International Journal of Engineering Research-Online
A Peer Reviewed International Journal
Articles available online http://www.ijoer.in

Vol.2., Issue.6, 2014

REVIEW ARTICLE

ISSN: 2321-7758

A SURVEY PAPER ON LOW POWER PACKET CLASSIFICATIONBASED ON DECISION TREE BASED ALGORITHM

SWAPNA S.R[1], Mrs.SHIJI A.S[2], Dr.SREEJA MOLE S.S[3],

[1]PG Student,ECE Department, Narayanaguru College of Engineering,Manjalumoodu, Tamil Nadu, KK District,India,

[2]Assistant Professor, ECE Department,Narayanaguru College of Engineering,Manjalumoodu, KK District, Tamil Nadu,India,

[3]Head of the Deparment ECE, Narayanaguru College of Engineering,Manjalumoodu Tamil Nadu, KK District,India,

Article Received: 28/11/2014 | Article Revised on:19/12/2014 | Article Accepted on:22/12/2014

SWAPNA S.R

ABSTRACT

The growing number of tasks that need to be carried out, which places the network processor under great pressure. Thedesign and implementation of an energy-efficient packet classification hardware accelerator that can relievea network processor's processing engines of the difficult and power hungry networking task of packet classification.Packet classification is used by networking equipmentto sort packets into flows by comparing their headers to a list of rules, with packets placed in the flow determined by the matched rule. A flow is used to decide a packet's priority and the manner in which it is processed. Packet classification is a difficult task due to the fact that all packets must be processed at wire speed.Here after literature survey this paper concluded that hardware acceleratoruses a modified version of the HyperCuts packet classification algorithm uses new pre-cutting process which reduces theamount of memory needed to save the search structure for large ruleset. This allows higher clock speeds and thus obtaining higher throughputs. This modified algorithm also removes the needfor floating point division to be performed when classifying a packet and it is small enough to fit in the on-chip memory of an FPGA.

Keywords–Packet Classification, Low Power, Accelerator, HiCuts, Hyper Cuts, Rules, Modular

INTRODUCTION

Usage of internet increases day by day because of its ease of access through a widerange of devices such as desktops, notebooks, tablets, and smartphones. These results in real strainon the networking equipment needed to inspect and process the resultant traffic. A survey showed that[1] this simple access has allowed Internet penetration to reach 32.7%of the world's population by December 2011, with the number of Internet users growing by528% between 2000 and 2011. This survey also showed that the U.S. had over 108 million internet users in 2000and in 2001, it becomes in billion range.Thus

when considering that the total amount of energy used in the year 2000 by various networking devices in the U.S. equated to the yearly output of a typical nuclear reactor unit. This means that the currentamount of energy used by networking devices worldwide could exceed the yearly output of 21nuclear reactor units. Therefore Power consumption should be a key concern when designingany new networking equipment for solving ever increasing amount ofnetwork traffic.Network processors are key components used to process packets as they pass through a network.Main functions were packet fragmentation and reassembly, encryption,forwarding, and classification. Reducing the pressure of Network processor by addition of extra processing capacity is not easy due to factors such as silicon limitations and tight power budgets. Ramping up clock speeds to gain extra performance is difficult due to physical limitations in the silicon used to create these devices, while increasing the number of processing cores can cause difficulty when it comes to writing the software needed to control the network processors. Both these approaches also lead to large increases in power consumption due to the extra heat generated by increasing the clock speed and the extra transistors needed to increase the number of processing cores.By using of hardware accelerators dedicated to the mostheavy tasks of a network processor can help toreduce power consumption while increasing processing capacity.This is because a hardware accelerator can be designedto have fewer transistors than that of the general-purposeprocessors used in multi-core network processors. It can also process more data than a general-purposeprocessor while running at slower clock speeds as they areoptimized to carry out specific tasks. Large savings in powerconsumption can occur due to high reduction in clockspeed and number of transistors.

PACKET CLASSIFICATION ALGORITHM

Large number of packet classification algorithms havebeen published in the past decade. Most of those algorithms fall into twocategories:

1) Decomposition-based Algorithm
2) Decision-tree-based Algorithm

Decomposition - based algorithms perform independent search on each fieldand eventually combine the search results from all fields. These types of algorithms are desirable for hardware implementation dueto their parallel search on multiple fields. Main disadvantage is that substantialstorage is usually needed to merge the independentsearch results in order to obtain the final result. Thus decompositionbasedalgorithms have poor scalability, and work well only for small-scale rule sets.

