Scalable Packet Classification on FPGA

[1]Deepak K. Thakkar, [2]Dr. B. S. Agarkar
[1]Student, [2]Professor
[1]Electronics and Telecommunication Engineering,
[1]Sanjivani college of Engineering, Kopargaon, India.

***Abstract*— Packet classification is used in networking for sorting out packets into flows by comparing packet headers with rules in classifier. A flow is used to decide what action is to be taken on incoming packet. Now a days it is great challenge to develop scalable solutions for advanced packet classification which is having higher performance, supports large rule sets as well as more packet header fields. This paper provides software as well as hardware implementation of HiCuts and HyperCuts algorithms by using Xilinx Vivado2016.1 software tool and Genesys 2 Kintex-7 FPGA development board with Xilinx Kintex-7 FPGA device X7K325T-2FG900C. Thus implementation results shows that HyperCuts algorithm is superior than HiCuts with respect to many parameters such as depth of decision tree, execution time and memory requirements with the different number of rules. The depth of decision tree is 1 and execution time for Hypercuts algorithms is 0.779ns for all cases which is less than that of HiCuts algorithms. Also in many cases the memory requirements of HyperCuts is less than HiCuts algorithm. To test these algorithms we consider ACL rule set.**

***IndexTerms* — ACL, Algorithm, FPGA, Packet classification, Quality of Service**

I. INTRODUCTION

Advanced, routers provide different network services such as firewall, quality of services (QoS), virtual private network, policy routing, traffic billing and some other value added services. In order to provide these services, router needs to classify packets on the basis of predefined rules and the process is called as packet classification. Flow is nothing but a set of packets which match with a particular rule and the collection of rules is called as classifier. Packet mainly consists of two fields: Packet header and payload. Packet header having multiple fields which specifies value ranges and process is called as multi-field packet classification. In traditional computer network applications, packet classification usually consists of fixed 2-tuple fields while advance packet classification consists of multiple fields that are more than 2-tuple fields. In this paper, we are implementing two decision tree based algorithms HiCuts and HyperCuts algorithms. These algorithms implemented on field programmable gate array (FPGA) using Xilinx Vivado2016.1 tool. The target device is Genesys2 kit.

This paper is organized in seven sections. Section I gives the introduction of the work and Section II states the problem definition. Section III reviews related work on basics of packet classification and FPGA based packet classification. Section IV explains decision tree based packet classification and detailed explanation of HiCuts as well as HyperCuts algorithm. In Section V FPGA implementation of our architecture is described in detail. Section VI describes the necessary results after implementation and Section VII concludes the work.

II. PROBLEM DEFINITION

In traditional packet classification, packet consists of mainly 2-fields and fields that mostly used are 32-bits source/destination IP addresses (SA/DA), 16-bits source/destination port numbers (SP/DP) and 8-bit protocol numbers. Rules are nothing but separate entries that classify the packet. Consider the example of 5-tuple classifier having different rule set as shown in table 1. In this example, there are total 10-rules from R1 to R10 and each rule consists of different fields and their associated values, priority as well as action is taken if particular rule is matched. Each rule having match conditions occurs if all 5-fields are matched.

Table 1. Example of 5-tuple classifier

Rule	SA	DA	SP	DP	Prot-ocol	Prio-rity	Acti-on
R1	*	*	2-9	6-11	*	1	act0
R2	1*	0*	3-8	1-4	10	2	act0
R3	0*	0110*	9-12	10-13	11	3	act1
R4	0*	11*	11-14	4-8	*	4	act2
R5	011*	11*	1-4	9-15	10	5	act2
R6	011*	11*	1-4	4-15	10	5	act1
R7	110*	00*	0-15	5-6	11	6	act3
R8	110*	0110*	0-15	5-6	*	6	act0
R9	111*	0110*	0-15	7-9	11	7	act2
R10	111*	00*	0-15	4-9	*	7	act1

Advance packet classification consisting larger number of header fields to be matched. Recently proposed one of the major processing engine that is OpenFlow switch for packet classification which consisting of 12-tuple header fields that are ingress port, source/destination Ethernet addresses, Ethernet type, VLAN ID, VLAN priority, source/destination IP addresses, IP protocol, IP type of service (ToS) bits and source/destination port numbers. Consider the example of 12-tuple fields which having 16-bit Ethernet source/destination addresses, 8-bit sorce/desination IP addresses and 4-bit protocol numbers. Table 2 shows example rule set for OpenFlow switch.

