Packet Classification Algorithms: A Survey

Evangeline Asha B[1], Kavitha S[2]
[1] *PG Scholar, Hindusthan Institute of Technology, Coimbatore, Tamil Nadu, India*
[2] *Professor, Hindusthan Institute of Technology, Coimbatore, Tamil Nadu, India*
[1]*ashabensam@gmail.com*, [2]*kavithamani2003@yahoo.com*

Abstract: Packet classification is used by networking equipment to sort packets into flows by comparing their headers to list of rules, with packets placed in the flow determined by the matched rule. Packet's priority is decided by a flow and the manner which is processed. Packet classification is a difficult task because all packets must be processed at wire speed and rulesets can contain tens of thousands of rules. The main aim is to classify the packet such that the packet is classified in a best case time by reducing the power consumption of the network elements and reducing the memory consumption. Packet classification improves security and the worst case amount of processing time during packet classification is reduced. Different techniques were used. All packets belonging to the same flow obey a pre-defined rule and are processed in a similar manner by the router. For example, all packets with the same source and destination IP addresses may be defined to form a flow. Packet classification is needed for non "best-effort" services, such as firewalls and quality of service; services that require the capability to distinguish and isolate traffic in different flows for suitable processing. Floating point division is removed in some algorithms when the packet is classified. The rule match is verified for the query data. Also, the power and the resources are estimated. This estimation helps to redesign the system if necessary.

Index Terms: best effort, firewalls, QoS, traffic

1. INTRODUCTION

Internet is today one of the most important part of our daily life. With the progress in the internet people are progressing in every sphere of life as it not only makes our tasks easier but also saves a lot of time. Today internet is used for different purposes depending upon the requirement. Internet access connects individual computer terminals, computers, mobile devices, and computer networks to the Internet, enabling users to access Internet services, such as email and the World Wide Web[7]. Every Internet router today can forward entering Internet messages (packets) based on the destination address. The 32-bit IP destination address is looked up in a table which then determines the output link on which the packet is sent. Packet classification is the process of categorization the packets according to its header fields. This process is applied in the forwarding machine (like router, firewall, Intrusion Detection System (IDS), Intrusion Prevision System (IPS), …, etc) to identify the context of the packets and to perform important actions. The action might include dropping unauthorized packets, coping, scheduling and prioritizing, and encrypting secure packets. In order to handle internet traffic to provide differentiated service, the routers for the Internet Service Provider (ISP) must have the ability to classify the packets by examining the values of header fields. Also, it must perform the suitable action for the packet according to the traffic services. The traffic services may deal with different service for the same path, for example packet filtering, preventing the malicious attacks, accounting and billing, and traffic rate limiting.

2. PERFORMANCE METRICS OF PACKET CLASSIFICATION

• *Search speed* — Faster links requires faster classification. For example, links running at 10Gbps can bring 31.25 million packets per second (assuming minimum sized 40 byte TCP/IP packets).
• *Low storage requirements* — Small storage requirements enable the use of fast memory technologies like SRAM (Static Random Access Memory). SRAM can be used as an on-chip cache by a software algorithm and as on-chip SRAM for a hardware algorithm.
• *Ability to handle large real-life classifiers*.
• *Fast updates* — As the classifier changes, the data structure needs to be updated. Data structures are categorized into those which can add or delete entries incrementally, and those which need to be reconstructed from scratch each time the classifier changes. When the data structure is reconstructed from scratch, it is called "pre-processing". The

update rate differs among different applications: a very low update rate may be sufficient in firewalls where entries are added manually or infrequently, whereas a router with per-flow queues may require very frequent updates.
• *Scalability in the number of header fields used for classification*.
• *Flexibility in specification* — A classification algorithm should support general rules, including prefixes, operators (range, less than, greater than, equal to, etc.) and wildcards. In some applications, non-contiguous masks may be required. [4]

3. PROBLEMS IN CLASSIFIYING THE PACKETS

The criteria for classifying packet is called rule R, and the set of finite rules R1, R2, , Rn contained in forwarding machine is called rule database or classifier. The fields of rule and packet header are related, For example, the rule that implement IPv4 consist of 5 fields (source IP address, destination IP address, protocol type, source port, and destination port). The incoming packet to router matches specific rule if the distinct fields in the packet match the corresponding fields in that rule.

