Research Journal of Recent Sciences _____ ISSN 2277-2502
Vol. **3(8),** 16-24, August **(2014)** *Res.J.Recent Sci.*

Modification on Packet Classification Algorithm Based on Geometric Tree by using Recursive Dimensional Cutting (DimCut) with Analysis

Hediyeh AmirJahanshahi Sistani[1], Sayyed Mehdi Poustchi Amin[1] and Haridas Acharya[2]
[1]Department of Computer Studies and Research, Symbiosis International University, Pune, INDIA
[2]Allana Institute of Management Science, Pune University, Pune, INDIA

Available online at: www.isca.in, www.isca.me
Received 18th September 2013, revised 13th April 2014, accepted 27th May 2014

Abstract

All concerning data which were forwarded or received, on networks, it was with great care maintained the sequence of packets. The significant characteristic is a packet consisting of a header and accompanied with data, and it is essentially required to be sent through networking system. It is seriously taken into consideration that this important medium of networking technology necessarily to be protected against growing piracy and to keep networking system secure, we have to develop intrusion detection system (IDS) along with firewalls which is competently able for controlling, categorizing and keeping a watchful check on network traffic. There is very essential thing to be kept in mind, these systems ought to be compared with every packet header, in comparison against a large set of rules, avoiding any unwanted disruption while analysing such related packets, often incur delay. Here, categorizing delay may possibly happen; therefore, this problem can be cut down with the help of quick packet classification method. This will help achieve conclusions faster in analysis of network packets. This paper presents a modified version of the packet classification algorithm, called Packet Classification Algorithm Based on Geometric Tree by using Recursive Dimensional Cutting (DimCut V2). We proposed, implemented and tested the DimCut algorithm in the previous article" Packet Classification Algorithm Based on Geometric Tree by using Recursive Dimensional Cutting (DimCut)", that classifies packets based on five header fields; it is found that the algorithm can classify packets quickly. In order to get extended DimCut algorithm in addition some discovery and result can be obtained. Modification could extend the DimCut algorithm with adding some new heuristics ideas and new implementing techniques which we propose in this paper.

Keywords: DimCut, Packet Classification, Firewalls, Heuristics, Rules, Cutting.

Introduction

In the latest computing system network, large scale packet classification is developed into better security area especially in policy based routing as well as quality of service (QoS) assurance. At the same time firewalls required to necessarily to making is ought to ascertain, where speed of decision making to deny or not to deny, is of utmost importance to classify packets. It is to accept that in the packet classification, there is fundamental problem involving large number of rule sets, in consequence with ever increasing network traffic, along with large dimensionality of the packet attributes database[1].

With the high speed internet services, and related services to Internet TV/Radio, Video on Demand (VoD) and e-businesses are on high demand, therefore causing a higher degree of transmission bandwidth and thus consequently inducing complexes security levels. As we are fully aware of the fact that packet classification demands packets to be clearly specified or characterized in accordance with multiple packet header areas to exactly ascertain particularly which flow is an incoming packet related to and what rules the packet is to be treated. This is most fundamental requirement as far as a range of networking management and controlling aspects connected with policy based networking traffic accounting and network address translation[2].

It is to notified here that in this publication, we intend to forward modification over existing nonlinear type of packet classification algorithm, named as "Packet Classification Algorithm Based on Geometric Tree by using Recursive Dimensional Cutting (DimCut)", it has been previously proposed as well[1]. The improvements have been validated through simulated trials. Here, in this paper in particular, we propose to examine absolute the new discovery of heuristic methods of packet classification, it gives satisfactory average for the practical performance and reasonable benchmarks results to our intended algorithm.

The present study paper is categorically arranged in following ways: Section one studies from high level perspective on the existing algorithms to obtain the fundamental ideas, and each one deals with its own advantages and disadvantages in accordance with throughput, cost, facility of implementation and scalability as well. Section two works out with details of our algorithm. Section three evaluates results pertaining to implementation. The section four deals with a comparative approached to concerned work, presents the results of

experiments and findings. In Section five, the work is duly concluded.

Related work

As far as packet classification is concerned, it is considered as a vital function related to certain of important network activities, e.g. routing and filtering and remains a basic part in networking. As with many of the functionalities provided by modern routers and firewalls, packet classification features must scale well in throughput, power, and memory size[1,3-13]. Packet classification must match a packet header along with a set of policy rules so as to control the acceptance or refusal of packets. As a matter of fact, the rules concerning in this respect codified as an ordered list, at the same time it is necessary that the process of classification must apply these rules from the top[1,14,15].

It is generally observed that the process for multidimensional packet classification trade memory usage especially for search speed is used for better result or through performance. When some of these methods are used, especially on small number of classification rules, they prove satisfactory mainly in search time or memory usage, but in the situation when the number of rules increases, they scale poorly in either search time or memory usage. The researchers are engaged to this problem in the area of academic as well as in the industrial; they prefer to adopt methods and solutions for better findings, as it is getting more important in today's high performance policy in ever increasing necessarily of networking utility[16].

