



(12)

CERERE DE BREVET DE INVENȚIE

(21) Nr. cerere: **a 2010 01285**

(22) Data de depozit: **08.12.2010**

(41) Data publicării cererii:
30.07.2012 BOPI nr. 7/2012

(71) Solicitant:
• **MARCTEL S.I.T. S.R.L.**,
STR. DR. LOUIS PASTEUR NR. 8,
SECTOR 5, BUCUREȘTI, B, RO

(72) Inventatori:
• **INVENTATORI NEDECLARAȚI, *, RO**

(54) **OPTIMIZAREA TRAFICULUI IP BAZATĂ PE ALGORITMI
GENETICI PENTRU REDIRECȚIONAREA FLUXURILOR
DE DATE**

(57) Rezumat:

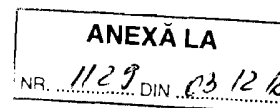
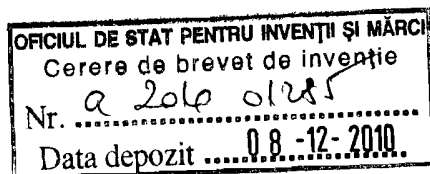
Invenția se referă la o metodă de optimizare a folosirii resurselor rețelelor de date prin decizii bazate pe algoritmi genetici, și permite o îmbunătățire a distribuției de încărcare a traficului în rețea. Metoda conform invenției constă în clasificarea datelor de intrare în fluxuri, în identificarea punctelor de optimizare a fluxurilor și, în final, în determinarea unor politici de

routare în conformitate cu noile modele de trafic, iar principalul efect este evitarea retransmisiilor datorate abandonării de date la nivelul interfețelor aflate la limita superioară de încărcare.

Revendicări: 5

Cu începere de la data publicării cererii de brevet, cererea asigură, în mod provizoriu, solicitantului, protecția conferită potrivit dispozițiilor art. 32 din Legea nr. 64/1991, cu excepția cazurilor în care cererea de brevet de invenție a fost respinsă, retrasă sau considerată ca fiind retrasă. Întinderea protecției conferite de cererea de brevet de invenție este determinată de revendicările conținute în cererea publicată în conformitate cu art. 23 alin. (1) - (3).





Descriere

Invenția „Optimizarea traficului IP bazată pe algoritmi genetici pentru redirectionarea fluxurilor de date” se referă la o metodă de optimizare a folosirii resurselor rețelelor de date prin decizii bazate pe algoritmi genetici. Invenția permite o îmbunătățire a distribuției de încărcare a traficului în rețea.

În prezent comutatoarele la nivel IP din rețelele de date (routere) acceptă trafic incident pe intrările interfețelor cu care sunt dotate și îl stochează în cozi, pentru a-l distribui către ieșiri, conform unor tabele de rutare. Cu excepția categoriilor privilegiate de trafic (determinate pe baza regulilor grupate în QoS – quality of service, calitatea serviciilor), dacă este depășită capacitatea cozilor, datorită limitărilor de bandă pe ieșirile interfețelor, volumul de trafic în cauză este abandonat (ignorat), urmând ca retransmisia porțiunilor pierdute să revină celor doi corespondenți implicați în trafic.

În cazul rețelelor cu administrare unică, la nivelul routerelor există posibilitatea modificării deciziilor de rutare luate de protocoalele dinamice de rutare, pentru redirectionarea fluxurilor de date prin secțiuni mai puțin utilizate ale rețelei. Metoda propusă se folosește de acest mecanism și, pe baza prelucrării informațiilor privind traficul din rețea, propune scheme de rutare alternative.

Așa cum se știe, un flux (cunoscut și ca flux de trafic sau flux de date) este o grupare de trafic (aplicație, protocol) ce are atribute asemănătoare precum adresa sursă/destinație, tip de informații, directive și alte informații. Fluxurile sunt caracterizate de informațiile despre capete, adică perechile sursă-destinație de la mai multe niveluri. Întrucât sunt identificabile prin capete, ele pot fi legate de o aplicație sau utilizator. Unui flux i se poate examina drumul legătură cu legătură sau rețea cu rețea. Acest mod de lucru este folositor când trebuie calculate necesitățile unui flux în rețele sau la nivel de rețea.

Un flux implică mai multe necesități de performanță: capacitate (bandă), întârziere (latență), fiabilitate (disponibilitate) și nivel de calitate al serviciilor. Fluxurile sunt folosite pentru a combina nivelul de performanță, necesitate și servicii cu informații despre locație pentru a înțelege unde este nevoie de acestea în rețea. Prin analiza de fluxuri se creează perspectiva traficului capăt la capăt ce arată unde este nevoie de îmbunătățire.

Un flux individual apare pentru o singură sesiune a unei aplicații și este unitatea de bază a traficului. Un flux compus este o combinație de necesități din partea mai multor aplicații sau fluxuri individuale ce împart aceeași legătură din rețea. Majoritatea fluxurilor din rețea sunt compuse.