Since a packet may match more than one rule in the database, assigning a cost to each rule can avoid this ambiguity. The packet classification problem is how to determine the lowest-cost matching for the incoming packet. The packet must match at least one rule. There are three matching [7] types.
1. Exact match: The values of rule fields and Packet header fields must be identical.
2. Prefix match: The rule fields values must be prefix for the header fields values.
3. Range match: The header fields values must lie in the range specified by the rule.

Table 1 lists some examples and application areas of packet classification. This table gives the related requirements of number of fields for matching classification types and filter examples. Based on the applications, the number of the fields varies. This can be demonstrated with examples of the filter.

Table 1Packet classification examples

Application	**Number of fields**	**Filter example**
Switching MPLS	Single / Exact match	Send packets directly to end hosts
Forwarding	Single / Longest prefix match	Send all packets to the ISP's router
Flow identification, IntServ	Multiple / Exact match	Give packets with highest priority
Filtering, DiffServ	Multiple / Prefix or Range match	Drop all packets
Load balancing	Multiple / Scan with Exact or Prefix match	Re-direct packets in DATA field to audio server
Intrusion Detection	Multiple / Scan and match Reg. Expressions	Create alarm in DATA field.

.

4. CLASSIFICATION ALGORITHMS

The overall structure of the packet classification algorithm can be summarized as given below in table 2. All the algorithms fall into seven classes

Table 2 Classification Algorithms

No	**Class**	**Algorithms**
1.	Data structure	Linear search, Caching
2.	Two dimensional	Hierarchial trie, Set Pruning Trie, Grid of Tries
3.	Divided Conquer	BV, ABV, RFC, HSM, Cross-producting
4.	Decision Tree	Hicuts, Hypercuts, D-cuts, Expcuts, Hypersplit
5.	Tuple Space and hash table	Tuple Space Search, BSOL, Hybrid approach to packet classification
6.	Heuristics at bit level	DBS
7.	Hardware	TCAM, BV-TCAM

The naive algorithms depend on the primary working principals offered by the available techniques, for example, linear search, and caching techniques. The linear search algorithms are characterized by efficient storage since it requires only O(N) memory locations, and the time to classify the packet grows linearly with the number of rules N. The algorithms that depend on Caching techniques are characterized by not working well in practice because of poor hit rate [16], and they still need a fast classifier as a backup when cache fails [9]. The two dimensional algorithms handle the rules that contain two fields; they are use to handle flow aggregation for MPLS and VPN, and these algorithms use in firewall where many rules contain distinct protocol ranges [15]. The extended two dimensional algorithms extend two dimensions algorithms to multiple dimensions based on source-destination matching, and pruning based on source destination fields will reduce the number of rules to be searched [3].

The divide and conquer algorithms are used to divide the complex problem into simpler subproblems and then efficiently combining the results into final stage [3]. Decision tree algorithms are characterized by difficult to do incremental update [14], low efficiency with large number of wildcard, better tradeoff between speed and memory, and they are efficient with edge routers [15]. Tuple space and hash table algorithms are characterized by dividing the search space into regions that can be searched in parallel, using exact matching [16], inefficient with large number of rules, tuple space and hash table algorithms are difficult to make updating [17], and the linear search on the tuples is more efficient than linear search on rules [15].

4.1. *Linear search*

The simplest data structure is a linked-list of rules stored in order of decreasing priority. A packet is compared with each rule sequentially until a rule is found that matches all relevant fields. While simple and storage-efficient, this algorithm clearly has poor scaling properties; the time to classify a packet grows linearly with the number of rules[4].

4.2. *Hierarchial tries*

This algorithm suffers from wasted time because of using backtracking, and it is scalable for 2-Dimension. The storage complexity is of O(NdW) [3]. This algorithm is also called as “multi-level tries”, “backtracking search”, or “trie-of- tries” [4].

4.3. *Set-pruning tries*

This algorithm has reduced query time obtained by replicating rules to eliminate recursive traversals. It suffers from prefix replication and it is scalable for 2 dimensional [22]

4.4. *Grid-of-tries*

This algorithm reduces the storage space by allocating a rule to only trie node as in hierarchial trie, with look-up time O(W) [12].

4.5. *Crossproducting*

Cross-producting [12] is suitable for an arbitrary number of dimensions. Packets are classified by composing the results of separate 1-dimensional range lookups for each dimension scalable for data base smaller than 50 rules [16], it Requires caching for larger classifiers [8], and it suffers from redundancy [18].

