



## Packet Classification Algorithms and Methods: A Critical Review

Midde Adisheshaiah<sup>1</sup> and M. Sailaja<sup>2</sup>

<sup>1</sup>Research Scholar, Department of Electronics and Communication Engineering,  
JNTUK, Kakinada (Andhra Pradesh), India.

<sup>2</sup>Professor, Department of Electronics and Communication Engineering,  
JNTUK, Kakinada (Andhra Pradesh), India.

(Corresponding author: Midde Adisheshaiah)

(Received 14 October 2019, Revised 17 December 2019, Accepted 24 December 2019)

(Published by Research Trend, Website: [www.researchtrend.net](http://www.researchtrend.net))

**ABSTRACT:** Packet classification is process of Categorization of packets interested in flows. It holds a significant role in network applications like network security, firewalls, virtual network, Quality of service (QoS) and some other network services. To facilitate these services, a predefined set of rules are used for classification of data packets. It can be done either hardware or software means. The procedure encompasses the usage of database of rule sets. In the High speed networks with wire links, the reliability of software based classification is uncertain. In general, packet classification on multiple fields is a difficult problem. The systematic problem handling is dependent on how well the protocol and multiple fields are analyzed. Also it is quite challenging to attain high throughput with wire-speed classification which depends on rule set characteristics. The fore coming classification procedures are in great need of hardware implementation, which is limited by the need of abundant power and memory utilization. This paper aims at delivering many state of art in methods developed in the recent past on packet classification. These methods are focused on various parameters like throughput, latency, power consumption, memory utilization, update of rule set, etc. The significance and flaws of these methods are critically reviewed.

**Keywords:** Packets, Quality of Service, router, rule set, throughput, latency, power consumption, memory utilization.

### I. INTRODUCTION

Packet Classification is a primary utility in Internet routers, which assists services like quality of service (QoS) and network security. A packet classifier uses a predefined set of rules to return identity of a highest-priority rule of the received packet. This is done by matching the packet header by examining multiple header fields. It is central to Software-Defined Networks. Software-Defined Networking (SDN) is a promising architecture, which is dynamic, convenient, cost-effective, and flexible. First SDN standards are considered as Open Flow, which divides hardware based data path from the software based control path and also used to manage network traffic between data and control planes. It performs the flow table lookup. In an Open flow, SDN allows the system to make the change of processing tasks. The software solutions are flexible to perform optimum packet classification on multi-core network processors. But they suffer with inherent issues like need of abundant memory and uneconomical high parallelism. A maximum of 10 Gbps throughput is achieved on multi-core network processors even with best software classification [1, 2]. This is less when compared with DC (Data Center) networks and large ISP (Internet Service Provider) backbones which offers 100Gbps [3]. Whereas the hardware procedures preferably use TCAM (Ternary Content Addressable Memory) and meets the near performance to wire-speed. But these methods are limited with a trade off of power efficiency, programmability and scalability. Also the TCAM based methods are unable to deal with complex rule sets since they suffer with a problem of range to prefix [4, 5]. The

performance of software methods employing multiple field inspection is inadequate for wire speed processing. Therefore hardware solutions perform packet classification by using multiple fields and plays prominent role in wire-speed processing and secure networking. Their memory requirement which vary limits the classifier performance [6]. In general they are realized using Field Programmable Logic Array (FPGA) having inadequate on chip memory and thus the task of classification is quite challenging. Another prime constraint for packet classification architecture is Power consumption, which is quite increasing with the rise of throughputs of the order of trillions of bits per second. Heavy power consumption at router components drowns the power efficiency and lead to the rise of operating costs [7].

Memory efficient look up solutions are need to be given prime consideration. They are in general classified as: decision tree approaches [8, 15, 16, 45] Bit Vector (BV)-based approaches [6, 9, 46] and ternary content-addressable memory (TCAM) [4, 10, 11]. The priority of adding new rules at random times has great priority than single priority. TCAM suffers with high overhead since they employ top to bottom decreasing priority and rearrangement of existing entries. The BV procedures return the results in terms of bit vectors or tables by performing the lookup on each field exclusively. They sacrifice the memory utilization and aim to maximize the classification speed. As the bit vector size and rule set size increases this approach will be complex to design on hardware setup. In the Decision tree approaches, traverses from root to leaf to perform the lookup. Then a tree is generated from small search spaces obtained by

splitting large search space. This tree construction needs many modifications and need the storage of child pointers at each node and thus it increases the system cost.

Even though the inexplicable collection of previous works, many requirements have met by packet classification techniques, such as high throughput, low power consumption, nominal cost and lesser memory density. Hardware based packet classifiers use TCAM for to store the rules in associative memories and perform a parallel match of these rules [11]. Therefore continuity in classification time is attained. But these methods are quite complex to develop and leads to a rise of system cost [4]. Whereas the software methods builds distinguished in-memory data structures, typically decision trees and can perform efficient classification. But they are slower than TCAM methods since they need decision tree from the root to the matching leaf. The software methods are inefficient to maintain minimum memory usage in order to achieve maximum system speed for packet classification. Therefore a rise of memory usage is observed with the exponential rise of rules set. For high throughput, TCAM is better to use. But it suffers from power consumption and cost. For both throughput and power consumption, SRAM is used. Based on the particular needs either TCAM or SRAM are used to combine with multi core and FPGAs in order to satisfy these constraints. Due to reconfigurable feature of FPGAs, which are mainly adopted for mid-sized and flexible solutions. FPGA offers abundant parallelism and is flexible to develop reconfigurable architecture. They utilize multiple strategies to perform parallel processing and are feasible to develop application of scientific computing. It full fills the needs of packet classification like high performance, line speed processing, multi match support and dynamic updating capability. In current scenarios, researchers are focused on the needs of above requirements in packet classification.