Fluxurile prezintă o perspectivă diferită a traficului din rețea pentru că prezintă distribuția acestuia prin asocierea cu aplicații. Din acest motiv fluxurile au o importanță crescută pentru procesul de analiză, concepție și optimizare a rețelelor.

Metoda de optimizare inventată ce rezolvă problema alocării fluxurilor multiple într-o rețea cu constrângeri se bazează pe algoritmi genetici. Aceasta problemă este de tip NP-complet, deci se rezolvă în timp nepolinomial, iar prin folosirea algoritmilor genetici se pot obține soluții bune într-un timp foarte scurt.

Metoda de optimizare a traficului are ca date de intrare căile urmate de fluxurile din rețea **1** și încărcarea generată de acestea **2**, pentru momentul în care se vrea optimizarea. Algoritmul cunoaște întreaga configurație a rețelei **3**, constând în informații despre noduri **4** și legăturile dintre acestea **5**.

În urma optimizării unele fluxuri(**1**) vor face legătura între nodul sursă **6** și nodul destinație **7** folosind alte drumuri disponibile în rețea, fără a se modifica fizic rețeaua inițială.

Se urmărește obținerea unui grad de încărcare uniform, prin încărcarea echilibrată a legăturilor de rețea existente, dar și prin oferirea unei calități pentru serviciile clienților în funcție de necesitățile acestora.

Optimizarea urmărește ca încărcarea rețelei în aceleași condiții de trafic să nu mai depășească un prag considerat critic **8**. Încărcarea pe legăturile din rețea va fi mai uniformă și nu vor mai exista interfețe supraîncărcate sau interfețe puțin folosite.

Invenția încearcă să administreze mai bine situațiile de congestie ce apar în condiții de trafic similar. Traficul similar presupune că scenariile în care apar congestii sunt asemănătoare, adică sunt perechi de fluxuri ce apar în același timp și încarcă legăturile din rețea.

Metoda de optimizare inventată urmărește micșorarea încărcării maxime de pe legăturile din rețea. Traficul în exces din anumite noduri ce poate fi redirecționat prin alte noduri pentru a aduce rețeaua într-o stare în care încărcarea este rezonabilă pentru toate legăturile din rețea. Din acest motiv funcția de fitness pentru algoritmul genetic este maximul încărcării din toate

legăturile din rețea. Funcția calculează încărcarea totală maximă **9** pentru muchiile din rețea pe baza reprezentării unui cromozom **10** și încărcările asociate drumurilor **11** din cromozom.

Încărcarea asociată unui drum (**11**) este adunată la valoarea încărcării pentru fiecare muchie din drum **12**, iar acest lucru se repetă pentru toate drumurile unui cromozom primit ca parametru de funcție.

Un individ al populației **13** algoritmului genetic trebuie să poată reține drumul pentru fiecare flux ce transferă date în momentul de optimizare.

Un cromozom al populației **14** este reprezentat printr-o lista de drumuri. Un drum **15** este o înșiruire de numere ce reprezintă ID-urile routerelor prin care trece un flux de informații de la sursa la destinație. Toți indivizii unei populații(**13**) vor avea același număr de drumuri, iar acestea pot avea lungimi diferite. Drumurile dintre perechile sursă-destinație vor fi în aceeași ordine în cromozomii populației.

În algoritmul genetic de optimizare se folosește o populație de 100 indivizi la fiecare generație. Între indivizii din populație au loc încrucișări **16** cu probabilitate de 40% și mutații **17** cu probabilitate de 20% pe perioada a 700 de generații.

Încrucișarea se face cu doi indivizi **18** aleși aleatoriu, cu un punct de intersecție. Astfel, se creează un individ nou ce are o parte din fluxuri de la un părinte și o parte de la al doilea părinte.

Mutația **19** unui individ se face pe un flux al acestuia ales aleatoriu. Acest flux este transformat între două noduri ce îi aparțin. Se încearcă ca aceste două noduri să se unească prin muchii ce au încărcarea cât mai mică.

Operația de mutație se folosește de un flux, încărcarea acestuia și încărcarea tuturor muchiilor din rețea. Selecția nodurilor între care se face mutația este aleatoare și acestea formează un interval de muchii între capetele drumului. Muchiile ce se află între cele două noduri alese pentru mutație sunt adăugate într-un vector cu muchii interzise **20**, iar costul lor este scăzut din matricea **21** ce conține încărcarea totală a legăturilor din rețea. Nodurile drumului ce nu se află între cele două noduri sunt adăugate într-un vectorul cu noduri interzise **22**. Toate muchiile **23** din matricea cu încărcările legăturilor din rețea ce nu se află printre muchiile interzise sunt ordonate crescător în funcție de cost.

Se folosește muchia **24** cu cea mai mică încărcare ce are un capăt ce nu a fost adăugat în arbore. Se selectează o muchie cu cost cât mai mic și este testată dacă poate fi conectată la arborele ce se construiește. Se parcurg toate nodurile arborelui iar dacă muchia are nodul stâng în arbore și nodul din dreapta nu este în arbore și poate fi ales (nu este printre nodurile interzise) adaug nodul drept în arbore și creez legătura cu nodul stâng. Dacă nodul din dreapta este nodul final atunci ies din crearea arborelui. În mod similar se procedează și dacă considerăm nodul din dreapta legătura pentru noua muchie.

