IPHV8I1042X
International Journal Of Advanced Research and Innovation -Vol.8, Issue .I
ISSN Online: 2319 – 9253
Print: 2319 – 9245

HIERARCHICAL PACKET CLASSIFICATION USING BINARY SEARCH ON LENGTH

[#1]NALE PREETHAM, M.Tech Student,
[#2]K.KUMARASWAMY, Associate Professor,
Dept of CSE,
MOTHER THERESSA COLLEGE OF ENGINEERING & TECHNOLOGY, KARIMNAGAR, T.S. INDIA.

ABSTRACT: Decision-tree-based packet classification algorithms such as Hi Cuts, Hyper Cuts, and Effi Cuts show excellent search performance by exploiting the geometrical representation of rules in a classifier and searching for a geometric subspace to which each input packet belongs. However, decision tree algorithms involve complicated heuristics for determining the field and number of cuts. Moreover, fixed interval-based cutting not relating to the actual space that each rule covers is ineffective and results in a huge storage requirement. A new efficient packet classification algorithm using boundary cutting is proposed in this paper. A packet classifier possesses a set of rules for classifying packets based on header fields. To classify a packet belonging to a particular flow or set of flows, network nodes like routers or firewalls must perform a search over a set of filters using multiple header fields of packet as a search key. Routers classify packets to determine their respective flow and the services they should receive. The paper deals with fast packet classification algorithms, Recursive Flow Classification (RFC) and Hierarchical Space Mapping (HSM). Packet classification is based on header fields of packet. RFC and HSM deal with header fields namely source and destination IP addresses as well as source and destination port number. Using those header fields mapping tables are computed and finally a decision is made about packet classification of individual packet. The RFC and HSM algorithms are implemented and the analysis of space required and time taken for classification is done.

Keywords: *Decision tree algorithms, Packet classification, Boundary cutting, Priority matching, Binary search.*

I. INTRODUCTION

The process of categorizing packets into "flows" in an Internet router is called packet classification. All packets belonging to the same flow obey a predefined rule and are processed in a similar manner by the router. Packet classification is an enabling function for a variety of internet applications including Quality of service (QoS), security, monitoring, multimedia Communications [1]. Growing and changing network traffic requirements invokes need of larger filter with more complex rules, which in turn gives rise to different fast packet classification algorithms. Packet classification is needed for non-best-effort services, such as firewalls and intrusion detection, routers, ISPs and usually in the most computation intensive task among others. Services such as bandwidth management, traffic provisioning, and utilization profiling also depend upon packet classification. Packet consists of header and information data and header consists of MAC address, IP address, port number etc. Traditionally, the Internet provided only a "best-effort" service, treating all packets going to the same destination identically, servicing them in a first come-first-served manner. However, internet users and their demands for different quality services are increasing day by day. So, Internet Service Providers are seeking ways to provide differentiated services (on the same network infrastructure) to different users based on their different requirements and expectations of quality from the Internet. For this, routers need to have the capability to distinguish and isolate traffic belonging to different flows. The ability to classify each incoming packet to determine the flow it belongs to is called packet classification and could be based on an arbitrary number of fields in the packet