In given classifier, there are 10- rules R1 to R10 and 12-header fields. If a particular packet is considered for matching rule then all the fields should be matched. There are so many rules matching at a time but rule which having highest priority is considered first and according to that action is taken. In particular there are two types of rules: simple rule and complex rule. Simple rule is nothing but all values are given, no wildcard entries occurred while complex rules which having larger wild card entries. Tables 2 shows that R10 acts as simple rule and R1 to R9 acts as complex rules.

Table 2. Example of OpenFlow rule set.

Rule	Ingr port	Eth Src	Eth Dst	Eth type	VLAN ID	VLAN Priority	IP src (SA)	IP dst (DA)	IP Proto-col	IP ToS	Port src(SP)	Port dst(DP)	Acti-on
R1	*	00:13	00:06	*	*	*	*	*	*	*	*	*	act0
R2	*	00:07	00:10	*	*	*	*	*	*	*	*	*	act0
R3	*	*	00:FF	*	*	*	*	*	*	*	*	*	act1
R4	*	00:1F	*	0x8100	100	5	*	*	*	*	*	*	act1
R5	*	*	*	0x0800	*	*	*	01*	*	*	*	*	act2
R6	*	*	*	0x0800	*	*	001*	11*	TCP	*	10	15	act0
R7	*	*	*	0x0800	*	*	001*	11*	UDP	*	2	11	act3
R8	*	*	*	0x0800	*	*	100*	110*	*	*	5	6	act1
R9	5	00:FF	00:00	0x0800	4095	7	0011*	1100*	TCP	0	2	5	act0
R10	1	00:1F	00:2A	0x0800	4095	7	01000001	10100011	TCP	0	2	7	act0

III. Related Work

In recent years, variety of packet classification schemes has been proposed to solve the general problem of multi-field packet classification. Multi-field packet classification is evolved from traditional 2-tuple packet classification by simply adding packet header fields. A brief review of major research work is carried out in the field of packet classification and various algorithms that are implemented on software as well as hardware are given below.

Taylor [1] proposes survey and taxonomy of packet classification techniques; in different packet classification technique they combined algorithmic and architectural approach to solve packet classification problems. Also by using taxonomy based high level approach they provide recent solution to the problems as well as various packet classification algorithms by using particular example are easily understand.

Gupta and McKeown [2] introduced algorithm, hierarchical intelligent cutting that is HiCuts algorithm. HiCuts performs well on any classifier available and builds the decision tree from the particular classifier. Every time packet arrives, it can traverse decision tree and find out leaf node which containing small number of rules. A linear search is used to find out best matching rule among the different rules. This algorithm is suitable for hardware implementation as well as it can implement by using software also.

Singh *et al*. [3] proposed new algorithm, HyperCuts algorithm in which we have to take multiple cuts simultaneously. HyperCuts algorithm is decision tree based algorithm and it is better than HiCuts algorithm in most of the ways. In HyperCuts algorithm the depth of decision tree is reduced upto 1, it requires less memory, optimized for speed as well as fast updates as compared to the HiCuts algorithms.

Balajee Vamanan *et al*. proposed new decision tree based algorithm known as EffiCuts [4]. This algorithm proposed new four ideas: 1. to reduce overlap of small and large rules in classifier they try to build separate decision tree for separate rule sets. 2. to reduce the multiple trees because that degrades the throughput so that they mixing small or large rules in one dimension. 3. usually we take cut along any field in unequal ways so that distribute rules in different leaf nodes. Thus they try to make cut in equal way along the fields. 4. to achieve fewer accesses per node than HiCuts and HyperCuts, they co-locate parts of node and its children. Thus by using these different ideas EffiCuts requires less memory than other decision tree based algorithms and performance is also get improved.