We have some important personalities such as Srinivasan, Varghese, Suri, and Waldvogel to mention who recommend the Grid-of-Tries and Cross Producting algorithms for packet classification. The data structure of 'Grid of Tries' algorithm is extended to two fields which uses a decision tree approach for packet classification on source and destination address prefixes. This is good but not beyond two fields, as good solution may not be achieved. The authors suggest the Cross Producting solution for multiple fields and bigger classifiers; also proposea caching technique with then on-deterministic classification time[5].

We have the researchers of Baboescu, Singh, and Varghese who have proposed on Extended Grid-of-Tries (EGT) which sustains multiple fields essentially. It is essentially important issue to note that the EGT alters the switch pointers to be jump pointers that maneuver the search to all feasible matching filters, rather than the filters with the longest matching destination and source address prefixes[17].

For additional information on proposed frame work for packet classification, Feldman and Muthu krishnan expound independent field searches on Fat Inverted Segment (FIS) Trees. It supplies a geometric view of the rule set and map rules into d-dimensional space[10].

On the other hand these three outstanding researchers, Srinivasan, Suri, and Varghese put forwarded the Tuple Space Search algorithms. Since, the number of distinct tuples is quite lower than the number of rules; it is significant to note that it searches the tuple space or a subset of the tuples in the space. In order to separate tuples it is possible to perform separately, therefore we can say that tuple space techniques can benefit of parallelism. A parallel designing to make it effective is a challenging endeavor, in the sense; it is highly unpredictable due to its size of the tuple space to investigate. Consequently, the lookup implementation for tuple space techniques differs very much. Lookups in individual tuple can be conducted through a simple hash table[13].

The Rectangle Search algorithm was originally suggested to provide theoretically optimal performance for packet classification on two fields irrespective of adopting assumptions in reference to the structure of the filter set. Rectangle Search uses the principle of markers and also presumption introduced by the Binary Search on Prefix Lengths technique for longest prefix matching[11,18].

The capable algorithms contain rational tradeoff between storage and through put and the tunable parameters to provide expected results. The poor form of the Cross Producting algorithm has substandard storage quality; with the inefficient result even for the moderate sized filter set. On the other hand, the Recursive Flow Classification (RFC) algorithm, it is fully efficient to reduce the storage in a better way. It is observed that while maintaining the high through put, the storage remarkably reduced. Notwithstanding, for larger filter sets, the storage capability of RFC remains low. These recommended additional tradeoffs ought to be realized to obtain better performance[5,8].

There is yet another technique known as, rule set intersecting technique; in this, a partial rule match is rather easier than that a full rule match. The characteristics of this technique is that, the packet header can be split into substring and matched with a subset of rules, whose intersection will give rule to match the whole packet header. The Bit Vector (BV) algorithm provides the subset of rules for every partial match by using bit vectors then Baboescu and Varghese proposed the Aggregated Bit-Vector (ABV) algorithm that helped improve the performance function of the Parallel BV technique[6,9]. The Hierarchical Intelligent Cuttings (HiCuts), the Multidimensional Cuttings (Hyper Cuts) and Woo's modular packet classification have geometrically packet classification view, they use this view for data structures and rule representation[4,14,7].

Still no technique can be good and perfect in its performance. It is expected that the good algorithm ought to have all possible approaches, for example in the area of the time-space tradeoff, but still it is not possible to achieve all the good characteristics. Yet there is room for the packet classification algorithms to improve in satisfactory performance.

DimCut: It is expounded that a packet classification algorithm must support prefixes, ranges, exact values and wildcards as well as performance metrics for testing purpose. It is expected to include of search speed, storage requirements, fast updates, scalability, and flexibility.

Gupta and Mc Keown propounded an algorithm, known as Hierarchical Intelligent Cuttings (HiCuts). This is an example of decision tree-based packet classification algorithm. Itdeals with the geometric view of the packet classification problem and establishes the ground for our new algorithm[7]. The decision tree packet classification algorithms, certain construction decision tree are followed by better improvement searches.

The DimCut[1] algorithm has furnished with some modifications and improvements on the HiCuts algorithm. Certain features are mentioned here: Each rule in the rule set defines a d-dimensional rectangle in d-dimensional space, where d is the number of fields in the rule. The algorithm pre-processes the rule set implied to establish a decision tree. It is well explained by mentioning that the leaves are containing a subset of rules with number of rules bound by a predefined threshold.

In DimCutthe wc (H) is defined to count of wild card entries in the column H in the whole of the rule set and the gd (H) that is the geometric distance associated with column H in the whole of the rule set.

To choose the proper cut dimensions, two fields Ha, Hbare selected which have the least wc ( ) values as the two dimensions. Or alternatively one can select Ha, Hb which have the least gd ( ) values. To compute the number of cuts and the bucket size threshold, the formulaNC1 = [20 + (N/1000)] = Number of cuts, and B = [N/ (20 + (N/1000))] = Bucket size (The threshold) are used. Here N = Total Number of rules.

