A Novel Signature-based Packet Classification

Yie-Tarng Chen and Ya-Hsin Yang

Department of Electronic Engineering, National Taiwan University of Science and Technology

***Abstract*- This paper describes a novel signature-based packet classification that can achieve gigabit speed at limited memory consumption. The innovative aspect of signature-based scheme is to extract the rule into an equivalent signature, a unique vriable-length bit string with shorter width. Therefore, only a small fraction of a rule is inspected in search, resulting in considerable saving in lookup time as well as providing an effective solution for high dimensional rule. Moreover, the signature-based packet classification can perform well at high dimensions. By running a simulation model that incorporates the publicly available packet traces, we show that the performance of the signature-based scheme can reach 11 million packets per second even in the worst case, when implemented by 3.96-M 10-ns SRAM for 10000 rule four-dimensional classifier. This result demonstrates signature-based scheme is superior to previous approaches.**

I.INTODUCTION

Multi-field packet classification has attracted much attention recently. This is probably due to the development of new network services, such as firewall, differentiated service, policy-based network and usage-based billing, which makes the multi-field packet classification as an important network component.

The rule of multi-field packet classification consists of arbitrary prefix or range specifications on the following fields: the source network address, the destination network address, the source port, the destination port, the protocol identifier, and possible other fields. When a packet arrives, the packet header is extracted and then compared to the rule's corresponding fields. If they match, this rule is considered as a candidate rule. Logically, the packet header is compared against every rule in this way. Finally, the candidate rule with the highest priority matches, and the packet operates according to the action specified in this rule. If none of the rule match the packet's fields, the packet results in a miss.

Packet classification on multiple fields is recognized a hard problem. Previous researches show that multi-field packet classification has poor worst-case performance, in either storage or time complexity. Moreover, scaling is a particular problem for packet classification. Most proposed algorithms neither work well for more than two dimensions nor scale to rules larger than a few thousand. Furthermore, the explosion of high-speed network requires an efficient packet classification at OC 48c or OC 192c rate. Consequently, packet classification on multiple fields poses a challenging issue, and this is exactly the problem addressed in this paper.

To motivate our work, we first observe the classifiers in Table 1. In rule F_2, prefix {140.0} in source network address is a unique bit string in the classifier. If a packet's source network address matches the prefix {140.0}, the remaining fields of the packet can immediately compare with rule F_2 without further inspection. If they match, then the packet is forwarding based on rule F_2. Therefore, each rule can be encoded as a unique bit string, called the signature. For example, {140.0} is the signature of Rule F_2, and {140.118, 112.110} is the signature of Rule F_6.

TABLE 1 Four-field Cassifiers.

Rule	Source Address	Destination Address	Source Port	Destination Port
F1	140.118.*	192.2	80	21
F2	140.0.*	140.112	80	*
F3	140.110.*	112.110.*	*	21-24
F4	140.118.*	192.2.*	21	80
F5	88.120.2.1	65.2.2.2	20-30	80
F6	140.118.*	112.110.*	21	21

In this paper, we proposed a novel signature-based packet classification. The proposed approach can be thought as an extension of the multi-bit trie to multi-field packet classification. To reduce memory requirements and long lookup time, we proposed the signature, unique bit string extracted from each rule, instead of the whole rule information, to prune unnecessary branch in the multi-bit trie. In comparisons with other packet classification schemes, the signature-based scheme has the following features.

(1) Fast lookup time

Only a small fraction of multi-field information is inspected for signature-based packet classification, resulting in considerable saving in lookup time, especially a fairly good average lookup time.

(2) Scale to high dimensions

Based on our observation, the distribution of the signature width for high dimensional rule concentrates below two fields. Therefore, the signature-based packet classification provides a good solution for high-dimensional packet classification.

0-7803-7206-9/01/$17.00 © 2001 IEEE

(3) Generic

General rule sets can achieve search-path compression by identifying rule signature and this approach is not limited to any particular rule set.

The rest of the paper is organized as follows. Section 2 introduces previous work. Section 3 presents data structure and initial construction. Section 4 presents search algorithm. Section 5 desrcibes signature transposition algorithm. Section 6 presents performance analysis, and the conclusions are presented in Section 7.

