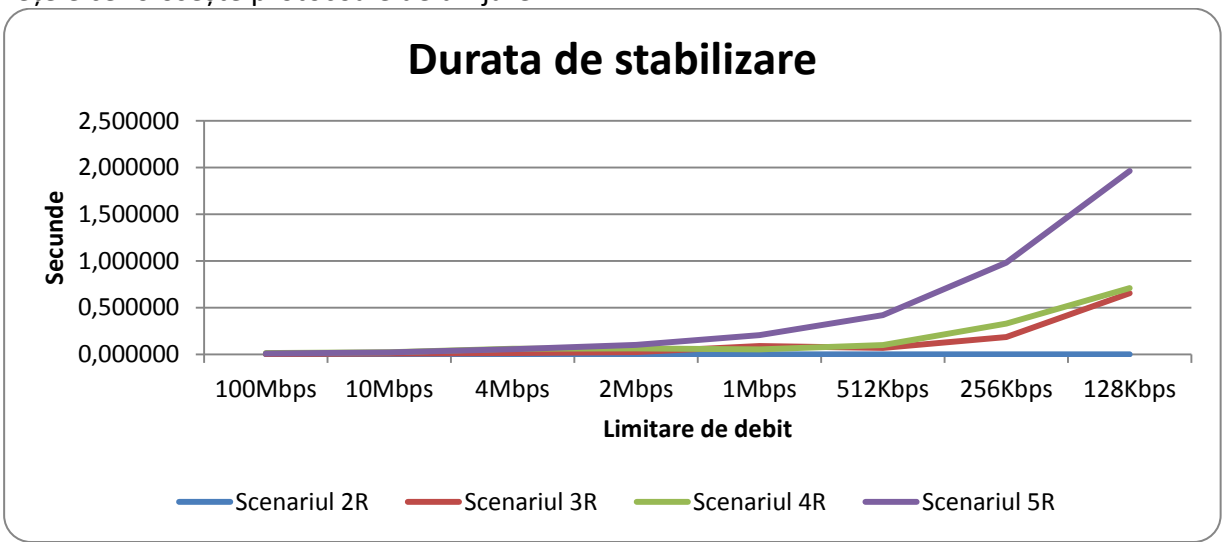

Fig. 5.29 Timpul de stabilizare a rețelei pentru scenariile de test

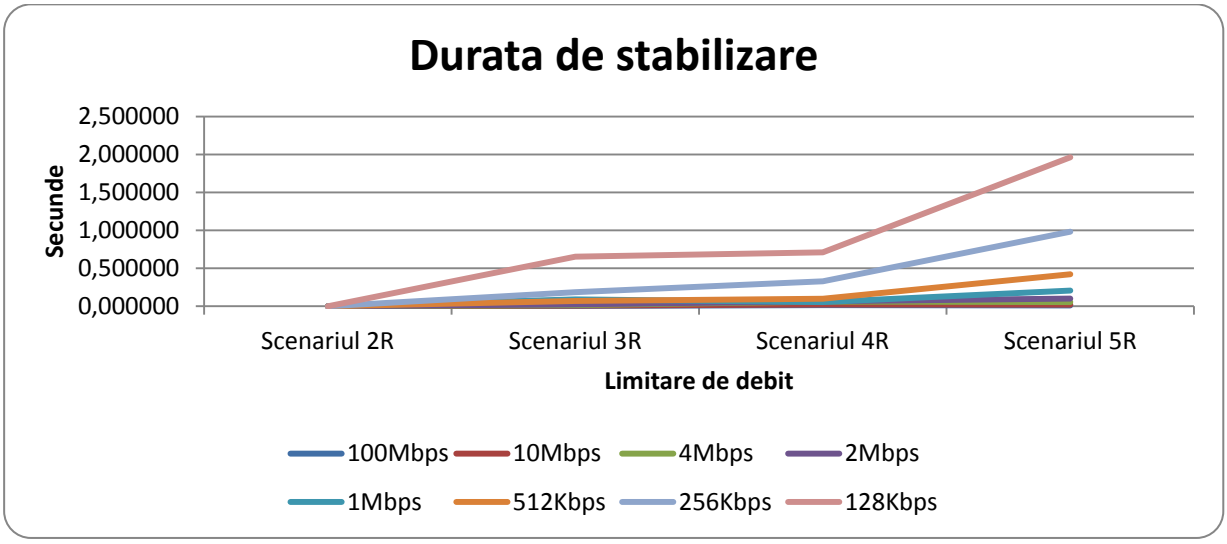

Fig. 5.30 Timpul de stabilizare a rețelei pentru limitările de debit aplicate

Mărirea limitării de debit la valoarea de 1 Mbps aduce o creștere sporită a timpului de stabilizare al rețelei. Comparând timpii cuprinși în intervalul 100 Mbps ÷ 1Mbps cu intervalul 1Mbps ÷ 128Kbps, observăm că, pentru cel din urmă interval, creșterea timpului de stabilizare este mai mare decât pentru primul interval. Mai mult, odată cu creșterea numărului de rutere din rețea, timpul de stabilizare începe să aibă o creștere sporită (Fig. 5.29, Fig. 5.30).

Timpul de stabilizare al rețelei, luând în considerare toate ruterele, nu poate fi formulat concis din punct de vedere matematic. Acesta depinde de o multitudine de factori,cum ar fi:

- Momentele de timp la care ruterele își schimbă starea;
- Momentele de timp la care legăturile își schimbă starea;
- Momentele de timp la care pornesc ruterele;
- Limitările de debit specifice fiecărei legături în parte;
- Numărul de rutere;
- Numărul de legături;
- Structura rețelei.

6. Concluzii și perpective

6.1 Concluzii generale

Evoluția Internetului a determinat o dezvoltare nebănuită a rețelelor de date IP. Sunt foarte puține cazurile în care o rețea de date nu este conectată la Internet. Această evoluție a adus cu sine și o serie de provocări care puneau problema conectării a din ce în ce mai mulți utilizatori. Din punctul de vedere al diponibilității de adrese IP a fost necesară dezvoltarea de noi scheme prin care să fie acomodați noi utilizatori în rețea. În momentul în care s-a ajuns la concluzia că aceste scheme nu mai sunt suficiente s-a decis dezvoltarea următoarei generații a protocolului IP care să acopere necesarul de adrese. Astfel a apărut IPv6, iar, în zilele noastre, migrarea este în plin proces de desfășurare.