The pre-processing level (tree construction and making index table) and search level are two separate levels to implement this algorithm; it uses the Link list data structure and works on large rule sets.After constructing the decision tree of rules, to classifying any packet, searching continues until reaching a leaf node storing the best matching rule for the packet.

Rule classification example is shown in Table 1, 2 and Figure 1, 2. In this simple example rules have only two fields (source Ip address and destination Ip address); each field has 4 bits.

Table-1
In this example rules are shown in priority and have only two fields (source Ip address and destination Ip address)

Rule	SCR Add. (SA)	DST Add. (DA)
R1	00**	000*
R2	010*	00**
R3	10**	00**
R4	100*	01**
R5	1100	0100
R6	0010	0100
R7	00**	01**
R8	001*	10**
R9	011*	01**
R10	10**	1***
R11	10**	10**
R12	111*	01**
R13	011*	101*
R14	1101	*
R15	01**	1*

Table-2
Shows length calculation for SA and DA to select the best cutting dimension

Rule	SCR Add. (SA)	Length SA	DS Add. (DA)	Length DA
R1	00**	0011-0000=0011	000*	0001-0000=0001
R2	010*	0101-0100=0001	00**	0011-0000=0011
R3	10**	1011-1000=0011	00**	0011-0000=0011
R4	100*	1001-1000=0001	01**	0111-0100=0011
R5	1100	0000	0100	0000
R6	0010	0000	0100	0000
R7	00**	0011-0000=0011	01**	0111-0100=0011
R8	001*	0011-0010=0001	10**	1011-1000=0011
R9	011*	0111-0110=0001	01**	0111-0100=0011
R10	10**	1011-1000=0011	1***	1111-1000=0111
R11	10**	1011-1000=0011	10**	1011-1000=0011
R12	111*	1111-1110=0001	01**	0111-0100=0011
R13	011*	0111-0110=0001	101*	1011-1010=0001
R14	1101	0000	*	1111-0000=1111
R15	01**	0111-0100=0011	1*	1111-1000=0111
Sum	SA	=24		=40

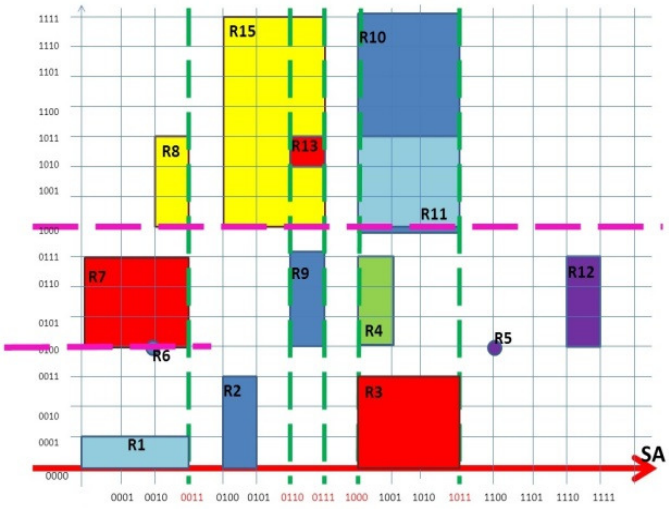

Figure-1
Each rectangle represents a geometric view of a rule and shows partitioning the rule set space into small rules buckets.

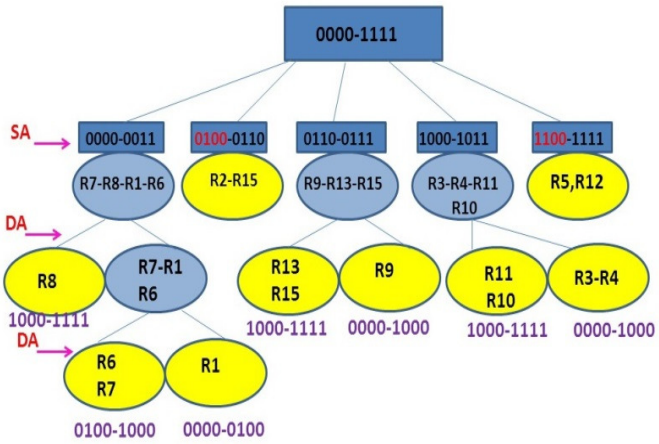

Figure-2
Shows the constructed tree according cutting the geometric view of rule set

The DimCuthaving discussed all its various aspects, we come to the conclusion that it is a remarkable algorithm. It has certain outstanding feature such as scalability, low memory consumption and reasonable speed. It is fully considered as one of the fully efficient packet classification algorithms.

Modified Algorithm

In this paper our aim is to put forward some important modification in relation to the DimCut and propound the new version of DimCut with reference to efficiency and performance enhancement. Our efforts have been to search for a better formula in order to calculate the number of cuts, along with implementing a suitable technique and data structure to affect the algorithm.

