International Journal on Recent and Innovation Trends in Computing and Communication
Volume: 3 Issue: 7

ISSN: 2321-8169
4636 - 4640

Packet Classification based on Boundary Cutting analysis by using Bloom Filters

Ms. Namita N. Kothari
ME Information Technology2nd,
Amrutvahini College of Engineering,
Sangamner, India
namitakothari8@gmail.com

Prof. S. E. Pawar
H. O. D. Department of IT
Amrutvahini College of Engineering,
Sangamner, India
Pawar.suvarna@gmail.com

Abstract:- Packet classification has received a great deal of attention over the half decade in applications such as Quality of Service (QoS), security, firewalls, Network Intrusion Detection System (NIDS), multimedia services, differentiated services. They perform different operations at different flows. Existing decision-tree-based packet classification algorithms, HiCuts and HyperCuts perform search by geometrical representation of rules in a classifier by searching for a geometric space to which packet belongs. These decision tree algorithms have complications in finding number of cuts and the field. Also fixed interval-based cutting not covers the actual space for each rule. Hence it is ineffective and requires huge storage requirement. In recent years, Bloom Filter, which is space-efficient and probabilistic data structure for membership queries, becomes popular in many network applications. It requires small amount of memory and used to avoid lookups to sustain high throughput. It handles the large database and provides security in network applications like NIDS. This paper presents a boundary cutting (BC) scenario which exploits the structure of classifiers. It finds out the space that each rule covers and perform cutting according to rule boundary. Hence it is deterministic, and more effective in providing improved search performance and efficient in memory requirement. Security roles are also considered during classification.

Keywords:- *packet classification; decision tree algorithms; bloom filters; boundary cutting; binary search*

*****

I. INTRODUCTION

Packet classification is a key building block for many network devices which demand efficiency and robustness of classification operation [11]. Marking an incoming packet allow or disallow is known as process of packet classification [2]. As shown in fig. 1, a packet classifier must compare header fields of predefined rules and return the identity of highest-priority rule which matches the packet header of packet.

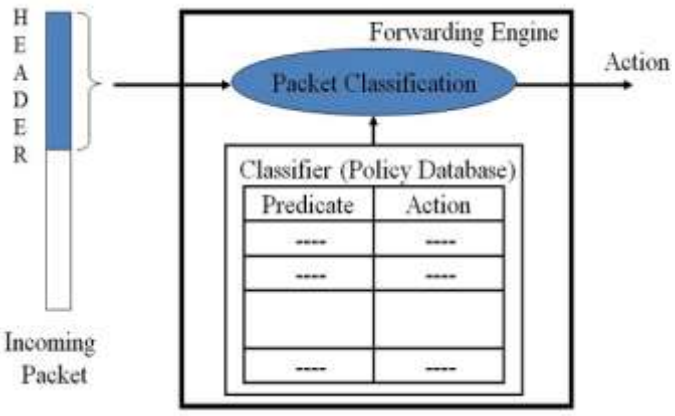

Figure 1. Packet Classification

The main problems in packet classification are huge rule sets (size of rule set), traffic intensity (heavy network traffic), and large dimensionality of the packet attributes database (large item sets) [2]. In past years, many algorithms and architectures have been proposed to identify an effective solution for fast classification. Number of network services requires packet classification and in each case, it is very necessary to find which flow an incoming packet belongs to. For each incoming packet it should be determined whether to forward or filter it (NIDS or firewall), where to forward it(router), which class of service it should receive(QoS) and how much should be charged for transporting that packet (traffic billing) [11]. The main drawback of the above applications is the classification stage. However, packet classification must cover the criteria of throughput, storage, classification time, incremental update support, power dissipation, flexibility, scalability and adaptability to the structure of filter sets [3]. The different performance metrics should be carried out to evaluate the performance of classification algorithm.

Most of the network applications require multimatch classification and highest priority concept because of the need for security in NIDS, worm detection, firewalls and packet level accounting to identify the context of the packets and to perform actions which include dropping an unauthorized packets, scheduling, coping, prioritizing and encrypting secure packets [1]. However, the previously well-known algorithms such as HiCuts [8], HyperCuts [9] select the field and number of cuts on a locally optimized decision, which reduces the search speed and requires the large storage. This process requires pre-processing which is the slowest operation in packet classification, consumes much memory and construction time. Hence it is difficult for those algorithms to be extended to large rule sets because of memory problems while building the decision tree. Moreover, the fixed interval-

 ISSN: 2321-8169
4636 - 4640

based cutting, which does not consider the actual space that each rule covers, hence finds ineffective [1].

