2016 International Conference on Computing, Networking and Communications, Green Computing, Networking and Communications

GreenTCAM: A Memory- and Energy-efficient TCAM-based Packet Classification

Xianfeng Li Yuanxin Lin Wenjun Li
Engineering Lab on Intelligent Perception for Internet of Things(ELIP)
School of Electronic and Computer Engineering, Peking University, Shenzhen, China
Email: lixianfeng@pkusz.edu.cn, keyyuanxin@sz.pku.edu.cn, liwenjun@sz.pku.edu.cn

***Abstract*—Ternary Content Addressable Memory (TCAM) is the de-facto standard device used for high-speed packet classification. Despite its capability for line-speed queries, it is very power hungry and area inefficient. The latest TCAM devices by leading vendors come with an power saving mechanism where a subset of its TCAM blocks can be selectively activated. Recent research efforts exploit this feature to reduce power consumption with pre-classification steps. However, the state-of-the-art technique achieves power savings at the expense of poor utilization of TCAM capacity, and the potential of power reduction is not fully exploited in many cases. In this paper, we propose GreenTCAM, an optimized two-stage design for TCAM-based packet classification. Based on common characteristics of rule sets, our design is able to group rules more compactly into TCAM blocks, and activates a minimum subset of these blocks for each incoming packet. Experimental results show that our design achieves a 93.6% power reduction with a TCAM storage overhead of only 5.6% on average.**

***Index Terms*—Packet Classification; TCAM; Power Reduction**

I. Introduction

Modern network routers and switches enforce policy-based forwarding with packet classification, which searches a combination of relevant packet header fields against a set of forwarding rules, and takes corresponding actions upon a rule match. Packet classification is the bottleneck of advanced forwarding, and has attracted research attentions for many years [1].

However, despite extensive research on algorithmic solutions, multi-field packet classification at line-speed remains to be a challenging problem. The common practice in industry is using Ternary Content Addressable Memory (TCAM), which enables parallel lookups against all the rules for the best match in a single pass. However, this brutal force hardware solution is not only expensive, but also very power-hungry. A lot of research efforts have been made on reducing the cost and power consumption of TCAMs, including both architectural designs and algorithmic solutions. In recent years, leading TCAM vendors have provided block-based TCAM designs, where the TCAM device is partitioned into fix-sized blocks, and a subset of them can be activated for lookups as desired. This improved design provides a good substrate for potential power reductions. The problem is how to organize the rules in the TCAM, and restrict packet classification to a small number of TCAM blocks for each incoming packet.

The state-of-the-art work making use of the blocked TCAMs is SmartPC [2]. It performs packet classification in two stages: in the first stage, a given packet is classified by a pre-classifier that identifies the TCAM block to be activated; in the second stage, only the identified block along with a few general blocks are activated and searched in parallel for a match for the packet. SmartPC achieves a very impressive power reduction on average (88%). However, it wastes considerable TCAM capacity in that many TCAM blocks are not fully occupied by rules. In addition, it generates considerable *general rules* for some rule sets, which must be searched for each incoming packet, leading to degraded power reductions.

In this paper, we propose GreenTCAM, an optimized two-stage TCAM-based packet classification, which achieves more power reduction without severe TCAM storage waste suffered by SmartPC. We first separate the rules into a small number of subsets by projecting them into different dimensions. Each subset exhibits a common characteristic on their projected dimension (field). We then arrange the rules into TCAM blocks by using a simplified single-dimensional grouping heuristics. Thanks to the projecting in the first step, this single-dimensional grouping produces very few general rules and holes in the non-general TCAM blocks. Finally, the two-stage TCAM lookups are performed by activating only a few TCAM blocks.

The experimental results show that for ClassBench [3], a popular rule set benchmark also used in SmartPC and other packet classification research, GreenTCAM achieves an average of 93.6% TCAM power reduction; and the TCAM storage overhead is only 5.6% on average, which is almost negligible in practice.

The rest of the paper is organized as follows. Section II gives a very brief introduction on the problem of packet classification. Section III reviews the related work on TCAM-based solutions. Section IV presents the technical details of our work. Section V provides our experimental results. Finally, Section VI draws conclusions on this work.