## II. LITERATURE REVIEW

Based on research, packet classification is divided into two classes. Algorithmic classification and another one is architectural classification. From existed works, Researchers have been noticed that classification of packets design architecture, it is observed that researchers have designed the packet classification architecture using algorithmic based, the following methods are:

- Exhaustive search
- Tuple search
- Decision tree
- Decomposition based method.

In exhaustive search, it includes two familiar approaches such as linear search and parallel search. Linear search verifies every rule in the classifier until and unless a match is found. The parallel search implied by Ternary Content Addressable Memory (TCAM) where each rule is assigned by a processor. It provides Effective utilization of memory and does not require any pre-processing elements. Those are updated with stability and simplicity. Though, they require memory accesses per each search are  $O(N)$ . Here  $N$  represents the total count of rules present in a rule set. A linear search becomes unaffordable slow even for moderately sized rule sets. Packet classification algorithm based on

Decision tree method focuses on two aspects. Firstly, how to choose the cut dimension and the second is how to make a decision. The cut-point for divide into subspaces from address space. The main problem of this method is memory requirement is high, and extended pre-processing time. Thereafter, demonstrated as in Gupta & McKeown [15] and Singh *et al.*, [16], demonstrated a decision tree based packet classifier using Hicuts and Hypercuts respectively. In the navigation tree process, a leaf node is used for multi rules accumulation. Then the incoming packets are allowed for a linear search. Vamanan *et al.*, [23] introduced a new decision tree based algorithm called as. This algorithm projected based on four proposals:

- To design separate decision tree for making different rule sets intended for reduction in overlap of large and small rules in a classifier.
- To decrease the multiple trees as that degrades the throughput, so as to incorporate the large or small rules in one dimension.
- More often than not to get cut along any field in irregular ways so with the purpose of distributing rules in different leaf nodes. Hence, aim to make cut in equal way along the fields.
- To attain less access per node, compare to both HiCuts and HyperCuts, they have co-located the parts of node and its children.

Thus the optimal memory utilization and improvement in performance is obtained using the proposed EffiCuts method. The major difference between Hicuts and Hypercuts is that in an internal node one dimensional cutting is used for Hicuts, where as multi dimensional cutting is used for Hypercuts. Hence Hicuts tree depth is longer than Hypercuts. Independent searches are executed on each packet fields, which are decomposed into multiple single field instances. It is analogous to multiple fields based Recursive Flow Classification [17] and Aggregated Bit Vector (ABV) and BV (Bit Vector) based classification [14]. At the trade off of fast classification, these methods fail in optimal utilization of memory and consume lot of simulation time for the preprocessing stage. Thus it is suitable for the areas involving frequent updating of rules set.

A novel Multi filed packet classification based on Recursive flow classification (RFC) is proposed by Gupta and McKeown [17]. The RFC maps packet header bits (S) to bits (T) of pre-computed class ID ( $T \ll S$ ) using valid rules. Though, change in rule characteristics, the results is unstable. This approach has given less significance to different overlapping regions i.e.,  $O(nF)$  for rule sets of 'n' number having F dimensions. Under usual conditions (with rare change of rules) this approach consumes much time for pre computation. And therefore it suits very less for the applications involving dynamic rules set update. Also with the liability of distinct overlapping regions, it even suffers with exponential memory usage.

Lakshman and Stiliadis [14] proposed bit vector algorithm which is enhanced by Singh *et al.*, [16]. Memory utilization is minimized through Aggregated bit vector algorithm (ABV). Radical memory usage, poor classification resulted with the dependency of arrived packet value. This tends to tree path variations [19]. Srinivasan *et al.*, [20] demonstrated packet classification on multiple fields based on Tuple Space method. The specified rules govern various tuple search categories.

The rules are varied by prefixes and implements hashing within unique rules set and thus minimized the search of multiple fields. Taylor *et al.*, [12] proposed a simple precise match using a novel partition based classifier. At a trade off of heavy pre computation time, this method is credited with optimal memory utilization and fast average query time. Moreover based on the rule set characteristics, the classification time can be altered.

