FAST PACKET CLASSIFICATION USING PERFECT HASH FUNCTIONS

Viktor Puš
Master Degree Programme (2), FIT BUT
E-mail: xpusvi00@stud.fit.vutbr.cz

Supervised by: Jan Kořenek
E-mail: korenek@fit.vutbr.cz

ABSTRACT

This paper deals with packet classification in computer networks. After the analysis of synthetic and real rulesets, we propose new algorithm suitable for hardware implementation. Unique property of this algorithm is constant time complexity in terms of external memory accesses. We also provide method for reduction of occupied memory space. Expected maximal throughput of the algorithm is 32 Gbit/s in the worst case.

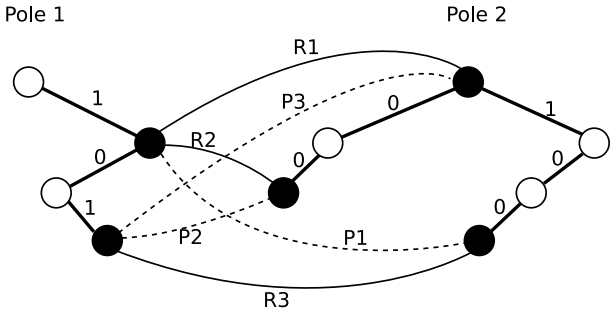
1 ÚVOD

Spolu s rozšiřováním počítačových sítí neustále stoupá potřeba jejich zabezpečení. Proto většina dnešních síťových zařízení musí klasifikovat pakety podle jejich parametrů a následně určit způsob dalšího zpracování. Díky rostoucím přenosovým rychlostem sítí při tom také stoupají nároky na rychlost klasifikačního algoritmu. Současný stav výpočetní techniky neumožňuje provádět klasifikaci paketů dostatečnou rychlostí na obecných procesorech, proto je nutné hledat algoritmy vhodné pro implementaci ve specializovaném hardware. Pro výzkumné účely je výhodné použití technologie FPGA, která je velice flexibilní díky možnost rekonfigurace.
Klasifikace se často provádí podle pěti polí z hlavičky paketu: zdrojová a cílová IP adresa, zdrojový a cílový port a protokol. Potom mluvíme o klasifikaci v pětidimenzionálním diskrétním prostoru. Klasifikační algoritmus má k dispozici množinu pravidel uspořádanou podle priority. Každé pravidlo definuje podmínku pro každou dimenzi. Podmínka může být rozsah (např. rozsah portů), prefix (např. 24 bit adresa sítě, dolních 8 bitů libovolných), přesná hodnota nebo libovolná hodnota. Často se provádí převod rozsahů na prefixy - každý rozsah je možné převést na jeden či více prefixů.
Každé pravidlo také obsahuje akci, kterou je třeba vykonat s daným paketem. V případě firewallu to může být prosté rozhodnutí, zda paket zahodit nebo propustit.

2 SOUČASNÉ PŘÍSTUPY

Dosud nejlepších výsledků dosahují metody založené na dekompozici problému ([1], [2]). V prvním kroku je provedeno vyhledání nejdelšího shodného prefixu (Longest Prefix Match

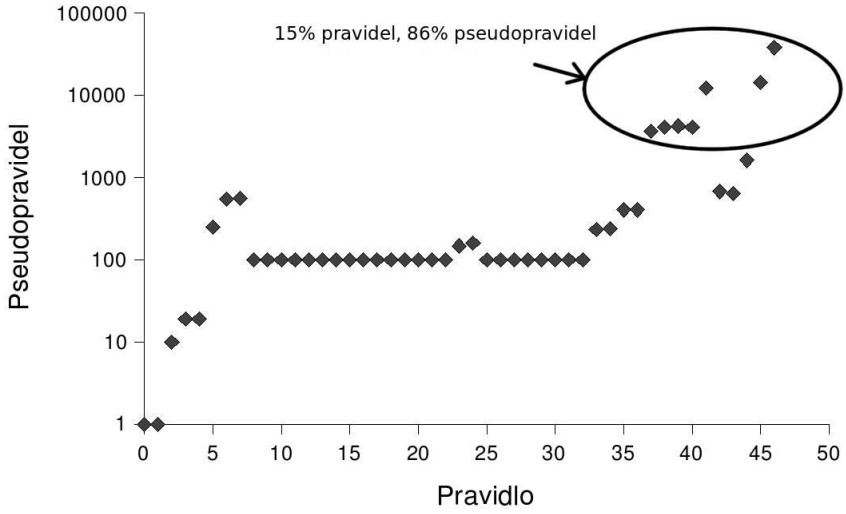
- LPM) odděleně v každé dimenzi. Pro tento proces se často používají stromové algoritmy vycházející z datové struktury trie.
Další krok dekompozičních metod se musí vypořádat s problémem tzv. pseudopravidel. Expanze pseudopravidel má charakter kartézského součinu, takže může generovat velké množství pseudopravidel. Obrázek 1 ilustruje vznik pseudopravidel.



