

A Survey Paper of Novel Approach towards Multidimensional Packet Classification

Priti Rathi¹, Prof. Ganesh Bandal²

¹M.E. Student, G. H. Raisoni College of Engineering and Management, Ahmednagar, 414008, India
²Associate Professor, G. H. Raisoni College of Engineering and Management, Wagholi, Pune, 412207, India

Abstract: Packet classification plays an important role in both edge and core routers to provide advanced network services. Despite the vast body of existing work, packet classification remains an open and challenging problem. Existing algorithms such as HiCuts, HyperCuts, and boundary cutting show good search performance by exploiting the geometrical representation of rules in a classifier and searching for a geometric subspace to which each input packet belongs. New efficient splitting criteria which are memory and power efficient as compared to other mentioned techniques. Its approach produce a set of different-sized cuts at each decision step, with the goal to balance the distribution of filters and to reduce the filter duplication effect.

Keywords: Packet classification, splitting criteria, multicuts

1. Introduction

Packet classification is an essential function in Internet routers that provides value-added services such as network security and quality of service (QoS) [1]. A packet classifier should compare multiple header fields of the received packet against an arrangement of predefined principles and return the personality of the highest-priority rule that matches the packet header. Packet classification assumes a vital part in both edge and core routers to provide advanced network services. Despite the incomprehensible collection of existing work, packet grouping remains an open and challenging problem. On the one hand, network security, system virtualization, and system nature of service (QoS) are the driving factors for large-scale packet order including thousands to many thousands of filters in a single router.

Although packet classification has been broadly concentrated on for quite a while, specialists are still motivated to seek novel and efficient packet classification arrangements because of the accompanying reasons:

- Continual growth of network bandwidth.
- Ever-increasing complexity of network applications
- Technology innovations of network systems.

Number of packet classification algorithms have been proposed as of late, the greater part of them stay in scientific examination and/ software simulation stage, and few of them have been actualized in business items as a non specific arrangement. The gap between theory and practice in existing work can be compressed by diverse examination inspirations. Some algorithms focusing on extensive mathematical analysis have been proposed and some of them are accounted for to have excellent temporal/spatial complexity. However, algorithms of this kind can scarcely be found to have any execution in real-life network devices. These algorithms often work well with particular sort of principle sets. Be that as it may, on the grounds that packet classification rules for different applications have diverse highlights [22], couple of calculations are sufficiently keen to completely exploit the

redundancy lying in different types of rule sets to get stable execution under different conditions.

Packet classification is still an important problem and there is an incredible requirement for novel arrangements. The difference between theory and practice motivates our research.

2. Related Work

• **TCAMrazor: A systematic approach towards minimizing packet classifiers in TCAMS [2]**, author states that packet classification is the core mechanism that enables many networking services on the internet such as firewall packet filtering and traffic accounting. Using ternary content addressable memories (TCAMS) to perform high-speed packet classification has become the de facto standard in industry. TCAMS classify packets in constant time by comparing a packet with all classification rules of ternary encoding in parallel. Despite their high speed, TCAMS suffer from the well-known range expansion problem. As packet classification rules usually have fields specified as ranges, converting such rules to TCAM-compatible rules may result in an explosive increase in the number of rules. This is not a problem if TCAM have large capacities. Unfortunately, TCAMS have very limited capacity, and more rules mean more power consumption and more heat generation for TCAMS. Even worse, the number of rules in packet classifiers has been increasing rapidly with the growing number of services deployed on the internet. In this paper, they consider the following problem: given a packet classifier, how can they generate another semantically equivalent packet classifier that requires the least number of TCAMS entries? In this paper, a systematic approach, the TCAMS razor that is effective, efficient, and practical in terms of effectiveness is proposed. TCAMS razor achieves a total compression ratio of 29.0%, which is significantly better than the previously published best result of 54%. In terms of efficiency, our TCAMS razor prototype runs in seconds, even for large packet classifiers. Finally, in terms of practicality, this TCAMS razor approach can be easily deployed as it does

not require any modification to existing packet classification systems, unlike many previous range encoding schemes.

- **Topological transformation approaches to TCAM-based packet classification [3]**, Author studies the existing methods to mitigate the effect of range expansion and the limitations of small capacity, large power consumption, and high heat generation of ternary content addressable memory (TCAM)-based packet classification systems. However, they all disregard the semantics of classifiers and therefore miss significant opportunities for space compression. This paper proposes new approaches to range reencoding by taking into account classifier semantics. Fundamentally different from prior work, they view reencoding as a topological transformation process from one colored hyper rectangle to another, where the color is the decision associated with a given packet. Stated another way, they reencode the entire classifier by considering the classifier's decisions rather than reencode only ranges in the classifier ignoring the classifier's decisions as prior work does. It presents two orthogonal, yet composable, reencoding approaches: domain compression and prefix alignment. These techniques significantly outperform all previous reencoding techniques. In comparison to prior art, experimental results show that current techniques achieve at least five times more space reduction in terms of TCAM space for an encoded classifier and at least three times more space reduction in terms of TCAM space for a reencode classifier and its transformers. This, in turn, leads to improved throughput and decreased power consumption.