Decision-tree-based algorithms takethe geometric view of the packet classification problem. Here eachrule defines a hypercube in a *d*-dimensional space where *d*is the number of header fields considered for packet classification.Each packet defines a point in this *d*-dimensionalspace. The decision tree construction algorithm employsseveral heuristics to cut the space recursively into smallersubspaces. Each subspace ends up with fewer rules, which helpsto a point a low-cost linear search to find the bestmatching rule.

DECISION TREE BASED ALGORITHM

To store ten- thousands of unique rules in the on-chipmemory of a single FPGA, needs to reduce the memoryrequirement of the decision tree.Here integrate two optimizationtechniques such as rule overlap reduction and precise range cuttings into the decision tree construction algorithm.Starting from the root node with the full rule set, recursivelycut the tree nodes until the number of rule in allthe leaf nodes is smaller than a parameter named listsize.At each node, we need to figure out the set of fields to cutand the number of cuts performed on each field. Therefore restrictthe maximum number of cuts at each node to be 64. In otherwords, an internal node can have 2, 4, 8, 16, 32 or 64 children.For the port fields, instead of the number of cuts need to determine the precise cut points. we restrict the number of cuts on port fields to beat most 2since more bits are needed to store the cut points than to store the number of cuts,. For example, we can have 2 cuts on source addresses(SA), 4 cutson destination addresses(DA), 2 cuts on source port(SP), and 2 cuts on destination port(DP). We do not cut onthe protocol field since the first 4 fields are normally enoughto distinguish different rules in real life [2].

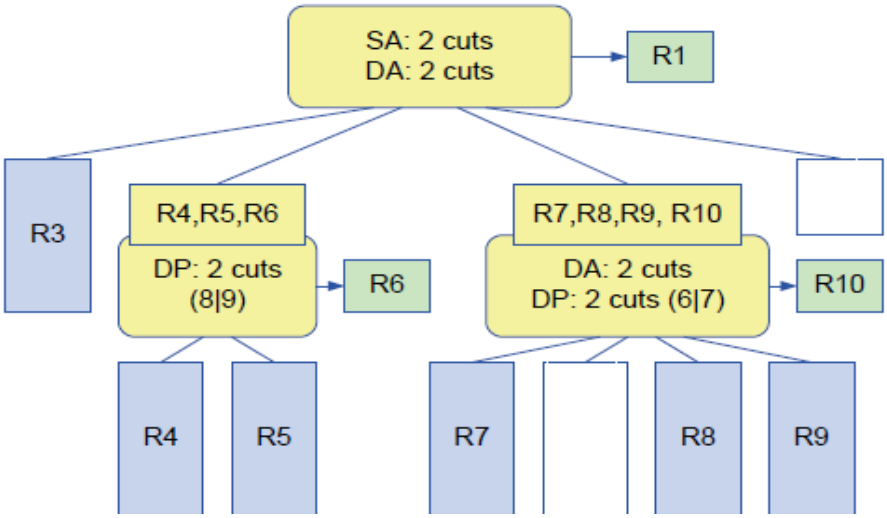
Algorithm To Building A Decision Tree[14]

: Initialize the root node and push it into *nodeList*.
2: **while** *nodeList* _= *null* **do**
3: *n* ← *Pop*(*nodeList*)
4: **if** *n.numRules* < *listSize* **then**
5: *n* is a leaf node. Continue.
6: **end if**
7:*n.numCuts* = 1
8: **while** *n.numCuts* <64 **do**
9: *f* ← *ChooseField*(*n*)
10: **if** *f* is SA or DA **then**
11: *numCuts*[*f*] ← *OptNumCuts*(*n, f*)
12: *n.numCuts* *= *numCuts*[*f*]
13: **else if** *f* is SP or DP **then**

14: *cutPoint*[*f*] ← *OptCutPoint*(*n, f*)
15: *n.numCuts* *= 2
16: **end if**
17: Update the duplication counts of all *r* ∈*n.ruleSet*: *r.dupCount* ← # of copies of *r* after cutting.
18: **while** *n.internalList.numRules* < *listSize* **do**
19: Find *rm* which has the largest duplication count among the rules in *n.ruleSet**n.internalList*.
20: Push *rm* into *n.internalList*.
21: **end while**
22: **if** All child nodes contain less than *listSize* rules **then**
23: Break.
24: **end if**
25: **end while**
26: Push the child nodes into *nodeList*.
27: **end while**

Rule	SA	DA	SP	DP	Protocol	Priority	Action
R1	*	*	2-9	6-11	Any	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	Any	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	Any	6	act0
R9	111*	0110*	0-15	7-9	11	7	act2
R10	111*	00*	0-15	4-9	Any	7	act1

Table 1 shows a simplified example, where each rule contains match conditions for 5fields: 8-bit source and destination addresses, 4-bit source and destination port numbers, and a 2-bit protocol value.Figure 1 shows the decision tree constructed for the rule set given in Table 1.