Luo et al. [5] propose new method for packet classification called as explicit range search that allows more cuts per node than HyperCuts algorithm. So that tree height is dramatically reduces but the memory consumption increases. At every internal node different amount of memory needed to traverse the child node which is not feasible to pipelining. Also authors do not implement this method on FPGA so the actual performance results are not clear.

Kennedy and Wang [6] implemented HyperCuts algorithms on Stratix-III FPGA. For packet classification they used networking equipment that for sorting the packets into flows by comparing headers to rules and find out best matching rules among the list of rules. By using this they can classify up to 433 million packets per second which containing tens of thousands of rules set and peak

power consumption is 9.03 W. Hardware accelerator uses modified version of HyperCuts algorithm which reduces amount of memory needed for larger rule sets so that which is best fit in the on chip memory of FPGA.

Jang and Prasanna [7] implemented decision tree based algorithm by using linear multi-pipeline architecture on virtex-5 FPGA for multi-field packet classification. They considered advance packet fields problems which consisting of more than 5-tuple header fields would classified. By using different techniques for decision tree based packet classification they reduces memory requirements such that 10K 5-tuple rules or 1K 12-tuple rules could fit into on-chip memory of single FPGA. Simulation and FPGA implementation shows the best results among the different solutions.

From the literature survey, it is observed that there is a need to develop a system which classify large packet and provides improved performance with less memory requirement. For achieving this, system uses mainly two algorithms named as HiCuts and HyperCuts. The work is more effective by using advanced Genesys 2 kit uses Kintex-7 FPGA as well as latest software Xilinx Vivado2016.1 uses for simulation of design.

IV. DECISION TREE BASED PACKET CLASSIFICATION

Next generation packet classification is extension of traditional 5-tuple packet classification which can studied in past decade. Here mainly decision tree based packet classification algorithms that are HiCuts and HyperCuts algorithms are explained in detail. The simplest way to match the header fields of packet to rules is linear search through one rule at a time starting from highest priority rule to lowest priority rule. But for this process large amount of processing time is required and which is difficult to classify packets at the speeds required for core of network. This large amount of processing time is reduced by HyperCuts algorithms. HyperCuts packet classification algorithms is mainly works by breaking the larger rule sets into groups and each group consists of small number of rules which is suitable for linear search. Each group of rules stored in leaf node of decision tree which is suitable for finding best matching rule through linear search.

Consider the example of Hicuts and Hypercuts decision trees for 2-fields in Table 1. The fields are either sip and dip or sp and dp. The rules which can be used to build decision trees are R1-R5. Figure 2 and Figure 3 shows the HiCuts and HyperCuts decision tree respectively builds by using rule classifier. In HiCuts algorithms only one cut taken along the any field while in HyperCuts algorithms simultaneously cutting takes place along all fields.

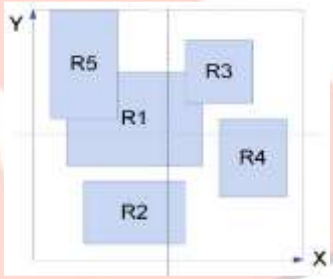

Fig.1 An example of 5-rule classifiers in 2-dimensions.

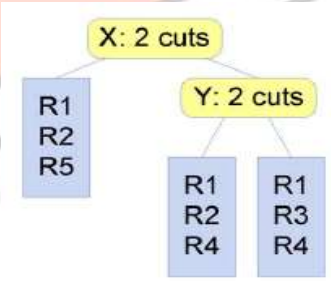

Fig.2 A possible tree (For HiCuts) with binth=3 for the example classifier in figure.1.

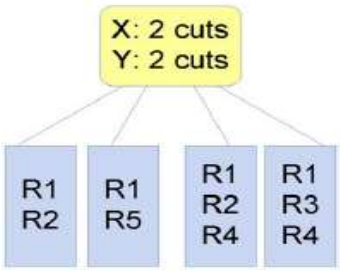

Fig.3 A possible tree for HyperCuts) with binth=3 for the example classifier in figure.1.

