Vyhledání nejdelšího shodného prefixu
Longest Prefix Match

Jiří Tobola
itobola@fit.vutbr.cz

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Osnova

- Úvod
- Motivace
- Současné metody
- Knihovna experimentů
- Plán dalších prací

Úvod - LPM

- Vyhledání nejdelšího shodného prefixu
- Vstup algoritmu: množina prefixů různé délky a jedna konkrétní hodnota
- Výstup algoritmu: prefix ze vstupní množiny, který odpovídá dané hodnotě a je nejdelší, tedy nejspecifičtější

- Příklad:
  - Databáze prefixů: 1*, 0*, 110*, 11*, 10111, 011*
  - Vstupní hodnota: 11011
  - Výstup: 110* (P3)

Motivace

- LPM je úloha, kterou je nutné řešit v každém síťovém zařízení (směrovač, firewall, IDS/IPS, monitorovací sonda)
- Rostoucí přenosové kapacity nových síťových technologií
- Potřeba zpracovat a klasifikovat až 30 miliónů paketů za vteřinu na duplexní 10 GE lince
- Vhodná aplikace pro doplnění platformy pro tvorbu vysokorychlostních aplikací NetCOPE
- Vylepšení klasifikačního algoritmu představeného Viktorem Pušem – Perfect Hashing Crossproduct Algorithm
- Úzce definovaný problém s jasně vymezenými hranicemi a možnostmi srovnání
- Nástup IPv6 přináší nové výzvy a potřeby

6 kroků k dizertaci

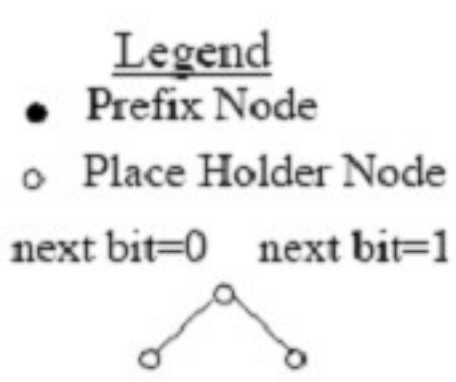
- Krok 1 – studium současného stavu
  - Hotovo (99%)
- Krok 2 – hypotéze a formulace cílů
- Krok 3 – knihovna experimentů
  - Rozpracováno (60%)
- Krok 4 – implementace vlastních metod
  - Návrh metod (50%)
  - Implementace (0%)
- Krok 5 – publikace
  - Na základě experimentů (0%)
- Krok 6 – dizertace
  - Soupis veškerého (0%)

Krok 1: LPM - současné metody

- Algoritmy založené na struktuře Trie
  - Trie
  - Controlled Prefix Expansion
  - Lulea Compressed Tries
  - LC Tries
  - Tree Bitmap
  - Shape Shifting Tree
- Ostatní algoritmy
  - Binární vyhledávání na intervalech
  - Binární vyhledávání na prefixech

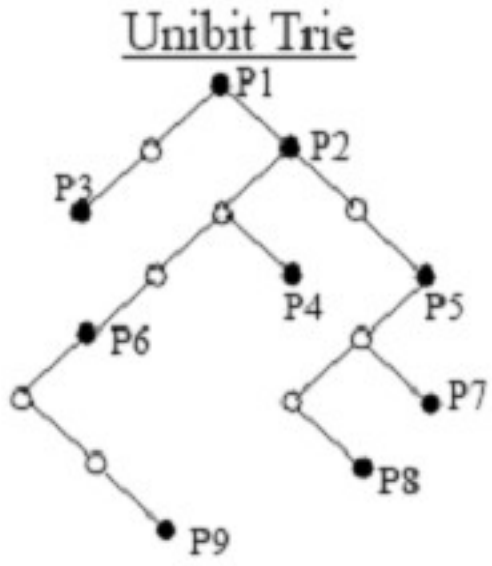
Trie – prefixový strom

- Datová struktura obsahující vyhledávané prefixy přímo ve své konstrukci
- V každém kroku výpočtu zpracován jeden bit a dle něj se postupuje hranou doleva či doprava (0/1), ukončení po zpracování všech bitů nebo při nemožnosti dalšího postupu
- Výhody: jednoduchá dat. struktura, nízké paměťové nároky
- Nevýhody: rychlost