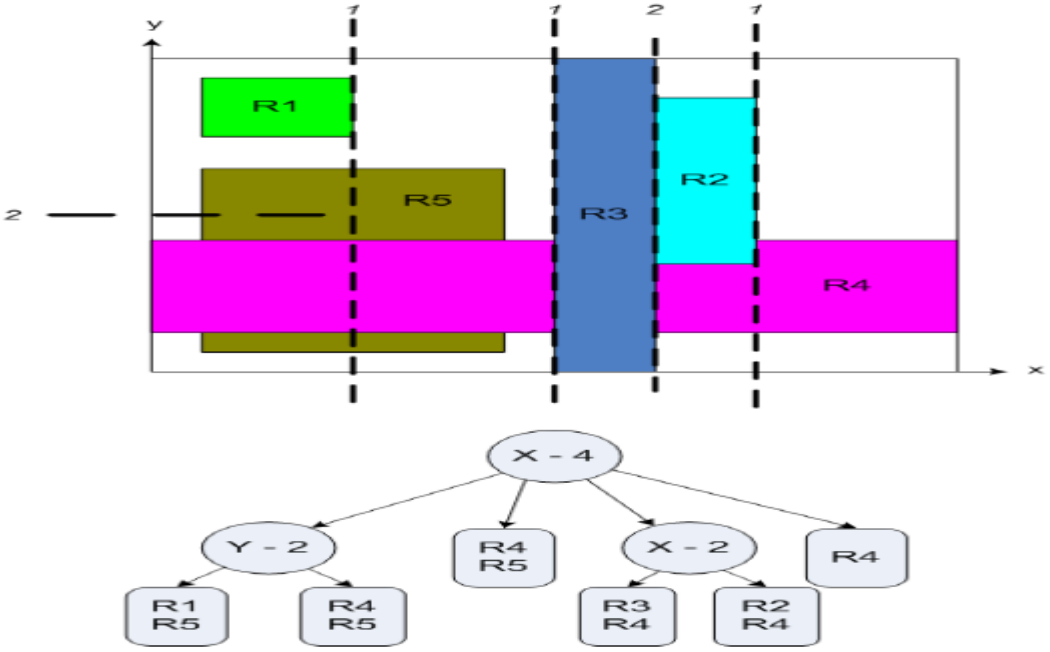


Various Decision tree-based algorithms were:
1)Hicuts
2) Modular Packet classification
3)HyperCuts

PACKET CLASSIFICATION USING HIERARCHICAL INTELLIGENT CUTTINGS (HICUTS)

The HiCut algorithm[2] works by carefully preprocessing the classifier to build a decisiontree data structure. Each time a packet arrives, the decision tree is traversed to find a leaf node, where leaf node stores a small number of rules. By linear searching among these rules provides the desiredmatching. The shape and depth of the decision tree as well as the local decisions to be made ateach node in the tree are chosen when the search tree is built.The following **[Figure-2]**illustrates an example of the decision-tree construction for a 2D filter set.There are five rectangles on the plane, each of them representing a filter. First step, cut is made along the *x*-axis to generate 4 sub-regions. After that, select two of these sub-regions to cut along the *y*-axis and *x*-axis,.

Now each sub-region overlaps less than or equal to 2 rectangles. The cuttingcan be stopped, if it is affordable to do a linear search on at most 2 filters.The number of decision tree nodes and the number of stored filters determine the storage of the algorithm data structure, and the depth of the decision tree and the number of filters in the leaf nodes determine the worst-case lookup throughput.