II.PREVIOUS WORK

Extensive collections of papers have addressed the packet classification problem [2-10]. Borg et al [3] presented a data structure based on a combination of trie with path compression and binomial trees. Srinivan et al. [2] presented grid of trie and cross-product approaches. However, grid of trie restricts to process filters with two fields and cross-product approach provides fast lookup but potentially requires large storage. An algorithm, called Recursive Flow Classification (RFC), [4] has been proposed by Gupta. For packet classifiers in the backbone, the number of distinct value in each field could be large, and memory requirement for RFC could blow up in the early phase. Special case of RFC is identical to cross-prodct. Gupta et al [6] also presented another heuristic, called hierarchical intelligent cutting, to exploit structure present in classifiers. Tuple space search [7] has been proposed for packet classification. The scheme partitions filters into distinct field length combinations, called tuple and searches through each tuple by hashing. The disadvantage of tuple space search is non-deterministic search time. Woo [8] developed an approach based on divide-and-conquer. A heuristic tree search separates a large set of filters into fixed size filter bucket and another linear search procedure through fixed size filter buckets. Some researchers have cast packet classification as the range search problem in computational geometry. Bit-level parallelism [5], a hardware-oriented scheme, has been proposed by Lakshman. The storage requirements scale quadratically with the size of the classifier. Buddhikot et al [10] developed a novel data structure, Area-based Quadtree (AQT) and used a recursive space decomposition to support incremental updates. This approach cannot extend to more than two-dimensional rules. Feldmann et al [9] propose the FIS-tree data structure for two-dimensional classification as a modification of the segment tree data structure. Overall, most existing studies work well for two dimensions, but do not extend to multiple dimensions. No previous studies have applied the concept of signature in packet classification. The proposed approach is excellent in average search time and can provide deterministic worst-case search time.

III.DATA STRUCURE AND INITIAL CONSTRUCTION

A. Sequential Signature Extraction

The key idea behind the proposed scheme is to extract the rule into an equivalent signature, a unique bit string with shorter length. Therefore,only a small fraction of a rule is inspected in search. The width of each signature is dependent on characteristic of classifiers. To create the signature for a rule, the straightforward approach is sequential signature extraction. First, extract the most significant r bits from the rule as a signature candidate, C. If C is a unique in packet classifiers, then C is the rule signature. Otherwise, Repeat to append next s bits of the rule to C until C is a unique bit sting. Consequently, the C is the rule signature. Different rules are not necessarily uniform in the signature width. For sequential signature extraction, r and s are design parameters to decide the width of rule signature. If the width of a signature is arbitrary, i.e. r=1 and s=1, packet classification scheme becomes complicated. To incorporate with the proposed data structure, 2^m-ary search tree, the signature width is selected as a multiple of 8, i.e. 16bits, 24bits, 32bits, 40bits etc. We choice r = 16 and s = 8.

B. Data Structure

The 2^m-ary search tree is the major data structure for storing rule signature. Each rule signature is represnted by a leaf in the tree strcutre, and the valaue of the signature corresponds to the path from the root of the search tree to the leaf. Each node in the 2^m-ary search tree is an index table. Each entry in the index table contains an *entry type* (2 bit), and a *ptr* (16 bit).

There are four entry types:

(1) Invalid Entry (00): The entry has no valid value. The default value of each entry is an invalid entry.

(2) Signature entry (01): The entry itself is a signature, and no entry in the next level index table is required for this rule. *Ptr* contains an index to a tag, which stores the remaining rule information.

(3) Internal entry (10): The entry itself is not a signature and entries in the next–level index table(s) are required to store the signature(s). *Ptr* contains the location of the next level index table.

(4) Wildcard entry (11)**:** The remaining bytes of a rule are wildcard and *Ptr* contains rule ID.

Both signature entry and wildcard entry are leaf entries.

In the last level of the index table, only two entry types.

(1) Terminal entry (11): The entry contains an entire rule, instead of a signature. *Ptr* contains a rule ID.

(2) Invalid entry (00)**:** The entry has no valid value.

Tag, an array structure, contains the remaining rule information and a 16-bit rule ID.

C. Initial Construction and Rule Insertion

A rule signature is not time-invariant. The original rule signature may not be unique when a new rule is inserted. For example, the original signature of Rule F_2 is {140.0} in Table1. After a new rule [140.0, 140.114, *, *] is added in

Table 1, {140.0} is not a unique bit string. Consequently, {140.0, 140.112} could be a new signature of Rule F_2. Therefore, signature modification is a necessary process in the rule insertion.