Rețelele de date ale organizatiilor și instituțiilor au deja o mărime suficientă ca să impună folosirea unui protocol de dirijare. Cu toate că se ușurează configurarea și mentenanța procesului de dirijare, s-a constatat că pot apare o serie de probleme. Defectele pot induce instabilități mari în funcționarea protocoalelor de dirijare care pot afecta într-o măsură mai ridicată decât în mod normal traficul prin rețea. Prin studiul realizat în Capitolul 2 au fost identificate metodele existente care au fost dezvoltate să ajute la revenirea rapidă a unui protocol de dirijare dintr-o stare instabilă către o stare stabilă. Aceste metode au fost gândite ca module adiționale care pot fi atașate protocoalelor consacrate de dirijare.

Pentru a se clarifica impactul pe care instabilitatea îl poate avea asupra protocoalelor de dirijare a fost efectuat un studiu cuprinzător asupra modului în care reacționează protocoalele de dirijare clasice la evenimente care pot induce instabilitate. Astfel, în urma simulărilor realizate în scenarii în care au fost introduse pierderi de pachete sau limitări de debit pe legături, s-a constatat sensibilitatea protocoalelor de dirijare actuale la anumite condiții nefavorabile. Mai mult, pentru ca utilizatorii rețelei să nu fie afectați de instabilitatea care apare în protocoalele de dirijare, a fost dezvoltată o metodă prin care să fie asigurată robustețea procesului de creare și menținere a adiacenței și procesului de trimitere a actualizărilor între rutere. Principiile care stau la baza metodei dezvoltate au creat oportunitatea implementării principiului de dirijare ce folosește vectorii distanță. Astfel, s-a obținut un nou protocol de dirijare rapid și robust. Pentru a se asigura o dezvoltare ulterioară a noului protocol realizat implementarea s-a realizat ca modul de kernel pentru o rapiditate sporită în procesarea datelor.

Activitatea științifică desfășurată s-a concretizat în trei articole la conferințe internaționale dintre care una indexată IEEE-Xplore. Lista acestor articole se regăsește în Anexa 1.

6.2 Contribuţii aduse de lucrare

- *Studiul metodelor existente de recuperare după o stare de instabilitate*

Dezvoltarea unei metode transmitere a informațiilor de actualizare a impus efectuarea unui studiu, în secțiunea 3.2, asupra preocupărilor recente în zona recuperării unei rețele de comunicații, după o starea de instabilitate.

