

# Taxonomy of Packet Classification Algorithms

Safaa O. Al-Mamory      Wesam S. Bhaya  
College of Computer Technology, University of Babylon

Anees M. Hadi  
Department of Computer, College of Sciences for Girls, University of Babylon

## Abstract

The process of categorizing Internet traffic in forwarding machine called packet classification. This process becomes very important in the last years, due to the huge evolution for the network services.

This paper explains a taxonomy for the most popular and modern packet classification algorithms with its distinct features. As a result, this paper will guide the interested in packet classification field how can choose a suitable algorithm according to network service requirements.

### Keywords

Packet Classification, Packet Classification Algorithms, matching types, implementation types, field dependency, Network Intrusion Detection System (NIDS).

## الخلاصة

ان عملية تمييز بيانات الانترنت ضمن اجهزة الشبكات تسمى بتصنيف الحزم. اصبحت تلك العملية مهمة جدا في السنوات الاخيرة، نتيجة التطور الكبير في خدمات الشبكات وخصوصاً شبكة الانترنت. يوضح هذا البحث اكثر خوارزميات تصنيف الحزم شيوعاً وحدائثاً، وكذلك يوضح البحث ما هي الصفات المميزة لكل خوارزمية. وبالنتيجة، فإن هذا البحث يرشد المهتمين في مجال تصنيف الحزم كيفية اختيار الخوارزمية المناسبة طبقاً الى متطلبات خدمات الشبكة.

## 1. Introduction

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 [Madhi D. *et al*,2007].

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 [Meiners C. R. *et al*,2010].

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 [Madhi D. *et al*,2007][Gupta P. *et al* a, 1999].

In section 2, We describe the packet classification problem. Section 3 describes the implementation of packet classification algorithms. Section 4 describes the packet classification NIDS. Section 5 describes the taxonomy for algorithms and the specific features for each algorithm.

## 2. Packet Classification Problem

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 [Madhi D. *et al*,2007].

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 [Varghese G., 2005].

Since a packet may match more than one rule in the database, assigning a cost to each rule can avoid this ambiguity [Varghese G., 2005].

The packet classification problem is how to determine the lowest-cost matching for the incoming packet [Madhi D. *et al*,2007][Varghese G., 2005].

The packet must match at least one rule. There are three matching types [Varghese G., 2005]

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.

## 3. Implementation of Packet Classification Algorithm

Packet Classification Algorithm can be implemented by two major types: Software-based and Hardware-based implementations [Yang B. *et al*,2009].

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 [Yang B. *et al*,2009]:

- **Field-independent algorithms:** These algorithms will build the index tables independently for each field in the rule. Then, the rules are grouped together. HSM [Xu B. *et al*, 2005], and RFC [Gupta P. *et al* a, 1999] 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 [Gupta P. *et al* b,1999], and Hypercuts [Singh S. *et al*,2003] 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 [Sherwood T. *et al*,2003][Jiang W. *et al*,2009].

In spite of the high classification speed achieved by hardware implementation, there are several reasons lead to use software implementation [Yang B. *et al*,2009].

1. Programmability: ASIC architectures have less programming ability because ASIC have special design.

2. Special chips requiring: ASIC require special chips called TCAM (Ternary Content Addressable Memory) to accelerate packet processing speed. TCAM suffer from some problems (density scaling, power scaling, time scaling, extra chips needed, and rule multiplications for range matching) [Baboescu F. *et al*,2003]. Thus, These problems will lead to higher cost and make it difficult to upgrade the algorithm.

#### **4. Packet Classification with NIDS**

NIDS use to protect computer networks. These systems are demand on high throughput and ability to handle new threats [Song H. *et al*,2005].

NIDS classify packets based on its header fields and the strings in the packet payload. Rules in NIDS database usually contain 5-tuple or fields associated with packet header (source IP address, destination IP address, protocol type, source port, and destination port), in addition to some strings in the packet payload called signatures [Song H. *et al*,2005].

If the incoming packet matches the specific rule that contain the same signature, we can classify this packet as a malicious packet.

Snort is a popular open source NIDS which uses signatures to detect the malicious packets [<http://www.snort.org>]. This software or called network sensor uses many efficient and high speed string matching algorithms to match strings in parallel [Wu S. *et al*,1994][Aho A. V. *et al*,1975].However, This software cannot keep up with high speed networks [Song H. *et al*,2005].