Warkhede *et al.*, [21] proposed two-dimensional conflict-free filters for Fast packet classification. In this scheme, to solve the classification problems a set of conflict free rules are used. In addition, they also prepared the assumption that the rule ranges are IP prefixes. Making a note of this assumption, data structure along with  $O(n \log^2 w)$ ,  $O(\log^2 w)$  query time and have been obtained. This scheme rapidly works for conflict resolved rules. Lee & Shieh [22] developed a Diagonal Based Tuple Space search, this system concentrates on minimizing the classification time even as maintain a requirement of reasonable memory. Familiar shortfalls of multiple packet classification algorithms suffered includes, less speed raised with the heavy access of memory and the dependency of classification on a typical kind of rules set. Also all these methods consumes considerable amount of simulation time during initial stages of data processing. Furthermore, nearly all of the published algorithms require substantial amount of pre-processing. Ahmed *et al.*, [19] used SHA-1 or MD5 and advanced cryptographic hash functions. Through this they limited the memory access that cause attributable to collisions. But the rapid computation has become hard while rescuing the hash lookup table. Thus the response speed is improved. So these hash functions are not having better performance under the consideration of uniform hashing. But the enforcement of practical unsophisticated hash functions may tend to suffer from high collision rates.

Srinivasan *et al.*, [13] and Van Lunteren *et al.*, [24] limited the collision bounds by the use of "semi-perfect" hash function. Broder and Karlin [25] suppressed the data collisions with Parallel computation of multiple hash tables having distinct hash functions. Furthermore, it uses only one hash table than d-hash functions utilized by Azar *et al.*, [26]. D-Hash function hashed each item, which acquiesce independently and equally distributed buckets for each item, and the least loaded bucket stored by each item. The bucket's average load is reduced while examining d- buckets. With a simple variation of d-left and d-random schemes, an improvement is observed for the IP lookups. Broder & Mitzenmacher *et al.*, [27] supposed the 2-left scheme as in Vöcking [28]. This method involves the insertion of least loaded bucket while separating the buckets into d-sections. And compared with d-random a superior performance has been attained.

Bloom, demonstrated a multi-hashing approach based on Bloom filters [29] and improved by Fan *et al.*, [30] using Counting Bloom Filter. This method substitute a count for every bit in a rule set and allowed for further hashing. Dharmapurikar *et al.*, [31] avoided off-chip hash table redundant search through lookup schemes based bloom filters. It is a simple classification method that involves prefix grouping and storage of these groups in a hash table. While performing the lookups in software, the best choice in the binary search on hash

table is prefix length demonstrated by Waldvogel *et al.*, [32]. For reducing number of searches from linear to logarithmic, binary search is used. As a result it improves the lookup time performance in the  $O(\log W)$ , where  $W$  is the number of unique prefix lengths.

Packet classification strictly by software procedures are observed to be suffered with various short falls. Pure software based classification methods suffers with limited speed of memory access, generalizability while developing a specific rule set having specific features and hectic simulation times. Cumulating these simulation practices with hardware component will try to conceal few more flaws or shortcomings present in existing methods. Several researchers' underway exploring hardware architectures to improve the packet classification process speed. To have a discussion regarding the Content-Addressable Memories (CAMs) represented by SRAMs usually, demonstrated as in Qin *et al.*, [33]. It stores word by 0 and 1. Compared to traditional procedures, hardware search engines called CAMs have attained superior performance. A TCAM (Ternary Content Addressable memory) stores words by 3 digital values, 0, 1, and X, where X denotes both 0 and 1. Both TCAM and ASIC perform the similar operation, only in hardware. In view of the fact that these TCAMs are designed for perform packet classification, because of high-speed and capable. While classifying the packet, the TCAM performs a assessment of all its entries in parallel manner, and the priority encoder picks up the best matching rule as proposed as in Chen & Oguntoyinbo [34]. TCAMs are efficient for small size rule set of the file, while using large applications it consume more power and a few number of operators have limited support and need additional initial processing stages. Hence the memory utilization is found inefficient. It is observed by some vendors sustain their rules in TCAM, demonstrated as in Juniper Networks [35]. Cong *et al.*, [36] developed a hardware accelerator called Electronic System Level (ESL) which has the capability of parallel execution of codes. Thus the realized Register Transfer Level (RTL) design procedures have achieved superior performance than pure software approach. Further a General Purpose Processors (GPPs) is allowed for packet classification at a tradeoff of flexibility and performance. Xtensa *et al.*, [37] proposed another ASIP method which obtained potentially better compared with GPP software approach. It is even highly flexible compared to pure RTL methods. Shah & Gupta [38] developed two methods called Chain-ancestor ordering constraint (CAO-OPT) and Prefix-Length Ordering Constraint algorithm (PLO-OPT). Both are used to reduce the TCAM update time. Liu *et al.*, [39] presented two methods. First method is based on pruning; redundant prefixes are identified and then removed. The second technique, based on mask extension, expands the mask to be any arbitrary combination of zeros and ones. Panigrahy & Sharma [40] developed large TCAM modules, in which different TCAM blocks are partitioned equally using prefix ranges and thus power consumption issue is more concentrated. Zane *et al.*, [41] demonstrated another partitioning algorithms which is realized into two strategies termed as trie-based and bit selection based techniques. If the TCAM block is full then the first method requires total reconstruction. In Song *et al.*, [42], TCAM is combined with Bit Vector.

