OPTIMIZATION OF CROSSPRODUCT-BASED CLASSIFICATION ALGORITHMS

Michal Kajan
Master Degree Programme (2), FIT BUT
E-mail: xkajan01@stud.fit.vutbr.cz

Supervised by: Viktor Puš
E-mail: ipus@fit.vutbr.cz

ABSTRACT

This paper deals with packet classification algorithms and their possible optimizations. It is aimed on algorithms based on crossproducting of field labels providing good results with the eventual hardware implementation. Main problem for these methods remains amount of used memory. In this paper several techniques providing a certain level of optimization are proposed.

1 ÚVOD

Rozvoj a rozširovanie počítačových sietí a Internetu dosiahol do súčasnej doby takej úrovne, že sa stal ideálnou platformou pre poskytovanie množstva rozmanitých služieb prístupných veľkému počtu pripojených užívateľov. Spolu s ich vzrastajúcim počtom a s rastúcimi požiadavkami na prenosové rýchlosti sa do popredia dostáva otázka zabezpečenia sieťovej prevádzky. Problematika klasifikácie paketov sa môže týkať oddelenia sieťovej prevádzky generovanej rôznymi užívateľmi, zabránenie neautorizovaného prístupu či odrazenie útokov k určitým segmentom danej siete, prispôsobovanie dostupnej šírky pásma a iné. Prenos dát na vysokorýchlostných sieťach kladie určité nároky aj na klasifikačné algoritmy. Algoritmy implementované na programovej úrovni nedokážu nároky na rýchlosť uspokojiť, preto sa pozornosť čoraz viac upriamuje na obvodové implementácie najčastejšie s využitím technológie FPGA.

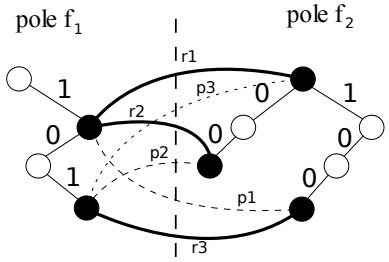
2 KLASIFIKÁCIA PAKETOV

Klasifikátor reprezentuje konečnú množinu pravidiel, vstupom klasifikácie sú pakety a pravidlá s určenou prioritou. Výstupom je určenie vyhovujúceho pravidla pre daný paket. Klasifikácia prebieha na základe zvolených položiek hlavičky paketu, tieto sa následne porovnávajú s jednotlivými položkami pravidla. Položka pravidla je definovaná hodnotou, tá vlastne reprezentuje príslušnú podmienku pre porovnanie. Podmienku je možné reprezentovať niekoľkými spôsobmi: vo forme presne definovanej hodnoty (pre definovanie typu protokolu) či ľubovoľnej hodnoty, potom vo forme prefixu (pre definovanie podsiete) či vo forme rozsahu (pre definovanie rozsahu hodnôt portov). Všetky tieto možnosti je možné previesť do prefixovej formy, čím sa dosiahne jednotný spôsob spracovania hodnôt. Po určení vyhovujúceho pravidla sa vykoná akcia, ktorá je k príslušnému pravidlu priradená — zahodenie paketu či jeho preposlanie na určité rozhranie.

3 ALGORITMY ZALOŽENÉ KARTÉZSKOM SÚČINE POLÍ

Tieto algoritmy patria do skupiny algoritmov založených na dekompozícii problému, pri ktorých klasifikácia prebieha v dvoch základných krokoch. V prvom dôjde k určeniu najdlhšieho zhodného prefixu (*LPM — Longest Prefix Match*) pre každé uvažované políčko hlavičky paketu zo sady pravidiel. Jedná sa vlastne o rovnakú operáciu ako pri vyhľadávaní záznamov v smerovacej tabuľke. V druhom kroku nasleduje určenie správneho pravidla.

Jednoduchý algoritmus kartézskeho súčinu je založený na rozdelení sady pravidiel do stĺpcov. Každý takýto stĺpec obsahuje všetky odlišné prefixy v danom políčku. Výsledky z LPM vyhľadávacej fázy sa skombinujú a vytvorí sa vyhľadávací kľúč zložený z jednotlivých najdlhších zhodných prefixov jednotlivých položiek hlavičky. Kľúč sa používa ako vstup do hash tabuľky, ktorá reprezentuje tabuľku kartézskych súčinov. Tento algoritmus však ukladá všetky možné kombinácie výsledkov LPM fáze, s čím súvisí zavedenie pojmu *pseudopravidiel*. Tie predstavujú špecifické prípady pôvodných pravidiel a je nutné ich do sady doplniť. Problém je znázornený na obrázku 1 a v tabuľkách 1, 2.



