

## High Performance Architecture for Secure and Delayless Packet Classification

**J.Jaiveni**

M.Tech Student,  
Dept of ECE,

Christ College of Engineering and Technology.

**G.K. Senthil Kumar**

Assistant Professor,  
Dept of ECE,

Christ College of Engineering and Technology.

### ABSTRACT:

In the emerging world, packet classification act as a main function in network infrastructure. The increasing of throughput will become demand while performing in wire-speed packet classification has more challenge. Nowadays the packet classification solutions are performed depends on the characteristics of rule set. A novel modular bit- vector (BV) based on architecture to perform high speed packet classification on FPGA. We introduce an algorithm named StrideBV and modularize the BV architecture to achieve better scalability than traditional BV methods.

Further, we incorporate range search in our architecture to eliminate ruleset expansion caused by range-to-prefix conversion. The post place-and-route results of our implementation on a state-of-the-art FPGA show that the proposed architecture is able to operate at 100+ Gbps for minimum size packets while supporting large rulesets up to 28 K rules using only the on-chip memory resources. Our solution is ruleset-feature independent, i.e. the above performance can be guaranteed for any ruleset regardless the composition of the ruleset.

### IndexTerms:

Packet Classification, FPGA, router, networking.

### INTRODUCTION:

Packet classification means categorizing the packet into different flows based on one or more fields in its header. It is widely used in the intermediate network devices, such as router, to perform the admission control, flow queuing and appropriate networking services. This technique is widely used for firewall packet filtering and can provide required service differentiation for virtual private networks (VPNs) and quality of service (QoS) guarantees.

As more and more advanced services are required in the future networks, the performance of packet classification is quite critical to the performance of the networks. There are several packet classification algorithms invented in the research area, such as linear search, HiCut, HyperCut, BDD, FDD, TCAM based methods. They all present the evaluation of their algorithms based on the real networks. However, very limited work has been done to provide the network simulation based on these algorithms.

### Existing System:

Performing packet classification is challenging since it involves inspection of multiple fields against a rule set possibly containing thousands of rules. Performing such operations at wire-speed is even more challenging with the increasing throughput demands in modern networks. Various hardware platforms have been employed for packet classification in the past. While numerous solutions exist for packet classification, the effectiveness of these solutions rely on the characteristics of the rule set. To perform packet classification, one or more header fields of an incoming packet is checked against a set of predefined rules, usually referred to as a rule set or a classifier.

### Proposed System:

The trie-based algorithms require small memory since each rule is stored exactly once, but they do not provide high throughput because of rule comparison at every rule node. The decision tree-based algorithms provide high throughput since the number of rules compared with an input packet can be controlled as a limited number, but they require excessive amount of memory because of high degree of rule replication. This paper proposes to combine these two types of algorithms. The proposed algorithm primarily constructs a trie and then applies a decision tree for nodes having more rules than a threshold value.

A. Area-Based Quad-TrieA binary trie is a bitwise data structure, representing a bit of a prefix using an edge of the trie. A prefix is stored into a node of the trie, in which the level and the path of the node relative to root node correspond to the length and the value of the prefix, respectively. The AQT is conceptually an area decomposition algorithm, but it can be described using a quadruple trie structure as well. As a two-dimensional trie, the quadruple trie uses two prefix fields at the same time. By concatenating each bit of source prefix and each bit of destination prefix of a rule, a code word is constructed, and the code-word is used to determine a node storing the rule in building an AQT trie. Figure 1 shows an example set of rules for packet classification and the corresponding AQT trie. For example, the code-word of rule  $R_0$  is 00 11 01, which is 031 in decimal. The rule  $R_0$  is stored in the node of level 3 and path 031 from root node.

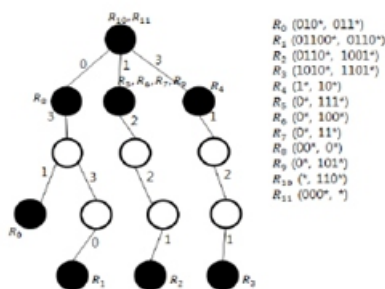


Fig. 1. AQT structure.

The decision tree construction process of the HiCuts algorithm is to split a given rule set recursively by using partial header information. At each tree node, a decision is made to split the current rule set into a number of subsets, in which each subset represents a child node. For the set of rules in Figure 1, Figure 2 shows the decision tree of the HiCuts algorithm. Each internal node handles the cut field, the number of cuts, and child pointers. Rules are stored at leaf nodes. The shape of the decision tree is determined by two pre-defined parameters; binth and space factor.