Prefix Database

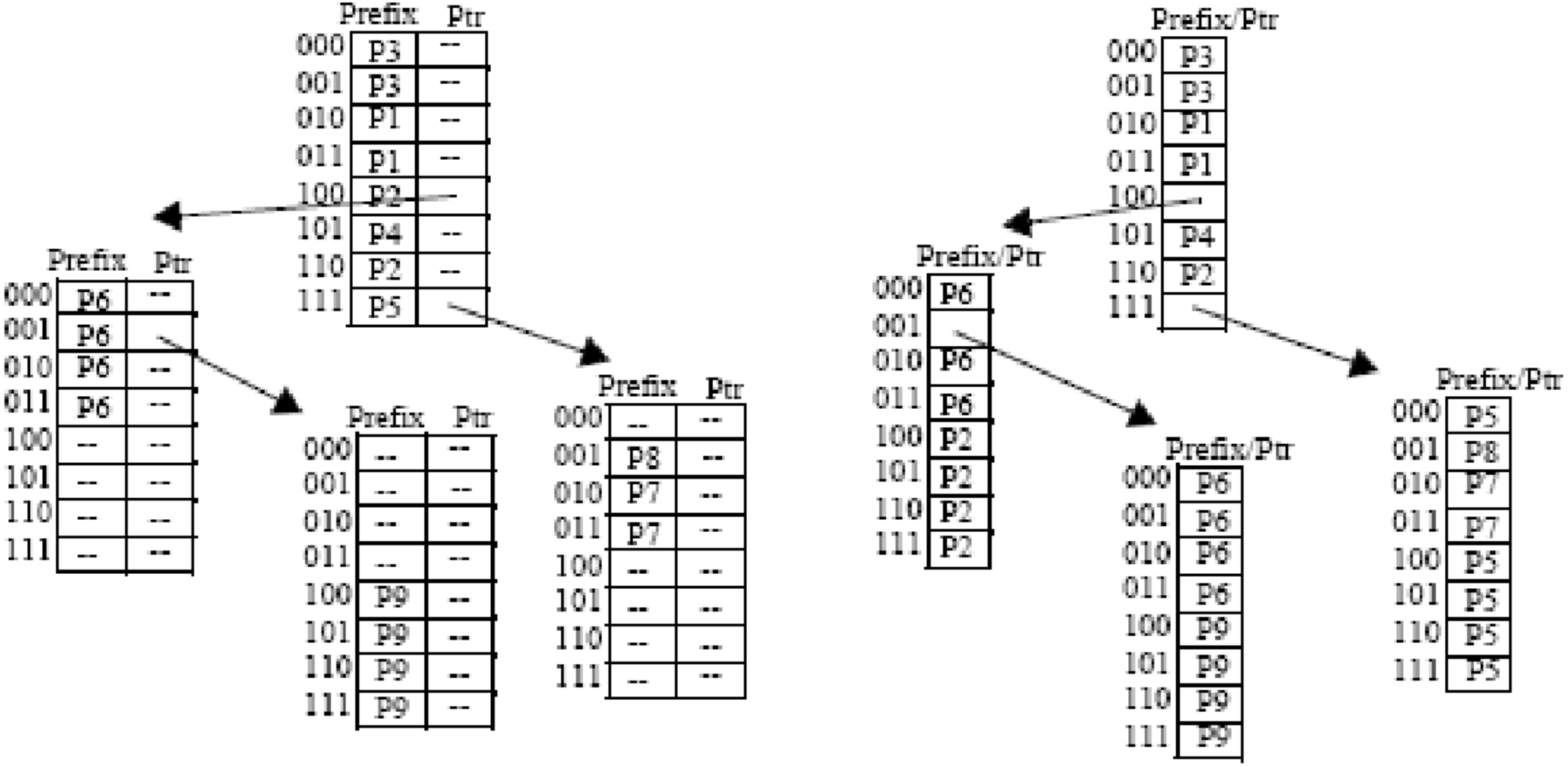
P1	*
P2	1*
P3	00*
P4	101*
P5	111*
P6	1000*
P7	11101*
P8	111001*
P9	1000011*