Obrázok 1: Pravidlá a pseudopravidlá, klasifikácia podľa dvoch políčok

pravidlo	**pole** f_1	**pole** f_2
r1	1*	*
r2	1*	00*
r3	101*	100*

Tabuľka 1: Pôvodná sada pravidiel

pseudopravidlo	**pole** f_1	**pole** f_2	**pravidlo**
p1	1*	100*	r1
p2	101*	00*	r2,r1
p3	101*	*	r1

Tabuľka 2: Sada s pseudopravidlami

Obrázok znázorňuje zjednodušený prípad klasifikácie podľa dvoch polí na základe jednoduchej sady pravidiel. Pre každé pole sa vytvorí stromová štruktúra prefixov. Platné prefixy sú v tejto štruktúre vyznačené čiernou farbou a pravidlá hranami medzi uzlami. Do sady je však potrebné doplniť nové pravidlá a asociovať ich s pôvodnými pravidlami, aby bolo možné správne klasifikovať každý prichádzajúci paket [1]. Zväčšenie pôvodnej sady pravidiel je najväčším problémom tejto skupiny algoritmov, nárast môže byť teoreticky exponenciálny vzhľadom na charakter kartézskeho súčinu. Experimentálne výsledky analýzy pravidiel poukazujú na fakt, že pravidiel spôsobujúcich enormný nárast počtu pseudopravidiel je veľmi málo oproti celkovému počtu (1%-2%) [1]. Určité riešenie vychádzajúce z tohto poznatku predstavuje použitie ternárnej asociatívnej pamäte (TCAM), ktorá podporuje priame vyhľadávanie prefixov. Z pôvodnej sady pravidiel by sa "problémové" pravidlá odstránili a umiestnili do TCAM pamäti. Tento prístup je navrhnutý napr. v [2], [1].

Ďalším príkladom z tejto skupiny algoritmov je PHCA (*Perfect Hashing Crossproduct Algorithm*) založený na použití algoritmu perfektnej hash pre nájdenie hashovacej funkcie, ktorá re-

alizuje mapovanie pseudopravidiel na pravidlá. Ten už neukladá všetky možné kombinácie vyhľadania najdlhšieho zhodného prefixu, čím je pamäťovo úspornejší, ale problém s pamäťovou náročnosťou súvisiaci s množstvom pseudopravidiel sa týka aj tohto algoritmu [2].

4 REDUKCIA PAMÄŤOVEJ NÁROČNOSTI

Spôsob výberu vhodných pravidiel do TCAM pamäte nebol ešte dostatočne preskúmaný a publikovaný, je však možné definovať niekoľko základných prístupov:

- Prvou možnosťou je odhad počtu možných pseudopravidiel pre každé pravidlo, ktorý spočíva v nájdení všetkých špecifickejších prefixov v danom políčku. Následne sa určí súčin zo všetkých takto získaných hodnôt a výsledok reprezentuje požadovaný odhad pre dané pravidlo. Táto metóda je veľmi rýchla, pretože počet pseudopravidiel sa neurčuje ich generovaním z pôvodných pravidiel. Takýto odhad však nemusí byť presný, čo môže vyústiť k najmenej uspokojivým výsledkom.
- Ďalšou možnosťou je vygenerovanie všetkých pseudopravidiel z danej sady a výber tých pravidiel, ku ktorým je asociované najväčšie množstvo pseudopravidiel. Takéto generovanie prebehne iba raz, je to však časovo náročnejšie ako odhad v predchádzajúcom spôsobe. Je však presnejšia, ale nie úplne presná a nemusí eliminovať pravidlá spôsobujúce nárast počtu pseudopravidiel spôsobený počtom unikátnych prefixov v jednotlivých políčkach, podľa ktorých sa klasifikuje.
- Ďalšou možnosťou je postupné skúšanie odoberania vždy jedného pravidla, čím nakoniec dôjde k otestovaniu všetkých možností. Tento prístup je však založený na použití hrubej sily, pretože pseudopravidlá je nutné generovať opakovane. Je úplne presná, ale časová zložitosť môže brániť jej praktickému použitiu.

5 ZÁVER

Uvedené prístupy k eliminácii počtu pseudopravidiel ešte vyžadujú vykonanie množstva experimentov, aby bolo možné určiť, ktorá z nich je pre praktické použitie najvýhodnejšia. Naďalej však existuje priestor pre vývoj nových metód, ktoré by prispeli k riešeniu uvedeného problému. Cieľom je totiž nájdenie takej heuristiky, ktorá by riešila predovšetkým problém s množstvom pseudopravidiel týkajúci sa množstva unikátnych prefixov v jednotlivých políčkach a ktorá by dokázala optimalizovať danú sadu pravidiel v prijateľnom čase. Takáto optimalizácia totiž môže byť významná z hľadiska bezpečnosti, kedy je potrebné novú sadu pravidiel spracovať a nasadiť čo najrýchlejšie. Nové prístupy by nakoniec umožnili lepšie využitie tejto skupiny algoritmov aj v praktickom použití.

REFERENCIE

[1] Lockwood, J., Turner, J., Dharmapurikar, S., Song, H.: Fast packet classification using Bloom filters. In: ANCS '06: Proceedings of the 2006 ACM/IEEE symposium on Architecture for networking and communications systems, New York, NY, USA, 2006, s. 61-70

[2] Puš, V.: Klasifikace paketů s využitím technologie FPGA, FIT VUT v Brně, 2008 diplomová práce