The Number of cuts and Bucket size compute with the new formula as, NC = 495.22 + (.034 * N) + (9 * 10^{-7} * N^2) + (6.23*10^{-12} *N^3) = Number of cuts, B = 2 if N<=10000 and B=5 if 10000<N<40000 and B= 8 if 40000<=N<=100000 = Bucket size (The threshold),Here N = Total Number of rules, in the complete rule set of any Node.

Very important precaution must be followed strictly when the number of cuts is to be confirmed, while applying the NC formula only one time the bucket size threshold of the algorithm is to be notified, this is vitally important in order not to split the rules. This is the process at the initial stage. The next cut level, the earlier method is to be repeated, this is essentially required to maintainthe best result. This new algorithm has a significant identification i.e. consisting two clear levels,the first as pre-processing or tree construction and making index table, second level is known as search level. In this algorithm the array and the pointer data structure are applied and it works with large amount of rules.

The Briefed Preprocessing Algorithm: i. Read rules and create an Array pointer structure to preserve them. ii. Find the cut dimension by applying any of 2 heuristics (any dimension that has the smallest geometric length (Broadcast Address - Network Address) or any dimension that has the smallest number of wildcards). iii. Calculate the number of cuts by using of (NC = [495.22 + (.034 * N) + (9 * 10^{-7} * N^2) + (6.23*10^{-12}*N^3)] = Number of cuts, B = 2 if N<=10000 and B=5 if 10000<N<40000 and B= 8 if 40000<=N<=100000 = Bucket size (The threshold)) .This is important to avoid splitting of rules while cutting as much as possible. iv. Separate those rules that has wildcard value in the same chosen field as cut dimension in the bucket0. v. Construct the tree, For i=1 to NC do: i. Create buckets (nodes), ii. Assign the rules that covered by buckets (nodes) region, iii. If the number of rules in bucket > Threshold, iv. Split buckets (nodes), v. Create the index table for rules in buckets, vi. Optimize and compress the tree.
END

The Briefed Search Algorithm: i. Use Search part: Read Packets. ii. For each Packet: i. Find the buckets that cover the packet, using new technique in place selection. ii. Search in the related index table of those buckets. iii. Find the specific matched rules. iv. Select the higher priority one as a target. v. Act as its action.
End

All rules have been categorically arranged in priority order in accordance with the network administrator. The decision tree will extend across to search the buckets covering the incoming packet; practically it will jump to the first bucket regions of its origin. Next, first match bucket is available, a packet will forward to all regions possibly of the bucket then all the header

fields of packet will compare to all governing rules linearly. In this process, the most prioritized rule is selected which perfectly matches.

Finally, the action (Accept/Deny/Log/Forward/Nothing) will be accepted for that incoming packet and the search will complete. It is notable here to observe i.e. optimization of the decision tree will be concluded by excluding the empty nodes, combing the nodes that belong to the same set of rules, also if the region covered by the rules is smaller than the overall size of the region that controls the node then one shrinks the region that accompanied with the node to minimum cover, and if the same rule is repeated with all nodes keeping the same level, then separate that rule and make a bucket of that so that it can be used at the time of search[14,7].

To compare the previous DimCut with the new one, we run the full test, provided the data analysis and draw the suitable graphs, so it is possible to prove the modified DimCut makes packet classification faster. Search performance is evaluated by running through it large number of packets and rules. We used worst case scenario to provide the same condition for all tests and do the best evaluation.

Methodology

We have implemented the common linear algorithm (L) andour algorithms which arecalled DimCut V1 and DimCut V2 inlanguage C, (GCC 4.4). The codes are compiled with the Code::Blocks 10.05, which is a full-featured IDE all on an Intel Pentium processor 2.00 GHz with RAM 2.00 GB, 32-bit OS, running Microsoft Windows 7 Ultimate and the Oracle VM Virtual Box, asthe virtual environment are used for all the tests. In the Figure 3, the plan of the simulated experimented isshown.

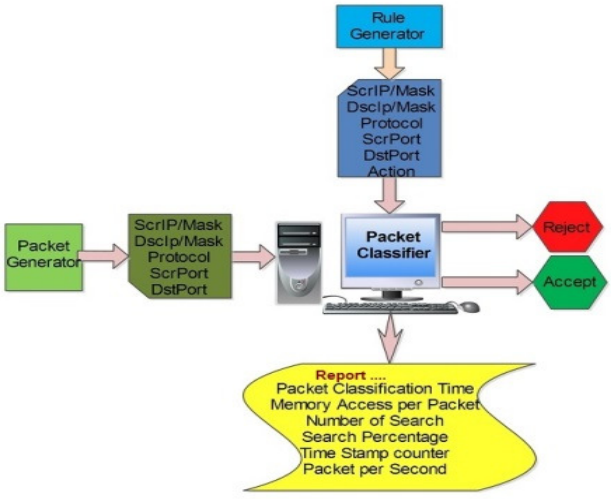

Figure-3
Simulated experimented methodology model

At the test step the packet generator generates packets, somehow to face worst case scenario only, to force the same fair test condition for all algorithms.The rule's headers are Source IP and Destination IP (Exact/prefix), Source Port and Destination Port: (Exact value, any, ranges), Protocol: (TCP, UDP, ICMP, ANY) and the Actions: (Accept; Deny, Log, Forward, Nothing).