- *Definirea explicită a noțiunilor de stare stabilă a unei rețele, proces de convergență către o stare stabilă, plecând de la informațiile de adiacență existente în rutere.*

În secțiunea 4.1, a fost explicit legată noțiunea de stare stabilă de informațiile de adiacență existente în rutere și a fost descrisă funcționarea unei rețele de comunicații în situația fluctuării informațiilor de adiacență. A fost reliefată importanța duratei de regăsire a unei stări stabile ca principal parametru al unui proces de convergență.

- *Analiza prin simulare a comportamentului protocoalelor de dirijare în condiții de pierderi de pachete și limitare de debit*

Situațiile reale de proiectare, mentenanță și exploatare a unei rețele de date au evidențiat nevoia unui studiu asupra comportamentului protocoalelor de dirijare. În urma simulărilor realizate în secțiunea 4.3, cu ajutorul mediului OPNET Modeler, s-a constatat că un grad sporit de pierderi poate induce instabilitate crescută în menținerea adiacenței dintre rutere. Nu sunt rare cazurile în care traficul care parcurge o rețea este mult mai mare decât debitul asigurat de anumite legături de date. În urma simulărilor realizate în secțiunea 4.4, la diferite valori crescute de limitări de debit s-a observat, ca și în situația precedentă, o instabilitate crescută în menținerea adiacenței dintre rutere.

- *Reliefarea faptului că limitarea puternică de debit a legăturilor de date este un factor major de instabilitate a unei rețele, chiar și când legăturile de date nu sunt afectate sau echipamente de rețea defecte*

În secțiunea 4.5, în urma analizei rezultatelor obținute după efectuarea simulărilor asupra comportamentului protocoalelor de dirijare în condiții de limitare de debit, îngreunează actualizarea informațiilor de dirijare, fără ca rețeaua să prezinte defecte majore.

- *O nouă metodă de transmitere a informațiilor de actualizare*

În Capitolul 5, a fost prezentată o nouă metodă de transmitere a informațiilor de actualizare, aplicabilă rețelelor IPv6, bazată pe utilizarea antetului extins Opțiuni pentru destinație. Prin metoda dezvoltată pachetele Hello proprii și informațiile de actualizare sunt transmise chiar și în condiții de pierderi mari de date sau limitări puternice de debit pe legături cu prețul unui grad de ocupare al legăturii mai mare ca la protocoalele de dirijare clasice.

- *Un nou protocol de dirijare*

În Capitolul 5, implementarea metodei propuse a folosit un algoritm de dirijare bazat pe vectori distanță. Pe baza actualizărilor comunicate, se creează rute și se actualizează tabela de dirijare a mașinii. Astfel s-a realizat un nou protocol de dirijare robust și rapid.

- *Implementarea protocolului de dirijare propus*

În secțiunea 5.3 este descrisă implementarea protocolului de dirijare în limbajul C++ sub sistemul de operare Linux. Pentru a dobândi prioritatea dorită, implementarea s-a realizat ca modul de kernel în Linux. Ulterior, în cadrul simulatorului de rețea, fiecare mașină virtuală care a emulat un ruter a conținut o astfel de implementare a protocolului.

- *Realizarea unui simulator de rețea de comunicații ce funcționează după protocolul de dirijare propus*

În Capitolul 5, este descris un mod de verificare și analiză a protocolului de dirijare propus, prin realizarea unui simulator bazat pe utilizarea unor mașini virtuale Linux. Fiecare mașină virtuală emulează funcționarea unui ruter din rețea. Ținând cont de modul de simulare, protocolul a fost implementat în limbajul C++ sub Linux.

6.3 Perspective

- *Construirea unei metrici complexe*

În varianta actuală, protocolul de dirijare dezvoltat folosește doar numărul de salturi până la destinație pentru a calcula costul unei rute. Pentru dezvoltare o serie de alți parametri pot fi luați în considerare: capacitatea legăturilor, latența pe legături, încărcarea legăturilor, stabilitatea legăturilor, stabilitatea vecinilor.

- *Nivel suplimentar de dirijare*

Datorită modului în care funcționează antetul extins folosit, metoda poate fi implementată pe rutere distante între care există o serie de alte rutere, care nu participă la dirijarea curentă. Astfel se poate face dirijarea propriilor pachete peste infrastructura existentă.

