

Soluții de optimizare a traficului într-o rețea folosind tehnologii Linux

Răzvan RADU, Gheorghe RADU

Abstract—This article presents some ways of optimizing network traffic by using Open Source tools.

Index Terms—Linux, internet provider, network clients, Open Source, traffic management

I. INTRODUCERE

DEZVOLTAREA explozivă a așa numitelor “rețele de cartier” apare ca un răspuns firesc la nevoia generală a populației de un serviciu de acces la Internet cu bandă largă, ieftin și rapid, rețelele de cartier devenind în scurt timp furnizori de Internet cu câteva sute sau chiar mii de clienți fiecare. Aceste mici afaceri au ca obiect de activitate un domeniu tehnologic complex, în care se utilizează unelte specifice, se lucrează în urma unei instruirii și a unei experiențe corespunzătoare. În mod evident, o rețea de cartier, construită în majoritatea cazurilor de studenți sau chiar de către elevi de liceu, nu beneficiază nicidecum de o implementare în cunoștință de cauză, de experiența și procedurile folosite în rețelele de telecomunicații și cu atât mai puțin de uneltele specifice acestui domeniu, echipamente electronice ce implică costuri mai mari, în unele cazuri, decât valoarea întregii rețele. În consecință, apare nevoia gestionării traficului rețelei prin tehnologii software cu costuri minime și eficiență maximă.

II. DESCRIERE

În contextul gestionării unei rețele de calculatoare cu acces la internet funcțiile principale sunt îndeplinite de un *router*, hardware sau software, în cazul nostru software. Mai exact, un PC cu arhitectura server ce rulează un sistem de operare Linux și toate uneltele necesare managementului de trafic.

Pentru o rețea de distribuție Internet se evidențiază următoarele categorii de cerințe, din punct de vedere al traficului:

- permiterea/restricționarea accesului diferitelor puncte de lucru, numite în continuare clienți ai rețelei sau, simplu, clienți;
- împărțirea lățimii de bandă existente în mod echitabil tuturor clienților rețelei;
- monitorizarea traficului realizat de fiecare client și de întreaga rețea.

Ne vom ocupa pe rând de fiecare din aceste secțiuni.

A. Gestionarea accesului la rețea

Traficul fiecărui client către Internet și dinspre Internet trece prin *router*-ul rețelei. Avem astfel posibilitatea aici de a decide asupra aceluiași trafic. Ca identifikatori unici ai unui client există adresa MAC a plăcii de rețea și adresa IP. Utilizând această pereche creăm reguli de acces în *router*-ul Linux care vor permite sau vor restricționa traficul clienților. Implementarea se face cu ajutorul regulilor *iptables*. Când rețeaua are câteva sute sau chiar mii de clienți vom avea tot atâtea reguli de acces active în *router*. Fiecare pachet de date care traversează *router*-ul va parcurge secvențial această listă de reguli pentru a vedea dacă poate trece sau nu. La câteva milioane de pachete zilnic transferate între rețeaua locală și Internet, parcurgerea secvențială a listei de reguli de acces poate crea mari întârzieri, astfel încât o optimizare este strict necesară. Se folosește în acest scop o unealtă special creată, numită *ipset*. Un *ipset* este un set de perechi MAC – IP împreună cu un mecanism de verificare rapidă a pachetelor ce îl traversează contra tuturor perechilor existente în set. Folosind *ipset* vom avea în implementarea noastră o singură regulă de acces *iptables* aplicată pe setul de perechi IP – MAC al întregii rețele.

B. Managementul lățimii de bandă

Managementul lățimii de bandă, împărțirea benzii existente echitabil între clienții rețelei, alocarea lățimii de bandă dorite fiecărui client, optimizarea consumului, redundanța benzii de Internet reprezintă problema fundamentală în gestionarea traficului unei rețele. Este aspectul care influențează cel mai mult eficiența rețelei, calitatea traficului/serviciului oferit, costurile financiare și de altă natură. Un progres substanțial în această privință l-a reprezentat dezvoltarea *shaper*-ului HTB, *Hierarchical Token Bucket*. Această politică de *shaping* este organizată pe clase de limitari. Se creează o clasă, o „găleată” pentru întreaga lățime de bandă și câte o subclasă pentru fiecare client al rețelei, subclasă ce va avea o limită inferioară și o limită superioară exprimate în bytes pe secundă.

Primul rol al *shaper*-ului este să asigure banda minimă garantată fiecărei clase. Unicitatea metodei HTB constă în faptul că permite redistribuirea benzii nefolosite de unele subclase celorlalte subclase. Astfel fiecare subclasă primește tot timpul minimul garantat de bandă iar banda extra, nefolosită, se distribuie subclaselor care au nevoie, în limita superioară stabilită, proporțional, în funcție de limita inferioară a fiecărei clase.