To enhance the network speed problem, we can use hardware to perform parallel packet header classification and signature matching [Song H. *et al*,2005].

We suggest two alternative approaches: fast packet classification algorithm performs header classification and signature matching using field dependently manner, and processing each of packet header and signatures in the payload separately. Each of which can be processed on different machine. This can be done by using fast packet classification algorithm on machine for packet header and another machine contains one of NIDS for signature matching.

The next section explains the packet classification algorithms with some of data structure that are used by the algorithms, and the specific algorithm features.

## 5. Taxonomy of Packet Classification Algorithms

After the studding of Packet classification algorithms, we can categorize these algorithms into eight classes as shown in Table 1.

Table 1: show the taxonomy for packet classification algorithms.

No.	Class	Algorithms
1	Naive	Linear search, Caching
2	Two dimensional	Hierarchal trie, Set Pruning Trie, Grid of Tries
3	Extended two dimensional	EGT, EGT-PC, FIS
4	Divide and conquer	BV, ABV, Cross-producing, RFC, HSM,AHSM, C-HSM
5	Decision tree	Hicuts, Hypercuts, D-cuts, Expcuts, Hypersplit, sBits
6	Tuple space and hash Table	TSS, HaRP, Hybrid approach to packet classification, BSOL
7	Heuristic at bit-level	DBS
8	Hardware	TCAM, BV-TCAM

Before we discuss the important features for each algorithm, we will show some features that can be offered by the classes listed in table1.

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  $N$  [Madhi D. *et al*,2007]. The algorithms that depend on Caching techniques are characterized by not working well in practice because of poor hit rate [Baboescu F. *et al*,2005], and they still need a fast classifier as a backup when cache fails [Baboescu F. *et al*,2003].

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 [Madhi D. *et al*,2007].

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 [Varghese G., 2005].

The divide and conquer algorithms are divide the complex problem into simpler sub-problems and then efficiently combining the results into final stage [Varghese G., 2005].

Decision tree algorithms are characterized by difficult to do incremental update [Pong F.*et al*,2009],low efficiency with large number of wildcard, better tradeoff between speed and memory, and they are efficient with edge routers[Madhi D. *et al*,2007].

Tuple space and hash table algorithms are characterized by dividing the search space into regions that can be searched in parallel, using exact matching [Meiners C. R. *et al*,2010], inefficient with large number of rules, tuple space and hash table algorithms are difficult to make updating [Sun X. *et al*,2005], and the linear search on the tuples is more efficient than linear search on rules [Madhi D. *et al*,2007].

Heuristic at bit-level algorithm is adopts heuristic on a bit level to detect the inherent characteristics of the rule set, and The rule sets can be partitioned more efficiently. Thus the storage required for data structures is significantly reduced. This algorithm adopts two levels of flat structures which require only two memory access times while searching. This searching technique will guarantee the speed of classification [Yang B. *et al*,2009]. Hardware classifiers are characterized by very fast, they efficient with wildcards, and changing the search algorithm is expensive [Pong F.*et al*,2009].

Table 2 show the primary features for packet classification algorithms shown in table 1.