4.6. *Distributed Crossproducting of Field Labels (DCFL)*

This method DCFL [2] is a novel combination of new and existing packet classification techniques that leverages key observations of filter set structure and takes advantage of the capabilities of modern hardware technology

4.7. *Recursive Flow Classification (RFC)*

RFC [8] is a heuristic for packet classification on multiple fields. Classifying a packet involves mapping S bits in the packet header T to a bit action identifier, where T<<S. A simple, but impractical method could pre-compute the action for each of the different packet headers, yielding the action in one step. RFC attempts to perform the same mapping over several phases. at each stage the algorithm maps one set of values to a smaller set. In each phase a set of memories return a value shorter (i.e., expressed in fewer bits) than the index of the memory access.

4.8. *Hierarchical Intelligent Cuttings (HiCuts)*

HiCuts [23] partitions the multi-dimensional search space guided by heuristics that exploit the structure of the classifier. Each query leads to a leaf node in the HiCuts tree, which stores a small number of rules that can be searched sequentially to find the best match. The characteristics of the decision tree (its depth, degree of each node, and the local search decision to be made at each node) are chosen while preprocessing the classifier based on its characteristics

4.9. *Multidimensional Hierarchical Intelligent Cutting (HyperCuts) algorithm*

Hypercuts is characterized by using multi cuts in internal nodes to reduce the Decision Tree depth, it has high storage than Hicuts, it is efficient with edge routers [15], it performs well under practical conditions [19], and it is difficult to support incremental updates [15]. This algorithm provides more advantages when applied practically.

4.10. *Explicit Cutting (ExpCuts) algorithm*

This does not suffer from excessive memory access and worst case search time, and it works with multi-core Network Processors [23].

4.11. *HyperSplit algorithm*

HyperSplit algorithm is characterized by its suitability for various rule sets, and using binary search, and it has better preprocessing time than Hicuts and HSM [23].

4.12. ***Dynamic Cuts(D-Cuts) algorithm***

D-Cutsis characterized by achieving higher speed than Hicuts because it adopts a network statistics into decision tree, suffering from long term tree searching [18], adopting structural characteristics and network statistics, and focusing on reducing the depth.

4.13. ***Hierarchical Space Mapping (HSM) algorithm***

HSM is characterized by using balanced binary search tree [18], high preprocessing time, and using rule based space decomposition on each field to achieve deterministic worst case search time [23].

4.14. ***Adaptive Hierarchical Space Mapping (AHSM) algorithm***

AHSM is characterized by using alphabetic search tree with recursive intersecting table, and adopting network statistics [18].

4.15. ***Improved Hierarchical Space Mapping (C-HSM) algorithm***

C-HSM is characterized by using pruning trie, and using heuristic to compress the space and save the memory [24].

4.16. ***Discrete Bit Selection (DBS) algorithm***

DBS algorithm is characterized by higher performance than Hicuts and HSM, applying heuristic classification on bit level, performing well in both temporal and special performance, and it is more scalable than HSM and Hicuts [19].

4.17. ***Shifted Bits (sBits) algorithm***

sBits is characterized by combining the advantages of RFC and Hicuts, it has efficient update time, and it is more scalable than HSM, Hicuts, RFC, and Hypercuts [23].

4.18. ***Binary Search On Level (BSOL / O(log W)) algorithm***

BSOL is characterized by depending on Hash table and binary tree, multidimensional scheme, and it has better memory and time performance than EGT-PC [25].

4.19. ***Fat Inverted Segment Tree (FIS-Tree) algorithm***

FIS tree is characterized by efficient update, for two dimensional classification as a modification of a segment tree. A segment tree stores a set of possibly overlapping line segments, to answer queries such as, finding the highest priority line segment containing a given point. it scales well for 2-D, and it may adopt clustering to reduce memory storage when the number of dimensions is larger than 2 [3]. A FIS-tree is a segment tree with two modifications:
(1) The segment tree is compressed (made "fat" by increasing the degree to more than two) in order to decrease its depth and occupies a given number of levels, and
(2) Up-pointers from child to parent nodes are used.

4.20. ***Tuple Space Search***

The basic tuple space search algorithm [12] decomposes a classification query into a number of exact match queries. The algorithm first maps each –dimensional rule into a -tuple whose i^{th} component stores the length of the prefix specified in the dimension of the rule. Hence, the set of rules mapped to the same tuple are of a fixed and known length, and can be stored in a hash table. Queries perform exact match operations on each of the hash tables corresponding to all possible tuples in the classifier. The number of tuples could be very large, up to $O(W^d)$, in the worst case.

