Boundary Cutting for Packet Classification using Bloom Filters

Ms. Namita N. Kothari
ME Information Technology[2nd,]
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

Prof. A.R. Panhalkar
Department of IT
Amrutvahini College of Engineering,
Sangamner, India
archana10bhosale@rediffmail.com

***Abstract*— Packet classification plays an important role in many applications such as firewalls, intrusion detection services, multimedia services, differentiated services, QoS(Quality of Service) routing, security. Current decision-tree-based classification algorithms such as HiCuts, HyperCuts, EffiCuts shows search performance by exploiting the geometric subspace to which input packet belongs. However, these algorithms exhibit relatively poor performance, huge storage requirement; fixed-interval based and involves complicated heuristics for determining number of cuts and field. The proposed Boundary cutting analysis is based on rule boundaries rather than regular (fixed) intervals. It is deterministic and more efficient in memory requirement providing good search performance. It also avoids rule replication due to unnecessary cutting by providing refined structure and multimatch classification. The algorithm make use of the bloom filters to avoid unnecessary access to the off-chip memory and filtering large amount of packets in real time without any packet drop or missing with required optimal memory space and improves the overall throughput greatly. Security roles are considered in Network Intrusion Detection System (NIDS) using packet classification.**

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

I. INTRODUCTION

A packet classification process is marking a packet to allow or disallow [2]. As shown in fig. 1, packet classification is nothing but comparing the bit stream given in the different fields in data packets with the classifier, which consist of a rule set. The comparison is done with the prefix bits. After the matched rule is found in the classifier respective action is applied on the packet which is defined in the classifier with the matched rule.

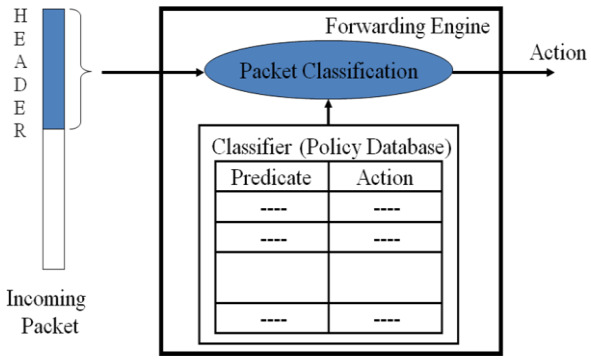

Figure 1. Packet Classification

Many algorithms and architectures have been proposed, the design of efficient classification system remain a challenging problem. Classification in today's network faces many challenges like inefficiency, inflexibility and hardness. Ideally, the classification should cover the criteria of throughput, storage, preprocessing time, incremental update support, scalability and adaptability to the structure of filter sets, power dissipation [3]. These different metrics should be carried out to evaluate the performance of classification algorithms. Most of the applications require highest priority matching and multimatch classification concept because of the interesting need for security in Network Intrusion Detection System (NIDS), load balancing, worm detection and packet level accounting to identify the context of the packets[1] and to perform important actions which might include dropping unauthorized packets, coping, scheduling, prioritizing and encrypting secure packets. Routers must perform the suitable action for the packet according to the traffic services to provide differentiated service [4].

However, existing decision-tree based classification algorithm such as HiCuts, HyperCuts and EffiCuts select

the number of cuts and field based on a locally optimized decision, which compromises the search speed and memory requirement [1]. For this, preprocessing is required which consumes much memory and construction time as it is the slowest operation in packet classification for large rule sets. Therefore many problems arise in building the decision trees. Moreover, it does not consider the actual space that each rule covers, hence finds ineffective.

A new efficient boundary cutting for packet classification using bloom filters improve overall performance with less storage space for the rule sets. This algorithm works on the principle of optimization of the rule set using region analysis. Hence, the amount of required memory is automatically reduced. It performs Binary search at internal nodes provides a good search performance for indexing [1]. Dynamic filter based analysis technique, Bloom filters provide most efficient solution for dynamic packet classification and filters large amount of packets in required time without any packet drop or missing with required optimal memory space. Bloom filters can suppress a large fraction of memory accesses and speed up rule matching [7]. Since rules are divided into multiple subsets. Bloom filters avoid lookups in subset containing no matching rules and gives high throughput [5]. Throughput can be calculated by dividing the total memory bandwidth by the memory bandwidth consumed per packet lookup.