The 2^m-ary search tree can be constructed as follows. When a new rule is added into classifiers, the new rule is served as a key to search the 2^m-ary search tree. New rule signature is decided by traversing the 2^m-ary search tree. The search starts at root and then traverses down the 2^m-ary search tree by inspecting m bits of the new rule in each node until a non-interior entry reaches. If it is an invalid entry, the new rule is stored in this entry. The entry type is modified as signature entry and the remaining rule information is stored in the corresponding Tag. On the other hand, if it is a signature entry, the signature of an existing becomes invalid. The entry type is modified as interior entry, and a new node, a 2^m –entry index table, in the search tree, is created. For the new rule and the rule with invalid signature, the next m bits index into the new index table. If both rules still map to the same entry, another new node, a 2^m -entry index table, is created. This step is repeated until each rule is mapped to a unique entry.

IV. SIGNATURE-BASED PACKET CLASSIFICATION

The signature-based packet classification works by carefully preprocessing the classifiers to build the 2^m-ary search tree. Packet classification is achieved by traversing 2^m-ary search tree. A search begins at root, level-1 index table, and then descends down the 2^m-ary search tree by inspecting m bits in each node until it reaches a leaf entry. Once a leaf entry is found, the remaining packet fields are compared with the remaining rule information stored in the corresponding tag. If there is a match, it yields the desired rule matching. In worst case, the lookup can be completed at the depth of the 2^m-ary search tree. Suppose that level-1 index table has 64k entries and remaining level has 256 entries, the depth of the 2^m-ary search tree equals to $W-1$, where W is the width of the rules in bytes. Hence, the worst–case time complexity is O(W). The worst-case time complexity of signature-based packet classifiers is independent of structures of the classifiers and the number of rules. Therefore, the proposed scheme can extend to large number of rules

V. BIT SELECTION ALGORITHM

The signature-based packet classification scheme can be further improved by minimizing the average width of the rule signature. Therefore the average lookup time can be reduced. The sequential signature extraction is simple in implementation, but it cannot minimize the average lookup time. The bit selection algorithm is proposed for this purpose. First, concatenate all fields of packet header as a bit vector, and divide the bit-vector into k slices of s bits each. Then, find a permutation for the k slices such that the average width of the rule signature is minimized.

The cost function of the average signature width can be expressed as

$$\sum_{i=1}^{k} i * N_i$$

where N_i is number of rule signature with i slices.

We use greedy approach to find the permutation with minimal signature width. First, select the slice with maximum number of signatures. Next, choose the next slice with maximum number of signatures in concatenation with the permutation constructed so far, and repeat it until the permutation contains all slices. After slice permutation, sequential signature extraction operates on the permutated bit vector. Figure 1 expresses the bit selection algorithm.

```
S = ϕ;
for (i=1; i ≤ k ; i ++) {
        maxnumber = 0;
        candidate = 0;
        for (j=1; j ≤k; j++) {
            currentnum = Snum(S, j);
        if (j ∉ S) {
            if (currentnum > maxnumber)
            {maxnumber = currentnum;
             candidate = j ;}
            }}
            S = S ∪ {candidate };
            }
```

Figure 1 bit selection algorithm

Assume k is the number of slices. S represents the set of slices chosen so far. Snum(S, j) is a function that calculates the number of signature including the j-th slice, given that the slices in S have already been chosen in the slice set.

VI. PERFORMANCE EVALUATION

To test the performance of signature-based packet classification, we built a simulator based on PALAC [14]. We assumes that 4 bytes of SRAM lookup cost 10ns. The major performance metrics in our experiments are the average lookup time, the experimental maximal lookup time and memory requirements. The structure of classifers can affect both memory requirements and lookup times. However, there are no large-scale real classifiers available in public domain. Therefore, three kinds of classifiers are considered in our experiments: real routing tables and two kinds of synthesized classifiers.

A. Performance Evaluation of Routing Table Lookup

Routing table lookup can be thought as a special packet classification. Moreover, based on signature-based scheme, characteristics of classifiers with distinct prefix in the source network address (or the destination network address), are similar to those of routing table lookup. The publicly available routing tables from IPMA[11] are used to

investigate the performance of signature-based schemes,.

(1) Memory Requirements:

Table 2 illustrates the memory requirements for routing table lookup under different number of routing entry. The memory requirement for signature-based scheme is about 1.49 Mbytes under 16000 routing entries. The signature-based scheme can achieve smaller storage requirements in comparisons with routing table lookup implemented in similar data structures [1].