A new efficient Packet Classification based on boundary Cutting analysis by using bloom filters improve overall performance with optimal storage space for the large rule sets. This classification algorithm works on the principle of optimization of the rule set by using analysis of region. Therefore, the amount of required memory automatically gets reduced. Packet classification table is deterministically built and does not contain the complicated heuristics used by current decision tree algorithms. It also performs binary search at internal nodes of decision tree which provides a good search performance for indexing [1]. Whereas, bloom filter provides most efficient solution for dynamic packet classification [6] and filters large amount of incoming packets in less time without any packet drop or missing with required optimal memory space in real time. They avoid lookups in subset which does not contain any matching rules and sustain high throughput [5]. Throughput is calculated by dividing the total memory bandwidth by the memory bandwidth consumed per packet lookup.

II. LITERATURE REVIEW

Packet classification algorithm should cover the features like, support general rules which has prefixes, range, exact values, wildcards, better data structures to rule bases, preprocessing and multiple matches [2]. Many architectures and algorithms have been proposed to evaluate an effective packet classification solution. Algorithms for packet classification are a vast body of literature review, are of four types: 1) Exhaustive search 2) Decision-Tree based 3) Decomposition type 4) Tuple space search [11].

Exhaustive search perform algorithmic based packet classification. In these techniques all entries in the filter set are examined sequentially which is similar to Ternary Content Addressable Memory (TCAM) approach. They perform well in terms of memory usage, can be updated incrementally and do not require preprocessing but they require more storage space and search is very slow. Hence it is very inefficient. They require O (*N*) memory accesses per lookup, where *N* is the number of rules. Exhaustive search also uses brute force method which is not effective because of linear search which becomes prohibitively slow.

Decomposition is algorithmic based packet classification which tends to decompose the multiple field searches into instances of single field searches and then combine the end results, similar to Recursive Flow Classification (RFC) and Bit Vector (BV) algorithm. RFC which grows exponentially in terms of memory use with the number of rules, they tend to have a high number of overlapping regions. Bit Vector algorithm has some drawbacks. Poor classification increase memory use dramatically and the classification time vary depending upon the incoming packet value, which may change the path inside the decision tree. Although, it provides high speed packet classification time and preprocessing time, which makes unsuitable for those systems who needs frequent rule set updates.

While Tuple space techniques divide the filter set according to the number of specified bits in the filter, then they examine the partitions using simple exact match search. But using tuples to partition the filter set, the tuple space approach can quickly narrow the scope for a multiple field search. The number of specifies bits in each field of the filters is known as Tuple. Though, this class of algorithm has the lowest memory uses, but requires highest preprocessing time and classification time. Sometimes it varies based on the nature of the rule set.

HiCuts [8] and HyperCuts [9] are decision tree based algorithms which construct a decision tree by finding number of cuts and fields and then use the packet fields to navigate the decision tree. All decision tree's leafs contain a rule or a subset of a rule and classification performed using search key to traverse through the decision tree [11]. The main drawbacks of these techniques are the high storage requirement and preprocessing time. Because of long preprocessing time, both do not support incremental updates. Classification time per packet also vary depending on the depth of the decision tree. TCAM based architectures are best solution for wire speed packet forwarding. Though it can be used to give high throughput but it exhibits relatively poor performance with respect to power and area efficiency. TCAMs impose more cost and more storage [1].

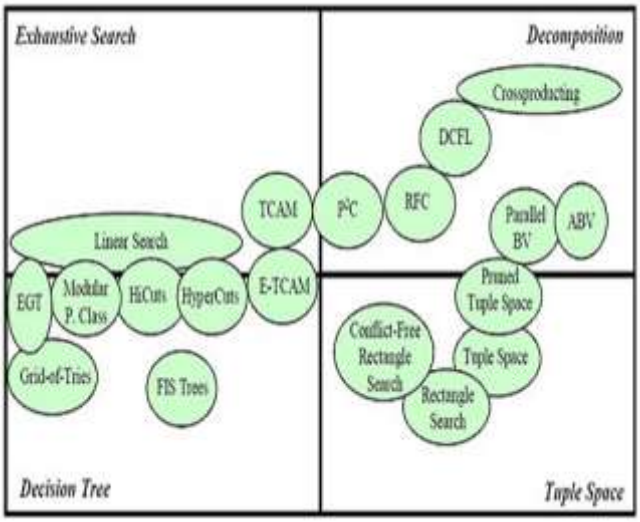

Figure 2. Categories of Packet Classification Techniques

III. EXISTING SYSTEM

An arriving packet belongs to a certain flow when all the packet fields are in the range of the rule flow. That means each rule has F components and the *i*th component of rule R, is known as R[i], which is a regular expression of the packet header of *i*th field [11].

