Discrete Bit Selection: Towards a Bit-level Heuristic Framework for Multi-dimensional Packet Classification

Baohua Yang[*,†] ,Yaxuan Qi[*,†], Fei He[*,†] ,Yibo Xue[†,‡] and Jun Li[†,‡]
[*]Dept. Automation, Tsinghua University
[†]Research Institute of Information Technology, Tsinghua University
[‡]Tsinghua National Lab for Information Science and Technology, Beijing, China
ybh07@mails.tsinghua.edu.cn, yaxuan@tsinghua.edu.cn, hefei06@mails.tsinghua.edu.cn, {yiboxue, junl}@tsinghua.edu.cn

***Abstract*—Packet classification is still a challenging problem in practice under large number of classification rules and constant growth of performance requirement. Most of the existing algorithms try to solve the problem heuristically by leveraging on the inherent field-level characteristics of the rules. This paper proposes a bit-level heuristic framework: Discrete Bit Selection (DBS) for multi-dimensional packet classification. Preliminary experimental results show that DBS-based algorithm gains much better performance both in search time and memory requirement than the well-known field-level algorithms with various real-life rule sets.**

I. INTRODUCTION

Algorithm for multi-dimensional packet classification is still critical due to its wide usage in high performance firewall, IDS and IPS, etc. Although it is theoretically hard to design a single algorithm that fits well for all cases [1], many effective algorithms were designed to improve performance in various practical cases, leveraging on the inherent characteristics of the real-life rule sets [2]. Those algorithms can be categorized as field-independent ones, such as RFC [3] and HSM [4], and field-dependent ones [5], including HiCuts [6] and HyperCuts [7]. For example, the IPv4 packet header contains 104 bits in 5 tuples, including source and destination IP addresses, source and destination ports, and protocol type. Field-independent algorithms build their search structure with two steps: first use the 5 tuples data to build separate structures independently, then group them together; while the field-dependent algorithms treat the 5 tuple data dependently, thus usually grouping is not necessary at the final stage. Since the field-dependent algorithms adopt more inherent relation between different tuples during the building, they typically achieve better tradeoff between time and space in practical cases.

In this paper, we introduce a light-weight, scalable, and effective framework for packet classification: Discrete Bit Selection (DBS), which aims for both high search speed and low memory requirement.

The framework is designed around two principles: (i) Use bit-level heuristics to split rules, and (ii) Adopt simple data structures to keep memory requirement low. To achieve highly effective classification, we design a structure *bit-string* to partition the rule set, which is built with bit-level heuristic scheme. To utilize memory bandwidth better, we take memory block structures to store rules continuously. The framework mainly consists of two phases: preparation and classification. The preparation phase contains two steps: bit-string generation and lookup table construction. The classification phase is straightforward, searching against the bit-string and lookup table. Preliminary experimental results show the framework is effective for classification on real-life rule sets.

The paper is organized as follows. The methodology is illustrated in section II; section III shows the experimental results while section IV gives the conclusion and future works.

II. METHODOLOGY

The two phases of the proposed method are described in more details in this section.

A. *Bit-string Generation*

Packet classification is to find out the best matched rule in a rule set **R** with the 104-bit (for IPv4 5-tuple match) packet header. However, not all the 104 bits information is necessary with a given rule set. Suppose the number of rules in **R** is N, the size of the selected bit-string b is l. With different value of b, **R** can divide into m subsets (m=2^l). We define a wrap value w for selected bit-string b as: $w = Max(N_i)/N$ ($0 \le i < m$), Where N_i is the number of rules in subset$_{[i]}$. Lower w means to partition the rule set into smaller subsets.

Our motivation is to select the most effective bit-string which splits **R** into subsets as small as possible. Since there are up to C_{104}^{l} different results, exhaustive selection is unpractical. Therefore, heuristic bit-string selection schemes are needed. Here are two intuitive schemes:

1) Incremental Selection Scheme: This method starts with l = 1. First we select the most effective bit and partition the current rule set into two subsets. Then set l = 2, and repeat the first step with the two sub-sets. Continue these first two steps until b is selected.

This work is supported by National High-Tech R&D 863 Program of China. *(No. 2007AA01Z468)*

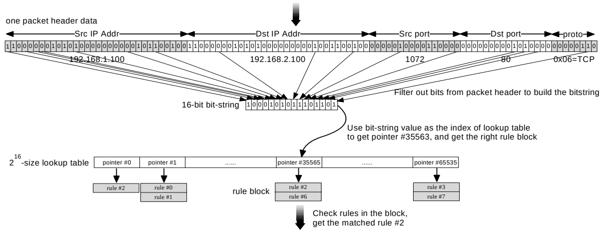

Figure 1. Example for Classification processing. A TCP packet from 192.168.1.100:1072 to 192.168.2.100:80 is received, with bits filtered out, we can count the index value 35563. Use the index to get the right rule block. After checking the rules in the block, we can get the best matched rule #2.