header. Packet classification is a multi-dimensional form of IP lookup and finding longest prefix matching to provide next-hop in routers. Classification of packet is an important function providing value-added services in Internet routers. Multi match classification notion is fetching an important research item because of rising need for network protection, for instance network intrusion detection systems and worm detection [1]. Usage of a high bandwidth and a tiny on-chip memory whereas rule database for packet classification resides in slower as well as superior capacity off chip memory by appropriate partitioning is enviable. The amount of memory necessary to accumulate packet classification table must be considered. Performance metrics in support of packet classification algorithms mainly comprise processing speed, while packet classification has to be carried out in wire-speed for each incoming packet. Processing speed is assessed by means of number of off-chip memory accesses necessary to find out class of a packet since it is the slowest procedure in packet classification. Our study analyzes a variety of decision-tree-based packet classification algorithms. Previous decision tree algorithms for instance HiCuts as well as Hyper Cuts select field as well as number of cuts based on a nearby optimized decision, which compromises search speed as well as memory prerequisite. This procedure requires a reasonable amount of preprocessing, which involves complex heuristics associated to each given rule set [2]. If a decision tree is appropriately partitioned with the intention that the internal tree nodes are accumulated in an on-chip memory and a huge rule database is accumulated in an off-chip memory, decision tree algorithm can make available extremely high-speed search performance. Decision tree algorithms obviously facilitate highest-priority match and multi-match packet classification. Innovative network applications have in recent times demanded a multipath packet classification in which the entire matching results along with highest-priority matching rule have to be returned. It is essential to discover competent algorithms to resolve classification problems. In our work a novel system of packet classification on basis of boundary cutting was put forward which finds out space that each rule performs cutting consistent with space boundary. Performance metrics for packet classification algorithms primarily include the processing speed, as packet classification should be carried out in wire-speed for every incoming packet. Processing speed is evaluated using the number of off-chip memory accesses required to determine the class of a packet because it is the slowest operation in packet classification. The amount of memory required to store the packet classification table should be also considered. Most traditional applications require the highest priority matching. However, the multi-match classification concept is becoming an important research item because of the increasing need for network security, such as network intrusion detection systems (NIDS) and worm detection, or in new application programs such as load balancing and packet-level accounting [8]. In NIDS, a packet may match multiple rule headers, in which case the related rule options for all of the matching rule headers need to be identified to enable later verification. In accounting, multiple counters may need to be updated for a given packet, making multi-match classification necessary for the identification of the relevant counters for each packet [5]. Our study analyzed various decision-tree-based packet classification algorithms. If a decision tree is properly partitioned so that the internal tree nodes are stored in an on-chip memory and a large rule database is stored in an off-chip memory, the decision tree algorithm can provide very high-speed search performance.