They performed multi-match packet classification using an architecture known as BV-TCAM for. In this architecture, tree Bitmap has implemented by using multi bit trie, so TCAM performs prefix or exact match, which are used for source or destination port lookup. Due to good speed and simple management, TCAM is an optimum hardware solution for 1D packet classification. It allows all fields check in unit time at greater speeds. Using TCAM, multi-dimensional packet classification. Chang & Su [43] proposed a scheme. This is a Gray-code based which employs an efficient range encoding trial. In past decades, many novel packet classification algorithms mapped onto FPGAs have been introduced [44] and Jiang & Prasanna [45]. Chang *et al* [44] developed Set Pruning Multi-bit Trie (SPMT) which is a pipelined architecture for multi-field packet classification.

Taylor and Turner [46] proposed a decomposed based algorithm called DCFL (Distributed Cross producing of Field Labels) algorithm. In Jiang & Prasanna [45] developed a decision-tree-based approach for packet classification. It is a 2D multi-pipeline architecture which involves current field-programmable gate arrays (FPGAs) to exploit abundant parallelism. In hardware circuits considering the advantage of parallel processing in addition to pipelining. This structure doesn't influence FPGA search speed, since it is developed in the initial processing stages. Packet classification has comprehensively studied over the past decade [50]. They performed packet classification on hardware using a decision tree and decomposition based procedures are desirable for hardware implementation of packet classification system. Most of the packet classification approaches developed on software or hardware, do classified into two foremost categories: decision-tree-based and decomposition-based [51, 52] algorithms. Decision-tree-based approaches demonstrated [45, 15]. These methods which allow recursive search of fields distributed into small subspaces.

Jiang & Prasanna [45] developed, pipelined architecture on FPGA mapped by a decision tree, however this tree based approaches is depend on rule set. Gupta & Mckeown [15] demonstrated a method to duplicate rules using a cut in one field approach. Consequently, up to  $O(Nd)$  memory used by a decision-tree; this approach can be unfeasible. Sun *et al* [48] proposed an algorithm based on decomposition, which includes two phase. Each field of packets is performed by individual searches in first phase and the results of first phase are combined in second phase. Implementation on FPGA on-chip memory has become a real challenge in this method. Qu *et al.*, [52], Pus & Korenek [53] addressed a decomposition based approach which involves individual search of packet header fields. Pus and Korenek [53] proposed hash-based merge techniques in which the final result is obtained by merging partial results. However they rely on additional hardware module to solve hash collisions or need abundant accesses to external memory accesses. Jiang & Prasanna [49] proposed a decomposition-based approach called Field-Split Parallel Bit Vector (FSBV) which makes use of FPGA block RAMs for multi-match packet classification. The utilization of memory is efficiently made due to range to prefix conversion. For decomposition-based approaches, the rule set features governs the specific field searching complexity, as

number of distinctive rules in a field [52, 53]. Researchers have developed various software solutions for packet classification, however hardware solutions supports dynamic updates and high performance. Qu *et al.*, [54] performed for packet classification on FPGA using a 2-dimensional pipelined architecture. This architecture consists of processing elements, which are self-reconfigurable. It does not require range to prefix conversion since prefix and range match is done in a modular processing element (PE). Further Multiple modular processing elements (PEs) are used to manage a large rules set on a 2D architecture. Then packet classification is done using Striding and clustering method by varying sub-field size and rules count. In this architecture, modification, deletion and insertion operations support by a set of algorithm.

Hatami & Bahramgiri [55] proposed RTST (Range based ternary search tree) to achieve fast lookup flow tables in SDN (software define network). In order to implement RTST, a parallel multi-pipeline architecture is presented to get low latency and high throughput. It also supports high clock rates and dynamic updates. Consequently, it achieved the memory efficiency. A 5 dimensional classification based dual stage bloom filter engine (2sBFCE) [56], this system uses 128k bytes of memory and is implemented using 4k rules. For to classify the throughput over 6Gbps, this system requires an average of 26 clock cycles. An Extended version of TCAM (ETCAM) called Distributed Cross producing of Field Labels (DCFL) is implemented on hardware reconfigurable device [57, 46]. They achieved a throughput of 50 million packets per second using Xilinx Virtex 2 and with 128 samples. The authors claim throughput 24 Gbps, if the same system is implemented on virtex5 FPGA. Jiang & Prasanna [58] performed multi field packet classification using a decision tree based HyperCut algorithms. It is a dual pipeline 2D FPGA architecture that employed precision range cutting and Rule overlap reduction methods for packet classification. This approach uses Xilinx Vertex 5 FPGA memory and achieved 80000 Mbps Throughput for 40 bytes minimum packet size and can support up to 10k rules.

Chang *et al.*, [44] used Xilinx Virtex5 FPGA to develop a novel Set Pruning Multi Bit Trie (SPMT) pipeline architecture. This approach invokes two stages. A wildcard field position based rules sub division is done in first stage. The lengths of prefix based rules sub division is done in second stage. With the Support of 10k rules, they achieved a throughput 100 Gbps. An enabled network virtualization based on open flow switch is addressed [62] and an enhanced 12 tuple fields is developed [45]. The reported multiple matches in Gbps using a novel architecture called BV-TCAM which operates on prefix or exact value of header fields. And a source to destination lookup is done through tree-bitmap. Using 222 rules they claimed a RAM block of less than 20% and power consumption less than 10% without actual implementation on FPGA. They further proclaimed that their method can achieve a throughput of 10 Gbps if it is implemented on advanced FPGA.