- **Efficient multimatch packet classification and lookup with TCAM [4]**, Today's packet classification systems are designed to provide the highest-priority matching result, such as the longest prefix match, even if a packet matches multiple classification rules. However, new network applications demanding multimatch classification - that is, requiring all matching results instead of only the highest-priority match - are emerging. Ternary content-addressable memory is becoming a common extension to network processors, and its capability and speed make it attractive for high-speed networks. The proposed TCAM-based scheme produces multimatch classification results with about 10 times fewer memory lookups than a pure software approach. In addition, their scheme for removing negation in rule sets saves up to 95 percent of the TCAM space used by a straightforward implementation.

- **Efficient multimatch packet classification for network security applications [5]**, Author presents new network applications like intrusion detection systems and packet-level accounting require multimatch packet classification, where all matching filters need to be reported. Ternary content addressable memories (TCAMs) have been adopted to solve the multimatch classification problem due to their ability to perform fast parallel matching. However, TCAMs are expensive and consume large amounts of power. None of the previously published multimatch classification schemes are both memory and power efficient. In this paper, develop a novel scheme that meets both requirements by using a new set splitting algorithm (SSA). The main idea behind SSA is that it

splits filters into multiple groups and performs separate TCAM lookups into these groups. It guarantees the removal of at least 1/2 the intersections when a filter set is split into two sets, thus resulting in low TCAM memory usage. SSA also accesses filters in the TCAM only once per packet, leading to low-power consumption. Here they compare SSA with two best known schemes: multimatch using discriminators (MUD) (Lakshminarayanan and Rangarajan, 2005) and geometric intersection-based solutions (Yu and Katz, 2004). Simulation results based on the SNORT filter sets show that SSA uses approximately the same amount of TCAM memory as MUD, but yields a 75%-95% reduction in power consumption. Compared with geometric intersection-based solutions, SSA uses 90% less TCAM memory and power at the cost of one additional TCAM lookup per packet. They also show that SSA can be combined with SRAM/TCAM hybrid approaches to further reduce energy consumption.

- **Efficient packet classification for network intrusion detection using FPGA [6]**, Using FPGA technology for real-time network intrusion detection has gained many research efforts recently. In this paper, a novel packet classification architecture called BV-TCAM is presented, which is implemented for an FPGA-based Network Intrusion Detection System (NIDS). The classifier can report multiple matches at gigabit per second network link rates. The BV-TCAM architecture combines the Ternary Content Addressable Memory (TCAM) and the Bit Vector (BV) algorithm to effectively compress the data representations and boost throughput. A tree-bitmap implementation of the BV algorithm is used for source and destination port lookup while a TCAM performs the lookup of the other header fields, which can be represented as a prefix or exact value. The architecture eliminates the requirement for prefix expansion of port ranges. With the aid of a small embedded TCAM, packet classification can be implemented in a relatively small part of the available logic of an FPGA. The design is prototyped and evaluated in a Xilinx FPGA XCV2000E on the FPX platform. Even with the most difficult set of rules and packet inputs, the circuit is fast enough to sustain OC48 traffic throughput. Using larger and faster FPGAs, the system can work at speeds greater than OC192.

3. Classification Techniques

Following table shows the various techniques used for packet classification purpose and algorithms comes under those techniques. Packet header is used by classification techniques to search the data architecture built in the preprocessing operation for finding packet matching rule set.

Packet Classification techniques:

Table 1: Classification Techniques

Sr. No.	Classification Technique	Algorithms
1.	Two Dimensional	Set pruning tree, Grid of trees, Hierarchical tree.
2.	Decision tree	HiCuts, HyperCuts, Effcuts, HyperSplit, Boundary cutting
3.	Hardware	TCAM, BV- TCAM, DCFL
4.	Data Structure	Caching, Linear search

Now we will take short overview of some of these mentioned classification techniques.

1. Set Pruning tree: This algorithm has reduced query time obtained by replicating rules to eliminate recursive traversals. It experiences prefix replication and it is scalable for 2 dimensional [15].

2. Hierarchical Tree: The backtracking method used in this algorithm degrades the performance of this algorithm which cause into the wasted time, also it is scalable for 2-Dimension. The storage complexity is of $O(NdW)$ [16]. This algorithm is also known as a "multi-level tries", "backtracking seek", or "trie-of-tries" [17].