4.21. ***Ternary CAMs***

TCAM stores each W-bit field as a (val, mask) pair; where val and mask are each bit numbers. For example, if W=5, a prefix 10* will be stored as the pair. An element matches a given input key by checking if those bits of val for which the mask bit is '1', match those in the key. The -bit bit-vector, matched, indicates which rules match and so the -bit priority encoder indicates the address of the highest priority match. The address is used to index into a RAM to find the action associated with this prefix.
There are, however, some disadvantages to TCAMs:
1. A TCAM is less dense than a RAM, storing fewer bits in the same chip area.
2. TCAMs dissipate more power than RAM solutions because an address is compared against every TCAM element in parallel.

4.22. ***Bitmap-intersection***

The bitmap-intersection classification scheme, proposed in [6], is based on the observation that the set of rules, S, that match a packet is the intersection of d sets, S_i , where S_i is the set of rules that match the packet in the i^{th} dimension alone. While cross-pro ducting pre-computes S and stores the best matching rule in S, this scheme computes S and the best matching rule during each classification operation.

4.23. ***Extended Grid of Tries (EGT)***

This algorithm is characterized by extending the two dimension Grid of Tries to process multidimensional fields, and using switch pointer and jump pointer techniques if the specific node is fail in matching [9] .

4.24. ***Extended Grid of Tries-Path Compression (EGT-PC) algorithm***

EGT-PC is more predictable than EGT, allowing improvement using multi bits tries, it can be implemented in SRAM, it removes the single branch path, and it is scalable for multi dimensional [9].

4.25. ***Bit Vector (BV) algorithm***

BV is characterized by slow dynamic update, bad memory using [15], it does not scale well for large data base and very high speed system [3], and it provides Parallel lookup header fields. It is also called as Lucent bitvector scheme or Parallel Bit-Vectors (BV). Here the initial assumption is made that the filters may be sorted according to priority. Parallel BV utilizes a geometric view of the filter set and maps filters into d-dimensional space. Searches are simple after constructing d data structures.

4.26. ***Aggregate Bit Vector (ABV)***

ABV algorithm seeks to improve the performance of the Parallel BV technique. ABV algorithm suffers from false positive [9], and from unpredictable average case search time, it uses rule aggregation to reduce memory access, it uses rule re-arranging to solve false positive problem [15], it can provide suitable throughput [19], and parallel lookup header fields. ABV converts all filter fields to prefixes, hence it incurs the same replication penalty as TCAMs [2].

4.27. ***HaRP algorithm***

This algorithm is characterized by parallel lookup for high performance, high memory efficiency, easy incremental update, applied on multi processor system, exhibiting Hash storage utilization, and efficient dynamic update [14].

4.28. Hybrid (Tuple + Top-Down Tree) algorithm

The hybrid algorithm is characterized by combining hash table with binary trie, and it is applicable with NP [7].

4 IMPLEMENTATION OF PACKET CLASSIFICATION ALGORITHM

Packet Classification Algorithm can be implemented by two major types: Software-based and Hardware-based implementations [19].

1. Software-based implementation: This type is used with general purpose processors and Network Processors (NP). The software-based algorithms can be categorized into two field's dependency types [19]:

Field-independent algorithms: These algorithms will build the index tables independently for each field in the rule. Then, the rules are grouped together. HSM [18], and RFC [8] algorithms use independent parallel search on index tables .The results of the searches are combined into a final result in several phases. Though these algorithms are fast in classification, they need large memory to store the search tables.

Field-dependent algorithms: These algorithms deal with the fields of the rule in dependently manner. Thus, there is no need to group the results in final stage. Hicuts [8], and Hypercuts [9] algorithms are examples of this type of field dependency. These algorithms use intelligent and simple decision tree classifier. Also, these algorithms require less memory than field-independent search algorithms. However, they cannot ensure stable worst case classification speed.

2. Hardware-based implementation: This type is used with ASIC (Application Specific Integrated Circuits) or with FPGA (Field Programmable Gate Array). This type of implementation is used with internet backbone routers for the high speed that support to handle the packets [20][21].