The Common linear, DimCut V1 and DimCut V2 algorithms are implemented and compared to provide the results for this paper. The observation recorded constituted of: the Packet Classification Time measurement (Linear-PC, DimCut-V1-PC, DimCut-V2-PC), the Number of Packet per Second Classification (Linear-PPS, DimCut-V1-PPS, DimCut-V2-PPS), the Number of Search till Packet Classification done (Linear-S#, DimCut-V1-S#, DimCut-V2-S#), the Percent of Search Numbers out of 100 till Packet Classification done(Linear-%Search, DimCut-V1-%Search, DimCut-V2-%Search), the Time stamp Counter per Packet, the Rule Memory access, and other parameters also are the number of buckets (leaves)/Number of Cuts, Depth of the tree structure and Threshold/Bucket size.

The number of rules is 100 to 100000; every test is repeated three times to arrive at the average amount in the final result. Then the collected data analyzed with the statistical software, SPSS.

Results and Discussion

It is proved that the proposed algorithms based on decision tree, make packet classification fast and also the DimCut V2 was faster than DimCut V1, by comparing the linear one with these new algorithms, and using data analysis and graphs. Search performance is evaluated by running through it large number of random packets that obey the weight specifications. All tests are in the worst case scenario to do the best evaluation.

It is notable here that our reference implementations are only for the purpose of simulation and evaluation; thus, the source code is not optimized assoftware. We choose the configurations that lead to the best overall performance.

Analysis Data collection: The Normality test (Kolmogorov–Smirnov), Paired sample T test and Correlation test are done on all data by the statistical package SPSS.It is duly cautioned that our models show results of average case performance.

The normality of the data distribution is tested by Kolmogorov–Smirnov test. It shows the samples are standardized by comparison with a standard normal distribution. The significant score is greater than 0.05, so our data is normal as showed in table 3. As the samples are the Normal data, the Paired sample T test and Correlation test can be done on the samples.

The correlation test is widely used in the sciences as a measure of the strength of linear dependence between two variables. The

two variables are perfectly correlated, meaning that one can predict the values of one variable from the values of the other variable. As the significant score is 0.000 (which is less than 0.05) as showed in table 4; therefore there is a significant correlation between the variables.

The t-test is a statistical method that compares the significant difference between the means of two groups of data. The dependent sample t-test (Paired sample T test) enables us to compare packet classification time, number of Packet per second, search number between Linear, DimCut V1 and DimCut V2. In this test as the significant score is 0.000 (which is less than 0.05) as showed in table 5; therefore there is a significant difference between the means of observed values from Linear, DimCut V1and DimCut V2.

Table-3
Shows the data significantly deviate from a normal distribution (test distribution is normal)

One-Sample Kolmogorov-Smirnov	Linear. PC	DimCutV1.PC	DimCutV2.PC	Linear. PPS	DimCutV1.PPS	DimCutV2.PPS	Linear.Search	DimCutV1. Search	DimCutV2. Search
N	13	13	13	13	13	13	13	13	13
Normal Parameters Mean	49838.587	1126.835	67.200	12771.923	73103.461	501705.769	43923076 9.230	30242754.000	1755135.000
Std. Deviation	44356.337	772.627	46.532	34797.713	91582.243	393754.661	33752682 7.043	22539806.842	1409462.577
Kolmogorov-Smirnov Z	.632	.588	.750	1.546	.922	.750	.524	.505	.540
Asymp. Sig. (2-tailed)	.819	.879	.628	.017	.363	.628	.946	.961	.933

Table 4
Shows significant correlation between the means of the linear and Tree attempts

Paired Samples Correlations	N	Correlation	Sig.
Pair 1: Linear.PCandDimCutV1.PC	13	.989	.000
Pair 2: Linear.PCandDimCutV2.PC	13	.976	.000
Pair 3: Linear.PPSandDimCutV1.PPS	13	.931	.000
Pair 4: Linear.PPSandDimCutV2.PPS	13	.750	.003
Pair 5: Linear.SearchandDimCutV1.Search	13	1.000	.000
Pair 6: Linear.SearchandDimCutV2.Search	13	.992	.000

Table-5
Shows significant differences between the means of the linear and Tree attempts

Paired Samples tests-paired differences	t	Sig. (2-tailed)
Pair 1: Linear.PCand DimCutV2.PC	4.050	.002
Pair 2: DimCutV1.PC and DimCutV2.PC	5.251	.000
Pair 3: Linear.PPS and DimCutV2.PPS	-4.786	.000
Pair 4: DimCutV1.PPS and DimCutV2.PPS	-4.943	.000
Pair 5: Linear.Searchand DimCutV2.Search	4.693	.001
Pair 6: DimCutV1.Search and DimCutV2.Search	4.857	.000

Graphs: The L means linear search algorithm, the DimCut V1 means new algorithm version one to construct tree and the DimCut V2 means the new algorithm version two to construct tree in all graphs, and the incoming packets are fixed to 20000 numbers at worst case condition that means, force the incoming packets to trace till the end of tree, by setting the Protocol field to X value. So the incoming packet doesn't match with any of the rules and provide the same fair condition for all algorithms

to comparison. In figure 4, the graph shows the measurement of Packet Classification Time in Millisecond (PC), as the rule numbers increases the PC time also increases, but the DimCut V2 is acting better in case of time consuming. The graph shows DimCut V1 is faster than linear one and the DimCut V2 is faster than DimCut V1 during packet classification action.