Controlled Prefix Expansion

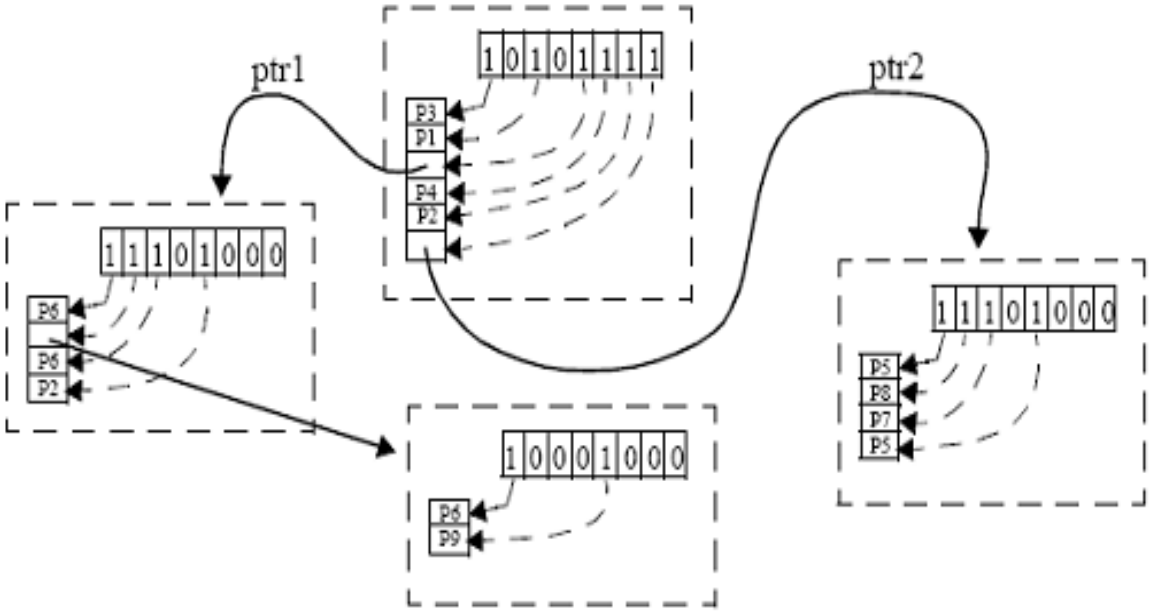
- Trie, která zpracovává více bitů najednou kvůli rychlosti
- Expandování prefixů (zarovnání prefixů na délku dle počtu zpracovávaných bitů v jednom kroku)
- Např. 0* při zpracování tří bitů najednou převedeme na: 000, 001, 010, 011
- V každém uzlu Trie uloženy prefixy a odkaz na následující uzel/podstrom
- Výhody: vyšší rychlost
- Nevýhody: plýtvání pamětí
- Optimalizace – technika „leaf pushing“