5 PERFORMANCE ANALYSIS OF VARIOUS ALGORITHMS

There are many classification schemes that are used for classifying the packets. Each scheme has its own advantage and disadvantage. The algorithm is chosen according to the requirement for classifying. The table 3 depicts some algorithms that has the worst time complexity as well as the worst space complexity. This table helps the designers to selct the algorithm according to the requirement in th real-time applications:

Table 3 Performance comparison

No	Algorithm	Worst time complexity	Worst space complexity
1	Linear search	N	N
2	Hierarchial trie	W^d	NdW
3	Pruning trie tree	DW	N^d
4	Grid of tries	W^{d-1}	NdW
5	Cross producting	DW	N^d
6	RFC	D	N^d
7	Tuple space	N	N
8	TCAM	1	N
9	Bitmap intersection	DW+N/ (memory unit)	DN^2

10	EGT	(H+2)T	-
11	EGT-PC	W^2+BP/C	N
12	BV	DW+N/M	DN^2
13	ABV	DW+N/M	DN^2
14	HiCuts	D	N^d
15	HyperCuts	-	-
16	HyperSplit	D*log(2N+1)	N^d
17	HSM	D(log(2N+1))	N^d
18	AHSM	2N+1* log(2N+1)	N^d
19	CHSM	DW+log(D-1)	N^d
20	DBS	D	-
21	BSOL/O(log W)	log(W)+S/C	N^d
22	FIS Tree	(L+1)W	$LN^{1+I/L}$
23	Independ-ent set	I	N
24	sBits	-	-
25	BV-TCAM	-	
26	HaRP	-	-
27	Hybrid	log (N/2^$\sum^b_k$V)+ V	N(1+2/V)- 2^$\sum^b_k$V

Where,

N	Number of Rules
D	Number of dimensions in the rule
W	Length of bit strings (for IPv4 is 32 bit, for IPv6 128 bit)
M	Memory width
C	Cache line size
L	Number of levels
I	Number of independent sets
T	Time to find the best prefix in the trie
H	Maximum length of the trie (32 for IP address)
S	Size of bucket (the bucket is source-destination prefix pair)
B	Number distinct source-destination prefixes pairs matching a packet
P	Maximum numbers of rules that share the same source-destination prefix pairs
V	Maximum number of leaf chain
2^$\sum^b_k$	Number of selected bits to create index table

6 CONCLUSION

Today, internet has become more open and sharable requirement. There are many algorithms used for classifying the packet. But, the difficulty is to choose the best packet classification algorithm such that there is no worst case time and space complexity. This paper, presents an overview about the algorithms used for classification, its performance metrics and problems in classifying the packets.

REFERENCES

[1] Alan Kennedy, “Ultra high throughput low power packet classification”, IEEE journal on VLSI, VOL. 22, NO. 2, FEBRUARY 2014, pp 286-299

[2] David E Taylor, “Scalable and taxonomy of packet classification techniques”, ACM Computing Surveys, Vol. 37, No. 3, September 2005, pp 238-275.

[3] Florin Baboescu and George, “Scalable packet classification”, IEEE ACM transactions on networking, Vol 13, No 1, February 2005, pp 2-14

[4] Pankaj Gupta and Nick McKeown, “Algorithms for packet classification”, pp 1-29.

[5] Viktor, “Packet classification algorithm”, Information sciences and technologies bulletin of the ACM Slovakia, Vol 4, No4, 2012, pp 31- 42.

[6] Xiaohui, Yusheng, Yongcheng, “A hybrid approach to packet classification”, NII Journal No 8, 2004, pp 13-21

[7] Safaa, Wesam, Anees, “Taxonomy of packet classification algorithms”, Journal of Babylon University, Applied Sciences, No7, Vol 21, 2013, pp 2296 – 2307

[8] P. Gupta and N. McKeown, “Packet Classification on Multiple Fields”, Proc. Sigcomm, Computer Communication Review, Harvard University, vol. 29, no. 4, pp 147-60, Sep 1999.

[9] F. Baboescu, S. Singh, and G. Varghese, “Packet classification for core routers: Is there an alternative to CAMs?” in Proc. IEEE Int. Conf. Comput. Commun., Apr. 2003, pp. 53–63.

[10] M. Gupta and S. Singh, “Greening of the internet,” in Proc. ACM Special Interest Group Data Commun. Conf., Aug. 2003, pp. 19–26

[11] David Taylor, Spitznagel, “Packet classification using extended TCAMs”, ICNP, 2003, pp 1-12.

[12] V. Srinivasan, S. Suri, G. Varghese, and M. Waldvogel. “Fast and Scalable Layer four Switching,” Proceedings of ACM Sigcomm, pages 203-14, September 1998.