(2) Lookup Time:

Table 3 shows the distribution of signature length under different number of entries. More than 99 percentage of rule signatures is less than two level. With 16000 routing entries, the average memory reference for signature-based scheme can achieve 1.84 references.

Table 2 memory requirements for routing table lookup (in byte).

# of routing entry	Memory requirements
500	196.6K
1000	243.6K
4000	647.3K
8000	956.2K
16000	1.49M

Table 3 Distribution of Signature Length

# of routing entry	Level1	Level2	Level3
500	98.40%	1.60%	0.00%
1000	86.30%	13.70%	0.00%
4000	46.25%	53.75%	0.00%
8000	27.99%	72.00%	0.01%
16000	16.32%	83.64%	0.04%

B. Performance Evaluation for Multi-Field Packet Classification

We use two approaches to create synthesized classifiers.

(1) Two-dimensional synthesized classifiers: Each rule is generated by randomly picking a prefix from publicly available routing tables in IPMA [11] as the source network address and the destination network address respectively. Wildcard are added at random to each dimension. Each index table is implemented as 256 entries.

(2) Four-dimensional classifiers: Each rule is constructed by randomly picking flow from packet traces in NLANR [13]. Wildcard are added at random to each dimension. Each index table is implemented as 256 entries.

Memory Requirements

Table 4 illustrates the memory requirements for two-dimensional classifiers and four-dimensional classifiers. The results demonstrate that the signature-based scheme can be used in high dimensions (number of fields) and large number of rules without memory explosion. Specially, the memory requirements of four-dimensional classifiers are very close to that of two-dimensional classifiers under the same number of rules.

Lookup Time

Table 5 shows the lookup time under different number of rules for two-dimensional classifiers and four-dimensional classifiers respectively. The average lookup time and experimental maximum lookup time increase very slowly as the number of rules increases. 20000 four-dimensional classifiers can achieve the average case lookup time at 71 ns and the maximum lookup time at 95 ns. This corresponds to at least 11 million lookups per second; enough to process the packet on a 3.52 Gb/s line (assuming a minimum length of IP data gram of 40 bytes). Specifically, the lookup times for two-dimensional classifiers and four-dimensional classifiers are very close under the same number of rules. Table 6 and Table 7 show the distribution of signature length for two-dimensional classifiers and four-dimensional classifiers respectively. Most of packets can be classified within 3 level. Theoretically, two-dimensional classifiers can achieve eight memory references at the worst case, and four-dimensional classifiers can achieve twelve memory references at the worst case. However, the signature-based scheme takes advantage of the structures of classifiers, the experimental maximum lookup time is smaller than the theoretical worst-case lookup time. Moreover, the theoretical worst-case lookup time never occurs in our experiments.

VII. CONCLUSIONS

In this paper, we proposed a novel signature-based packet classification on multiple fields, which can achieve gigabit speed at limited memory consumption. It is motivated by observation that each rule can be encoded as a unique bit string with shorter lengths, called the signature. Therefore, the multi-field packet classification problem is equivalent to a specific signature search. Only a small fraction of multi-field information is inspected for signature-based packet classification, resulting in considerable saving in lookup time. The signature-based scheme is a generic scheme. It can combine with existing routing table lookups and packet classification algorithms to achieve performance improvement. Most previous researches work well for two dimensions, but do not extend to multiple dimensions. Nevertheless, the signature-based packet classification can perform well at high dimensions. Simultaneously, it provides low bound in the worst-case lookup time.

References

[1] P. Gupta, S. Lin, and N. McKeown, "Routing lookups in hardware at memory access speeds," in Proceedings of the Conference on Computer Communications (IEEE INFOCOMM), (San Francisco, California), vol. 3, pp.1241-1248, March/April 1998.

[2] V. Srinivasan, G. Varghese, S. Suri, and M. Waldvogel, "Fast and Scalable Layer Four Switching," in Proc. ACM SIGCOMM 1998, pp.203-214.

[3] N. Borg, E Svanberg and O. Schelen, "Efficient Multi-field packet Classification for QoS proposes," in Proc. ACM SIGCOMM Sep.1998, pp.191-202.

[4] P. Gupta, and N. Mckeown, "Packet classification on multiple fields," in Proc. ACM SIGCOMM 1999, pp.147-160.

[5] T.V. Lakshman, and D. Stiliadis, "High-speed policy-based packet forwarding using efficient multi-dimensional range matching," in Proc. ACM SIGCOMM 1998, pp. 191-202.