Controlled Prefix Expansion



Lulea Compressed Tries

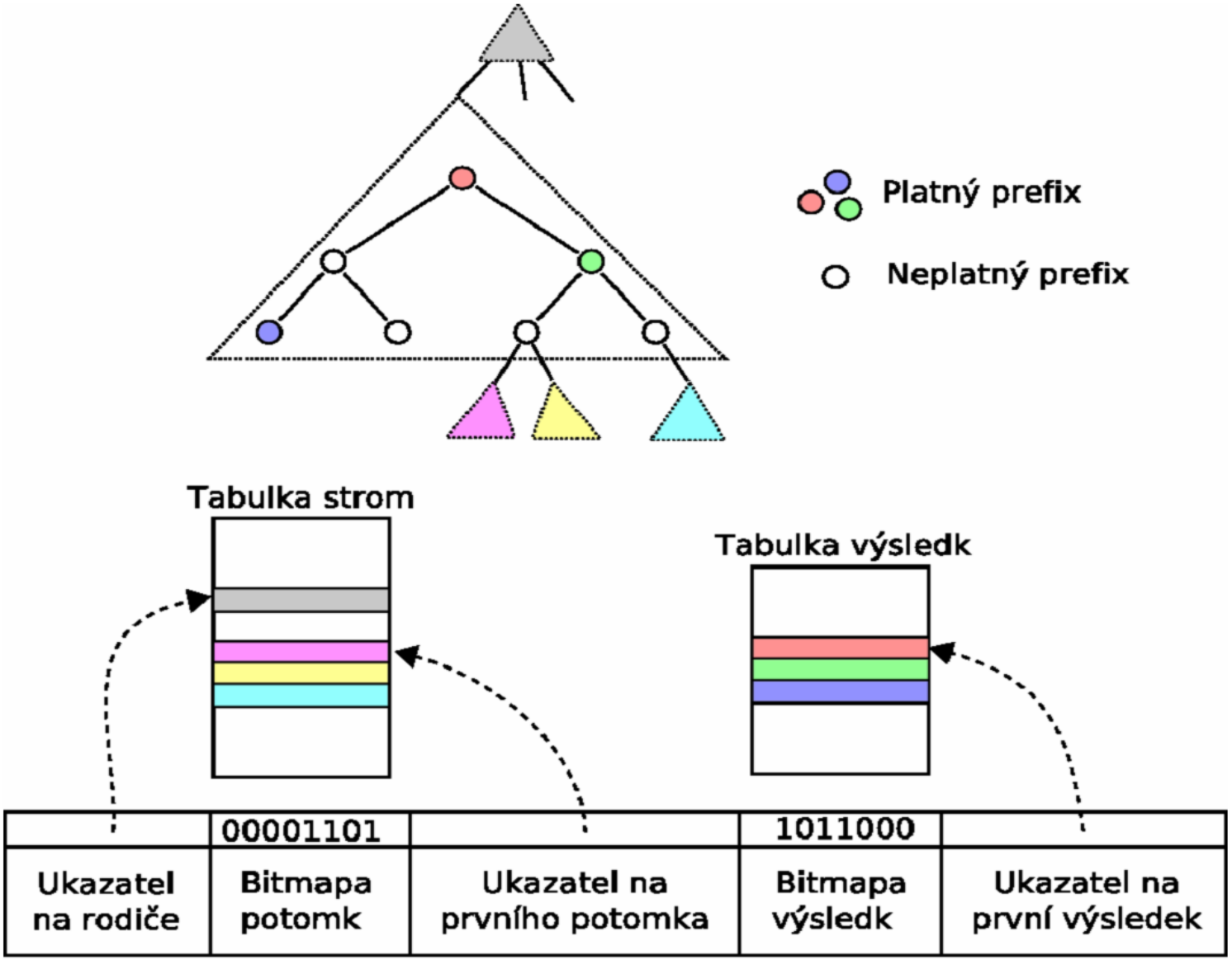
- Základ je metoda CPE a další paměťová optimalizace
- Nahrazení všech elementů se stejnou hodnotou, které jsou v paměti za sebou, jedinou hodnotou
- Využití bitmapy pro indikaci opakujících se hodnot
- Výhody: účinná paměťová komprese
- Nevýhody: pomalá operace přidání nového prefixu



Tree Bitmap

- V současnosti zřejmě nejpoužívanější algoritmus LPM, využívá se v dekompozičních klasifikačních metodách
- Datová struktura Trie, arita uzlu je mocnina dvou, každý uzel odpovídá binárnímu podstromu
- Následníci každého uzlu jsou uloženi v paměti za sebou a vynechány jsou ty podstromy neobsahující žádný prefix. Platnost podstromů/následníků je potvrzována bitmapou.
- Stejně jako následníci jsou uloženy platné prefixy uzlu v paměti za sebou a jejich platnost je potvrzována bitmapou
- Výhody: paměťová nenáročnost, snadná HW realizace, parametrizace arity uzlů (počtu zpracovávaných bitů za takt)