Obrázek 1: Trie pro dvě dimenze a ukázka vzniku pseudopravidel. K pravidlům R1, R2 a R3 je zde nutné připojit pseudopravidla P1, P2, P3, která jsou vždy speciálním případem některého pravidla.

2.1 ANALÝZA VZNIKU PSEUDOPRAVIDEL

Abychom mohli efektivně snižovat počet pseudopravidel, zkoumáme, kolik pseudopravidel může vzniknout z jednotlivých pravidel. Používáme syntetické množiny pravidel vygenerované nástrojem ClassBench [3], ale také několik databází z reálného síťového provozu. Příklad je znázorněn na obrázku 2.



Obrázek 2: Počet pseudopravidel vygenerovaných z jednotlivých pravidel.

Je vidět, že většina pravidel generuje pouze řádově stovky pseudopravidel. Existuje však několik pravidel, kvůli nimž je nutné vygenerovat enormní množství pseudopravidel. Proto přicházíme s myšlenkou oddělení těchto několika pravidel a jejich umístění do malé asociativní paměti (TCAM) přímo na čipu.

3 ALGORITMUS ZALOŽENÝ NA PERFEKTNÍM HASHOVÁNÍ

Jakmile vygenerujeme pravidla a pseudopravidla (s použitím výše zmíněné techniky výrazně omezíme počet pseudopravidel), uložíme je do externí paměti. Nyní musíme řešit problém efektivního přístupu k nim. Nabízí se použití hashovací funkce k získání adresy těchto záznamů. Avšak u hashovací funkce může docházet ke kolizím, jejichž řešení by mohlo zpomalit celý algoritmus.

Proto použijeme techniku nalezení perfektní (bezkolizní) hashovací funkce, popsanou v [4]. Tento algoritmus pracuje na principu konstrukce acyklického neorientovaného grafu, kde hrany odpovídají vstupním slovům, zatímco uzly odpovídají výsledkům dvou různých hashovacích funkcí. Uzly jsou ohodnoceny tak, že jsou-li spojeny hranou, potom jejich součet dá právě požadovaný výsledek, v našem případě číslo pravidla. Díky tomu musíme přistoupit do paměti vždy pouze dvakrát, vyčtené hodnoty sečíst, a získáme tak přímo číslo výsledného pravidla. Protože pravidel není mnoho, mohou být uložena v malé paměti na čipu, kde nakonec provedeme kontrolu, zda paket skutečně odpovídá danému pravidlu.

4 ZÁVĚR

Navrhl jsem nový algoritmus klasifikace paketů, který je zaměřen na dosažení vysoké výkonnosti při hardwarové implementaci. Tomu předcházela detailní analýza generování pseudopravidel v syntetických i reálných množinách pravidel. Pro řádové snížení paměťových nároků je zavedeno použití malé asociativní paměti na čipu. Zcela nová je také idea použití perfektních hashovacích funkcí v oblasti klasifikace paketů. Unikátní vlastností algoritmu je konstantní časová složitost s ohledem na počet přístupů do externí paměti.

V projektu Liberouter [5] se plánuje vytvoření prototypu algoritmu na specializované kartě s FPGA Virtex5 a jeho nasazení v akademické síti CESNET. Při použití jedné paměti typu QDRII na frekvenci 250 MHz lze klasifikovat 62,5 milionů paketů za sekundu. To při minimální velikosti paketu 64 B dává maximální propustnost 32 Gbit/s.

REFERENCE

[1] David E. Taylor, Jonathan S. Turner: Scalable Packet Classification Using Distributed Crossproducting of Field Labels, IEEE INFOCOM 2005

[2] Sarang Dharmapurikar, Haoyu Song, Jonathan Turner, John Lockwood: Fast Packet Classification Using Bloom filters, ANCS 2006

[3] David E. Taylor, Jonathan S. Turner: ClassBench: A Packet Classification Benchmark, IEEE INFOCOM 2005

[4] Zbigniew J. Czech, George Havas, Bohdan S. Majewski: An optimal algorithm for generating minimal perfect hash functions, Information Processing Letters, 1992

[5] Projekt Liberouter, www.liberouter.org