II. Background

The purpose of packet classification is to enable differentiated packet forwarding according to a set of rules, with each rule R consisting of a tuple of F field values and an action. The problem of packet classification is to find a matching rule for each incoming packet, and take the corresponding

978-1-4673-8579-4/16/$31.00 ©2016 IEEE

action for this packet. Nowadays, A typical rule is a 5-tuple of packet header fields, including source IP addresses (SA), destination IP addresses (DA), source port numbers (SP), destination port numbers (DP), and the protocol type (Prot). An example 5-tuple rule set is shown in Table I. The complexity of packet classification comes from three aspects. First, the multi-field rule set essentially represents a number of hypercubes in multi-dimensional space, and matching a packet corresponds to identifying a hypercube in which the point of the packet is located. Second, some of the fields in the rule set require longest prefix matching (LPM) or range matching (e.g., "*" as a wildcard in Table I means a match on any value), which prohibits fast hashing-based lookups available for exact matches. Last, the rules may have overlaps with each other, and packet classification must report the one with the highest priority.

TABLE I: AN EXAMPLE 5-TUPLE RULE SET

Rule	SA	DA	SP	DP	Prot	Action
R_1	10.0.8.8/32	64.1.8.20/32	*	80	TCP	$action_1$
R_2	10.0.3.8/32	*	80	*	UDP	$action_2$
...	...	...	...	...	...	...
R_N	*	*	*	*	*	$action_N$

Because of the complexity of algorithmic packet classification techniques, high-performance network routers and switches generally rely on TCAMs for line-speed packet classification. A TCAM is a specialized hardware for parallel comparisons. It consists of two parts: a TCAM array and a priority encoder. Rules are stored in the array in descending order of priorities. An input is compared against all entries of the TCAM array in parallel, and the result bit vector (with "1" being match and "0" otherwise) is fed to the priority encoder to select the matched entry with the highest priority.

TCAM is costly in area, and is very power-hungry due to its brutal force comparisons against all entries. To address this problem, leading TCAM vendors are now providing blocked TCAM designs for power reduction by enabling selective activation on a subset of TCAM blocks. In this paper, the proposed framework for TCAM power reduction is based on this substrate.

III. Related Work

Packet classification has received extensive research due to its importance and challenge. There are mainly two threads: algorithmic techniques using commodity SRAMs, and hardware solutions based on TCAMs. Algorithmic techniques, using either decision-tree [4], [5], [6] or field decomposition [7], [8], [9], are still unable to meet the line-speed requirement of every increasing network bandwidth. As a result, TCAMs are still the dominant device used by high-performance network devices for packet classification.

TCAM-based techniques are either concerned with storage [10], [11], [12], [13] or power reduction [14], [15], [2], [16]. Given the purpose of this work, we only make a brief review on the power reduction techniques. TCAM power reduction can be at the architectural/circuit level [14], or at the algorithmic level [15], [2], [16]. Circuit-level techniques are constrained by the market volume, and are unlikely to get commercialized. On the other hand, algorithmic TCAM techniques are based on commodity TCAM designs, thus are more promising on finding their applications in industry.

The state-of-the-art algorithmic TCAM technique is SmartPC [13], which makes use of the blocked TCAM design. The basic idea is to divide the whole 5-dimensional space into a number of non-intersecting sub-spaces, each covering a subset of rules within it. In particular, the two-stage packet classification framework in SmartPC works as follows. It first constructs a pre-classifier for deciding the candidate rules that may match with an incoming packet in the first stage. With the help of pre-classification, only the TCAM blocks containing the candidate rules will be activated in the second stage, thereby avoiding comparisons against all rules blindly.

Under this framework, the critical issue is how to build an effective pre-classifier, such that the rules (TCAM blocks) to be compared are as few as possible during the second stage. At the same time, since the pre-classifier itself is also implemented with TCAM, the size of the pre-classifier itself should be reasonably small to prevent incurring high overhead.