1. Hierarchical Trie algorithm is suffers from wasted time because of using backtracking, and it is scalable for 2-Dimension [Varghese G., 2005].
2. Set Pruning Trie algorithm is suffers from prefix replication, and it is scalable for 2-D [Varghese G., 2005].
3. Grid of Tries algorithm helps in avoiding the wasted time in backtracking using pre-computation to the path, it is scalable for 2-D [Varghese G., 2005], it may suffer from missing best matching rule, and it avoids reach to end path with no result using switch pointer [Madhi D. *et al*,2007].
4. Extended Grid of Tries (EGT) 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 [Baboescu F. *et al*,2003].
5. Extended Grid of Tries-Path Compression (EGT-PC) algorithm 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 [Baboescu F. *et al*,2003].
6. Bit Vector (BV) algorithm is characterized by slow dynamic update, bad memory using [Madhi D. *et al*,2007], it does not scale well for large data base and very high speed system [Varghese G., 2005], and it provides Parallel lookup header fields [Song H. *et al*,2005].
7. Aggregate Bit Vector (ABV) algorithm is suffers from false positive [Baboescu F. *et al*,2003], 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 [Madhi D. *et al*,2007],it can provide suitable throughput [Wang P. C. *et al*,2007], and parallel lookup header fields [Song H. *et al*,2005].
8. Cross-Producting algorithm is scalable for data base smaller than 50 rules [Baboescu F. *et al*,2005], it Requires caching for larger classifiers [Gupta P. *et al* b,1999], and it suffers from redundancy [Xu B. *et al*, 2005].
9. Recursive Flow Classification (RFC) algorithm is an improving form of cross-producting [Baboescu F. *et al*,2003], it is characterized by high storage requirement [Baboescu F. *et al*,2005], and Unpredictable preprocessing time [Madhi D. *et al*,2007], it does not support incremental updates [Baboescu F. *et al*,2005], and it performs stability at search time [Xu B. *et al*, 2005].
10. Ternary Contentable Address Memory (TCAM) algorithm is characterized by offering good solution in HW for small classifiers, consuming too much power and board area [Baboescu F. *et al*,2003], it has efficient in wildcards matching, not practical for PC-

based routers [Baboescu F. *et al*,2005], and it may suffer from rules blowup [Madhi D. *et al*,2007].

11. Hierarchical Intelligent Cutting (HiCuts) algorithm is characterized by defaulting support incremental updates, requiring more memory access with more depth, it is efficient with edge routers [Madhi D. *et al*,2007], sometime it has no explicit worst case time, using local optimized scheme to avoid unnecessary memory storage [Qi Y. *et al*,2009], and it performs well with no overlapped rules [Xu B. *et al*, 2005].

12. Multidimensional Hierarchical Intelligent Cutting (HyperCuts) algorithm 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 [Madhi D. *et al*,2007], it performs well under practical conditions [Wang P. C. *et al*,2007], and it is difficult to support incremental updates [Madhi D. *et al*,2007].

13. Explicit Cutting (ExpCuts) algorithm does not suffer from excessive memory access and worst case search time, and it works with multi-core Network Processors [Qi Y. *et al*,2007].

14. 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 [Qi Y. *et al*,2009].

15. Dynamic Cuts(D-Cuts) algorithm is characterized by achieving higher speed than Hicuts because it adopts a network statistics into decision tree, suffering from long term tree searching [Xu B. *et al*,2007], adopting structural characteristics and network statistics, and focusing on reducing the depth of D.T [Qi Y. *et al*,2004].

16. Hierarchical Space Mapping (HSM) algorithm is characterized by using balanced binary search tree [Xu B. *et al*,2007], high preprocessing time, and using rule based space decomposition on each field to achieve deterministic worst case search time [Qi Y. *et al*,2009].

17. Adaptive Hierarchical Space Mapping (AHSM) algorithm is characterized by using alphabetic search tree with recursive intersecting table, and adopting network statistics [Xu B. *et al*,2007].

18. Improved Hierarchical Space Mapping (C-HSM) algorithm is characterized by using pruning trie, and using heuristic to compress the space and save the memory [Cao C. *et al*,2006].

19. Discrete Bit Selection (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 [Yang B. *et al*,2009].

20. Shifted Bits (sBits) algorithm 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 [Qi Y. *et al*,2006].

21. Binary Search On Level (BSOL /  $O(\log W)$ ) algorithm is characterized by depending on Hash table and binary tree, multidimensional scheme, and it has better memory and time performance than EGT-PC [Lu H. *et al*,2007].

22. Fat Inverted Segment Tree (FIS-Tree) algorithm is characterized by efficient update, it scales well for 2-D [Gupta P. *et al*,2001], and it may adopt clustering to reduce memory storage when the number of dimensions is larger than 2 [Baboescu F. *et al*,2005].

23. Independent *set* algorithm is not affected by number of wildcards, it is not affected by the size of rule table, and it has very fast updating time [Sun X. *et al*,2005].
24. Tuple Search Space algorithm is not efficient with incremental updates [Pong F. *et al*,2009], it depends on the number of tuples in the classifiers [Sun X. *et al*,2005], and it has fast search time [Baboescu F. *et al*,2005].
25. HaRP 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 [Pong F. *et al*,2009].
26. Hybrid (Tuple+Top-Down Tree) algorithm is characterized by combining hash table with binary trie, and it is applicable with NP [ZHAO X. *et al*,2004].
27. BV-TCAM algorithm is scalable for Intrusion detection system, and it uses FPGA architecture [Song H. *et al*,2005].
28. Linear Search algorithm is scalable for small classifiers, and it has efficient update time [Qi Y. *et al*,2004].