II. RELATED WORK

In this section, we describe related research work on packet classification algorithms. There are various packet classification algorithms proposed so far. Algorithms for packet classification can be categorized on various bases such as

1) Hardware based: They use Ternary content addressable memories (TCAMs).

2) Software based: Trie base, Decision tree, Hash based etc.

Different algorithms for packet classification are as follows:

- GoT: Grid of Tries
- EGT: Extended Grid of Tries
- HiCuts: Hierarchical intelligent Cuts
- HSM: Hierarchical Space Mapping
- AFBV: Aggregated and Folded Bit Vector
- CP: Compression Path
- RFC: Recursive Flow Classification
- B-RFC: Bitmap aggregation Recursive Flow Classification
- H-Tries: Hierarchical tries
- SP-Tries: Set Pruning tries
- BV: Bit Vector
- ABV: Aggregated Bit Vector

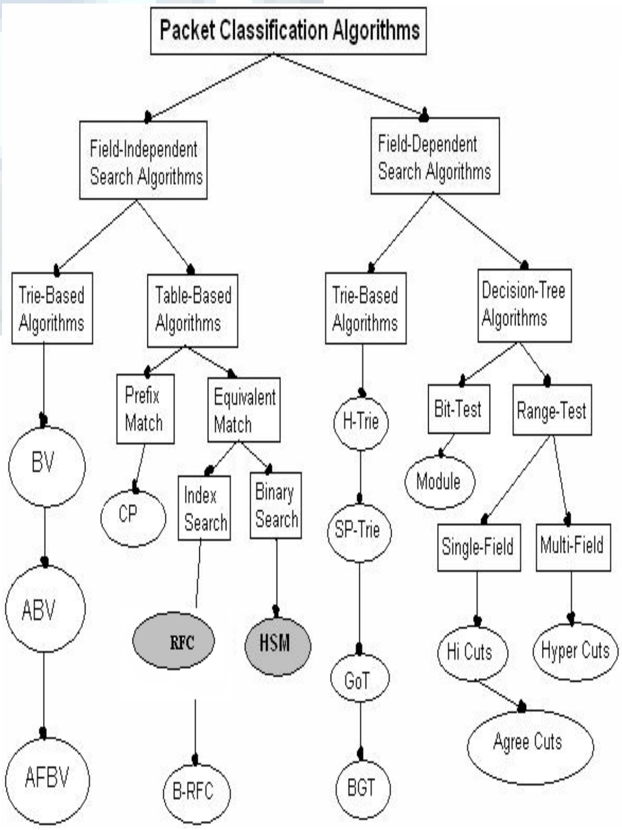

Figure-2 *Hardware Based Packet Classification*

A. Hardware Based Packet Classification

A high degree of parallelism can be implemented in hardware to increase the speed of classification. This can be achieved by using Ternary content addressable memories(TCAMs).But TCAMs cannot be used where flexible filter specifications are required as well as they have high power consumption and low scalability.

B. Software Based Packet Classification

Tri based algorithms has memory requirement of O(NW) and requires 2W-1 memory access per lookup, where N is number of filters and W is length of IP address area based quad tree (AQT) was proposed for two field filters.AQT supports efficient update time.The performance of trie-based algorithms are studied in. Schemes using decision tree to categarise filters into multiple sets is presented in papers and. The number of filters in each set is limited by predefined values and linear search is used to traverse the filter set. In the mechanism called cross producting, involving BMP lookups on individual fields and use of precomputed table to combine results of individual prefix lookups is presented. But in this scheme the memory requirements increase with the number of fields, O (*Nk*) where k is number of classified fields. The hash based idea [12] has given rise to 2-D filters. The filters with specific prefix length are grouped into a tuple, each tuple is then concatenated to create a hash key which is used for the tuple lookup. The matched filter can be found by probing each tuple alternately while tracking the least cost filter.

C. Software Based Packet Classification

Tri based algorithms has memory requirement of O(NW) and requires 2W-1 memory access per lookup, where N is number of filters and W is length of IP address . In area based quad tree (AQT) was proposed for two field filters.AQT supports efficient update time.The performance of trie-based algorithms are studied in . Schemes using decision tree to categorize filters into multiple sets is presented in papers. The number of filters in each set is limited by predefined values and linear search is used to traverse the filter set. In the mechanism called cross production, involving BMP lookups on individual fields and use of recomputed table to combine results of individual prefix lookups is presented. But in this scheme the memory requirements increase with the number of fields, O (N k) where k is number of classified fields. The hash based idea has given rise to 2-D filters. The filters with specific prefix length are grouped into a tuple, each tuple is then concatenated to create a hash key which is used for the tuple lookup. The matched filter can be found by probing each tuple alternately while tracking the least cost filter.

III. MULTI FIELD PACKET CLASSIFICATION

Firewall devices, traffic billing, QoS etc. are various applications in a network that requires multi field packet classification. Generally, multiple-field packet classification is not an easy problem.The categorization of packets into different flows is done by flow classifier which contains the set of rules. Packet classification requires that every packet is compared with the predefined database of rules and applying the action on the packet based on the rule of highest priority. Currently the order is increased for routers to supply QoS to various applications, hence the routers require new capabilities such as reservation of resources, per-flow queuing, admission control, and others. Distinguishing of packets of different flows is requirement for the router by the aforementioned mechanisms. As shown in figure 1 the instructions about the information carried by the packet are contained in its header, which include synchronization, length of packet, packet number, originating address, destination address, protocol, and port numbers of source/destination are used to find the matching rules in the database.