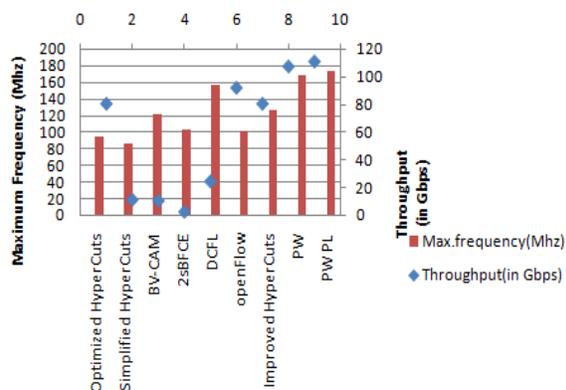
Ganegedara *et al.*, [59] performed a comparative analysis of three packet classification methods called Stride BV, brute force search method, Ternary Content Addressable Memory (TCAM) by varying rule set size from 32 to 2048. This method faces limited on chip

memory for to store larger set of rules. Taylor [12] devised a hard ware set up to achieve high performance using multiple packet classification methods like Distributed Cross Producting of Field Labels (DCFL) leverages filter set characteristics, decomposition, and a novel labeling technique.

Faezipour & Nourani [60] demonstrated a multi match search using two TCAM based architectures. The first or all matches in a packet filter set is done in the first stage and a multi match packet classification using intersection properties is done in the second stage. This method is able to reduce power consumption by an order of two and can perform all possible matches in exactly one TCAM cycle. Chang & Su [43] renovated interval- based range encoding schemes and developed a novel range encoding methods which consumes less TCAM storage space compare to existing techniques. Sun *et al.*, [48] addressed a packet classification problem using set intersections and compared it with bit vector algorithm. This method uses lesser resources to store large rule sets with small m value with faster clock rate. As m increases, it requires more resources and is less attractive than bit vector algorithm with smaller rules set. Khan *et al.*, [61] attained a higher throughput of +100 Gbps using an architecture which is independent of rule-set features. This method has achieved low latency and supports prefix, exact and range match ability.

### III. RESULTS AND DISCUSSION

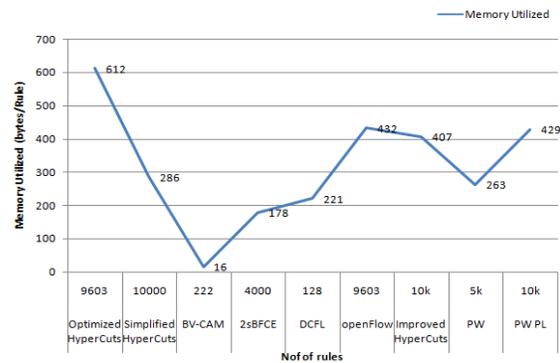
Many authors proposed various packet classification techniques based on various approaches like Optimized hypercuts, Simplified hypercuts, BV-CAM, 2sBFCE, DCFL, Openflow, Improved hypercuts, PW, PW PL, XnorBV etc. All these methods aimed to achieve the parameters like High throughput, Maximum frequency, less memory and power consumption, low latency, less area etc. The performance evaluation of all these methods is represented graphically in the following figures.



**Fig. 1.** Maximum Frequency (MHz) and Throughput (in Gbps).

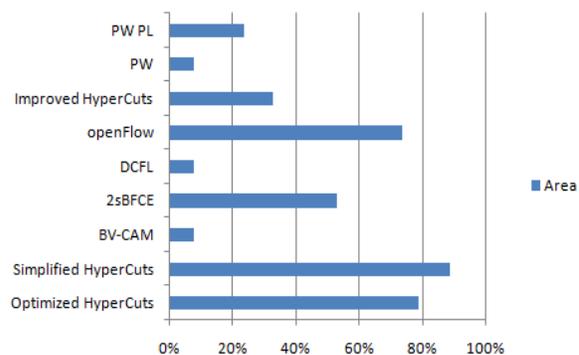
Fig. 1 reveals the throughput attained for various existing methods which are operated at different frequencies. It is observed that the PWPL method has achieved maximum throughput of 110.73 Gbps and operated at maximum frequency of 173.02 MHz with the use of 10k rules. But the 24% of the area consumed while usage of memory 429 bytes/rule. Also the

methods PW and DCFL have achieved considerable throughput and frequency.

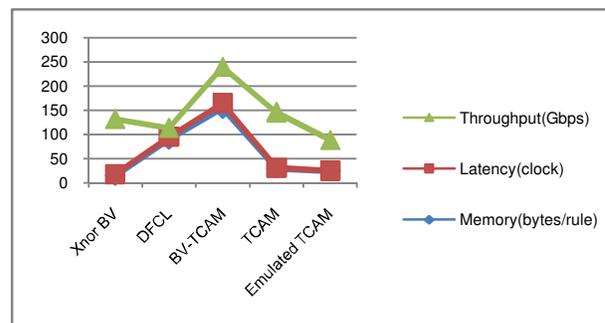


**Fig. 2.** Memory Utilization and No of rules.

Fig. 2 exhibits the memory utilization corresponding to various rules of various approaches. It is noticed that Simplified Hypercut method has utilized less memory of 286 bytes/rule for 10k rules. Even PW, 2sBFCE and BV-CAM has consumed less memory while using 5k, 4k and 222 rules respectively.