Tree Bitmap


Platný prefix
Neplatný prefix
Tabulka strom
Tabulka výsledek
00001101
1011000
Ukazatel na rodiče
Bitmapa potomk
Ukazatel na prvního potomka
Bitmapa výsledk
Ukazatel na první výsledek

Shape Shifting Tree

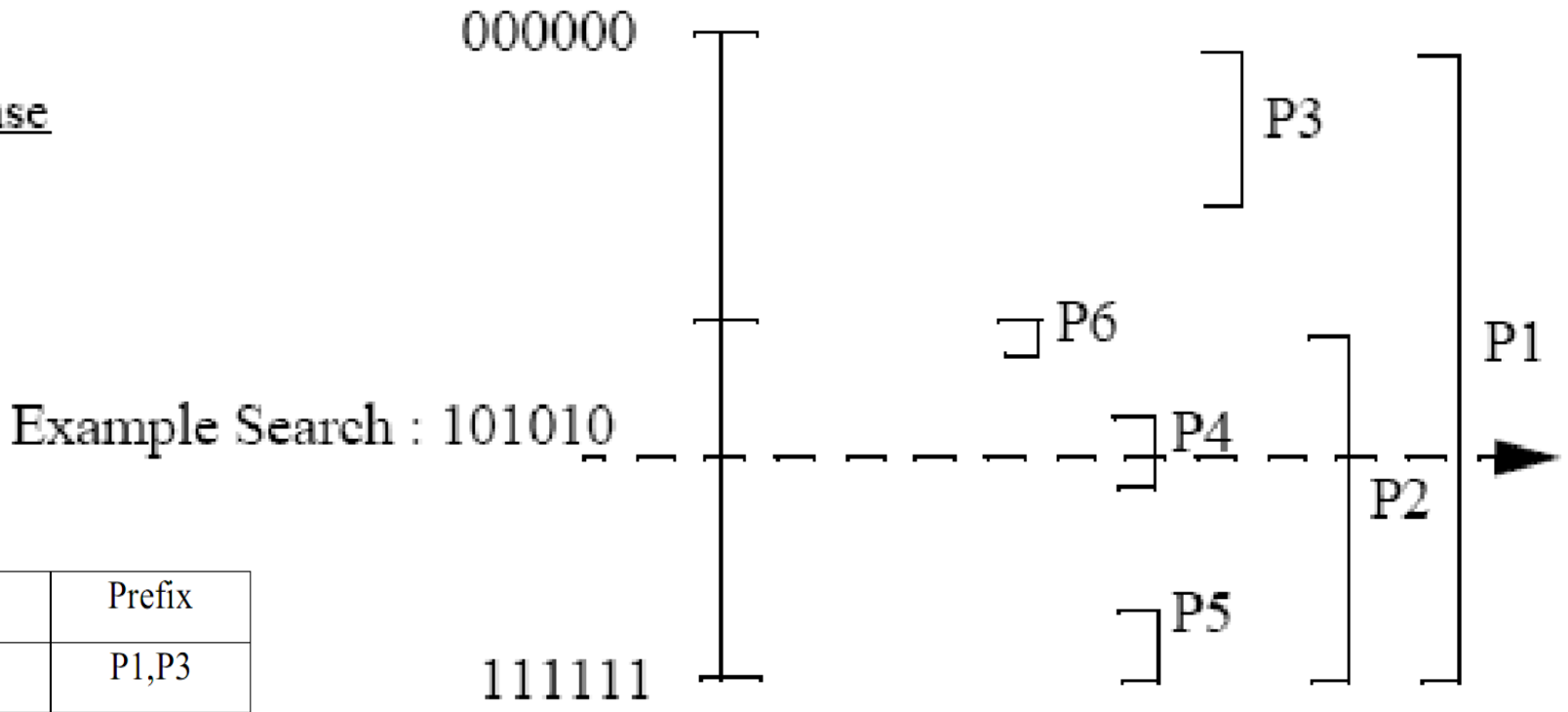
- Základem je Tree Bitmap metoda
- Zavedení další bitmapy Shape která kóduje tvar Trie
- Výhoda: eliminace dlouhých nevětvených cest, snížení paměťových nároků
- Zvýšení rychlosti díky omezení hloubky stromu
- Jedna z mála metod kde je diskutován nástup IPv6