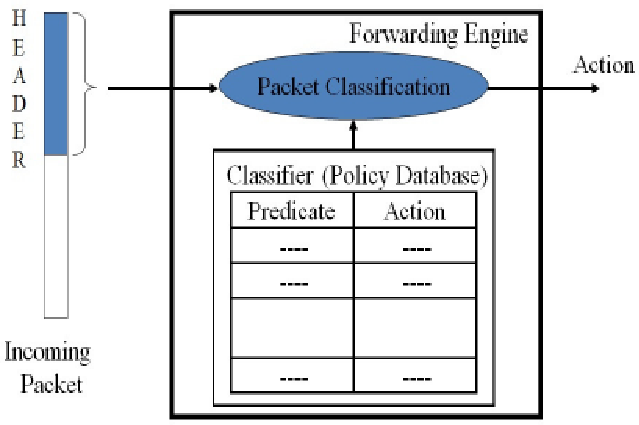

Figure-3 Packet classification

The database contains rule set labeled as R1, R2, R3…, RN where these rules are stored in a certain sequence and each rule consist of d values. Each field of the rule undergoes three types of matches3:

Exact match: where values of header field should be identical to the value of the rule filed. The exact match is used for protocol such as TCP and UDP.

Prefix match: The rule field should be a prefix of the header field where the prefix match is represented by using values followed by * wildcard. If the wildcard * occurred alone without values this means any value can be matched to this field.

IPHV8I1042X
International Journal Of Advanced Research and Innovation -Vol.8, Issue .I
ISSN Online: 2319 – 9253
Print: 2319 – 9245

Range match: The values of packet header fields are within a particular range defined by the rule. This is exploited for ranges of port number.

The packet matches the rule only if each packet field matches the corresponding rule field. Additionally, each rule in the classifier includes action which defines the process to be applied to the packet matching the rule. A tuple is basically the field in the header of a packet. 5 tuple4 is a term used in computer networks to refer to a set of five different values that make up a Transmission Control Protocol/Internet Protocol (TCP/IP) connection. The tuple is employed by network and system administrators in identifying the key requirements to create an operational, secure and bidirectional network connection between two or more local and remote machines.12 tuple are used in next generation packet classifications. The primary components of 12 tuple are the ingress port (router port number determine the ingress port width, as an example router with port number equal to 63 means it has 6 bit ingress port), address of Ethernet Source/Destination, type of Ethernet, ID of VLAN, priority of VLAN, address of IP Source/Destination, IP type of service bits, and port number of source and destination.

Multi-field packet classification requires high throughput along with maximum utilization of memory. For example, the cutting edge link rate has been pushed to 40Gbps, requiring that a packet is processed at the rate of 8 ns in the worst case (for packet having a size of 40 bytes minimum). Achieving such processing using available software processing method is not realistic. Therefore, finding new techniques to enhance the processing speed is popular research activity.

IV. PROPOSED SYSTEM:

The trie-based algorithms require small memory since each rule is stored exactly once, but they do not provide high throughput because of rule comparison at every rule node.The decision tree-based algorithms provide high throughput since the number of rules compared with an input packet can be controlled as a limited number, but they require excessive amount of memory because of high degree of rule replication. This paper proposes to combine these two types of algorithms. The proposed algorithm primarily constructs a trie and then applies a decision tree for nodes having more rules than a threshold value.

A. Area-Based Quad-TrieA binary trie is a bitwise data structure, representing a bitof a prefix using an edge of the trie. A prefix is stored into a node of the trie, in which the level and the path of the node relative to root node correspond to the length and the value of the prefix, respectively. The AQTis conceptually an area decomposition algorithm, but it can be described using a quadruple trie structure as well. As a two-dimensional trie,the quadruple trie uses two prefix fields at the same time. By concatenating each bit of source prefix and each bit of destination prefix of a rule, a code word is constructed, and the code-word is used to determine a node storing the rule in building an AQT trie. Figure 1 shows an example set of rules for packet classification and the corresponding AQT trie. For example, the code-word of rule R0 is 00 11 01, which is 031 in decimal. The rule R0 is stored in the node of level 3 and path 031 from root node.