**Fig. 3.** Area utilization (in %).



**Fig. 4.** Throughput (Gbps), Latency and Memory utilization (bytes/rule).

Fig. 3 represents pictorial view of area utilized by the various existing methods. It is clear that PW, DCFL and BV-CAM has used the very less percentage of area (8%) compared to other methods.

The following Figs. (4, 5) represents the various approaches like Xnor-BV, DCFL, BV-TCAM, TCAM, Emulated TCAM that used 512 rules, latency, throughput maintained at 300 MHz along with power consumption.

The Throughput, Latency and Memory utilization of different existing methods in Fig. 4. Of them the Xnor-

BV method is observed to be optimum in attaining maximum throughput (114 Gbps), with low latency and memory utilization (15 bytes/rule) compared to remaining methods. And Fig. 5 represents that this method is also efficient in less power consumption (0.3 mw)

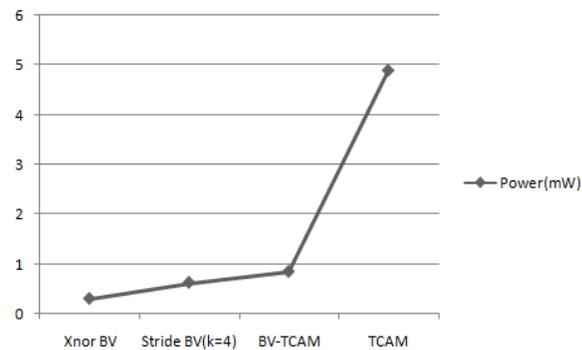


Fig. 5. Power Consumption (mW).

#### IV. CONCLUSION AND FUTURE SCOPE

In this paper, various packet classification algorithms and architectures with their advantages and disadvantages are described. It is observed that most of the state of art techniques are poorly scaled either in terms of search time or memory utilization. To conclude, in the existing algorithms, there is still huge possibility to improve the system performance. The prime significance challenges of packet classifications include (i) Support bulk rule sets, (ii) Sustainance of optimum performance, (iii) Make possible dynamic updates. Hence, tremendous research is to be done in academics and industry to resolve these problems through the adoption of robust solutions. Also there is a need of hybrid techniques, which are generated, based on various methods implemented on FPGAs and are described earlier to produce high performance packet classifiers.

Hence it is need a develop a much sophisticated systems for packet classification which makes use of maximum number of rules in producing high throughput, while using less memory and power consumption on an FPGA.

#### REFERENCES

[1]. Liu, D., Hua, B., Hu, X., & Tang, X. (2006). High-performance packet classification algorithm for many-core and multithreaded network processor. In *Proceedings of the 2006 international conference on Compilers, architecture and synthesis for embedded systems*, 334-344.

[2]. Qi, Y., Xu, L., Yang, B., Xue, Y., & Li, J. (2009). Packet classification algorithms: From theory to practice. In *IEEE INFOCOM 2009*, 648-656. IEEE.

[3]. D'Ambrosia, J., Law, D., & Nowell, M. (2008). 40 gigabit Ethernet and 100 gigabit Ethernet technology overview. *Ethernet Alliance White Paper*, 1-16.

[4]. Lakshminarayanan, K., Rangarajan, A., & Venkatachary, S. (2005). Algorithms for advanced packet classification with ternary cams. In: *ACM SIGCOMM Computer Communication Review*, 35, 193-204.

[5]. Meiners, C. R., Liu, A. X., & Torng, E. (2010). *Hardware based packet classification for high speed internet routers*. Springer Science & Business Media.

[6]. Ganegedara, T., & Prasanna, V. K. (2012). StrideBV: Single chip 400G+ packet classification. In *2012 IEEE 13th International Conference on High Performance Switching and Routing* (pp. 1-6). IEEE.

[7]. Agarkar, B. S., & Kulkarni, U. V. (2013). A Novel Technique for Fast Parallel Packet Classification. *International Journal of Computer Applications*, 76(4), 18-25.

[8]. Le, H., & Prasanna, V. K. (2011). Scalable tree-based architectures for IPv4/v6 lookup using prefix partitioning. *IEEE Transactions on Computers*, 61(7), 1026-1039.

[9]. Qu, Y. R., & Prasanna, V. K. (2016). High-performance and dynamically updatable packet classification engine on FPGA. *IEEE Transactions on Parallel and Distributed Systems*, 27(1), 197-209.

[10]. Yu, F., Katz, R. H., & Lakshman, T. V. (2005). Efficient multimatch packet classification and lookup with TCAM. *IEEE Micro*, 25(1), 50-59.

[11]. Vamanan, B., & Vijaykumar, T. N. (2011). Tree CAM: decoupling updates and lookups in packet classification. In *Proceedings of the Seventh Conference on emerging Networking EXperiments and Technologies* (pp. 1-12).

[12]. Taylor, D. E. (2005). Survey and Taxonomy of Packet Classification Techniques. *ACM computing Surveys*, 37(3), 238-275.