Table 2 shows the important features for the most popular and modern packet classification algorithms.

No.	Algorithm name	Type of Match	Field dependency	Time Complexity	Space Complexity	HW / SW
1	Hierarchical Tries	Prefix	Dependent	$O(N^F)$	$O(N)$	SW
2	Set Pruning Tries	Prefix	Dependent	$O(FW)$	$O(N^F)$	SW
3	Grid of Tries	Prefix	Dependent	$O(W^{F-1})$	$O(N)$	SW
4	EGT	Prefix	Dependent	$O(H+2)T$		SW
5	EGT-PC	Prefix	Dependent	$O(W^2+BP/C)$	$O(N)$	SW
6	BV	Range	Independent	$O(FW+N/M)$	$FN^2$	HW/SW
7	ABV	Prefix	Independent	$O(FW+N/M)$	$FN^2$	HW/SW
8	Cross-Producing	Range	Independent	$O(FW)$	$O(N^F)$	SW
9	RFC	Range	Independent	$O(F)$	$O(N^F)$	SW
10	TCAM	Prefix	independent	$O(1)$	$O(N)$	HW
11	HiCuts	Range	Dependent	$O(F)$	$O(N^F)$	SW
12	HyperCuts	Range	Dependent			SW
13	ExpCuts	Range	Dependent			SW
14	HyperSplit	Range	Dependent	$O(F*\log(2N+1))$	$O(N^F)$	SW
15	D-Cuts	Range	Dependent			SW
16	HSM	Range	Independent	$O(F(\log(2N+1)))$	$O(N^F)$	SW
17	AHSM	Range	Independent	$O(2N+1*\log(2N+1))$	$O(N^F)$	SW
18	C-HSM	Prefix / Range	Independent	$O(FW+\log(F-1))$	$O(N^F)$	SW
19	DBS	Prefix	Dependent / Independent	$O(F)$		HW/SW / Combined
20	sBits	Range	Dependent			SW
21	BSOL / $O(\log W)$	Prefix / Range	Independent	$O(\log(W)+S/C)$	$O(N^F)$	SW
22	FIS-Tree	Range	Dependent	$O(L+1)W$	$O(LN^{1+1/L})$	SW
23	Independent set algorithm	Range	Independent	$O(I)$	$O(N)$	HW/SW
24	Tuple Search Space	prefix	Independent	$O(N)$	$O(N)$	SW
25	HaRP	prefix	Independent			SW
26	Hybrid (Tuple+Top-Down Tree)	Prefix / Range	Independent	$O(\log(N/(2^{\sum^k V})) + V)$	$O(N(1+2/V) - 2^{\sum^k})$	SW
27	BV-TCAM	All types	Independent			HW
28	Linear Search	Exact	Dependent	$O(N)$	$O(N)$	SW



Where

N	Number of Rules
F	Number of dimensions (Fields) 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

$$2^{\sum^h k} \quad \text{Number of bits selected to create index table}$$

## Conclusion

In this paper we present taxonomy for packet classification algorithms with a survey for the modern and most popular algorithms. Also, this paper includes the basic principles that are used with packet classification algorithms, for example field dependency, matching type and type of implementation. In addition, we suggest how to enhance the network speed for the NIDS.

This paper will help the designer of packet classification algorithm hoe to specify what is the algorithm needed according to the networks service requirement.