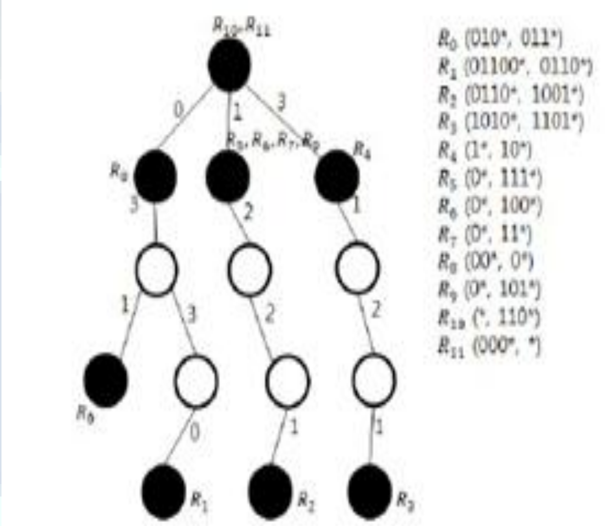

Fig-4 AQT structure

The decision tree construction process of the HiCuts algorithms to split a given rule set recursively by using partial header information. At each tree node, a decision is made to split the current rule set into a number of subsets, in which each subset represents a child node. For the set of rules in Figure 1, Figure 2 shows the decision tree of the HiCuts algorithm. Each internal node handles the cut field, the number of cuts, and child pointers. Rules are stored at leaf nodes. The shape of the decision tree is determined by two predefined parameters; binth and space factor.

The binth defines the maximum number of rules stored at a leaf node. In the tree construction process, if the number of rules included in the current node is not greater than the binth, the node becomes a leaf node. Otherwise the node is further split and becomes an internal node. A large binth value produces a decision tree with a small depth. The space factor controls the degree of rule replication caused in recursive splitting process.

Since more cuts are allowed in a large space factor, a rule set can be divided into a large number of subsets and a rule can be included in many leaf nodes, which results in large rule replication.

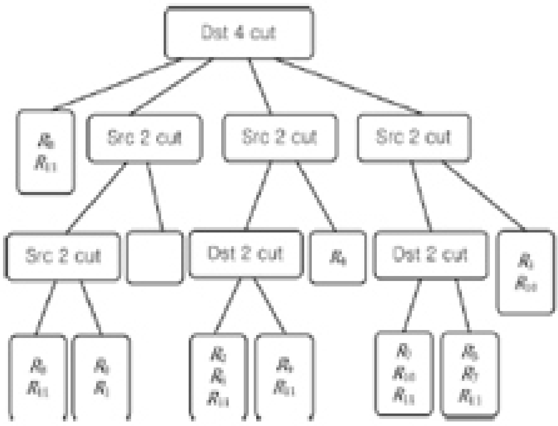

Fig-5 HiCuts structure (abinth=3 space factor= 1.5)