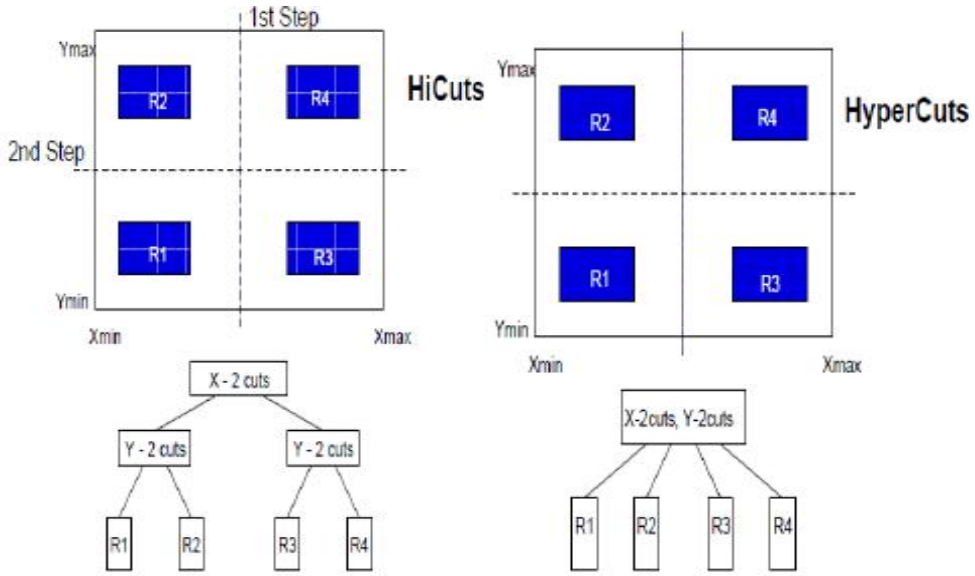


PACKET CLASSIFICATION USING MODULAR PACKET CLASSIFICATION

This algorithm[3] approaches the problem of packet classification very practically. Algorithm proposes which combines heuristictree search with the use of filter buckets. It has high performance and economic storagerequirement, algorithm is unique in the sense that it can adapt to the input packet distribution bytaking into account the relative filter usage.By examining specific bit positions algorithm tries to eliminates as many filters as possible.when the set ofremaining filters is less than some pre-specified maximum, instead of eliminating all terminated the first step. This set of filters is called as filter bucket. This early termination avoids steps of completely differentiate between a few “similar” filters. In the second step, the filter bucket isprocessed to find a match. A completely differentprocedure can be used due to the limited size of a filter bucket . Thereforethis algorithm is a modular composition of two procedures: thefirst to decompose large filter table into small filter buckets of a fixed maximum size, and thesecond procedure is to process filter buckets of limited size to find a match.

PACKET CLASSIFICATION USING HYPERCUTS ALGORITHM

HyperCuts [4] is based on a decisiontree structure like HiCuts. In HiCuts, each node in the decision tree represents a hyperplane. But inHyperCut each node in the decision tree represents a k--dimensional hypercube. HyperCuts can provide an order of magnitude improvement over existingclassification algorithmsusing thisextra degree of freedom and a new set of heuristics to find optimal hypercubes for a given amount of storage. HyperCuts uses less memory than HiCuts which is optimized for memory. The worst case search time of HyperCuts is 50-500% better than that of HiCuts.so HyperCuts is optimizedfor speed.An example of a two dimensional classifier is shown in **[Figure – 3]** with 4 rules: R1….R4.Each rule is represented by a rectangle in two dimensional space. The left figure represents theaction of HiCuts. At each node HiCuts builds a decision tree using local optimization decisionsto choose the next dimension of test inorder to find how many cuts to make in the chosen dimension.The leaves of the HiCuts tree store a list of rules. These rules may match the search path to the leaf. Theleft part of **[Figure - 3]** shows how the HiCuts algorithm works on the example rule set.



No matter how many cutsare going to be executed at a time,assuming the maximum number of rules held in a leaf is 1. HiCuts algorithm requires at least two levels in thedecision tree. By introducing one more degree of freedomHyperCuts algorithmeliminates this limitation in HiCuts. Each node in the decision tree represents adecision taken on the most representative dimensions, as opposed to using only a singledimension. For each of the chosen dimensions, the number of cuts is computed based onconditions dependent on the amount of space that is available for the search structure. In theexample in **[Figure – 3]** Hyper-Cuts (on the right) cuts the plane into four squares with onedirect cut, reducing the height of the decision tree to 1.

PROPOSED METHOD

Here proposed method is a modified version of HyperCuts algorithm.The hardware accelerator uses a modified version of the Hypercuts packet classification algorithm ,with a new pre-cutting process used to reduce the amount of memory needed to save the search structure for large rulesets so that it is small enough to fit in the on-chip memory of an FPGA.The modified algorithm also removes the need for floating point division to be performed when classifying a packet, allowing higher clock speeds and thus obtaining higher throughputs.It implements a modified version of the Hypercuts packet classification algorithm,which breaks a ruleset into groups, with each group containing a small number of rules that can be searched linearly .A decision tree is used to guide a packet based on its headervalues to the correct group to be searched.Also explains decision tree-based packet classification and gives a detailed explaination of the Hypercuts algorithm .This is done so that the changes made here to make the algorithm more suited to hardware acceleration can be better understood.The performance results including memory usage, throughput,and power consumption.