V. IMPLEMENTATION

The architecture of packet classification engines having two modules as shown in Figure 4. The first module is tree traverser that is used to traverse decision tree until the empty node is reached means that there is no matching rule or leaf node is reached. Leaf node is reached means that tree traverser passing header and information to the second module that is leaf node searcher. The second module compares header and rules in the leaf node until best matching rule get or no matching rule condition occurs. The leaf node searcher consists of two comparator block works in parallel that allows two rules to be searched on every memory access.

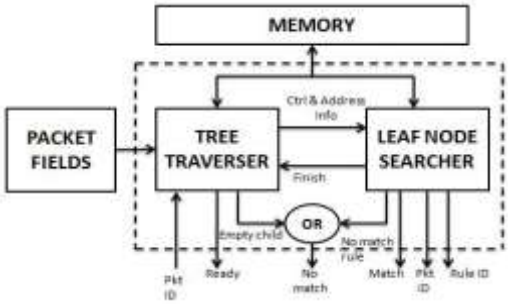

Fig.4 Architecture of packet classification engine

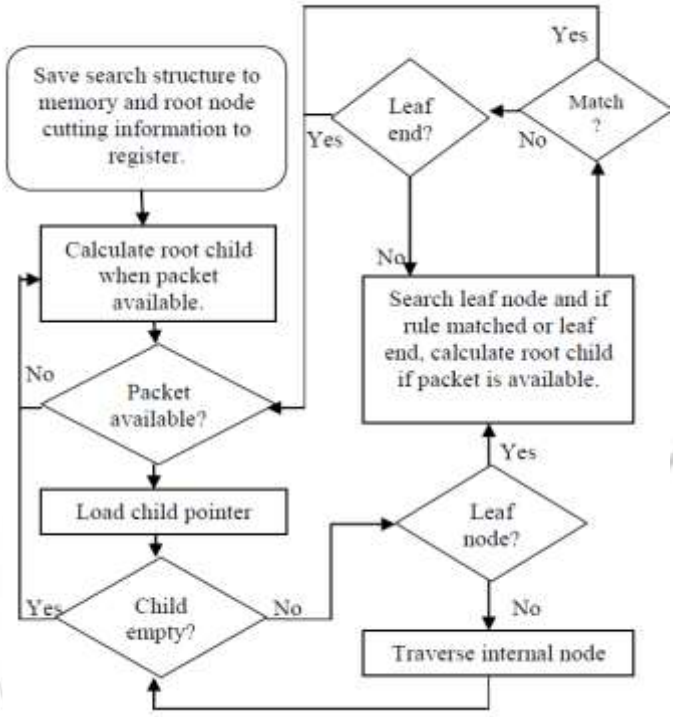

Fig.5 Operation of packet classification engine

In tree traverser, information on root node of decision tree is stored in registers so that tree traverse is classify new packets and old packets compared with rules in leaf nodes. Also pipelining increases throughput having single packet with two clock cycles if root and leaf node of decision tree doesn't having more than two rules.

Figure 5 shows the Flowchart explain the detail operation of packet classification. The engine designed in such a way that it traverses root node or internal node in one memory access. Also it can search leaf node with two rules on every memory access.

The classifier having multiple packet classification engines works in parallel. The maximum speed of engine is much slower than speed of the internal memory of FPGAs because of the comparator blocks used in engine that generates logic delays. So that multiple engines needed and classifier maximizes throughput. Another reason of using multiple engines is that rule sets having many wildcard entries that are divided into groups. After dividing the rule sets it is easier to build decision tree with small leaf nodes which ultimately increases throughput and reduces memory.

Figure 6 shows the architecture of classifier containing total eight packet classification engine working in parallel. Classifier takes advantage of FPGA which having dual internal memory so that two classifiers working in parallel and uses same memory. Each classifier reads data from data port and packet header buffers stores the header fields of incoming packet. It works on the basis of first come first served basis and produces packet ID output that with matching rules are outputted in same order of incoming packet. The four packet classification engines of classifier working in parallel at the same clock speed but out of phase. Sorter logic block are used for matching rule IDs are outputted in the order of packet input. Also sorter block accepts Match, No Match, Rule ID, Packet ID signals and according to the highest priority rule it produces different results.