[6] P. Gupta, N. McKeown, "Packet Classification Using Hierarchical Intelligent Cuttings" Hot Interconnections VII, 1999

[7] V. Srinivasan, G. Varghese, and S. Suri,. "Fast Packet Classification Using Tuple Space Search", ACM SIGCOMM 1998pp 203-214,

[8] T. Woo, "A Modular Approach to Packet Classification: Algorithms and Results" INFOCOM 2000

[9] A. Feldmann and S. Muthuskishnan, "Tradoffs for Packet Classifcation", INFOCOM 2000

[10] M.M. Buddhikot, S. Suri and M. Waldvogel, "Space Decomposition Techniques for Fast Layer-4 Switching", Protocols for High Speed Network, Vol. 66, No. 6, pp.277-283, 1999

[11] Michigan University and Merit Network, Internet Performance Measurement and Analysis (IPMA) Project, http://nic.merit.edu/~ipma/.

[12] NLANR Network Analysis Infrastructure http://moat.nlanr.net

[13] Blake et. al., "A Architecture for Differentiated Services," RFC 2475, Dec. 1998

[14] http://Klamath.Stanford.edu/tools/PALAC/SRC

Table 4 memory requirements for multi-field packet classifiers (in KB)

#of rules	100	500	1000	2000	4000	8000	10000	15000	20000
2-dimensions	36.20	139.9	312.6	656.3	1380.	3101.	3960.	6346.	9065.
4-dimensions	22.66	129.9	317.0	645.5	1392.	3091.	4037.	6522.	9231.

Table 5 lookup time for multi-field packet classifiers.(in ns)

# of rules	100	500	1000	2000	4000	8000	10000	15000	20000
2-d maximal lookup time	55.0	65.0	75.0	75.0	75.0	85.0	85.0	85.0	85.0
2-d average lookup time	46.6	48.4	50.4	52.5	54.6	57.2	58.0	59.6	61.0
4-d maximal lookup time	65.0	85.0	85.0	85.0	85.0	105.0	95.0	95.0	95.0
4-d average lookup time	54.7	58.1	60.6	62.4	64.7	67.2	68.0	69.7	71.0

Table 6 distribution of signature length for two-dimensional classifiers.

#of rules	100	500	1000	2000	4000	8000	10000	15000	20000
DA level 1	9.00%	3.20%	1.10%	0.45%	0.23%	0.03%	0.05%	0.03%	0.05%
DA level 2	66.00%	60.20%	46.30%	31.60%	20.23%	10.89%	9.35%	6.38%	4.28%
DA level 3	21.00%	33.00%	46.60%	55.95%	59.23%	51.19%	46.43%	36.32%	27.92%
DA level 4	0.00%	0.00%	0.00%	0.10%	0.03%	0.08%	0.06%	0.03%	0.04%
SA level 1	4.00%	3.20%	5.20%	11.60%	18.55%	34.24%	40.45%	52.18%	60.26%
SA level 2	0.00%	0.40%	0.80%	0.40%	1.75%	3.51%	3.60%	4.99%	7.37%
SA level 3	0.00%	0.00%	0.00%	0.00%	0.00%	0.08%	0.06%	0.08%	0.10%
SA level 4	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%

Table 7 distribution of signature length for four-dimensional classifiers.

#of rules	100	500	1000	2000	4000	8000	10000	15000	20000
SA Level 1	11.00%	3.20%	1.00%	0.45%	0.23%	0.05%	0.06%	0.03%	0.03%
SA Level 2	81.00%	63.20%	45.60%	31.55%	19.45%	11.14%	9.25%	6.09%	4.19%
SA Level 3	8.00%	32.00%	45.40%	57.80%	59.43%	51.25%	46.03%	36.89%	27.51%
SA Level 4	0.00%	0.00%	0.00%	0.05%	0.03%	0.03%	0.08%	0.04%	0.03%
DA Level 1	0.00%	1.20%	7.60%	9.55%	18.98%	34.14%	40.98%	51.42%	60.90%
DA Level 2	0.00%	0.40%	0.40%	0.60%	1.90%	3.35%	3.52%	5.42%	7.25%
DA Level 3	0.00%	0.00%	0.00%	0.00%	0.00%	0.05%	0.08%	0.11%	0.10%
DA Level 4	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
SP Level 1	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
SP Level 2	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
DP Level 1	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
DP Level 2	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%

Where DA: destination address SA: source address SP: source port DP: destination port