Binární vyhledávání - intervaly

- Expanze prefixů na intervaly do tabulky (problém vnoření)
- Následně jakékoliv vyhledání – např. binární, B-strom
- Výhody: dobrý poměr použité paměti k použitým prefixům, počet paměťových přístupů má logaritmický nárůst
- Nevýhody: dlouhá doba aktualizace prefixů, nutno při každém vložení přepočítat celou tabulku

Binární vyhledávání - intervaly

Prefix Database
P1 *
P2 1*
P3 00*
P4 101*
P5 111*
P6 1000*



Adresa	Prefix
000000	P1,P3
001111	P1
100000	P6
100011	P2
101000	P4
101111	P2
111000	P5

Binární vyhledávání - prefixy

- Hlavní myšlenkou je modifikace tabulky prefixů tak, ať lze využít ověřený, rychlý a úsporný způsob pro uložení a vyhledání polí – hashování.
- Protože (zatím) neumíme použít hashování na prefixy, používá se tato metoda při hledání prefixů stejné délky, postupuje se přitom stejně jako u binárního vyhledávání
- Důležité je rozšíření tabulky prefixů o odkazy umožňující výběr správného směru vyhledávání.
- Nevýhody: větší objem spotřebované paměti, dlouhá doba pro aktualizaci pravidel
- Výhody: rychlost (počet přístupů do paměti)

Shrnutí

- Metody založené na Trie mají lineární časovou složitost
- Binární metody složitost logaritmickou
- Vzhledem k pevné délce IPv4 adresy 32 bitů se používají metody založené na Trie, složitost je konstantní
- Příchod IPv6 je v literatuře diskutován pouze okrajové, přičemž adresa je 4x delší, což u Trie metod znamená 4násobné zvýšení časové složitosti

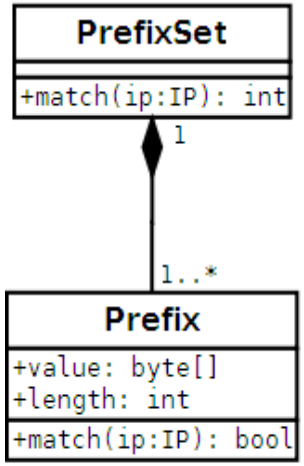
Krok 2 – cíle dizertační práce

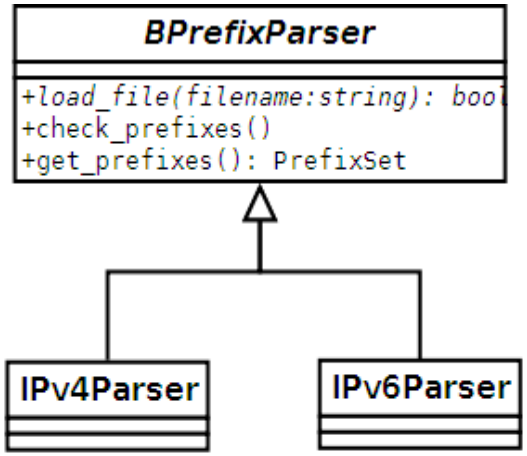
- Nalezení nového či vylepšení stávajícího LPM algoritmu
- Zaměření na
  - IPv6
  - 100 Gb/s
  - interní paměti v FPGA
- Nalezení vhodné metodiky pro optimalizaci poměru mezi obsazenou pamětí a rychlostí vyhledávacího algoritmu