CONCLUSION

In this paper different algorithms for software approaches of packet classification are discussed. Packet classification can be implemented in the core of the network and henceimproving the speed and security. Packet classification is usually limited to use by routers at the edge of a network where line speeds do not typically exceed a few gigabit per second. This paper also introduced a new algorithm called modified HyperCuts algorithm. Packet classification hardware accelerator with enough processing power to allow packet classification to be implemented at the core of a network, thus improving security. It worked with rulesets containing tens of thousandsof rules at speeds of up to 138.56 Gb/s, allowing Internet service providers to perform a large plethora of tasks. Theclassifier consumed only 9.03 W when classifying packetsat its maximum throughput of 433 Mpps. This is low whencompared to other FPGA-based classifiers. The classifier rana modified version of the HyperCuts algorithm that has been modified so that it is better suited to hardware implementation.These modifications included changing the cutting schemeso that the need for slow and logic intensive floating pointdivision is removed when classifying a packet. This was doneby replacing the region compaction scheme used by HyperCutswith a new scheme that uses pre-cutting.

REFERENCE

[1]. Usage and population statistics(2012,jun.) [online].Available:http://www.internetworldstats.com/stats.htm

[2]. P. Gupta and N. McKeown, "Packet classification using hierarchical intelligent cuttings,"IEEE Micro, vol. 20, no. 1, pp. 34–41, Feb. 2000.

[3]. T. Woo, "A modular approach to packet classification: Algorithms and results," in *Proc.*IEEE Int. Conf. Comput. Commun., Mar. 2000, pp. 1213–1222.

[4]. S. Singh, F. Baboescu, G. Varghese, and J. Wang, "Packet classification usingmultidimensional cutting," in Proc. ACM Special Interest Group Data Commun. Conf., Aug.2003, pp. 213–224.

[5]. P. Gupta and N. McKeown, "Packet classification on multiple fields," in Proc. ACM SpecialInterest Group Data Commun. Conf., Sep. 1999, pp. 147–160.

[6]. T. V. Lakshman and D. Stiliadis, "High-speed policy based packet forwarding using efficient multi-dimensional range matching," in Proc. ACM Special Interest Group Data Commun. Conf., Sep. 1998, pp. 203–214.

[7]. V. Srinivasan, S. Suri, and G. Varghese, "Packet classification using tuple space search, " in Proc. ACM Special Interest Group Data Commun.Conf., Sep. 1999, pp. 135–146.

[8]. A. Kennedy, D. Bermingham, X. Wang, and B. Liu, "Power analysis of packet classification on programmable network processors," in Proc. EE Int. Conf. Signal Process. Commun.,Nov. 2007, pp. 1231–1234.

[9]. Weirong Jiang and Viktor K. Prasanna, "Scalable Packet Classification on FPGA". IEEE transactions on Very Large Scale Integration (VLSI) Systems, vol. 20, no. 9, September 2012.

[10]. K. Lakshminarayanan, A. Rangarajan, and S. Venkatachary, "Algorithms for advanced packet classification with ternary CAMs," in Proc.SIGCOMM, 2005, pp. 193–204.

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

[12]. P. Gupta and N. McKeown, "Algorithms for packet classification," IEEE Network, vol. 15, no. 2, pp. 24–32, 2001.

[13]. S. Dharmapurikar, H. Song, J. S. Turner, and J. W. Lockwood, "Fast packet classification using bloom filters," in Proc. ANCS, 2006, pp.61–70.

[14]. W. Jiang and V. K. Prasanna, "Large-scale wire-speed packet classification on FPGAs," in Proc. FPGA, 2009, pp. 219–228.

[15]. M. E. Kounavis, A. Kumar, R. Yavatkar, and H. Vin, "Two stage packet classification using most specific filter matching and transport level sharing," Comput. Netw., vol. 51, no. 18, pp. 4951–4978, 2007.

[16]. P. Gupta and N. McKeown, "Classifying packets with hierarchical intelligent cuttings," IEEE Micro, vol. 20, no. 1, pp. 34–41, 2000.

[17]. T. V. Lakshman and D. Stiliadis, "High-speed policy-based packet forwarding using efficient multi-dimensional range matching," in Proc. SIGCOMM, 1998, pp. 203–214.

[18]. I. Papaefstathiou and V. Papaefstathiou, "Memory-efficient 5D packet classification at 40 Gbps," in Proc. INFOCOM, 2007, pp. 1370–1378.

[19]. A. Kennedy, X.Wang, Z. Liu, and B. Liu, "Low power architecture for high speed packet classification," in Proc. ANCS, 2008, pp. 131–140.