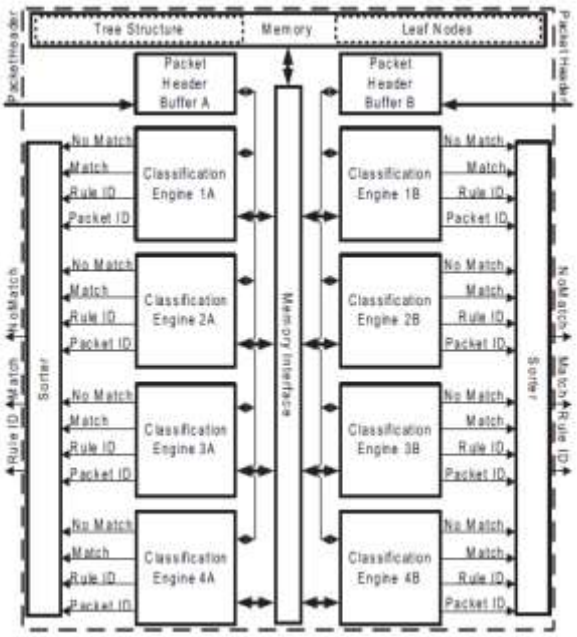

Fig.6 Architecture of classifier

VI. EXPERIMENTAL RESULTS

The proposed system are simulated on software tool such as Xilinx Vivado2016.1 using two decision tree based algorithms namely HiCuts and HyperCuts as explained in earlier chapter. The example of this algorithm as seen earlier for 5 rules, that same example is simulated and finding out best matching rules. Also the number of rules is increases from 5 rules to 25 rules for both HiCuts and HyperCuts algorithm. The same design is done on the hardware and getting matching rules on led.

Fig.7 Simulation result for Hicuts and HyperCuts algorithms for 5 rules

Fig. 8 Simulation result for Hicuts and HyperCuts algorithms for 10 rules

The simulation result shown in figure 7 indicates that at the rising edge of the clock we get output and that store in led register. The rulecnt indicates that total number of rules that can be used to simulate the both the algorithms. Here rulecnt shows that 5 rules are used. Also by giving input values sip and dip in the program then it can generates output. In this particular simulation sip and dip values are matching with the rule 1 so led output indicates value as "00000001". In particular 5 rules example $binth$ values is 3 for both HiCuts and HyperCuts algorithms but depth of decision tree is more that is 2 in HiCuts while that of HyperCuts is1.

The above figure shows the simulation of these algorithms for 10 rules which is indicated by the rulecnt "1010". In this simulation by using sip and dip values that are best match with the rule 10 so the led output is given as "00001010" at the rising edge of clock signal. The depth of decision tree is mostly same as that of 5 rules but only $binth$ value is increases that are 6 for HiCuts and 7 For HyperCuts.

Thus similar results are obtain as number of rules increases with respect to binth value, depth of decision tree, execution time as well memory requirements are given in summary table 3. Also graph is plotted for execution time with respect to number of rules as shown in figure 9.

Table 3. Summary of Hicuts and HyperCuts simulation results
(B: binth value, D: Max. tree depth, Execution time in ns and memory in kilo bytes)

No. of rules	HiCuts Algorithm				HyperCuts Algorithm			
	B	D	Execution time (ns)	Memory (KB)	B	D	Execution time(ns)	Memory (KB)
5	3	2	0.8324	428719	3	1	0.779	432802
10	6	2	0.9388	442056.8	7	1	0.779	440904.8
15	7	2	0.8614	443490.4	10	1	0.779	424328
20	10	1	0.7944	441045.6	10	1	0.779	426492
25	12	1	0.7884	430447.2	12	1	0.779	434261.6