Hardware based Application Specific Integrated Circuits (ASICs) with off-chip TCAM is the good solution for wire-

ISSN: 2321-8169
4636 - 4640

speed packet forwarding but the high cost and power consumption of TCAM, making the exploration of some other algorithmic solutions. Moreover, TCAM algorithm's throughput is limited to a single character per clock tick. To scan multiple characters at a time, multiple TCAM chips would require. TCAM may require 2(*L*-1) TCAM entries for an *L*-bit port range field, for making the exploration necessary [1].

Packet classification using decision tree algorithm is nothing but constructing a decision tree and leaves of the tree have rules or subset of rules. If a decision tree is properly divided so that the internal nodes of the tree area stored in an on-chip memory whereas large rule database is stored in an off-chip memory so that decision tree algorithm can provide high-speed search performance. HiCuts [8], HyperCuts [9] and EffiCuts [10] naturally enable the highest priority match. HiCuts and HyperCuts algorithms select the number of cuts and field by taking locally optimized decision, which decreases the search speed and the memory requirement. The HiCuts algorithm works by preprocessing the classifier to build a decision tree. Each time a packet receives, the decision tree traversed to find a leaf node, which contains a small number of rules.

A linear search is performed on these rules yields the desired matching. When search tree is built, the shape and depth of the decision tree and local decisions are made at each node in the tree are chosen. HiCuts gives high-speed performance but the memory overhead for larger rule sets and with wildcard rules makes its use impractical. While HyperCuts algorithm considers multiple fields at a time. HyperCuts algorithm generally has a smaller depth of decision tree as compared to the HiCuts algorithm. Multiple fields are used at the same time on a single internal node.

While EffiCuts gives several new ideas such as tree separation and *equi-dense* cut. The tree separation is making small rules from large rule sets and makes decision trees so that there is no replication of larger rules. *equi-dense* known as unequal-sized cuts on rule density to divide rules evenly in each subspace [10].

IV. DRAWBACKS OF EXISITING SYSTEM

Extensive simulations using classbench databases (firewall, access control list, internet protocol chain) for the existing decision tree algorithms, HiCuts [8] and HyperCuts [9], discovered that the performance of these algorithms highly dependent on the rule set characteristics. Moreover, fixed-interval based cutting does not cover the actual space that rule covers consumes large storage and inefficient. For preprocessing, computation is required which consumes large memory and construction time. HiCuts algorithm gives high-speed performance but the memory overhead for larger sets and with many wildcard rules makes it nondeterministic. While HyperCuts algorithm does not give high-speed search performance and requires a huge amount of memory.

V. PROPOSED SYSTEM

Proposed algorithm improves over existing nonlinear type of packet classification algorithms. HiCuts takes the geometric view of the packet classification problem therefore it form the basis for new algorithm. A new efficient proposed packet classification algorithm is based on boundary cutting scenario by using bloom filters. Since boundary cutting is based on the disjoint space that each rule covers and perform the cutting according to space boundary is called as boundary cutting. It is more effective and deterministic which provides improved search performance and efficient in memory requirement.

Moreover, bloom filter which is an efficient data structure, used for representing set in order to support membership queries [7]. It becomes popular in real time networking applications like Network Intrusion Detection System (NIDS). It detects variants of attack signatures inside packet payloads by finding suspicious portion and analyzing them based on database of known attack signatures which are encoded inside bloom filter. Bloom filter avoid lookups in subset that contain no matching rules and sustain high throughput.