Ca **exemplu**, să presupunem că avem 3 subclase A, B, C și clasa mamă X. Clasa mamă are disponibilă o lățime de bandă de 10 MB, clasa A are un minim de 2 MB și un maxim de 5 MB, clasa B are un minim de 3 MB și un maxim de 7 MB iar clasa C are un minim de 5 MB și un maxim de 10 MB. Să presupunem că clientul aferent clasei B nu face trafic, prin urmare: se asigură minimul de 2 MB clasei A, minimul de 5 MB clasei C și rezultă un surplus de bandă disponibilă de 3 MB. Aceasta se va distribui proporțional cu minimele claselor A și C astfel: clasa A va primi 0.85 MB și clasa C va primi restul de 2.15 MB din surplusul de 3 MB.

Astfel, din punct de vedere structural mecanismul HTB satisface cu succes nevoile de management al benzii. Din punct de vedere funcțional însă întâmpinăm aceleași provocări ca și în cazul regulilor de acces, fiecare pachet de date care circulă prin router va traversa secvențial întreaga listă de clase pentru a vedea căreia îi aparține în funcție de adresa IP sursă, respectiv destinație. În cazul unei rețele cu 1000 de clienți, deci 1000 de subclase de limitări, orice pachet destinat celei de a 1000-a clase din listă va parcurge de fiecare dată toate cele 999 de clase până va intra în clasa de care aparține și va putea trece astfel mai departe, fapt ce poate crea mari întârzieri de trafic în rețea.

Se folosește, în acest sens, un mecanism de *hashing*. Prin intermediul acestui mecanism, crieriile de selecție ale pachetelor și trimiterea lor către clase sunt organizate într-un alt mod decât secvențial, permițând găsirea unei potriviri între pachet și clasa corespunzătoare din maxim 3 încercări.

C. Monitorizarea traficului

Nevoia de monitorizare a traficului fiecărui client, al rețelei per total, cunoașterea tipului de trafic care se face, a orelor din zi sau perioadelor din săptămână sau din lună când se face mai mult trafic, toate acestea sunt informații fundamentale pentru luarea deciziilor și constuirea strategiilor pe termen scurt și mediu pentru dezvoltarea rețelei. Fără o monitorizare cât mai detaliată a ceea ce se întâmplă efectiv în rețea, a urmărilor măsurilor pe care le luăm, nu putem eficientiza funcționarea rețelei.

În privința monitorizării traficului într-o rețea se evidențiază următoarele aspecte/nevoi:

- viteza cu care s-a făcut trafic într-un anumit moment și statistici;
- cantitatea de date consumată într-o anumită perioadă de timp;
- evidențiere completă, instant, a traficului care se realizează în orice moment;

- tipul de trafic: extern, metropolitan, upload, download, web, ș.a.m.d.

Clasele de limitări împreună cu traficul care le traversează ne furnizează suficiente informații pentru a putea construi analizele necesare.

Pentru statisticile de viteză, cantitatea de trafic și tipurile de trafic se folosește *RRD Tool*, o unealtă ce permite transpunerea facilă a unor date statistice într-o formă grafică foarte ușor de urmărit și înțeles.

Pentru urmărirea traficului instant, live, în orice moment se folosește *lqdump* ce ordonează clasele în ordinea celor mai mari consumatori și are o rată de refresh de o secundă.

Toate aceste tehnologii, împreună cu multe alte facilități și funcționalități, atât tehnice cât și economice, sunt integrate și disponibile într-un produs software comercial, *Murphy Manager de Rețele*, un software complet de gestiune a unei rețele de distribuție de Internet cu până la 5000 de utilizatori, ce se bucură de un real succes, fiind primul jucător pe piața soft-ului pentru rețele de cartier.

III. CONCLUZII

În domeniul rețelisticii, un domeniu extrem de dinamic și provocator, cerințele cresc cu o viteză extraordinară, noi și noi provocări sunt lansate către profesioniștii acestui domeniu în fiecare zi. Pe lângă soluțiile profesionale, dar și extrem de costisitoare, oferite de cele mai mari companii din domeniu, există o zonă paralelă cel puțin la fel de dinamică și de hotărâtă să țină pasul, care se regăsește în lumea Linux a software-ului pentru rețele, în lumea *Open Source*. Cele prezentate sunt doar un exemplu despre ceea ce se poate crea, pornind de la niște unelte în totalitate gratuite, și cu un rezultat foarte bun, prin mobilizarea energiilor și creativității individuale.

WEBOGRAFIE

- [1] www.solutiiretele.ro
- [2] <http://luxik.cdi.cz/~devik/qos/htb/>
- [3] <http://oss.oetiker.ch/rdrtool/>
- [4] <http://www.linux360.ro>
- [5] <http://www.netfilter.org>