II. LITERATURE REVIEW

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

In Exhaustive search technique, all entries in the filter set are examined sequentially that is similar to the Ternary Content Addressable Memory (TCAM) approach. these techniques perform well in terms of memory usage, do not require preprocessing, and can be updated incrementally easily but they require *O(N)* memory accesses per lookup, where *N* is the number of rules in the rule set. However, Exhaustive search uses brute force method which is most inefficient as linear search becomes prohibitively slow. Decomposition is decomposing multiple fields into several instances of single fields and then combining the end results, similar to RFC (Recursive Flow Classification) and BV (Bit Vector) algorithm. In RFC, memory use grows exponentially with the number of rules as they tend to have a high number of overlapping regions. BV algorithm has two drawbacks. First, poor classification increase memory usage dramatically. Secondly, the classification time vary depending on the arriving packet value, which may change the path inside the tree. Though it provides a high speed classification time, yet needs more memory usage and preprocessing time, which makes it unsuitable for systems that need frequent rule-set updates. Tuple space divide the filter according to the number of specified bits in the filter, then a subset of partition using simple exact match searches. The tuple space approach can narrow the scope of a multiple-fields search, by using tuples to partition the filter set. Tuple defines the number of specified bits in each field of the filters. This technique is effective when the number of distinct tuples is much less than the number of filter in the filter set. These classification algorithms has the lowest memory usage but requires a high preprocessing time and classification time could vary based on the nature of the rule set. HiCuts [8] and HyperCuts [9] techniques construct a decision tree from the filters and use the packet fields to navigate the decision-tree. Decision tree's leafs contain a rule or a subset of a rule. Classification is done by constructing a search key from the packet header fields and using this key to traverse through the tree. The main disadvantages

of this method are the high memory requirement and long preprocessing time. Due to the large amount of preprocessing required, both algorithms do not support incremental updates. Moreover, Classification time per packet can vary depending on the depth of the tree. Therefore, a majority of the above algorithms do not work well with incremental updates, which is a key problem that is resolved in proposed packet classification [2].

III. EXISTING SYSTEM

An incoming packet belongs to a certain flow when all the fields of the packet are in the range of that rule flow. In other words, each rule has F components and the *ith* component of rule R, referred to as R[i], which is a regular expression of the packet header on the *ith* field. That means, a packet (P) matches a particular rule (R) if, and only if, P is in the range of R[i] for every *ith* field of the header [11].

Application-specific integrated circuits(ASICs) with off-chip ternary content addressable memories(TCAM) is the best solution for wire-speed packet forwarding but the cost and high power consumption of TCAM made to explore some other algorithmic solutions. Furthermore, the throughput of TCAM algorithm is limited to a single character per clock tick. Multiple TCAM chips would require for scanning multiple characters at a time. It may require 2(L-1) TCAM entries for an L-bit port range field, making the exploration necessary [1]. Packet classification by Decision tree algorithm is nothing but constructing decision tree where the leaves of the tree have rules or subset of rules. If internal nodes are stored in an off-chip memory, then decision tree algorithm can provide high speed search performance. Decision tree based structure algorithms such as HiCuts, HyperCuts and EffiCuts [10] enables the highest priority match. HiCuts and HyperCuts algorithms select the field and number of cuts on a locally optimized decision, which compromises the memory requirement and search speed. In HiCuts, each rule defines a d-dimensional rectangle in space, where d=number of fields in the rule. It recursively cuts the space into subspace with fewer overlapped rule. A linear search is performed using rules to find a match for the incoming packets. It uses two parameters, threshold (*binth*) and space factor (*spfac*). HiCuts algorithm considers one field at a time while selecting the dimensions of the cuts. HyperCuts considers multiple fields at a time by minimizing the depth of decision tree and divide it into multiple fields [1].

While EffiCuts employs several new ideas such as tree separation and *equi-dense* cut. The tree separation is nothing but making small rules from large rules and makes multiple decision trees so that there is no chance of replication of large rules. *equi-dense* means unequal-sized cuts on rule density to distribute rules evenly in each subspace [10].

IV. DRAWBACKS OF EXISTING SYSTEM

Existing decision tree algorithms [8], [9], [10] select field and number of cuts based on locally optimized decisions. Moreover, regular-interval based cutting does not relate the actual space that each rule covers is ineffective and consumes large storage. For preprocessing, computation is required which consumes much memory and construction time. HiCuts algorithm gives high-speed performance but the memory overhead for larger sets and with many wildcard rules makes its use impractical. While HyperCuts algorithm either does not give high-speed search performance or requires a huge amount of memory depending on how to implement the pushing upward optimization. Performance of these decision tree algorithms is highly dependent on rule characteristics, especially a short-length prefix or number of rules with a wildcard.

V. PROPOSED SYSTEM

A new efficient proposed packet classification algorithm is based on boundary cutting using bloom filters. Boundary cutting is based on the disjoint space covered by each rule that is it finds out the space that each rule covers and performs the cutting according to space boundary, termed as boundary cutting. It is deterministic and more effective in providing improved search performance and much efficient in memory requirement.