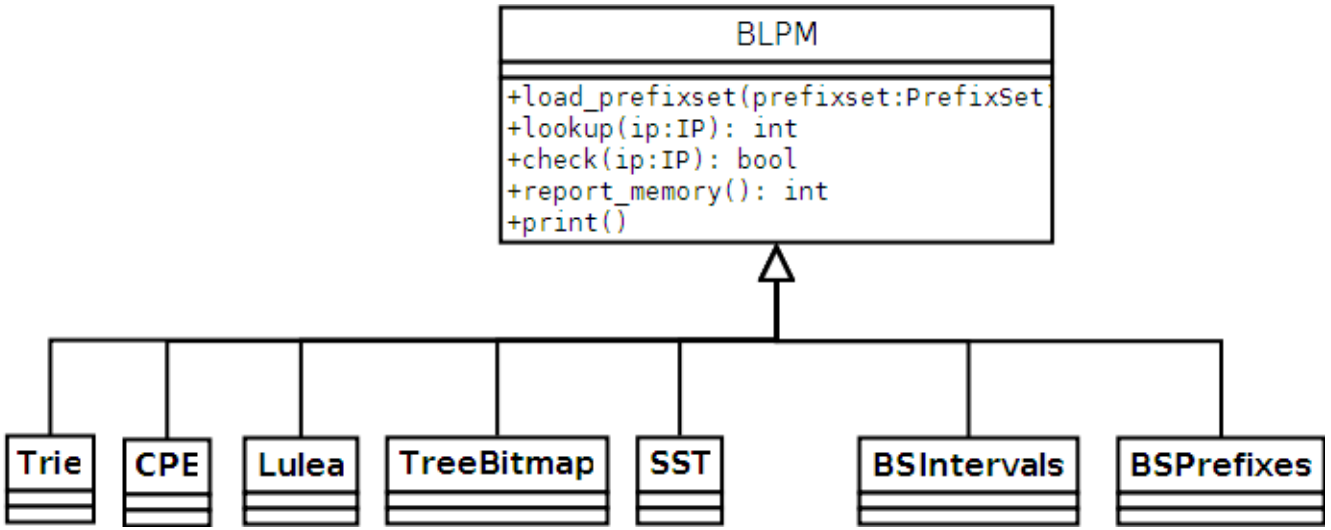
Krok 3 - knihovna experimentů

- Řešeno v rámci projektu ANT@FIT
- Cíle:
  - implementace všech současných LPM metod
  - programovací jazyk Python, objektový návrh
  - volně dostupná knihovna pro všechny zabývající se tématikou
  - objektivní možnosti srovnání nových LPM metod
- Tým:
  - Jura Tobola
  - Martin Skačan
  - Jaroslav Suchodol