3. HiCuts: This algorithm is the abbreviation of Hierarchical Intelligent Cuttings which was introduced in 2000. This algorithm recursively cuts the space into subspaces using one dimension per step. Each subspace ends up with fewer covered standard hypercube that consider a straight pursuit. In the construction of a decision tree of the HiCuts algorithm, an extensive number of cuts expend more stockpiling, and a little number of cuts causes slower search performance [18]. There are no. of characteristics of the decision tree such as its depth, degree of each node, and the local search decision to be made at every hub are picked while pre-processing the classifier based on its characteristics.

4. HyperCuts: HyperCuts is known as advanced version of HiCuts algorithm. It was introduced in 2003. In this algorithm multiple fields are considered at a time for cutting. Hence which results in fast searching time and decision trees are shorter in depth [18].

5. EffiCuts: This calculation radically decreases the overhead. To address the problem of variation in rule size, this technique eliminate overlap among small and large rules by separating small and large rules in each dimension into distinct separable trees so that each dimension can be cut finely or coarsely without incurring replication. To reduce the multiple trees' extra accesses which degrade throughput, it selectively merges separable trees mixing rules that may be small or large in at most one dimension [19].

6. Hypersplit: HyperSplit algorithm is known for its suitability for various types of rules set. HyperSplit achieves

better performance in terms of classification speed, memory usage and pre-processing time. HyperSplit is the main calculation that can effectively handle all the rule sets; HyperSplit is also the only algorithm that achieves more than 6Gbps throughput on the Oocteon3860 multi-core platform when tested with 64-byte Ethernet packets against 10K ACL rules [20].

7. Boundary Cutting: This algorithm finds out the space that each rule covers and performs the cutting according to the space limit. Thus, the cutting in the proposed algorithm is deterministic rather than involving the complicated heuristics, and it is more powerful in giving made strides search performance and more efficient in memory requirement [18].

8. TCAM: TCAM is better known for the straightforward design, speed and good management associative. Here the all rules are checked at the same time using parallel hardware. Search operation in TCAM involves the input data comparing with all TCAM contents and the result appear in one clock cycle. Over the past few years TCAM as device for subject been investigated by many. Fundamental issue considered has been that of the improvement of TCAM range representation space efficiency.

9. BV – CAM: The [21] present a novel architecture called BV-CAM in which multiple matches in Gbps are reported. TCAM perform the look up of header fields which can be a prefix or exact value, while the tree-bitmap s utilized to execute source and destination port gaze upward.

10. DCFL: This algorithm has two important phases, in the first phase each field of packet is independently searched and the results from this phase are added in the second phase. The fundamental challenge in performing this technique is in the second phase that means how the searches results of single field combined efficiently [21].

Below table shows different algorithm with their advantage and disadvantage comparatively.

Table 2: Advantages and Disadvantages of classification algorithms

Algorithms	Advantages	Disadvantages
HiCuts	Each query leads to a leaf node in the HiCuts tree, which stores a small number of rules that can be searched sequentially to find the best match.	The computation required for the preprocessing consumes much memory and construction time.
HyperCuts	HyperCuts is characterized by using multicuts in internal nodes to reduce the Decision Tree depth, it has high storage than HiCuts, it is efficient with edge routers.	It is difficult to Support incremental updates.
BV-CAM	Searches are simple after constructing d data structures.	It has low dynamic update, bad memory using, it does not scale well for large data base and very high speed system.
Hypersplit	HyperSplit algorithm is characterized by It's suitability for various rule sets, and using binary search, and it has better preprocessing time than HiCuts and HSM.	Not Memory efficient.
DCFL	It takes advantage of the capabilities of modern hardware	The data structure only supports local

technology.	reconstruction.
-------------	-----------------

4. Performance Analysis

We have study in previous section that there are many algorithms used for packet classification. Each technique has it's own advantages and disadvantages. In below table III we will see the how these algorithms performance with different types of parameters such as memory consumption, throughput, area etc.

Table 3: Comparative analysis of different algorithms with important parameters

Sr. No.	Technique	Memory Utilized	Throughput	Area
1	Optimized HyperCuts	612	80.23 gbps	78%
2	Simplified HyperCuts	286	10.80 gbps	89%
3	BV-CAM	16	10gbps	8%
4.	DCFL	221	8gbps	24%
5	Boundary cutting	200	6gbps	20%

5. Conclusions and Future Work

In this paper we have studied decision tree based packet classification algorithm. We have studied the existing algorithms like HiCuts, HyperCuts, BV-CAM, Hypersplit and DCFL for improve the results of proposed algorithm and overcome drawbacks of these existing algorithms. In future work we will consider the matching of multiple rules. We will study or extend further algorithm for multidimensional packet classification. It will work for large rule set with effective memory efficiency, search speed and avoiding the rule duplication.