The binth defines the maximum number of rules stored at a leaf node. In the tree construction process, if the number of rules included in the current node is not greater than the binth, the node becomes a leaf node. Otherwise the node is further split and becomes an internal node. A large binth value produces a decision tree with a small depth. The space factor controls the degree of rule replication caused in recursive splitting process.

Since more cuts are allowed in a large space factor, a rule set can be divided into a large number of subsets and a rule can be included in many leaf nodes, which results in large rule replication.

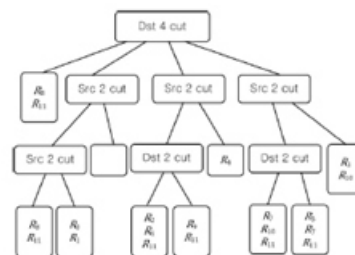


Fig. 2. HiCuts structure (binth = 3, space factor = 1.5).

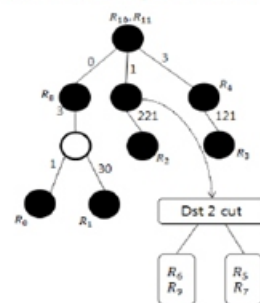


Fig. 3. Proposed structure (An AQT trie merged with a HiCuts tree).

### PROPOSED ALGORITHM:

The proposed algorithm combines a trie-based algorithm and a decision tree-based algorithm. This section explains the proposed algorithm using the AQT as a trie-based algorithm and the HiCuts as a decision tree-based algorithm. Figure 3 shows the proposed structure for the same example rule set. (In this figure, path-compression is applied, in which single-child empty nodes are removed and the path is compressed.) Instead of linearly storing the rules included in a CFS in the AQT algorithm, our proposed idea is to apply the HiCuts algorithm if the number of rules in the CFS is greater than a pre-defined threshold value. Since HiCuts uses arbitrary number of fields, fields other than the prefix fields can also be used for splitting the current CFS into different subsets. Hence, the number of rules linearly compared with a given input can become much smaller.

### PERFORMANCE EVALUATION:

In this section, we compare the performance of the AQT, and the Stride BV, the BC, and the proposed algorithm (represented by AQT-HiCuts) in throughput and Delay requirement.

Since rules are usually stored in off-chip memories and accessing off-chip memories is the slowest operation in the packet classification process, throughput is measured by the worst-case and the average number of rule comparisons. Simulation has been performed using rule sets generated by Classbench. We have generated three different sets of rules: access control list (ACL), IP chain (IPC), and fire-wall (FW).

## CONCLUSION:

This paper described a new approach to combine two differential algorithms for packet classification which show trade-off between throughput performance and memory requirement. Our future work will include the exploration of various combinations of different approaches and find a composite measure evaluating the performance of classification algorithms.

## REFERENCES:

- [1] H. Song and J. S. Turner, "ABC: adaptive binary cutting for multidimensional packet classification," *IEEE/ACM Trans. Networking*, vol. 21, no. 1, pp. 98–109, Feb. 2013.
- [2] M. M. Buddhikot, S. Suri, and M. Waldvogel, "Space decomposition techniques for fast layer-4 switching," *Protocols for High Speed Networks*, pp. 25–41, Aug. 1999.
- [3] F. Geraci, M. Pellegrini, P. Pisati, and L. Rizzo, "Packet classification via improved space decomposition technique," in *Proc. 2005 IEEE INFOCOM*, vol. 1, pp. 304–312.
- [4] P. Gupta and N. McKeown, "Classifying packets with hierarchical intelligent cuttings," *IEEE Micro*, vol. 20, no. 1, pp. 34–41, Jan./Feb. 2000.
- [5] S. Singh, F. Baboescu, G. Varghese, and J. Wang, "Packet classification using multidimensional cutting," *2003 ACM SIGCOMM*.
- [6] B. Vamanan, G. Voskuilen, and T. N. Vijaykumar, "EffiCuts: optimizing packet classification for memory and throughput," in *Proc. 2010 ACM SIGCOMM*, pp. 207–218.
- [7] W. Pak and Y. J. Choi, "High performance for packet classification for network-device platforms," *IEEE Commun. Lett.*, vol. 17, no. 6, pp. 1252–
- [8] Y. Chang and Y. Wang, "CubeCuts: a novel cutting scheme for packet classification," in *Proc. 2012 International Conference WAINA*, pp. 274–279.
- [9] H. Lim, N. Lee, G. Jin, J. Lee, Y. Choi, and C. Yim, "Boundary cutting for packet classification," *IEEE/ACM Trans. Networking*, vol. pp, no. 9.