[13]. Srinivasan, V., Varghese, G., Suri, S., & Waldvogel, M. (1998). Fast and scalable layer four switching. In *Proceedings of the ACM SIGCOMM'98 conference on Applications, technologies, architectures, and protocols for computer communication*, 191-202.

[14]. Lakshman, T. V., & Stiliadis, D. (1998). High-speed policy-based packet forwarding using efficient multi-dimensional range matching. *ACM SIGCOMM Computer Communication Review*, 28(4), 203-214.

[15]. Gupta, P., & Mckeown, N. (2000). Classification using Hierarchical Intelligent Cuttings. *IEEE Micro*, 20(1), 34-41.

[16]. Singh, S., Baboescu, F., Varghese, G., & Wang, J. (2003). Packet classification using multidimensional cutting. In *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, 213-224.

[17]. Gupta, P., & McKeown, N. (1999). Packet Classification on Multiple Fields. *Proc. ACM SIGCOMM*, 147-160.

[18]. Srinivasan, T., Dhanasekar, N., Nivedita, M., Dhivyakrishnan, R., & Azeezunnisa, A. A. (2006). Scalable and parallel aggregated bit vector packet classification using prefix computation model. In *International Symposium on Parallel Computing in Electrical Engineering (PARELEC'06)*, 139-144. IEEE.

[19]. Ahmed, O., Areibi, S., & Grewal, G. (2013). Hardware accelerators targeting a novel group based packet classification algorithm. *International Journal of Reconfigurable Computing*, 1-33.

[20]. Srinivasan, V., Suri, S., & Varghese, G. (1999). Packet classification using tuple space search. In *Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communication*, 135-146.

- [21]. Warkhede, P., Suri, S., & Varghese, G. (2001). Fast packet classification for two-dimensional conflict-free filters. In *Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (Cat. No. 01CH37213)*, 3, 1434-1443.
- [22]. Lee, F. Y., & Shieh, S. (2006). Packet classification using diagonal-based tuple space search. *Computer Networks*, 50(9), 1406-1423.
- [23]. Vamanan, B., Voskuilen, G., & Vijaykumar, T. (2010). Efficuts: Optimizing packet classification for memory and throughput. In *Proc. ACM Special Interest Group Data Commun. Conf*, 207-218.
- [24]. Van Lunteren, J. (2001). Searching very large routing tables in wide embedded memory. In *GLOBECOM'01. IEEE Global Telecommunications Conference (Cat. No. 01CH37270)*, 3, 1615-1619.
- [25]. Broder, A., and Karlin, A. (1990). Multilevel adaptive hashing. *Proceedings of the 1st ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 43-53.
- [26]. Azar, Y., Broder, A., Karlin, A., & Upfal, E. (1994). Balanced Allocations, In *Proceedings of the 26th ACM Symposium on the Theory of Computing*, pp. 593-602.
- [27]. Broder, A., & Mitzenmacher, M. (2001). Using multiple hash functions to improve IP lookups. In *Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (Cat. No. 01CH37213)*, 3, 1454-1463. IEEE.
- [28]. Vocking, B. (1999). How Asymmetry Helps Load Balancing. In *Proc. of the 40th IEEE Symp. on Foundations of Computer Science*, pp. 131-141.
- [29]. Bloom, B. H. (1970). Space/Time Trade-offs in Hash Coding with Allowable Errors. *J. Communication of the ACM*, 13(7), 422-426.
- [30]. Fan, L., Cao, P., Almeida, J., & Broder, A. Z. (2000). Summary cache: a scalable wide-area web cache sharing protocol. *IEEE/ACM transactions on networking*, 8(3), 281-293.
- [31]. Dharmapurikar, S., Krishnamurthy, P., & Taylor, D. E. (2003). Longest prefix matching using Bloom filters. In *ACM Sigcomm*, 202-212
- [32]. Waldvogel, M., Varghese, G., Turner, J., & Plattner, B. (1997). Scalable high speed IP routing lookups. In *Proceedings of the ACM SIGCOMM'97 Conference on Applications, technologies, architectures, and protocols for computer communication*, 25-36.
- [33]. Qin, H., Sasao, T., & Butler, J. T. (2007). On the Design of LPM Address Generators Using Multiple LUT Cascades on FPGAs. *International Journal of Electronics*, 94(5), 451-467.
- [34]. Chen, Y., & Oguntoyinbo, O. (2009). Power efficient packet classification using cascaded bloom filter and off-the-shelf ternary CAM for WDM networks. *Computer Communications*, 32(2), 349-356.
- [35]. Juniper Networks E320 Broadband Services Router, [https://www.apac-juniper.net/juniper\\_public/campaign/pdf/iptv/en/100103.pdf](https://www.apac-juniper.net/juniper_public/campaign/pdf/iptv/en/100103.pdf).
- [36]. Cong, J., Liu, B., Neuendorffer, S., Noguera, J., Vissers, K., & Zhang, Z. (2011). High-level synthesis for FPGAs: From prototyping to deployment. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 30(4), 473-491.
- [37]. Xtensa Instruction Set Architecture (ISA), (2012) Technical publications, Tensilica Inc., Santa Clara, CA 95054.
- [38]. Shah, D., & Gupta, P. (2001). Fast updating algorithms for TCAM. *IEEE Micro*, 21(1), 36-47.
- [39]. Liu, H. (2002). Routing table compaction in ternary CAM. *IEEE Micro*, 22(1), 58-64.
- [40]. Panigrahy, R., & Sharma, S. (2002). Reducing TCAM power consumption and increasing throughput. In *Proceedings 10th Symposium on High Performance Interconnects*, 107-112. IEEE.
- [41]. Zane, F., Narlikar, G., & Basu, A. (2003). CoolCAMs: Power-efficient TCAMs for forwarding engines. In *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No. 03CH37428)*, 1, 42-52.
- [42]. Song, H., & Lockwood, J. W. (2005). Efficient packet classification for network intrusion detection using FPGA. In *Proceedings of the 2005 ACM/SIGDA 13th international symposium on Field-programmable gate arrays*, 238-245.
- [43]. Chang, Y. K., & Su, C. C. (2007). Efficient TCAM encoding schemes for packet classification using gray code. In *IEEE GLOBECOM 2007-IEEE Global Telecommunications Conference*, 1834-1839.
- [44]. Chang, Y. K., Lin, Y. S., & Su, C. C. (2010). A high-speed and memory efficient pipeline architecture for packet classification. In *2010 18th IEEE Annual International Symposium on Field-Programmable Custom Computing Machines*, 215-218.
- [45]. Jiang, W., & Prasanna, V. K. (2012). Scalable packet classification on FPGA. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 20(9), 1668-1680.
- [46]. Taylor, D. E., & Turner, J. S. (2005). Scalable packet classification using distributed crossproducing of field labels. In *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, 1, 269-280.
- [47]. Li, J., Chen, Y., Ho, C., & Lu, Z. (2013). Binary-tree-based high speed packet classification system on FPGA. In *The International Conference on Information Networking 2013 (ICOIN)*, 517-522. IEEE.
- [48]. Sun, L., Le, H., & Prasanna, V. K. (2011). Optimizing decomposition-based packet classification implementation on FPGAs. In *2011 International Conference on Reconfigurable Computing and FPGAs*, 170-175.
- [49]. Jiang, W., & Prasanna, V. K. (2009). Field-split parallel architecture for high performance multi-match packet classification using FPGAs. In *Proceedings of the twenty-first annual symposium on Parallelism in algorithms and architectures*, 188-196.
- [50]. Gupta, P., and McKeown, N. (2001). Algorithms for packet classification. *IEEE Network*. 15(2), pp. 24-32.
- [51]. Gupta, P., & McKeown, N. (2000). Dynamic Algorithms with Worst Case Performance for Packet Classification. *Proc. of European Commission International Conference (IFIP-TC6)*, 528-539.
- [52]. Qu, Y., Zhou, S., & Prasanna, V. K. (2013). Scalable many-field packet classification on multi-core processors. In *2013 25th International Symposium on Computer Architecture and High Performance Computing*, 33-40. IEEE.