SmartPC adopts a locally greedy heuristic to construct the pre-classifier. For each pre-classifier entry, it starts with a randomly selected free rule (not yet included in any pre-classifier entry), and the initial sub-space covered by this entry corresponds to the selected rule. Any rules intersecting with current sub-space will be candidates for inclusion. However, for a rule to be included, either it is completely covered by the current sub-space, or the sub-space should be expanded to satisfy the first condition. This space expansion will lead to more intersecting rules, and this process will continue until the subset of rules reaches the limit of the TCAM block size. When this process completes, a pre-classifier entry can be generated for the sub-space, and a corresponding TCAM block is allocated for the contained subset of rules.

There are two situations where a rule cannot be included in any pre-classifier entry: (1) the TCAM block of the pre-classifier entry covering this rule becomes full, or (2) its inclusion results in a space expansion that intersects with the sub-space of another pre-classifier. In either case, this rule will be marked as a *general rule*, which has to be compared unconditionally for each incoming packet. Therefore, general rules are highly undesirable, and should be as few as possible. In addition, the second case may also lead to a potential waste on TCAM storage if all the intersecting rules for a non-full pre-classifier entry have such a problem.

An example is shown in Figure 1 (the same one used in SmartPC), where 14 five-dimensional rules are projected on the two-dimensional space along the SA and DA fields. Assume the block size is 5, starting from R0, SmartPC gets an initial sub-space corresponding to R0. The rules that intersect with this sub-space will lead to space expansions. Among them, assume the rule first picked is R5, which will expand the sub-space to the upper red rectangle. Now there are 8 rules that are completed covered by this expanded sub-space, more

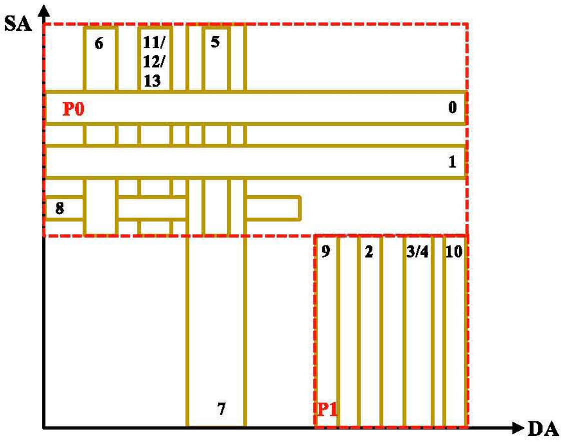

Fig. 1: An example rule set from SmartPC

than the capacity of a TCAM block. As a result, some of them have to be treated as *general rules*, e.g., R11, R12, and R13. In addition, since the pre-classifier entry is already full, the intersecting rule R7 also needs to be marked as a general rule. The subset of rules for the first pre-classifier entry P0 would be {R0, R1, R5, R6, R8}. SmartPC then starts with a free rule to generate the second pre-classifier entry P1. Assume R9 is picked, and when this process completes, P1 will have a subset of {R2, R3, R4, R9, R10}.

From this example, we may discover that the space expands at a fast pace due to rule intersections, especially when the intersected rule is big along some dimension. This easily leads to many general rules because of limited TCAM block size. In this paper, to reduce the number of general rules and to have better utilization of TCAM storage, we propose a framework that reduces the impact of rule intersections on the space expansion. With this framework, we are able to construct a pre-classifier more efficiently.

IV. GreenTCAM

Like other packet classification techniques, we first make some observations on common characteristics of rule sets. With this insight, we propose an improved two-stage TCAM-based packet classification architecture.

A. Observations on Rule Sets

Existing research shows that rule sets for packet classification exhibit some common characteristics, which can be exploited with efficient algorithms [7], [8], [4], [9], [5], [6]. In this paper, we use the publicly available ClassBench [3] for study. From Table II, we can see that the number of unique field values of distinct port ranges and protocol is much smaller than that of the IP address field values (especially for larger rule sets), which is consistent with observations in previous work. The statistical results show that it is hard to distinguish rules by SP, DP or Protocol fields, but the large number of SA/DA values indicate that these two fields may provide good separability for rules, as long as these values are not heavily overlapped.