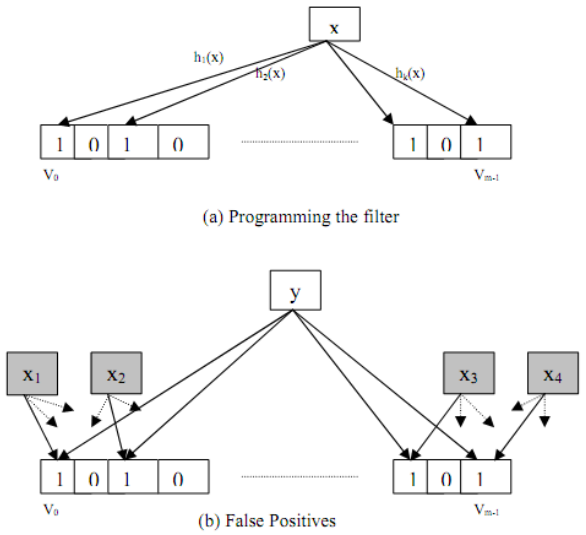

Figure 2. Bloom filter

Moreover, a space-efficient probabilistic data structure, used for concisely representing a set in order to support approximate membership queries [7]. It obtained so much popularity in a variety of fields especially large-scale network applications. Bloom filters are used to avoid unnecessary access to the off-chip memory. They also avoid lookups in subsets that contain no matching rules, making it possible to give high throughput. Bloom filters are used to represent a set of strings. As shown in fig. 2, Bloom filters can also be queried to check if a given string is stored in it or not and answer these queries quickly but with some false positives. False positive matches are possible, thus bloom filters has 100% recall rate. It provides most efficient solution for dynamic packet classification and filters large amount of packets in real time without any packet drop or missing with required optimal memory space.

VI. ADVANTAGES OF PROPOSED SYSTEM

Proposed system has many advantages. It is more effective than that of earlier decision tree algorithms since it is based on rule boundaries rather than regular intervals. It is deterministic and more efficient in terms of memory requirement. Boundary cutting uses binary search at internal nodes provides good search performance. Moreover, bloom filters, an efficient data structure filters large amount of packets in real time without any packet drop/ missing with required optimal memory space maintaining high throughput and 100% recall rate. They can be used to store the source and destination port prefixes.

VII. ARCHITECTURE

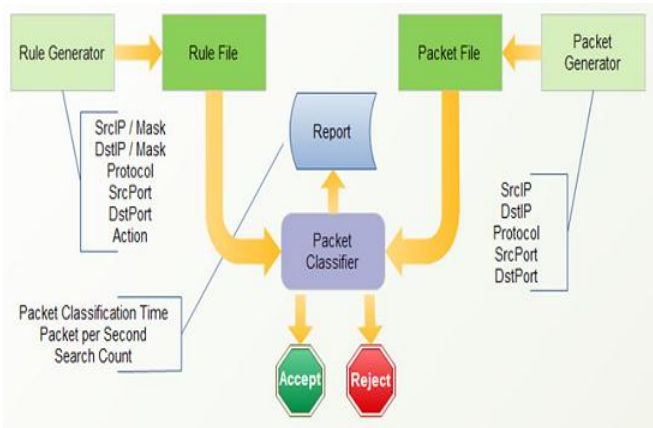

Figure 3. System Architecture

As shown in fig. 3, A set of policy rules control the action of packets whether to accept or reject. During preprocessing, all incoming packets need to be checked at fields. While packet classifier compare header fields of every incoming packet against a set of rules to assign flow identifier. The decision tree traverse to find the appropriate buckets that cover the incoming packet. When first match index found with highest priority, a packet will check the all fields of rules by binary search. The most prioritized packet is selected via those that match completely and final action(Accept/Reject) will be taken for that incoming packet and in this way search will end.

VIII. RESULT

The expected result of proposed system is design and evaluation of packet classification which has low storage, high throughput, classification speed, and scalability, flexibility in specification and security in real time applications like NIDS. Bloom filters accelerate the computational process of classification. For rule sets with 1000-10000 rules, proposed algorithm provides a packet classification through 20-30 on-chip memory and 1-4 off-chip memory access in average. It consumes a lot less

space compared to earlier decision-tree algorithms providing good search performance and no overlapping of rules. It is up to several kilobytes per rule. It is better alternative for TCAM for lower cost, flexibility and less power consumption.

IX. CONCLUSION

Here, proposed approach is for packet classification in network applications to provide value-added services such as security and quality of service (QoS) [6]. HiCuts and HyperCuts decision tree algorithms performance depend on the rule set characteristics in number of rules with a wildcard or a short-length prefix. For example, HiCuts can provide good search performance, but the memory overhead which has large set with many wildcard rules. Hence it is impractical in use. In HyperCuts algorithm, it does not provide high-speed performance and requires large amount of memory access depending on how to implement pushing upward optimization of rules.

While the cutting in previous algorithms is based on fixed interval, the cutting in proposed algorithm is based on rule boundaries; hence cutting in proposed algorithm is deterministic and effective. Refined structure avoid rule replication due to unnecessary cutting, consumes a lot less memory space compared to previous algorithms [1]. Packet Classification uses bloom filters to avoid unnecessary access to the off-chip memory and filtering large amount of packets in real time without any packet drop or missing with required optimal memory space sustaining high throughput. Bloom filters can be used to store the source and destination port prefixes [6]. Proposed algorithm enables multimatch classification and highest-priority match in which all matching results with highest priority rule must be returned.

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] Safaa O. Al-Mamory, Wesam S. Bhaya, Anees M. Hadi, “Taxonomy of packet classification algorithms,” vol. 21, no. 7, 2013.

[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