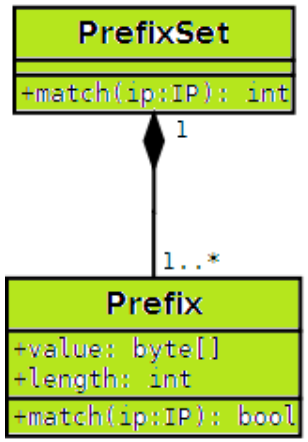
Objektový návrh

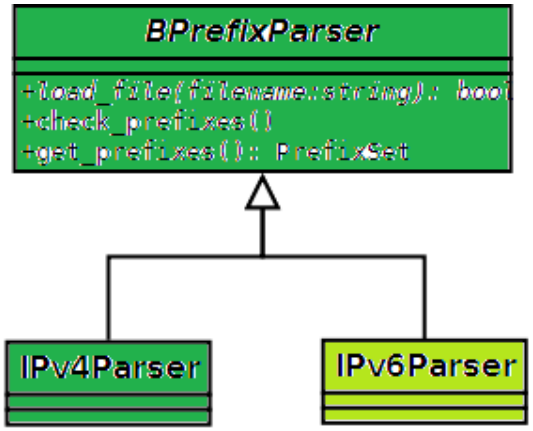


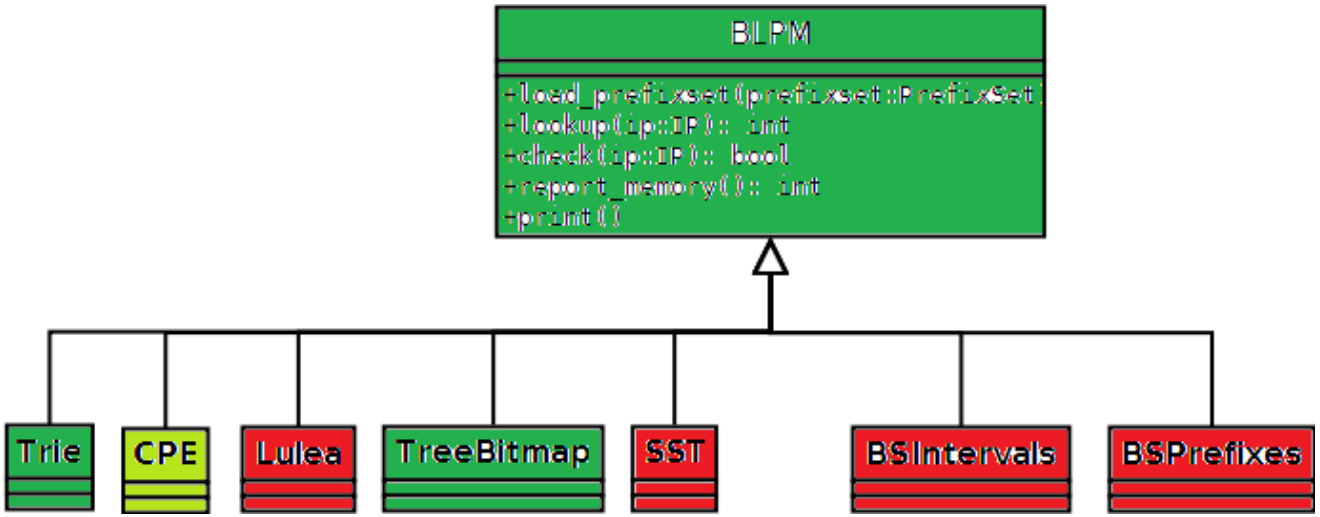




Aktuální stav implementace







Experimenty

- Porovnání všech metod z pohledu paměťové a časové složitost na dvou typech vstupních množin
  - 100 – 1000 prefixů: filtry
  - > 10 000 prefixů: směrovače
- Vstupní data
  - benchmark sety pro klasifikaci
  - výstupy ClassBench
  - routovací tabulky
  - pravidla firewallů
- Zaměření na IPv6

Krok 4 – vlastní metody 1

- Fixní MultiMatch metoda
  - Rozdělení vstupní množiny prefixů na skupiny pevné délky
  - Expanze prefixů jiných délek do těchto množin
  - Využití standarní hash funkce počítané paralelně pro všechny skupiny délek nad vstupní IP adresou
  - Generický parametr – počet skupin pro dělení, např. parametr 4 pro IPv4 → prefixy délky 8, 16, 24, 32
- Dynamická MultiMatch metoda
  - Skupiny pevné délky jsou zvoleny dynamicky na základě analýzy vstupní množiny prefixů

Krok 4 – vlastní metody 2

- Hybridní metoda
  - odstranění všech prefixů délky 32/128 (IPv4/IPv6) do samostatné množiny, na tyto je aplikována standardní hash funkce. U IPv6 postačí hash aplikovat na posledních 64 bitů adresy.
  - paralelně je pro kratší prefixy vystavěn TreeBitmap strom a provedeno vyhledání ve struktuře

Kroky 5 a 6

- Další dílčí úkoly:
  - Experimentální vyhodnocení vlastností navržených metod a porovnání s ostatními
  - Tvorba optimalizačních metrik na základě parametrů vstupní množiny prefixů – adaptace, rekonfigurace
  - Praktické ověření funkce na 10GE kartách COMBO v rámci filtrujícího zařízení NIFIC
- Publikace
  - PAD09
- Dizertace
  - 2012

Děkuji za pozornost

Otázky, diskuze...