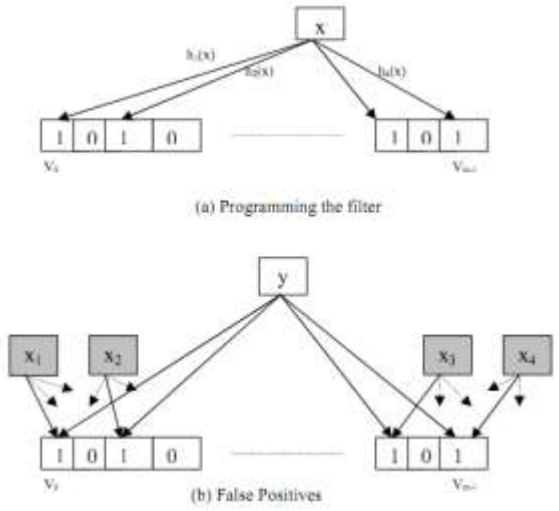

Figure 3. Bloom filters

As shown in fig.2, bloom filters are used for membership queries on a small subset and it is the way of compactly representing a set of items. To check if given item *y*, belongs to set S or not, the *k* independent hash functions are applied to *y* in a set of locations. The bloom filter accepts *y* if and only if all these locations are 1's. Then the filter accepts *y* (with highest probability) as it belongs to set S and if any of the mapped locations contain zero then *y* is rejected as being not belonging to set S [4]. Hence, a bloom filter could result in false positives. But the space saving controls the rate of false positives. False positives are tolerated if they occur with small probability in large scale applications. Algorithm represents the pseudocode for membership test of an element in bloom.

 ISSN: 2321-8169
4636 - 4640

Probability of False positive can be approximated as:

$$\Pr(\text{false positive}) = (1\text{-}P)^{k} \quad (1)$$

Data: x is the object key for which membership is tested.
Function: *ismember(x)* returns true or false to the membership test

$m \leftarrow 1$;
$j \leftarrow 1$;
while $m == 1$ *and* $j \leq k$ **do**
  $i \leftarrow h_j(x)$;
  if $B_i == 0$ **then**
    $m \leftarrow 0$;
  end
  $j \leftarrow j + 1$;
end
return m;

Algorithm: Pseudo code for bloom membership test

VI. ADVANTAGES OF PROPOSED SYSTEM

Proposed system has many advantages over existing system. It is more effective and efficient than that of current decision tree algorithms, especially in rule set characteristics. This algorithm is based on rule boundaries (starting and ending boundaries) rather than regular intervals. Moreover, it is deterministic and improved in terms of memory requirement. Boundary Cutting analysis uses binary search at internal nodes gives better search performance. Bloom filter, space efficient data structure filters large scale of packets in real time without any packet drop or missing with required optimal memory storage maintaining high throughput and gives 100% recall rate. They are used to store the source prefixes and destination prefixes.

VII. METHODOLOGY MODEL

As shown in fig. 4, the rule and packet generator randomly generate the rules and packets. The packet header fields contain the standard IP5- tuple as Source IP, Destination IP, Protocol (TCP, UDP any), Source Port, Destination Port, Action (Accept, deny, log, forward) and generates the database contains in bloom filter data structure.

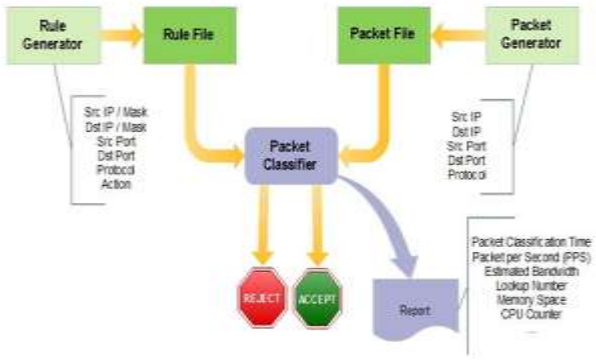

Figure 4. Experimental methodology model setup

Packet generator generates packets with header fields contain in that packet. Packet file contains all the packets which will be classified according to the rule set and policy to be in use. Now, packet classifier searches for the high-priority rule set matching the packet where each rule set identifies the prefix in the IP address, an exact match or wildcard and accordingly identifies the type of action to be performed.

The recorded observations are the number of cuts, fields, rule classification time, packet classification time, bandwidth, PPS(Packet Per Second), rule memory access(bytes), bucket memory access(bytes), number of bytes accessed per packet, number of search and search percentage used for evaluation.