Acknowledgment

We would like to thank all the authors of different research papers referred during writing this paper. It was very knowledge gaining and helpful for the further research to be done in future.

References

[1] H. J. Chao, "Next generation routers," Proc. IEEE, vol. 90, no. 9, pp. 1518–1588, Sep. 2002

[2] A. X. Liu, C.R.Meiners, and E.Torng, "TCAM razor: A systematic approach towards minimizing packet classifiers in TCAMs," IEEE/ACM Trans. Netw., vol. 18, no. 2, pp. 490–500, Apr. 2010

[3] C. R. Meiners, A. X. Liu, and E. Torng, "Topological transformation approaches to TCAM-based packet classification," IEEE/ACM Trans. Netw., vol. 19, no. 1, pp. 237–250, Feb. 2011

[4] F. Yu and T. V. Lakshnam, "Efficient multimatch packet classification and lookup with TCAM," IEEE Micro, vol. 25, no. 1, pp. 50–59, Jan. –Feb. 2005

[5] F. Yu, T. V. Lakshman, M. A. Motoyama, and R. H. Katz, "Efficient multimatch packet classification for

network security applications," IEEE J. Sel. Areas Commun., vol. 24, no. 10, pp. 1805–1816, Oct. 2006

[6] H. Yu and R. Mahapatra, "A memory-efficient hashing by multi-predicate bloom filters for packet classification," in Proc. IEEE INFOCOM, 2008, pp. 2467–2475

[7] H. Song and J. W. Lockwood, "Efficient packet classification for network intrusion detection using FPGA," in Proc. ACM SIGDA FPGA,2005, pp. 238–245

[8] P. Gupta and N. Mckeown, "Classification using hierarchical intelligent cuttings," IEEE Micro, vol. 20, no. 1, pp. 34–41, Jan.–Feb. 2000

[9] S. Singh, F. Baboescu, G. Varghese, and J. Wang, "Packet classification using multidimensional cutting," in Proc. SIGCOMM, 2003, pp. 213–224

[10] P. Gupta and N. Mckeown, "Algorithms for packet classification," IEEE Netw., vol. 15, no. 2, pp. 24–32, Mar.–Apr. 2001

[11] B. Vamanan, G. Voskuilen, and T. N. Vijaykumar, "EffiCuts: Optimizing packet classification for memory and throughput," in Proc.ACM SIGCOMM, 2010, pp. 207–218

[12] H. Song, M. Kodialam, F. Hao, and T. V. Lakshman, "Efficient trie braiding in scalable virtual routers," IEEE/ACM Trans. Netw., vol. 20, no. 5, pp. 1489–1500, Oct. 2012

[13] J. Treurniet, "A network activity classification schema and its application to scan detection," IEEE/ACM Trans. Netw., vol. 19, no. 5, pp. 1396–1404, Oct. 2011

[14] L. Choi, H. Kim, S. Ki, and M. H. Kim, "Scalable packet classification through rule base partitioning using the maximum entropy hashing," IEEE/ACM Trans. Netw., vol. 17, no. 6, pp. 1926–1935, Dec. 2009

[15] P. Tsuchiya. "A search algorithm for table entries with non-contiguous wildcarding," unpublished report, Bellcore

[16] Florin Baboescu and George, "Scalable packet classification", IEEE ACM transactions on networking, Vol 13, No 1, February 2005, pp 2-14

[17] Pankaj Gupta and Nick McKeown, "Algorithms for packet classification", pp 1- 29

[18] Hyesook Lim, Nara Lee, Geumdan Jin, Jungwon Lee, Youngju Choi and Changhoon Yim, "Boundry cutting for packet classification", IEEE/ACM transactions on networking, vol.. 22, no. 2, April 2014

[19] Balajee Vamanan*, Gwendolyn Voskuilen* and T. N. Vijaykumar, "EffiCuts: Optimizing Packet Classification for Memory and Throughput", SIGCOMM 2010, August 30-September 3, 2010

[20] Yaxuan Qi, Lianghong Xu and Baohua Yang and Yibo Xue and Jun Li, "Packet Classification Algorithms: From Theory to Practice", IEEE INFOCOM 2009

[21] Ammar Yahya Daeef Al-Nejadi and Nasir Shaikh- Husin, "Survey on Multi Field Packet Classification Techniques", Research Journal of Recent Sciences ISSN 2277-2502 Vol. 4(2), 98-106, February (2015)

[22] <http://www.arl.wustl.edu/~hs1/PClassEval.html>