[13] A. Feldman and S. Muthukrishnan. “Tradeoffs for packet classification,”

Proceedings of Infocom, vol. 3, pages 1193-202, March 2000.
[14] Pong F., and Tzeng N. F., 2009,"Hashing Round-down Prefixes for Rapid Packet Classification", Proceeding of the conference on USENIX Annual technical conference, pp. 1-15.
[15] Madhi D., and ramasamyK., 2007, "Network routing algorithms, protocols, and architectures", Morgan Kaufmann, USA, pp. 1-957.
[16] Meiners C. R., Liu A. X., and Torng E., 2010,"Hardware Based Packet Classification for High Speed Internet Routers", Springer, New York-USA, pp. 1-122.
[17] Sun X., Sahni S. K., and Zhao Y. Q., 2005, "Packet Classification Consuming Small Amount of Memory", IEEE/ACM Transactions on Networking, Vol 13, No 5, pp. 1- 11.
[18] Xu B, Zhou G., Xue Y., and Li J., 2007, "Ahsm: adaptive packet filtering with network traffic statistics", Proceeding of the 20th IASTED International Conference Parallel and Distributed Computing and System (PDCS 2008), pp. 258-263.
[19] Yang B., Wang X. , Xue Y., and Li J., 2009,"DBS: A Bit-level Heuristic Packet Classification Algorithm for High Speed Network ", In Proceeding of the 15th IEEE International Conference on Parallel and Distributed System (ICPADS), pp. 260-267.
[20] Sherwood T., Varghese G., and Calder B., 2003, " A Pipelined Memory Architecture for High Throughput Network Processors", In Proceedings of the 30th International Symposium on Computer Architecture (ISCA), pp.1-12.
[21] Jiang W., and Prasanna V. K., 2009, "A FPGA-based Parallel Architecture for Scalable High-Speed Packet Classification", In Proceeding of 20th IEEE International Conference on Application specific Systems Architectures and Processors, (ASAP '09) pp. 1-8
[22] P. Tsuchiya. "A search algorithm for table entries with non-contiguous wildcarding," unpublished report, Bellcore
[23] Qi Y., and Li J., 2004,"Packet Classification with Network Traffic Statistics ", Proceeding of the 3rd Trusted Internet Workshop (TIW), pp. 1-9.
[24] Cao C., Yuan J., Wang H., and Dong Y., 2006, "An Improved HSM Packet Classification Algorithm Using Pruning Trie and Space Compression", Proceeding of International Conference on Wireless Mobile and Multimedia Networks proceeding (ICWMMN), pp. 1-4.
[25] Lu H., and Sahni S., 2007, "O(log W) Multidimensional Packet Classification", IEEE/ACM Transactions on Networking, Vol 15, No 2, pp. 462-472.
[26] Shanliang Zhang and Lihua Zhu, "A Packet Classification Algorithm based on Improved Decision Tree", Journal of networks, Vol 8, No 12, Dec 2013, pp 2864-2871
[27] Nian, Xiaan, "An efficient hybrid hierarchial Trie packet classification algorithm based on No Prefix relationship", Journal of computational information systems 9:22, 2013, pp 9193-9202
[28] Francis Chang, Wu Chang Feng, "Appropriate caches for packet classification", IEEE INFOCOM, 2004.
[29] Kai Zheng, Hao Che, Zhijun Wang, Bin Liu, Xin Zhang, "DPPC-RE TCAM based distributed parallel packet classification with range encoding", IEEE Trans on computer, pp 1-35.
[30] D.Pao, Zhou, Liu, "Enhanced prefix inclusion coding filter encoding algorithm for packet classification with TCAM", IET computer digital technology, 2007, 1, (5), pp 572-580
[31] Yadi Ma, Suman Banerjee, "A smart pre-classifier to reduce power consumption of TCAM for Multi-dimensional Packet Classification".
[32] Xiaohui, Yusheng JI, Yongcheng LEI "A hybrid approach to packet clasiification", NII Journal No. 8 (2004), pp 13-21
[33] Alok, Sreenaath and Sekar, "Fast packet classification for Snort by native compilation of rules", LISA'08, pp 159-164
[34] Weirong, Viktor, "Large scale wire speed packet classification on FPGA".
[35] Fan, Carlos, Yiftachu, "Joint source coding and packet classification for real time video transmission over differentiated service networks", IEEE transanction on multimedia, Vol7, No4 August 2005, pp 716-727.