VIII. SYSTEM FLOW ARCHITECTURE

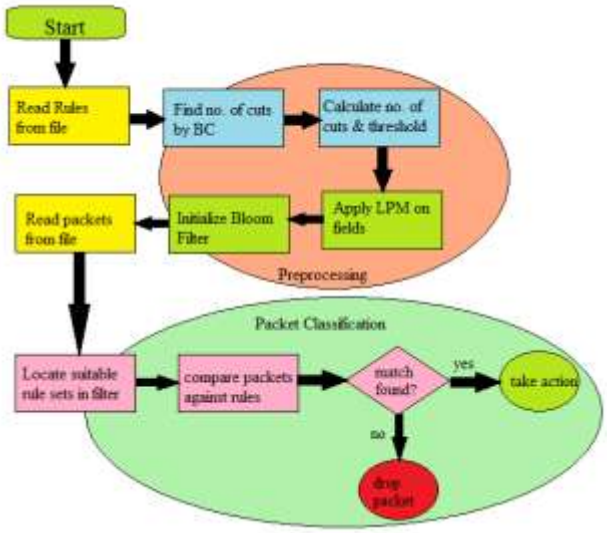

Figure 5. System flow architecture

Above figure shows the main whole procedures for packet classification using bloom filter and boundary cutting. First phase of preprocessing ends with rule classification and tree construction by calculating number of cuts and threshold. Number of cuts can be calculated as (NC = [20+ (number of rules / 1000)]) and threshold T = [(number of rules) / NC]. Then apply Longest Prefix Matching (LPM) on each field using bloom filter. Packet classification phase includes locating suitable rule set in filter by reading packets from file. All the header fields of packet will be checked with the governing rules. The highest priority one will be picked up out of those which completely match. So the final action (Accept/ Deny) will be taken and search will end [12].

IX. CONCLUSION

Large scale and multi-field packet classification is an important factor in NIDS, firewalls, network security, routers, and Quality of Service (QoS) assurance [12]. The packet classification needs the packet to be specified with the number

ISSN: 2321-8169
4636 - 4640

of packet headers, to inform which incoming flow is and which rule the packet is related with. To achieve a good performance, an algorithm must be designed in such a way that the best characteristics are getting combined in all approaches optimizing the time-space tradeoffs.

The proposed algorithm based on boundary cutting with a decision tree structure enables high performance packet classification. It would be helpful mainly in security, routers, intrusion detection systems, firewalls and other performance challenges in high speed environments. It is based on HiCuts by new heuristics and techniques by adding space-efficient data structure bloom filter in real time application. The recorded observations would be packet classification time, number of packet per second, number of search, percent of search numbers, rule memory access, preprocessing time, number of leaves, threshold and depth of tree. Proposed algorithm is deterministic and very effective in terms of speed and memory. It also avoids rule replication caused by unnecessary cutting enabling both highest priority match and multimedia classification [5].

REFERENCES

[1] H. Lim, N. Lee, G. Jin, J. Lee, Y. Choi, and C. Yim, "Boundary Cutting for Packet Classification," vol. 22, no. 2, pp. 443-456, April 2014

[2] H. A. J. Sistani, S. P. Amin, and H. Acharya, "Packet classification algorithm based on geometric tree by using Recursive Dimensional Cutting (DimCut)," vol. 2, no. 8, pp. 31-39, August2013.

[3] Haoyu Song, "Design and Evaluation of packet classification systems," September2006.

[4] M. A. Soliman and AMREL- HELW, "Network Intrusion Detection System using Bloom Filters," winter-2005.

[5] A.G. Alagu Priya and H. Lim, "Hierarchical packet classification using a Bloom filter and rule-priority tries," Comput. Commun., vol. 33, no. 10, pp. 1215–1226, Jun. 2010.

[6] S. Dharmapurikar, H. Song, J. Turner, and J. Lockwood, "Fast packet classification using Bloom filters," in Proc. ACM/IEEE ANCS, 2006, pp. 61–70.

[7] S. Dharmapurikar, J. Lockwood, "Fast and scalable pattern matching for Network Intrusion Detection Systems",vol.24, no. 10, pp. 1781-1792.

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

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

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

[11] Omar ahmed, "Towards efficient packet classification algorithms and architectures," Aug. 2013.

[12] H. A. J. Sistani and H. Acharya, "Fast Packet Classification, using the Recursive Dimensional Cutting by DimCut Packet Classification algorithm," vol. 7(5), pp. 600-613, May. 2014.