2) Heuristic Swap Scheme: First we choose l-bit b_0 from the 104 bits randomly, and divide **R** into 2^l subsets, obtaining the initial wrap value w_0. Try to swap each bit from b_0 with one from the left bits and get w_1. If $w_1 < w_0$, accept the swap, or else keep the bit and try the next one.

B. Lookup Table Construction

After the generation of the l-bit bit-string b, the lookup table can be constructed with the following steps:

- Set up an m-length table T ($m = 2^l$,too), where each $T_{[i]}$ stores a pointer to a block;
- Check each 104-bit length rule #r in the rule set which covers some ranges at the 2^{104}-length binary line. Pick out the same bits from #r as b, which forms a new l-length #r'. Rule #r' also covers some ranges at the 2^i-length binary line.
- Check each i ($0 \le i < m$). If ranges by #r' covers i, store #r into the block pointed by $T_{[i]}$.

C. Packet Classification Process

Filter out the same l bits as b from each packet header, which form an l bits value i, and index to a block by $T_{[i]}$. After that, check the rules in the block and find the best matched one. Fig. 1 shows an example for the process. Since most blocks only contain few rules, the searching inside a block will be fast. Moreover, heuristic searching technologies like DBS can be taken recursively. As the rules in a block are stored continuously in memory space, network processors with cache like Cavium's Octeon [8] can take advantage.

III. EXPERIMENT

We carry out our experiments on 12 real-life rule sets of three types: Access Control List (ACL), Firewall (FW) and IP Chains (IPC), which are publicly available at [9]. ACL1, ACL1_1-10K contains 752, 916, 4415, 9603 rules; FW1, FW1_1-10K contains 269, 791, 4653, 9311 rules and IPC1, IPC1_1-10K contains 1550, 938, 4460, 9037 rules. HiCuts and HSM are chosen for comparison. We choose the heuristic swap scheme to generate the bit-string (the size is set to 16) and use linear searching in block. Fig. 2 compares the storage among HiCuts, HSM and DBS, from which we can see that DBS is at least an order of magnitude less than HiCuts and HSM for most rule sets. Fig. 3 and Fig. 4 compares the memory access time in average-case and worst-case, which demonstrate that the

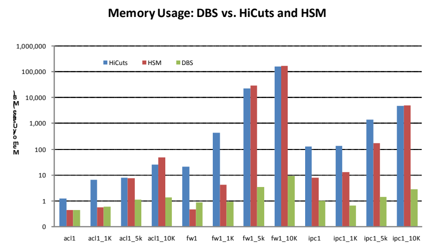

Figure 2. Memory Usage: DBS vs. HiCuts and HSM

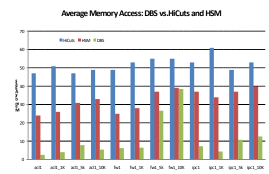

Figure 3. Average Memory Access: DBS vs. HiCuts and HSM

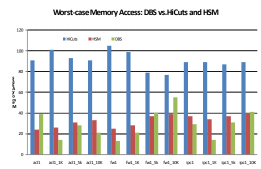

Figure 4. Worst-case Memory Access: DBS vs. HiCuts and HSM

average-case access time of DBS is only 5%-20% of other two, while the worst-case access time of DBS is 10%-50% of HiCuts and less than HSM for most rule sets. Above all, the results demonstrate that DBS achieves much higher performance than HiCuts and HSM on both time and storage.

IV. CONCLUSION AND FUTURE WORKS

In this paper, we propose a bit-level heuristic framework for multi-dimensional packet classification. The packet classification algorithm under our framework shows superior performance on both temporal and spatial measurements comparing with HiCuts and HSM.

Although the experimental results are encouraging, current work is still preliminary. Our future works include the bit-string generation schemes and evaluation in real system.

REFERENCES

[1] M. H. Overmars and A. F. van der Stappen, "Range Searching and Point Location among Fat Objects," Journal of Algorithms, 21(3), 1996.

[2] S. Singh, F. Baboescu, G. Varghese and J. Wang, "Packet Classification Using Multidimensional Cutting," Proc. of ACM SIGCOMM, 2003.

[3] P. Gupta and N. McKeown, "Packet Classification on Multiple Fields," Proc. of ACM SIGCOMM, 1999.

[4] B. Xu, D. Jiang and J. Li, "HSM: A Fast Packet Classification Algorithm," Proc. of the 19th International Conference on Advanced Information Networking and Applications (AINA), 2005.

[5] J. van Lunteren and T. Engbersen, "Dynamic Multi-Field Packet Classification," Proc. of IEEE GLOBECOM 2002, vol. 3, pp. 2215-2219, 2002.

[6] P. Gupta and N. McKeown, "Packet Classification Using Hierarchical Intelligent Cuttings," Proc. of Hot Interconnects, 1999.

[7] S. Singh, F. Baboescu, G. Varghese and J. Wang, "Packet Classification Using Multidimensional Cutting," Proc. of ACM SIGCOMM, 2003.

[8] [Online.] Available: http://www.caviumnetworks.com/ Table.html

[9] [Online.] Available: http://www.arl.wustl.edu/~hs1/PClassEval.htm