To confirm this property, we study the distribution of SA and DA fields in terms of their sizes, and the result is plotted in Figure 2. The X-axis represents the prefix length of the SA and DA fields. The smaller the X-axis value, the larger the SA/DA size, for example, 0 means a wildcard "*" (without any prefix). The Y-axis gives the percentage of rules with both SA and DA prefix lengths less than or equal to the corresponding number on the X-axis. For example, the IPC_10K rule set has about 27% rules with both SA and DA prefix lengths of 24 or less. The distribution in Figure 2 clearly shows that the vast majority of rules are very small in either the SA or DA fields. Take ACL_10K for example, 92% of its rules have an exact value in either the SA or DA fields.

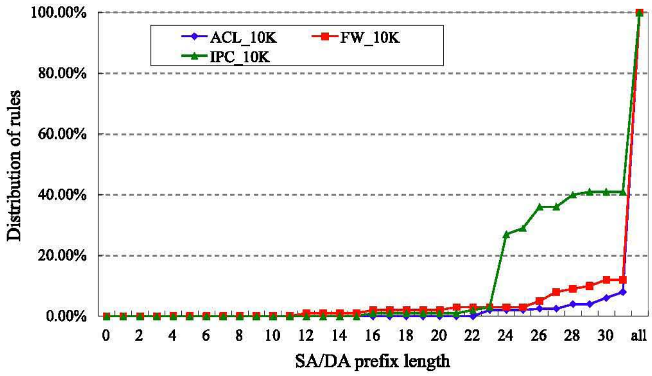

Fig. 2: Distribution of rules in terms of SA/DA prefix lengths

Combining the results in Table II and Figure 2, we get a very desirable observation: *there are many distinctive SA/DA fields, and the vast majority of rules are small either in the SA or DA fields. Therefore, the SA and DA fields provide good separability for rules.*

B. Separability-based Architecture

To exploit the properties observed above, we propose a separability-based two-stage packet classification architecture. The first step is to partition rules into three subsets: 1) SA-subset, which includes rules more specific on the SA dimension (i.e., with SA prefix longer than DA prefix); 2) DA-subset, which includes rules more specific on the DA dimension; and 3) general subset, which includes rules with both SA and DA fields being wildcards (note this subset will be expanded during subsequent steps). Intuitively, rules in the SA-subset have better separability along the SA field because

TABLE II: NUMBER OF UNIQUE FIELD VALUES

Rule sets	Size	Number of unique values				
		SA	DA	SP	DP	Prot
ACL_10K	9063	4784	733	1	108	4
FW_10K	9311	3638	6951	13	43	5
IPC_10K	9037	1515	2726	34	54	7
ACL_100K	98324	42404	78051	1	96	4
FW_100K	88517	21964	66027	13	43	5
IPC_100K	99294	86323	90448	35	55	7

of their smaller SA fields, similarly, rules in DA-subset have better separability along the DA field.

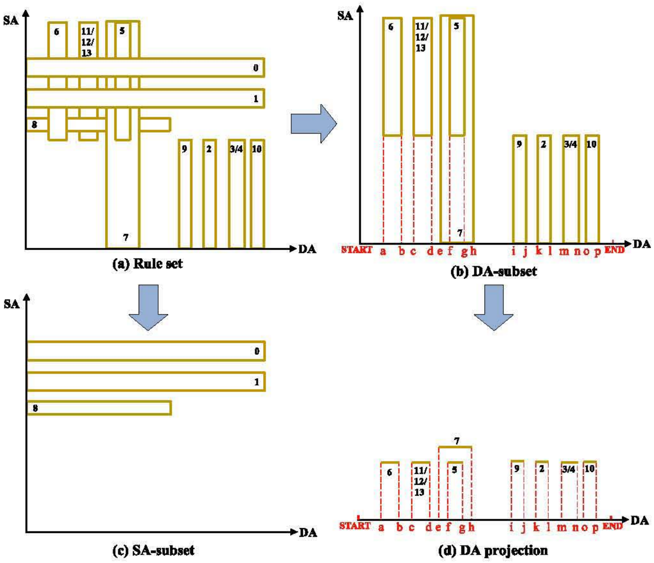

Fig. 3: Rule set partitioning

Take the example in Figure 3 for illustration. The rule set in Figure 3(a) is partitioned into SA-subset in Figure 3(c) and DA-subset in Figure 3(b). As we can see, the dominant intersection pattern is between rules with big fields in different dimensions, but when separated into subsets along the SA/DA fields, rule intersections in each individual subset is greatly reduced compared to the original rule set.