Muchiile folosite nu se regăsesc în drumul inițial dintre cele două noduri, iar algoritmul se oprește atunci când muchia aleasă **25** conține nodul al doilea al mutației. În acest fel se construiește un drum **26** ce diferă de cel inițial, cu proprietatea că muchiile au costul cel mai mic. Aceasta proprietate ajută la diversificarea mulțimii de indivizi din algoritmul genetic. Întrucât muchiile sunt salvate sub formă de matrice de adiacență, complexitatea este $O(V \cdot M)$ și depinde proporțional de V = numărul de vârfuri și M = numărul de muchii.

Arborele de acoperire **27** cu cost minim este stocat folosind două liste. Lista *parcursereNod* **28** salvează nodurile adăugate în arborele de acoperire a grafului, iar în lista *parcurserePrec* **29** se salvează poziția din lista *parcursereNod* cu care se leagă nodul. Stocarea în acest mod permite folosirea algoritmului de parcurgere în lățime pentru unirea celor două noduri. Folosind o variantă modificată a algoritmului Prim și parcurgerea în lățime avem siguranța că cele două noduri inițiale (sursă-destinație) sunt unite prin muchii ce au încărcarea cea mai mică posibilă și cel mai mic număr de noduri în drum.

Scopul practic este de a implementa o aplicație pentru echilibrarea consumului de resurse în rețea. Acest lucru se realizează prin clasificarea datelor în fluxuri, identificarea punctelor de optimizare a fluxurilor și, în final, determinarea unor politici de rutare în conformitate cu noile modele de trafic.

Se poate crea astfel un sistem de analiză și optimizare ce privește alocarea acelor fluxuri de date ce se înscriu în modelele optimizate în prealabil. Dificultatea optimizării complete a rutării este datorată imposibilității de control în timp real a deciziilor de rutare. În urma optimizării traficului folosind metoda inventată bazată pe algoritmi genetici rezultă alocarea eficientă a fluxurilor de date și o mai eficientă utilizare a resurselor de rețea prin balansarea traficului. Rezultatele optimizării cresc performanța rețelei atât ca volum de date transferate pe unitatea

de timp (throughput) cât și ca latență. Sistemul ce se bazează pe metoda inventată este benefic pentru funcționarea unei rețele pentru că ajută administratorul de rețea să reducă aglomerarea și, eventual, inaccesibilitatea rețelei, care afectează nivelul serviciilor oferite utilizatorilor.

Revendicări

Metodă de optimizare bazată pe algoritmi genetici ce rezolvă problema alocării fluxurilor multiple într-o rețea cu constrângeri, **caracterizată prin aceea că** are ca date de intrare un moment temporal descris prin căile urmate de fluxurile din rețea (1) și încărcarea generată de acestea (2), cunoaște întreaga configurație a rețelei și produce o nouă alocare a drumului pentru unele fluxuri (1) pentru a face legătura dintre nodurile sursă (6) și nodul destinație (7) cu scopul de a obține un grad de încărcare uniform, prin încărcarea echilibrată a legăturilor de rețea existente.

Aplicare a algoritmilor genetici, **caracterizată prin aceea că** funcția de fitness pentru algoritmul genetic este maximul încărcării din toate legăturile din rețea, funcția calculează încărcarea totală maximă (9) pentru muchiile din rețea pe baza reprezentării unui cromozom (10) și încărcările asociate drumurilor (11) din cromozom.

Aplicare a algoritmilor genetici, **caracterizată prin aceea că** un individ al populației algoritmului genetic trebuie să poată reține drumul pentru fiecare flux ce transferă date în momentul de optimizare, adică este reprezentat printr-o listă de drumuri, iar un drum (15) este o înșiruire de numere ce reprezintă ID-urile routerelor prin care trece un flux de informații de la sursa la destinație.

Mutația unui individ se face pe un flux al acestuia ales aleatoriu, iar acesta este transformat între două noduri ce îi aparțin, **caracterizate prin aceea că** se încearcă unirea acestor două noduri prin muchii ce au încărcarea cât mai mică.

Se folosește o variantă modificată a algoritmului Prim și parcurgerea în lățime, **caracterizată prin aceea că** cele două noduri inițiale (sursă-destinație) sunt unite prin muchii ce au încărcarea cea mai mică posibilă și cel mai mic număr de noduri în drum, se selectează o muchie cu cost cât mai mic și este testată dacă poate fi conectată la arborele ce se construiește, se parcurg toate nodurile arborelui iar, dacă muchia are nodul stâng în arbore și nodul din dreapta nu este în arbore și poate fi ales (nu este printre nodurile interzise), nodul drept este adăugat în arbore și este creată legătura cu nodul stâng, iar dacă nodul din dreapta este nodul final, se iese din crearea arborelui; din arborele generat se reconstruiește drumul sursă-destinație.