Memory Efficient IP Lookups in High-Speed Networks

Author: **Martin Skačan**
Supervisor: **Jan Kořenek**

BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

Motivation

The growth of Internet traffic and speed of network links has direct impact on the required performance of core routers. To handle network traffic with **100 Gbps throughput**, routers have to process millions of packets per second. The most time-critical operation in core routers is the **Longest Prefix Match (LPM)**. Current IP lookup algorithms have high memory demands when representing **IPv6** prefix sets or introduce very high pre-processing overhead. Therefore, we propose a **new LPM algorithm** which is able to provide **very low memory demands** for IPv4/IPv6 lookups.

Summary

- Proposed prefix set representation has the **lowest memory requirements** in comparison to all existing LPM algorithms
- Large routing tables can fit into the **on-chip memory**
- HW architecture for this algorithm was also designed and proposed
- The architecture was implemented on Xilinx Virtex-6 FPGA with **140 Gbps throughput**
- Published on **IEEE conferences**:
  - **DDECS**
  - **FPL**

Analysis

LPM algorithms analyzed:

- Unibit Trie
- Controlled Prefix Expansion
- Lulea Compressed Tries
- Level Compression Trie
- Tree Bitmap (**TBM**)
- Shape Shifting Trie (**SST**)
- Prefix Partitioning

IPv6 prefix sets analysis (using TBM):

Identified 2 significant groups of nodes:

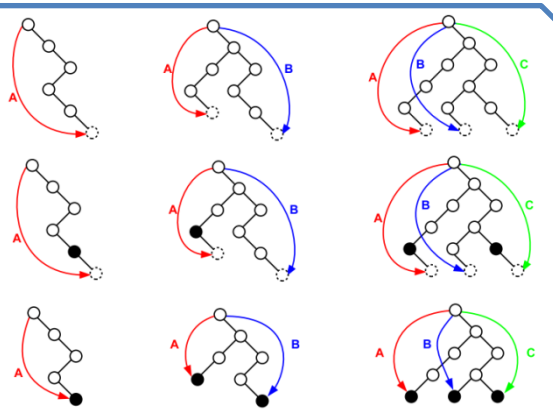
- **Leaf nodes with no children**
- **Internal nodes without prefixes**

Prefixes \ Child Nodes	0	1	2	3	4	5	6	7	8
0	0	11303	1666	812	538	184	145	131	249
1	8965	547	142	19	17	3	2	1	1
2	193	21	14	4	3	0	1	0	0
3	50	3	3	3	1	0	1	0	0
4	29	3	1	1	3	1	1	0	0
5	0	1	0	1	0	0	0	0	0

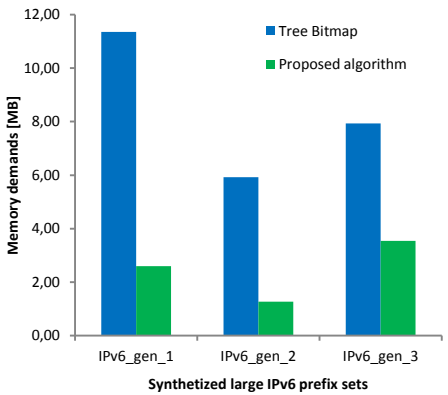
Design

Algorithm properties:

- **Using** fixed set of **node-types**
- Nodes optimized based on analysis results
- **Variable length** and **shape** of nodes
- Nodes are mapped to the prefix tree in order to find **best coverage** of the tree
- In every part of the tree is used the node with the **lowest price**



Results



Protocol	Prefixes	Memory requirements [kB]			Memory savings	
		TBM	SST	Proposed algorithm	Compared to TBM	Compared to SST
IPv4	220779	712,76	510,13	446,42	**37,37%**	**12,49%**
	442748	1492,76	1096,87	972,48	**34,85%**	**11,34%**
IPv6	10518	159,43	73,56	59,47	**62,70%**	**19,16%**
	10814	165,82	77,14	61,73	**62,77%**	**19,98%**

Real IPv4/IPv6 prefix sets