Based on this projection-directed partitioning, we propose a two-stage packet classification architecture in Figure 4 that exploits the good separability of the individual subsets. Compared to the framework in SmartPC, the main difference is that instead of a single pre-classifier covering all the five fields, our design has two pre-classifiers covering the SA or DA fields respectively. For an incoming packet, its SA and DA fields are fed to the two pre-classifier in parallel, each activating a specific TCAM block in the second stage in case of a match with a pre-classifier entry. At last, the respective matching results, along with the result from the general rules, are fed to the priority encoder for the final best match. Although it appears that our design activates one more pre-classifier and TCAM block, this little price will be paid off by less general rules, which must be unconditionally searched each time.

C. *Pre-classification Algorithm*

Now the critical problem is how to construct the pre-classifiers for the SA-subset and DA-subset. Fortunately, since our algorithm now works only on a single dimension, and the overlapping of rules on this dimension is much less severe compared to the multi-dimensional case in SmartPC, we can employ a simple, yet more effective heuristic for pre-classifier construction.

We take the DA-subset in Figure 3 as an example to describe the pre-classification algorithm in an intuitive manner. First, by

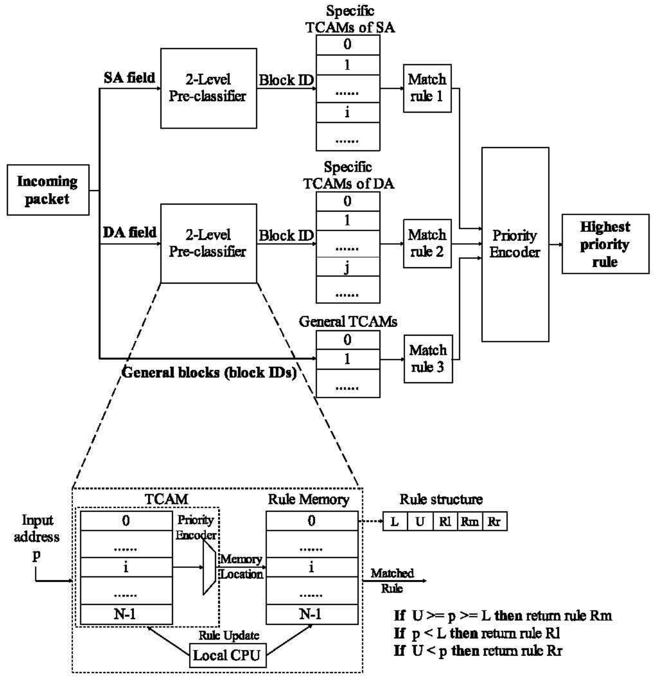

Fig. 4: The architecture of GreenTCAM

projecting the rules in the DA-subset onto the DA dimension, a number of intervals are formulated, with their boundaries being either the start or end points of the individual DA fields. In this example, the 11 DA-subset rules produce 17 intervals, with the first one being [START, a], and the last one being [p, END]. The resulted intervals have a favorable property: at any point in an interval, the rules covering it keep unchanged. It means that if an incoming packet's DA value falls in one interval, then we can identify the potential rules that may match with it.

With this projection in place, our algorithm marches from the first interval originated at the START point, and gradually expands its pre-classification range by absorbing subsequent intervals. Assuming a TCAM block size of 5, the pre-classification range expansion continues until the number of rules *completely covered* by the range exceeds the capacity of the TCAM block. For example, when the range has expanded to [START, g], the subset of rules *completely covered* by it is R6, R11, R12, R13, R5, reaching the TCAM capacity. Therefore, the first pre-classification range is determined, and a process for finding the next pre-classification range will get started at g+1 (not point h). But before that, any additional rules intersecting with the current pre-classification range need to be put into the pool of general rules. In this case, R7 will be treated as a general rule, as it intersects with [START, g].