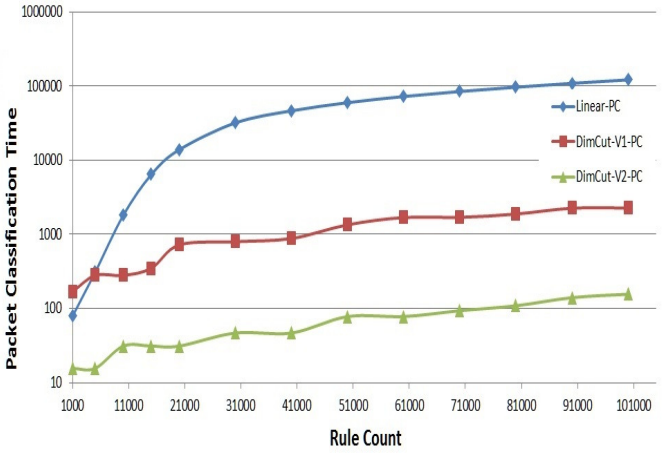

Figure-4
A comparison between the linear and the DimCut V1 and V2 algorithms, as we increase rules to measure the packet classification time (milli second)

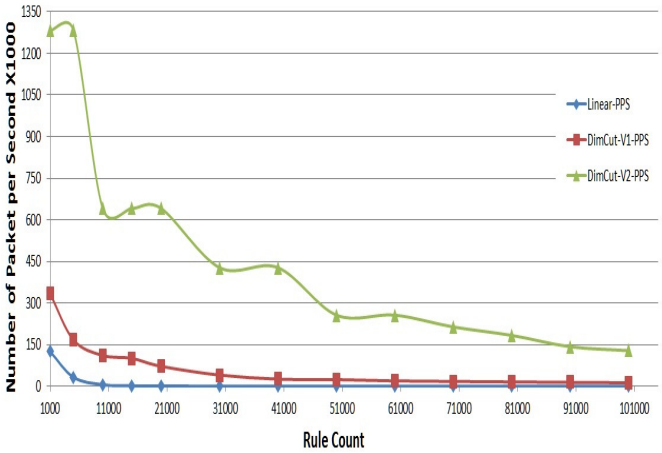

Figure 5: A comparison between the linear, the DimCut V1 and V2 algorithms, as we increase rules to measure the number of packet per second processing

In figure 5, the graph shows the Packet per Second processing (PPS), as the rules number increases, the numbers of packet processing ability is decreasing during the packet classification action. However the newer algorithms seem to be more efficient than the linear one, it's clear that the performance of DimCut V2 is higher. Search numbers is far higher in case the linear algorithm compared to DimCut algorithms as can be seen from figure 6.

The graph in figure 7 shows the Percentage of Search Numbers measurement for DimCut version two, which explain its efficiency and performance.It shows, to classify any packet during packet classification, there is no need to search linearly among all rules, it searches only among some amount of rules that there is possibility for matching, out of 100 percent of rules.

The graph in figure 8 shows the number of Time Stamp Counter per Packet, which counts the number of cycles for each packet till classification done, to measure the performance of the function. The graph in figure 9 shows the Rule Memory Access in Byte, which is the amount of memory that being accessed by rules. When the rule numbers increases the Time Stamp and Rule memory Access amounts also increases, but the graphs in figures 8 and 9 show the reasonable and acceptable slow grows.

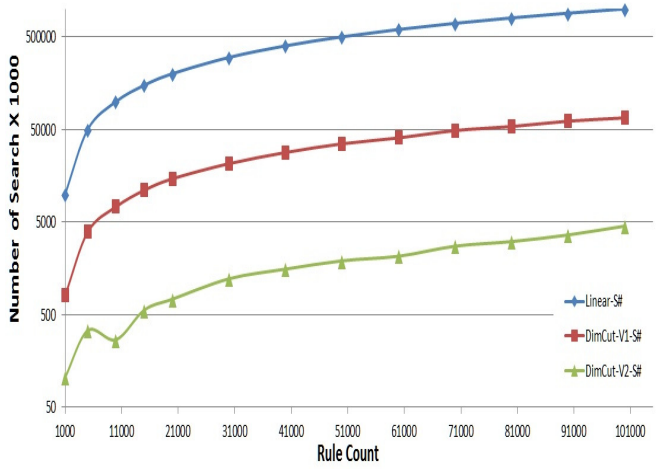

Figure-6
A comparison between the linear and the DimCut V1 and V2 algorithms, as we increase rules to measure the number of search times