- [53]. Pus, V., & Korenek, J. (2009). Fast and Scalable Packet Classification using Perfect Hash Functions. *Proc. of the ACM/SIGDA international symposium on Field Programmable Gate Arrays (FPGA)*, 229–236.
- [54]. Qu, Y. R., Zhou, S., & Prasanna, V. K. (2013). High-performance architecture for dynamically updatable packet classification on FPGA. In *Architectures for Networking and Communications Systems*, 125-136. IEEE.
- [55]. Hatami, R., & Bahramgiri, H. (2019). High-performance architecture for flow-table lookup in SDN on FPGA. *The Journal of Supercomputing*, 75(1), 384-399.
- [56]. Nikitakis, A., & Papaefstathiou, L. (2008). A Memory-Efficient FPGA-based Classification Engine, in *16th International Symposium on Field-Programmable Custom Computing Machines*, 53-62.
- [57]. Jedhe, G. S., Ramamoorthy, A., & Varghese, K. (2008). A scalable high throughput firewall in FPGA. In *2008 16th International Symposium on Field-Programmable Custom Computing Machines*, 43-52.
- [58]. Jiang, W., & Prasanna, V. K. (2009). Large-scale wire-speed packet classification on FPGAs. In *Proceedings of the ACM/SIGDA international symposium on Field programmable gate arrays*, 219-228.
- [59]. Ganegedara, T., Jiang, W., & Prasanna, V. K. (2013). A scalable and modular architecture for high-performance packet classification. *IEEE Transactions on Parallel and Distributed Systems*, 25(5), 1135-1144.
- [60]. Faezipour, M., & Nourani, M. (2009). Wire-speed TCAM-based architectures for multimatch packet classification. *IEEE Transactions on Computers*, 58(1), 5-17.
- [61]. Khan, A. U., Chawhan, M., Suryawanshi, Y., & Kakde, S. (2017). Design of High Performance Packet Classification Architecture for Communication Networks. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, 9(4), 109-115.
- [62]. McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., & Turner, J. (2008). OpenFlow: enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review*, 38(2), 69-74.

**How to cite this article:** Adisheshaiah, Midde and Sailaja, M. (2020). Packet Classification Algorithms and Methods: A Critical Review. *International Journal on Emerging Technologies*, 11(1): 454–461.