In this example, when this process completes, we get two pre-classification ranges: [START, g] covering rules {R6, R11, R12, R13, R5}, and [g+1, END] covering rules {R9, R2, R3 R4, R10}. With these pre-classification ranges, we construct their corresponding entries in the pre-classifier. However, since the range is not a single prefix, it cannot be directly represented by an ordinary TCAM entry. Usually, a range needs to be expanded into multiple prefixes for the TCAM, and may lead to considerable TCAM expansion.

Fortunately, the pre-classification ranges have a desirable property: they are single-dimensional contiguous (free of overlaps and holes) ranges. For this special set of ranges, there exists a very efficient architecture proposed by [17], which needs only the same number of TCAM entries as that of the ranges, plus a corresponding SRAM of the same size. We adopt this architecture for our pre-classifier, as illustrated in the lower part of Figure 4. The basic idea is to store the Longest Common Prefix (LCP) covering a range, instead of storing the range itself. Upon a matching on a specific LCP entry for an incoming packet, it consults the corresponding SRAM entry to further decide whether the packet falls in the range covered by the LCP, or in its left/right neighbors. This is done by comparing with the L and U fields (corresponding to the two end-points of the related range) in the SRAM entry, and returning one of the three range indices (Rl, Rm, Rr) depending on the result of the comparisons. With the range index, the corresponding downstream TCAM block will be activated. More details of this architecture can be found in [17].

V. Experimental Results

In this section, we evaluate the performance of GreenTCAM. Part of the rule sets used in our experiments are publicly available from *http://www.arl.wustl.edu/ hs1/PClassEval.html*, a web site of the Washington University at St. Louis. We choose the 10K rule sets from this site. The other 100K rule sets are generated using ClassBench [3]. There are three types of rule sets: Access Control List (ACL), Firewall (FW), IP Chain (IPC). Each rule set is named by its type and size, e.g., IPC_10K and IPC_100K refer to the IP Chain rule set with about 10,000 rules and 100,000 rules respectively.

The effectiveness is evaluated with two metrics: the portion of activated TCAM entries, and the storage overhead due to the pre-classifier and wastage on TCAM block capacities. To study the impact of TCAM block size and give guidance, our evaluation is performed with three different TCAM block sizes: 64, 128, and 256. Since the power consumption of TCAMs is proportional to the number of searched entries, instead of using real hardware or detailed circuit-level simulations, we use a simple linear power model to estimate the power reductions, though the real reductions may be slightly different.

A. Pre-classifier and General Rules

Like SmartPC, our work employs additional TCAM storage for pre-classifier entries, and the resulting storage and power overheads are a natural concern: pre-classifiers with modest sizes are desirable. In addition, we would like to see a small set of general rules, such that the searched TCAM entries can be as few as possible. For these purposes, we take a quantitative evaluation on the two important aspects under different TCAM block sizes.

Table III presents the ratio of general rules and the additional pre-classifier entries. It shows that in most cases, less than 5% of rules are treated as general, the only exception is ACL_10K under the TCAM block size of 64 and 128. Compared to SmartPC, which has 9% general rules on average, GreenTCAM has only 3.3% general rules on average. It means that with a reasonable TCAM block size, GreenTCAM works very well on separating rules apart.

The last three columns in Table III present the additional storage incurred by pre-classifiers in terms of the ratio compared to the size of the rule sets. The results are also very favorable: less than 1% of TCAM entries for 128 and 256 block sizes. If the TCAM width for the pre-classifier is taken into account (which is 32-bit, roughly one-third of a 5-tuple TCAM entrie), the storage overhead can be even lower (about 0.2% for TCAM block size of 128).

B. Power Reduction

To evaluate power reduction, we need to know the number of TCAM blocks activated in the two stages (pre-classification blocks, specific TCAM blocks, as well as general TCAM blocks), and compare their summation with the number of TCAM blocks in the original single-stage TCAM lookup.