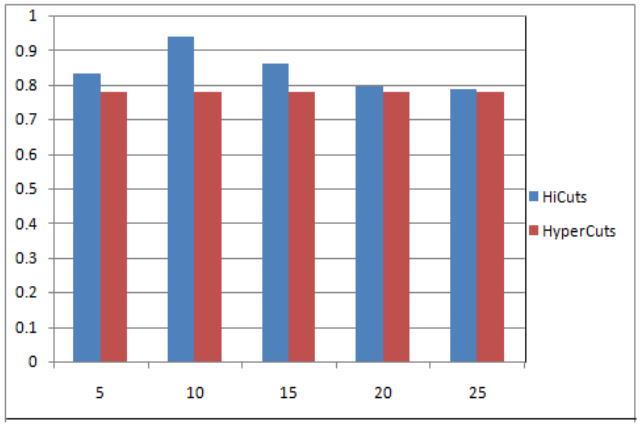

Fig. 9 Execution time for HiCuts and HyprCuts with respect to number of rules

The summary table and graph of these algorithms is indicated that execution time for HyperCuts is less as that of HiCuts algorithm for any number of rule count. Thus HyperCuts is faster than HiCuts algorithm. Also depth of decision tree is always 1 for HyperCuts that is less than HiCuts algorithm and in many cases memory requirements for HyperCuts is less than that of HiCuts algorithm. Thus we can say that in all ways HyperCuts decision tree algorithm is superior than HiCuts algorithm.

Also different experimental results are obtained of power consumption for both HiCuts and HyperCuts algorithms. The power consumption results for both these algorithms with respect to number of rules vary from 5, 10, 15, 20 and 25. We show only results of power consumption of both algorithms with respect to 25 rules and plot the graph of all rule sets.

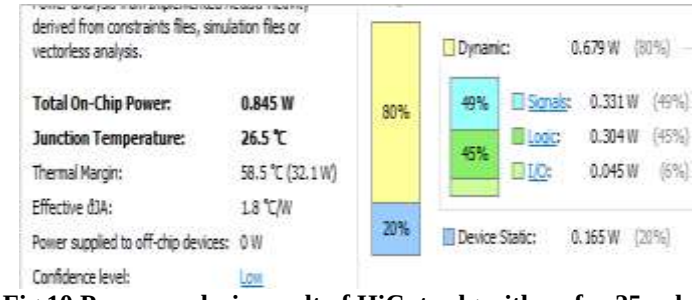

Fig.10 Power analysis result of HiCuts algorithms for 25 rules

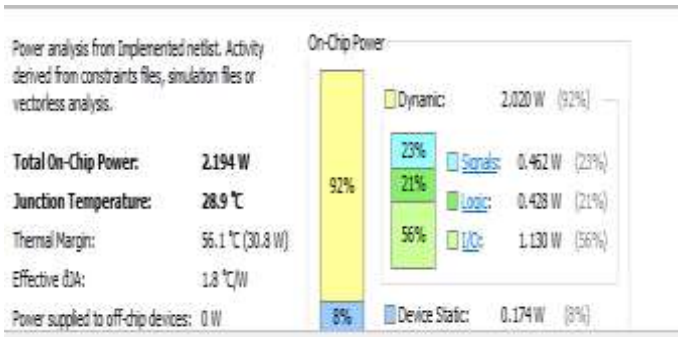

Fig.11 Power analysis result of HyperCuts algorithms for 25 rules

The graph of power in watt with respect to number of rules varies from 5 to 25 are shown in below.

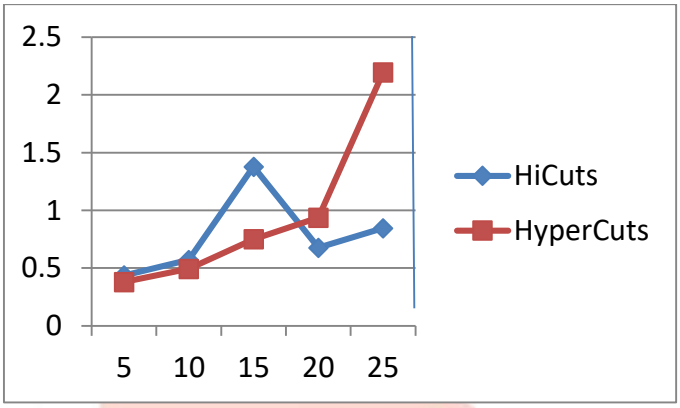

Fig.12 Power in watt for HiCuts and HyperCuts algorithms with respect to number of rules

From the above graph it is clear that, as the number of rules increases then power for HyperCuts algorithm is increases but in case of HiCuts algorithm for 15 numbers of rules power is increases with respect to number of rules then power is drastically reduces and further increases with small amount.