Bibliografie selectivă

[ACEV94] Albrightson B., Garcia-Luna-Aceves J., Boyle J.. „EIGRP - a fast routing protocol based on distance vectors”, Proceedings of Networld/Interop, May 1994

[ADO12b] Adomnicăi C., Mînzu V., „Method for adjacency information updating in IPv6 networks”, Proceedings of the 16th International Conference on System Theory, Control and Computing Joint Conference SINTES 16, SACCS 12, SIMSIS 16, 12 - 14 Octombrie, Sinaia, Romania, ISBN 978-606-8348-48-3

[AGAR05] Agarwal S., Nucci A., Bhattacharyya S.,” Measuring the shared fate of IGP engineering and interdomain traffic”, Proceedings of the 13th International Conference on Network Protocols (ICNP), 2005

[APPL03] Applegate D., Cohen E., „Making intra-domain routing robust to changing and uncertain traffic demans: Understanding fundamental tradeoffs”, SIGCOMM ’03, August 2003

[ATLA06]Atlas A., "U-turn Alternates for IP/LDP Fast-Reroute", Internet-Draft, Work in Progress, http://tools.ietf.org/id/draft-atlas-ip-local-protect-uturn-03.txt Internet Engineering Task Force, Februarie 2006

[BEN05] Benvenuti C., “Understanding Linux Network Internals”, O'Reilly, December 2005, ISBN: 978-0-596-00255-8

[BRYA07]Bryant S., Filsfils C., Previdi S., et al., „IP fast reroute using tunnels”, Internet Draft, http://tools.ietf.org/id/draft-bryant-ipfrr-tunnels-03.txt, Internet Engineering Task Force, 2008

[DUBO04] Dubois N., Fondeviole B., Michel N., „Fast convergence project,” presented at the RIPE47, Ianuarie 2004, http://www.ripe.net/ripe/ meetings/ripe-47/presentations/ripe47-routing-fcp.pdf

[FAIR02] Fairhurst G., Wood L., „Advice to link designers on link Automatic Repeat reQuest (ARQ)”, Request for comments 3366, Internet Engineering Task Force, 2002.

[FORTZ06] Fortz B., Rexford J., Thorup M., “Traffic engineering with traditional IP routing protocols,” IEEE Commun. Mag., Octombrie 2002.

[GJOK07] Gjoka M., Ram V., Yang X., „Evaluation of IP Fast Reroute Proposals”, IEEE International Conference on Communication System Software Middleware (COMSWARE), Bangalore, India, 2007

[HAGE06] Hagen S., „IPv6 Essentials”, O'Reilly, 2006, ISBN: 0-596-10058-2

[IANN04] Iannaccone G., Chuah C., Bhattacharyya S., Diot C., “Feasibility of IP restoration in a Tier-1 backbone,” IEEE Network Mag., Ian–Feb. 2004

[LABO08] Labovitz C., Malan Robert G., Jahanian F., „Internet Routing Instability”, IEEE/ACM TRANSACTIONS ON NETWORKING, VOL. 6, NO. 5, OCTOBER 1998

[LI05] Yuxi Li , Harms J., Holte R., „A simple method for balancing network utilization and quality of routing", Proceedings of ICCCN, San Diego, CA, 2005

[MARK04] Markopoulou A., Iannaccone G., Bhattacharyya S., Chuah C.-N., Diot C., "Characterization of failures in an IP backbone," Proceedinngs of IEEE INFOCOM, Hong Kong, 2004

[PETE12] Peterson Larry L., Davie Bruce S., "Computer Networks a system approach" Fifth Edition , Morgan Kaufmann, 2012, ISBN: 978-0-12-385059-1.

[RFC 2463] Conta A., Deering S., „Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", Request for comments 2463, Internet Engineering Task Force, 1998

[RFC 5286]Atlas A, Zinin A., „Basic specification for IP fast-reroute: loop-free alternates", Request for comments 5286, Internet Engineering Task Force, 2008

[SHAI02] Shaikh A., Isett C., Greenberg A., Roughan M., Gottlieb J., „A Case Study of OSPF Behavior in a Large Enterprise Network", Proc. ACM SIGCOMM Internet Measurement Workshop (IMW , 2002)

[TEIX06] Teixeira R., Rexford J., "Managing routing disruptions in Internet Service Provider networks," IEEE Communications Magazine, Volumul 44, pp 160 - 165 , 2006.

[WATS03] Watson D., Jahanian F., Labovitz C., „Experiences with Monitoring OSPF on a Regional Service Provider Network", Proceedings of International Conference on Distributed Computing Systems, pp. 204–213, May 2003