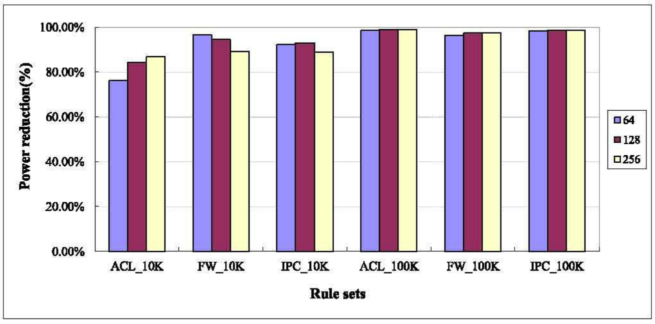

Fig. 5: Power reduction of GreenTCAM

Figure 5 shows the power reduction of the selected rule sets under different block sizes. It ranges from 78% to 99%, with an average of 93.6%. Compared to the average power reduction of 88% in SmartPC, GreenTCAM works even better.

C. Storage Overhead

Although GreenTCAM makes improvements over SmartPC in terms of power reduction, this is not its primary advantage, as the power reduction by SmartPC is already very impressive.

TABLE III: RATIO OF GENERAL RULES AND PRE-CLASSIFIER ENTRIES

Rule sets	general rules(%)			pre entries(%)		
	64	128	256	64	128	256
ACL_10K	22.8	12.4	3.1	1.5	0.8	0.5
FW_10K	0.9	0.9	0.9	1.6	0.8	0.4
IPC_10K	5.5	2.2	1.2	1.6	0.8	0.4
ACL_100K	0.7	0.4	0.1	1.6	0.8	0.4
FW_100K	3.0	2.0	1.6	1.6	0.8	0.4
IPC_100K	0.9	0.5	0.4	1.6	0.8	0.4

The major contribution of this work is the much lower storage overhead. In [2], it is reported that SmartPC achieves power reduction with a poor TCAM storage utilization. There are two reasons that lead to storage wastage: the additional TCAM for pre-classifier entries, and the low utilization of TCAM blocks for specific rules. The later one is the primary contributing factor in SmartPC, as its space expansion mechanism works at a fast pace, and in many cases it is unable to find suitable rules to fill an non-full TCAM block.

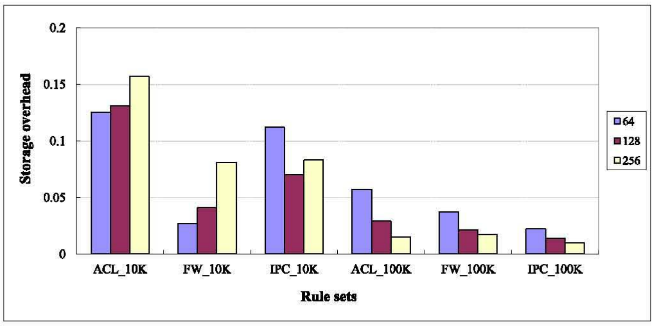

Fig. 6: Storage overhead of GreenTCAM

In contrast, we can see from Figure 6 that the storage overhead in our work ranges from 1% to 16%, with an average storage overhead of only 5.6%. SmartPC, however, has an average storage overhead of 84% for TCAM block size of 128, which means nearly half of the TCAM entries are wasted. As TCAMs are precious resources in high performance routers and switches, the very low TCAM storage overhead of GreenTCAM, along with better power reduction, gives it a clear advantage over SmartPC.

VI. Conclusion

TCAM-based packet classification is the de-facto standard for high-performance packet forwarding. The major problems with TCAMs are their high cost and power consumption. Although SmartPC, the state-of-the-art, has achieved very significant power reduction by selective activation of relevant TCAM blocks, it suffers a high TCAM storage overhead.

In this paper, we propose GreenTCAM, an improved two-stage packet classification architecture. With insights on the separability of rule sets, we design a simple, yet highly effective projection-based framework. For each projected subset, rule overlapping is greatly reduced compared to the original rule set. With this favorable feature, an effective pre-classifier is constructed for each individual subset of rules.

The experimental results show that GreenTCAM achieves greater power savings over SmartPC (93.5% versus 88% on average). More importantly, unlike SmartPC, which has an average of 84% storage overhead, the average overhead of GreenTCAM is only 5.6%, almost negligible in practice. The combined benefits on power consumption and storage efficiency give GreenTCAM a clear advantage over SmartPC, and it can find applications with high-performance requirements under stringent power constraints.