Finally we obtain the utilization summary for both HiCuts and HyperCuts algorithms.

Table 4. Utilization Summary:

Parameters	HiCuts algorithm	HyperCuts algorithm
Slice LUTs(203800)	114	153
Slice Registers(407600)	12	12
Slice (50950)	31	42
LUT as logic(203800)	114	153
LUT flip-flop pairs(203800)	6	6
Bonded IOBs(500)	18	18
IBUFDS(480)	1	1
BUFGCTRL(32)	1	1

VII. CONCLUSION

In this paper, decision tree based packet classification algorithm that are HiCuts and HyperCuts algorithms are implemented by using Genesys 2 Kintex-7 FPGA development board with Xilinx Kintex-7 FPGA device XC7K325T-2FFG900C.The HiCuts and HyperCuts both algorithms are simulated and implemented on kit. Thus it is observed that HyperCuts algorithm is superior than HiCuts with respect to many parameters such as depth of decision tree, execution time and memory requirements with the different number of rules. The depth of decision tree is 1 and execution time for Hypercuts algorithms is 0.779ns for all cases which is less than that of HiCuts algorithms. Also in many cases the memory requirements of HyperCuts is less than HiCuts algorithm. Thus both the algorithms work better way on software as well as on hardware.

REFERENCES

[1] D. E. Taylor, "Survey and Taxonomy of Packet Classification Techniques," ACM computing Surveys, Vol. 37, No. 3, Sept. 2005, pp.238-275.

[2] P. Gupta and N. Mckeown, "Classification using Hierachical Intelligent Cuttings," IEEE Micro, Vol. 20, No.1, Feb 2000, pp.34-41.

[3] S.Singh, F. Baboescu, G.Varghese and J. Wang, "Packet Classification using Multidimensional Cutting," in Proc. ACM special Interest Group Data Commun Conf., Aug. 2003, pp. 213-224.

[4] B. Vamanan, G. Voskuilen and T. Vijaykumar, "Efficuts: Optimizing packet classification for memory and throughput," in Proc. ACM Special Interest Group Data Commun. Conf., Aug. 2010, pp. 207–218.

[5] Y. Luo, K. Xiang and S. Li, "Acceleration of decision tree searching for IP traffic classification," in *Proc. ANCS*, 2008, pp. 40-49.

[6] Alan Kennedy and Xiaojun Wang, "Ultra High Throughput Low Power Packet Classification," IEEE Transaction on Very large scale integration (VLSI) systems, Vol.22, No.2, Feb. 2014, pp.286-299.

[7] Weirong Jiang and Vector K. Prasanna, "Scalable Packet Classification on FPGA," IEEE Transactions on very large scale integration (VLSI) systems, Vol. 20, No. 9, Sept. 2012, pp. 1668-1680.

[8] Pankaj Gupta and Nick Mckeown, "Packet classification on multiple fields," in Proc. ACM Special Interest Group Data Commun Conf., Sept. 1999, pp.147-160.

[9] H. Lim, N. Lee, Geumdan Jin, Jungwon Lee, Youngju Choi and ChanghoonYim, "Boundary Cutting for Packet Classification," IEEE Transactions on Networking, Vol.22, No.2, April 2014, pp.443-456.

[10] Yuxian Qi, LianghongXu, Baohua Yang, YiboXue and Jun Li, "Packet classification Algorithms: From Theory to Practice," in Proc. of IEEE INFOCOM 2009.

[11] Haoyu Song and John W. Lockwood, "Efficient Packet Classification for Network Intrusion Detection using FPGA", ACM1, Feb.2005.

[12] R. Sathesh Raaj and J. Kumarnath, "FPGA Based Packet Classification using Multi-pipeline Architecture", IJCSMC, Vol.4, Issue.4 , Apr. 2015, pp. 541-548.