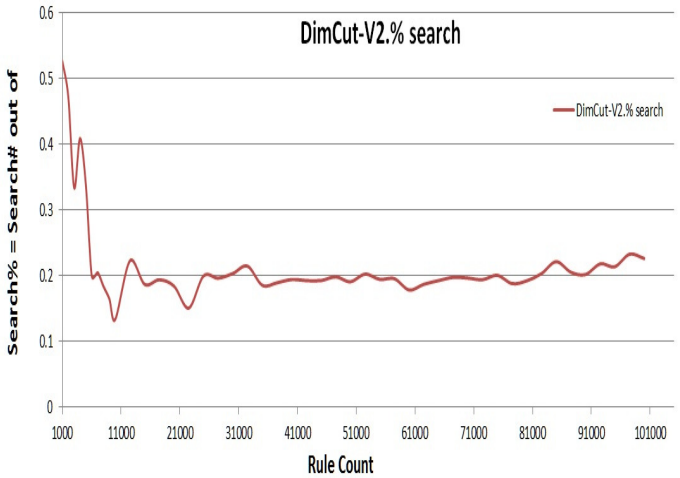

Figure-7
Shows the Percentage of Search Numbers measurement for DimCut version two

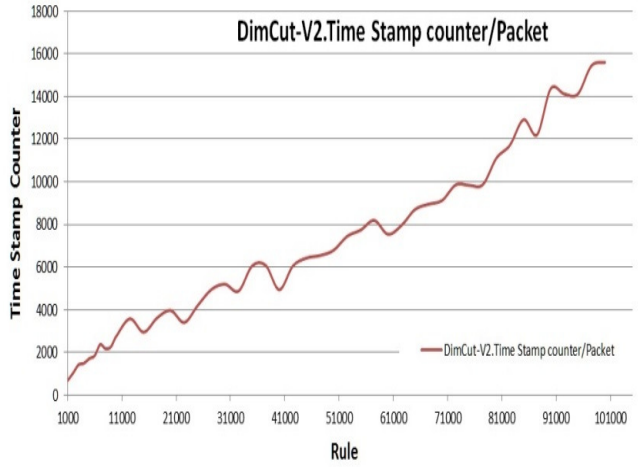

Figure-8
Shows the number of Time Stamp Counter per Packet, which counts the number of cycles for each packet

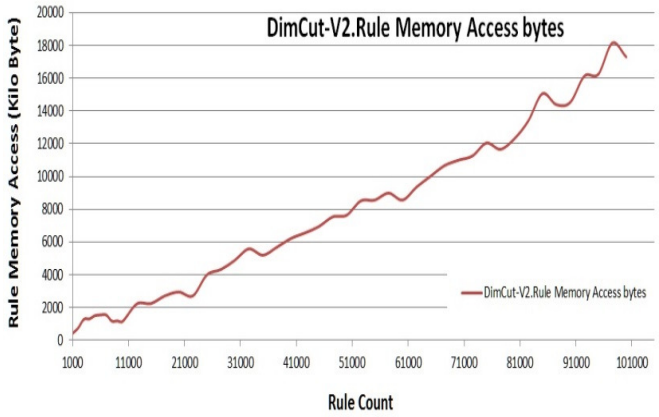

Figure-9
Shows the Rule memory Access Bytes, which is the amount of memory that being accessed by rules

The number of cuts and the dimension selection to cut at each internal decision tree node arethe critical sensitiveness for the algorithm performance. A larger bucket size or lesser number of cuts can help to reduce the size and depth of a decision tree, but it can induce a longer linear search time. Experimentally could determine the appropriate bucket size for the best tradeoff of storage and throughput. Generally, a larger bucket size means a worse search processing but this does not always hold.

When the number of rules increases the preprocessing time or the decision tree construction time increases also. Smaller bucket size requires significantly longer preprocessing time and bigger decision tree in size and depth. Experimentally we found that the algorithm consistently demonstrates better performance and scalability with the proposed parameters and formula setting. The evaluation results are normalized in a directly comparable way, as mostly packet classification algorithms are based on heuristics, different rule sets with different structures and sizes tend to give very different results.

It is expounded that the decision tree-based algorithm is flexible for a reasonable tradeoff between throughput and storage, but the overall performance is not so convincer, so more researching need to be done on better systematic ways to find out better parameters' configuration, adaptive decision-tree construction procedures and rule set structure.

Conclusion

The Dim Cut V2has been proposed in this paper, this is the modification over existing nonlinear type of packet classification algorithm, named as "Packet Classification Algorithm Based on Geometric Tree by using Recursive Dimensional Cutting (Dim Cut)" [1]. Proper implementation of this algorithm can help high performance packet classification to enable the firewalls and security challenges in high-speed environments.