Acknowledgments

This work is supported by the grant of ZTE Cooperation under the contract for SDN forwarding-plane chip architecture research, as well as the grant of Shenzhen municipal government for basic research on Information Technologies (No. JCYJ20130331144751105).

References

[1] David E. Taylor. Survey and Taxonomy of Packet Classification Techniques. *ACM Comput. Surv.*, 37(3):238–275, September 2005. 16.

[2] Yadi Ma and Suman Banerjee. A Smart Pre-classifier to Reduce Power Consumption of TCAMs for Multi-dimensional Packet Classification. *SIGCOMM Comput. Commun. Rev.*, 42(4):335–346, August 2012. 25.

[3] David E. Taylor and Jonathan S. Turner. ClassBench: A Packet Classification Benchmark. *IEEE/ACM Trans. Netw.*, 15(3):499–511, June 2007.

[4] Sumeet Singh, Florin Baboescu, George Varghese, and Jia Wang. Packet Classification Using Multidimensional Cutting. In *Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '03, pages 213–224, New York, NY, USA, 2003. ACM. 32.

[5] Balajee Vamanan, Gwendolyn Voskuilen, and T. N. Vijaykumar. EffiCuts: Optimizing Packet Classification for Memory and Throughput. In *Proceedings of the ACM SIGCOMM 2010 Conference*, SIGCOMM '10, pages 207–218, New York, NY, USA, 2010. ACM. 19.

[6] Wenjun Li and Xianfeng Li. HybridCuts: A Scheme Combining Decomposition and Cutting for Packet Classification. In *2013 IEEE 21st Annual Symposium on High-Performance Interconnects (HOTI)*, pages 41–48, August 2013.

[7] Pankaj Gupta and Nick McKeown. Packet Classification on Multiple Fields. In *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, SIGCOMM '99, pages 147–160, New York, NY, USA, 1999. ACM. 18.

[8] Florin Baboescu and George Varghese. Scalable Packet Classification. In *Proceedings of the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '01, pages 199–210, New York, NY, USA, 2001. ACM.

[9] Scalable Packet Classification using Distributed Crossproducting of Field Labels. 2005.

[10] Qunfeng Dong, Suman Banerjee, Jia Wang, Dheeraj Agrawal, and Ashutosh Shukla. Packet Classifiers in Ternary CAMs Can Be Smaller. In *Proceedings of the Joint International Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '06/Performance '06, pages 311–322, New York, NY, USA, 2006. ACM.

[11] Chad R. Meiners, Alex X. Liu, and Eric Torng. Topological Transformation Approaches to Optimizing TCAM-based Packet Classification Systems. In *Proceedings of the Eleventh International Joint Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '09, pages 73–84, New York, NY, USA, 2009. ACM.

[12] A.X. Liu, C.R. Meiners, and E. Torng. TCAM Razor: A Systematic Approach Towards Minimizing Packet Classifiers in TCAMs. *IEEE/ACM Transactions on Networking*, 18(2):490–500, April 2010. 23.

[13] Chad R. Meiners, Alex X. Liu, and Eric Torng. Bit Weaving: A Non-prefix Approach to Compressing Packet Classifiers in TCAMs. *IEEE/ACM Trans. Netw.*, 20(2):488–500, April 2012.

[14] Ed Spitznagel, David Taylor, and Jonathan Turner. Packet Classification Using Extended TCAMs. In *Proceedings of the 11th IEEE International Conference on Network Protocols*, ICNP '03, pages 120–, Washington, DC, USA, 2003. IEEE Computer Society.

[15] F. Zane, G. Narlikar, and A. Basu. Coolcams: power-efficient TCAMs for forwarding engines. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 1, pages 42–52 vol.1, March 2003.

[16] T. Banerjee, S. Sahni, and G. Seetharaman. PC-TRIO: A Power Efficient TCAM Architecture for Packet Classifiers. *IEEE Transactions on Computers*, 64(4):1104–1118, April 2015.

[17] Yeim-Kuan Chang. A 2-Level TCAM Architecture for Ranges. *IEEE Transactions on Computers*, 55(12):1614–1629, December 2006.