## References

- Aho A. V., Corasick M. J., 1975, "Efficient String Matching: An Aid to Bibliographic Search". In Proceedings of Communications of the ACM, Vol 18, No 6, pp. 333-340.
- Baboescu F., and Varghese G., 2005, "Scalable packet classification", In Proceeding of IEEE/ACM Transactions on Networking (TON), Vol. 13, No 1, pp. 1-13.
- Baboescu F., Singh S., and Varghese G., 2003, "Packet Classification for Core Routers: Is there an alternative to CAMs? ", In Proceeding of INFOCOM Twenty-Second Annual Joint Conference of the IEEE Computer and Communications, Vol. 1 , pp. 53 – 63
- Cao C., Yuan J., Wang H., and Dong Y., 2006, "An Improved HSM Packet Classification Algorithm Using Pruning Trie and Space Compression ", In Proceeding of International Conference on Wireless Mobile and Multimedia Networks proceeding (ICWMMN), pp. 1-4.
- Gupta P., and McKeown N. a, 1999, "Packet classification on multiple fields", In Proceeding of the conference on Applications, technologies, architectures, and protocols for computer communication SIGCOMM '99, pp. 147-160.
- Gupta P., and McKeown N. b, 1999, "Packet classification using hierarchical intelligent cuttings," In Proceeding of Hot Interconnects VII, pp. 34-41.
- Gupta P., and McKeown N., 2001, "Algorithms for Packet Classification ", IEEE Network Magazine, Vol 15, No 2, pp.24-32.
- 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.
- Lu H., and Sahni S., 2007, "O(log W ) Multidimensional Packet Classification ", IEEE/ACM Transactions on Networking, Vol 15, No 2, pp. 462-472.
- Madhi D., and ramasamyK., 2007, "Network routing algorithms, protocols, and architectures", Morgan Kaufmann, USA, pp. 1-957.
- Meiners C. R., Liu A. X., and Tornig E., 2010, "Hardware Based Packet Classification for High Speed Internet Routers", Springer, New York-USA, pp. 1-122.
- Pong F., and Tzeng N. F., 2009, "Hashing Round-down Prefixes for Rapid Packet Classification", In Proceeding of the conference on USENIX Annual technical conference, pp. 1-15.
- Qi Y., and Li J., 2004, "Packet Classification with Network Traffic Statistics ", In Proceeding of the 3rd Trusted Internet Workshop (TIW), pp. 1-9.

- Qi Y., and Li J., 2006, "An Efficient Hybrid Algorithm for Multidimensional Packet Classification", In Proceeding of the 3rd IASTED Conference on Communication, Network and Information Security (CNIS), pp. 185-190.
- Qi Y., Xu B., He F., Zhou X., Yu J., and Li J. , 2007," Towards Optimized Packet Classification Algorithms for Multi-Core Network Processors",in Proceeding of the International Conference on Parallel Processing (ICPP).
- Qi Y., Xu L., Yang B., Xue Y., and Li ., 2009 "Packet Classification Algorithms: From Theory to Practice", In Proceeding of INFOCOM 28th IEEE International Conference on Computer Communications, pp. 648-656.
- 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.
- Singh S., Baboescu F., Varghese G., and Wang J., 2003, "Packet Classification Using Multidimensional Cutting ",In proceeding of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications SIGCOMM'03, pp. 213-224.
- Song H., and Lockwood J. W., 2005, "Efficient Packet Classification for Network Intrusion Detection using FPGA", In Proceeding of the ACM/SIGDA 13th international symposium on Field-programmable gate arrays, pp. 238-245.
- 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.
- Varghese G., 2005, "Network Algorithmics An Interdisciplinary Approach to Designing Fast Networked Devices ", Morgan Kaufmann, USA. pp. 1-491.
- Wang P. C., and Chang C. M., 2007, "Scalable Packet Classification for Network Intrusion Detection", In Proceeding of the Fifth IASTED International Conference on Circuits, Signals and Systems, pp.64-69.
- Wu S., and Manber U., 1994, "A FAST ALGORITHM FOR MULTI-PATTERN SEARCHING", Tech. Rep. TR94-17,Department of Computer Science, pp. 1-11.  
WWW.SNORT.ORG
- Xu B, Zhou G., Xue Y., and Li J., 2007, "AHSM: ADAPTIVE PACKET FILTERING WITH NETWORK TRAFFIC STATISTICS ",In Proceeding of the 20th IASTED International Conference Parallel and Distributed Computing and System (PDCS 2008), pp. 258-263.

Xu B., and Li J., 2005, "HSM: A Fast Packet Classification Algorithm", In Proceeding of the International Conference on Advanced Information Networking and Applications (AINA), Vol 1, pp. 1-8.

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.

ZHAO X., JI Y., and LEI Y., 2004, "A hybrid approach to packet classification," NII Journal, Vol 8, pp. 13-21.