We intend to work on characteristics of real classifiers, tuning parameters and algorithm modification for a reasonable tradeoff between throughput and storage. We proposed and implemented the DimCut-V1, DimCut-V2 as the new algorithm, based on Hi Cuts by new heuristics and modifications. The evaluation has been done for the proposed algorithm by measuring the Packet Classification Time, Number of Packet per Second Classification, Number of Search, Percent of Search Numbers, Time stamp Counter per Packet and Rule Memory access and every test is repeated three times to arrive at the average amount for the final result.

The proposed algorithm was analyzed for the different tradeoffs between the bucket size, number of cuts, number of rules, and number of levels. The SPSS statistical software analyses data while the Normality test that is done by Kolmogorov–Smirnov, the Difference test that is the Paired sample T Test and the Correlation test, to check the final results then we drawseveral graphs for better representation.

Ultimately, the evaluation showed that the DimCut V2 is faster than the DimCut V1 and can be a preferable Packet Classification algorithm that gives the conclusive performance with flexibility to tradeoff the components. In a general sense this project can be useful for the packet classification problem researchers. Although, the decision tree-based algorithm is flexible for a reasonable tradeoff between throughput and storage, but the overall performance is not so convincer, so more researching need to be done to find out better parameters' configuration, adaptive decision-tree construction and rule set structure.

References

1. Amirjahanshahi H., Poustchi M. and Acharya H., Packet Classification Algorithm Based on Geometric Tree by using Recursive Dimensional Cutting (DimCut), in

proceeding of the *Research journal of Recent Sciences*, **2(8)**, 31-39, **(2013)**

2. Abdelghani M., Sezer S., Garcia E. and Jun M., Packet Classification Using Adaptive Rules Cutting (ARC), in proceeding of an advanced industrial conference on telecommunications/service assurance with partial and intermittent resources conference/e-learning on telecommunications workshop, 28-33, **(2005)**

3. Song H. and Turner J., Toward Advocacy-Free Evaluation of Packet Classification Algorithms, *IEEE Transactions on Computers,* **60**, **(2011)**

4. Woo T., A Modular Approach to Packet Classification: Algorithms and Results, in Proceedings of INFOCOM 2000, *Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies*, **3**, 26-30 **(2000)**

5. Srinivasan V., Varghese G., Suri S. and Waldvogel M., Fast and scalable layer four switching, in Proceedings of ACM Sigcomm, Vancouver, Canada, **98,** 191-202, **(1998)**

6. Stihdis D. and Lakslunan T.V., High-speed policy-based packet forwarding using efficient multi-dimensional range matching, in Proceedings of ACM Sigcomm, 203-214, Vancouver, Canada, August 31-September 4 **(1998)**

7. Gupta P. and McKeown N., Packet Classification Using Hierarchical Intelligent Cuttings, in Proceedings of IEEE Symp. High Performance Interconnects (HotI), 7, **(1999)**

8. Gupta P. and McKeown N., Packet Classification on Multiple Fields, in proceedings of the conference on Applications, technologies, architectures, and protocols for computer communication in ACM SIGCOMM**' 99,** 147-160, **(1999)**

9. Baboescu F. and Varghese G., Scalable Packet Classification, in Proceedings of the ACM SIGCOMM, 199–210, **(2001)**

10. Feldmann A. and Muthukrishnan S., Tradeoffs for Packet Classification, in Proceedings of the IEEE INFOCOM, *Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies*, **3**, 1193–1202 **(2000)**

11. Waldvogel M., Varghese G., Turner J. and Plattner B., Scalable High Speed IP Routing Lookups, in Proceedings of the ACM SIGCOMM, 25-38 **(1997)**

12. Srinivasan V. and Varghese G., Fast Address Lookups Using Controlled Prefix Expansion, in Proceedings of the ACM Trans. Computer Systems, **17**, 1-40, **(1999)**

13. Srinivasan V., Suri S. and Varghese G., Packet Classification Using Tuple Space Search, in Proceedings of the ACM SIGCOMM' **99,** 135-146, **(1999)**

14. Singh S., Baboescu F., Varghese G. and Wang J., Packet Classification using Multidimensional Cutting, in Proceedings of the ACM SIGCOMM '03 Conference on Applications, Tech., Archi., and Protocols for Computer Communication (SIGCOMM '03), 213–224 **(2003)**

15. Chen W., Wang W., Li Z. and Chen H., Dynamic Update of Firewall Policy Based on MFDT, in Proceedings of the Computational Intelligence and Security, International Conference, **2(3-6),** 1117–1120 **(2006)**

16. Qi Y., and Li J., An efficient hybrid algorithm for multidimensional packet classification, in Proceedings of the International Conference Communication, Network, and Information Security, MIT, Cambridge, MA, USA, 9–11, **(2006)**

17. Baboescu F., Singh S. and Varghese G., Packet Classification for Core Routers: Is there an alternative to CAMs?, in Proceedings of the IEEE Infocom, **(2003)**

18. Warkhede P., Suri S. and Varghese G., Fast Packet Classification for Two-Dimensional Conflict-Free Filters, in Proceedings of the IEEE Infocom, **(2001)**