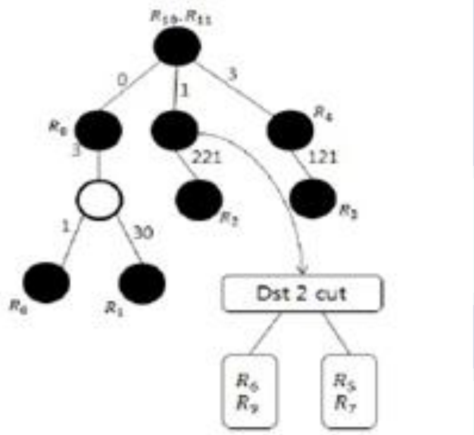

Fig-5 proposed structure (An AQT trie merged with a HiCuts tree

V. CONCLUSION

In recent times, multi match classification is fetching an important research item because of rising need for network protection. Pioneering network applications have demanded a multi-match packet classification in which the entire matching results along with highest-priority matching rule have to be returned and it is essential to discover competent algorithms to resolve classification problems. In our work an effective algorithm of packet classification based on boundary cutting was put forward. Our study has analyzed a number of decision-tree-based packet classification algorithms. The projected algorithm has two benefits such as boundary cutting algorithm is more effectual than that of earlier algorithms as it is based on rule boundaries to a certain extent than permanent intervals hence for this reason amount of necessary memory is considerably reduced. While boundary cutting loses indexing capability at internal nodes binary search advise advanced search performance. HiCuts as well as HyperCuts algorithms perform cutting based on a fixed interval, and hence partitioning is ineffective in dropping the number of rules that belong to a subspace. Cutting in proposed algorithm is deterministic to a certain extent than involving difficult heuristics, and it is more effectual in providing enhanced search performance and more competent in memory necessity. It is based on disjoint space covered by every rule therefore; packet classification table by means of projected algorithm is deterministically constructed and does not necessitate the complex heuristics used by previous decision tree algorithms.

REFERENCES

[1] B. Xu, D. Jiang, and J. Li, "HSM: A fast packet classification algorithm," *19th International Conference on Advanced Information Networking and Applications (AINA'05)*, vol. 1, pp. 987-992, 2005.
[2] T. V. Lakshman and D. Stiliadis, "High-speed policy-based packet forwarding using efficient multidimensional range matching," in *ACM SIGCOMM'98*, September 1998.
[3] A. Feldmann and S. Muthukrishnan, "Tradeoffs for packet classification," *AT & T Labs Research Florham Park, NJ*.
[4] T. Y. C. Woo, "A modular approch to packet classification: Algorithms and results," *Bell Laboratories, Lucent Technologies*.
[5] H. Kim, K. C. Claffy, M. Fomenkov, D. Barman, M. Faloutsos, and K. Lee, "Internet traffic classification demystified: Myths, caveats, and the best practicies," *ACM Conext*, 2008.
[6] P. Gupta and N. McKeown, "Packet classification using hierarchical intelligent cuttings," in *Hot Interconnects VII*, August 1999. *International Journal of Computer Theory and Engineering, Vol. 4, No. 6, December 2012*
[7] P. Gupta and N. McKeown, *Algorithms for Packet Classification, IEEE Network*, 2001.
[8] V. Srinivasan, G. Varghese, S. Suri, and M. Waldvogel, " Fast scalable level four switching," in *Proc. ACM SIGCOMM*, Sep. 1998, pp. 191-202
[9] M. Buddhikot, S. Suri, and M. Waldvogel, "Space decomposition techniques for fast layer-4 switching," in *Proc. IFIP 6th Int. Workshop on Protocols for High Speed Networks*, Aug. 1999, pp. 25-41.
[10] V. Sahasranaman and M. Buddhikot, "Comparative evaluation of software implementation of layer 4 packet classifcation schemes," in *Proc. IEEE ICNP*, Nov. 2001, pp. 220-228.
[11] D. E. Taylor, "Survey and taxonomy of packet classification techniques," *WUCSE* – 2004.
[12] V. Srinivasan, G. Varghese, and S. Suri, "Packet classifcation using tuple space search," in *Proc. ACM SIGCOMM*, Sep. 1999, pp. 135-146.

[13] P. Gupta and N. McKeown, “Packet classification on multiple fields,” in *ACM SIGCOMM*, Sept. 1999, pp. 147 160.
[14] F. Baboescu, S. Singh, and G. Varghese, “Packet classification for core routers: Is there an alternative to CAMs?,” in Proc. IEEE INFOCOM, 2003, pp. 53–63.
[15] S. Dharmapurikar, H. Song, J. Turner, and J. Lockwood, “Fast packet classification using Bloom filters,” in Proc. ACM/IEEE ANCS, 2006, pp. 61–70.
[16] P. Wang, C. Chan, C. Lee, and H. Chang, “Scalable packet classification for enabling internet differentiated services,” IEEE Trans. Multimedia, vol. 8, no. 6, pp. 1239 1249, Dec. 2006.
[17] H. Lim, M. Kang, and C. Yim, “Twodimensional packet classification algorithm using a quad-tree,” Comput. Commun., vol. 30, no. 6, pp. 1396–1405, Mar. 2007.