Computer Communications 54 (2014) 84–96

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

A multi-pipeline architecture for high-speed packet classification

Derek Pao [*], Ziyan Lu

Electronic Engineering Department, City University of Hong Kong, Hong Kong

ARTICLE INFO

Article history:
Received 23 May 2013
Received in revised form 5 August 2014
Accepted 7 August 2014
Available online 18 August 2014

Keywords:
Packet classification
Pipelined processing
Prefix inclusion coding

ABSTRACT

In typical algorithmic packet classification methods, the data structure is tailored for the given ruleset. It is common among published algorithmic methods that the worst case number of memory accesses per classification depends on the properties of the ruleset, such as the distribution of the address prefixes and port ranges. As a result, existing methods cannot assure constant classification rate. A novel multi-pipeline architecture for packet classification is presented in this paper. Our method has outstanding performance in both space and time. We incorporate the prefix inclusion coding scheme to achieve outstanding memory efficiency. For rulesets with 10 thousand rules, the storage cost of our method is between 16 and 24.5 bytes per rule. The hardware uses fixed-length linear pipelines. Hence, the classification rate is constant regardless of the ruleset properties. To demonstrate the feasibility of our method, the proposed architecture is implemented on a Virtex-6 FPGA and the device can achieve a classification rate of 340 million packets per second. Power dissipation of the device is about 1.43 W.

1. Introduction

When a packet arrives at a flow-aware router, it is subject to a number of lookup and classification operations. The packet is first checked against an access control list and/or firewall to determine whether it will be accepted or rejected. This classification function supports packet filtering and provides IPSec security associations. A second classification is used to regulate the traffic on a per-flow basis. This step supports quality-of-service provisioning, traffic shaping, and billing and accounting. Packet forwarding can have two choices, (i) the next-hop specified by the routing table or (ii) policy-based forwarding using layer 3 and layer 4 header information. The policy-based forwarding is used to support virtual private networking and tunneling. IP address lookup [1] uses the packet's destination IP address to look up the routing table to determine the packet's next-hop. Packet classification [2,3] uses multiple TCP/IP header fields to classify packets into flows. A typical classification rule involves five TCP/IP header fields, namely the source and destination IP address (SA, DA), the source and destination port number (SP, DP), and the protocol (PT) field.

The IP address lookup and packet classification functions have the same throughput requirement, i.e. the processing rate should match the packet arrival rate. The two operations are implemented on hardware in high-speed routers in order to support wire-speed packet processing. Ternary content addressable memory (TCAM) has been a popular candidate for implementing the IP lookup table and packet classifier in high-speed routers, e.g. the Cisco Catalyst 6500 series. However, TCAM is expensive in terms of transistor count per memory cell and power consumption. Another drawback of TCAM is that its operating speed cannot be scaled up to support 100 Gbps line speed, especially for IPv6. Hence, the industry is eager to explore more cost effective algorithmic RAM-based solutions to replace TCAM. There have been significant progresses in the design of RAM-based IP address lookup engines [4–7], but so far the research on the design of RAM-based architectures for packet classification is less fruitful.

The 5-dimension (DA, SA, PT, DP, SP) packet classification problem is studied in this research. An IPv4 address prefix is represented by a 32-bit value and a 6-bit length attribute. A port range is represented by a pair of 16-bit lower and upper bounds, and the protocol filed is an 8-bit value. Hence, the basic storage cost for a 5-tuple IPv4 classification rule is 18.5 bytes excluding the associated control information, e.g. the rule action. Memory efficiency and classification speed are the two major issues in the design of RAM-based hardware packet classifier. A classification rule corresponds to a rectangular region in the 5-dimensional space, and the classification problem can be modeled as the point location problem, i.e. finding the region(s) that encloses a given point. This problem can be solved using conventional searching algorithms in $O(\log N)$ time with $O(N^d)$ space, or $O((\log N)^{d-1})$ time with $O(N)$ space [2], where N is the number of rules in the ruleset and d is the number of dimensions. One can see that the minimization of the storage space and

* Corresponding author.
E-mail addresses: d.pao@cityu.edu.hk (D. Pao), ziyanlu2-c@my.cityu.edu.hk (Z. Lu).

http://dx.doi.org/10.1016/j.comcom.2014.08.004
0140-3664/© 2014 Elsevier B.V. All rights reserved.

the maximization of the classification speed are two conflicting requirements. The storage cost and classification speed of the published algorithmic methods depend to a large extend on the properties of the ruleset, e.g. the distributions of the address prefixes and port ranges. It is not uncommon among published packet classification methods that the storage cost can be well above 100 bytes per rule. Moreover, to the best of our knowledge, none of the published algorithmic methods can effectively assure constant classification rate that is independent of the ruleset properties.

In this paper we shall present a novel multi-pipeline (MP) architecture for packet classification that has outstanding memory efficiency and offers constant classification rate. The proposed method incorporates the *prefix inclusion coding* (PIC) scheme [8,9] previously proposed by the first author for optimizing the storage space of TCAM-based classifier. Our method is the first proposal to apply field label encoding to optimize the memory cost of RAM-based classifier. The advantages of using PIC in the RAM-based architecture are two folds. First, the length of the SA field can be reduced from 32 bits to 16 bits. The reduced field length of the SA enables an efficient implementation of the hierarchical trie for finding the best matching (DA, SA) pair. Second, the port range can be reduced from 32 bits (16-bit lower and upper bounds) to a 12-bit entity. It is common to have overlapping port ranges in the ruleset. An input port number may match multiple ranges. The inclusion property of PIC allows the system to represent all the ranges that match an input port number by a single codeword (label). This is a significant advantage over conventional field decomposition approach, where an input address or port number is mapped to multiple field labels. The prefix format of PIC also allows an efficient implementation of the hierarchical trie for finding the matching (DA, SA) pair, and a decision tree for handling the (DP, SP) pair or the (PT, DP, SP) triple.

The ruleset is divided into 5 partitions in our method as defined in Table 1. In general, most of the rules fall into partitions 1 and 2, and partition 5 contains a small number of rules. The hardware classifier has three processing phases. In the first phase, the classifier looks up the respective field labels and/or the control information for individual header fields using dedicated fixed-length pipelines. Parallel pipelines are used to process the 5 partitions in the second and third phases. The second phase is used to find the best matching (DA, SA, PT) triple for partition 1, and the (DA, SA) pair for partition 2. The computation involved in the second phase is similar to the hierarchical trie approach. The third processing phase is used to process the remaining fields, i.e. (DP, SP) for partitions 1 and 2, and (PT, DP, SP) for partitions 3–5. The third processing phase is implemented with a novel *decision tree* approach that supports two searching modes, namely *chaining* and *cutting*. If the hardware supports k parallel comparisons ($k = 8$ in this study), a decision tree with 3 levels can support a subset with up to k^3 (512 for $k = 8$) rules.

Table 1
Division of the ruleset into 5 partitions. The symbol * represents wildcard.

Partition No.	DA	SA	PT
1	Non-wildcard value	Non-wildcard value	Non-wildcard value
2	Non-wildcard value	Non-wildcard value	*
3	Non-wildcard value	*	
4	*	Non-wildcard value	
5	*	*	

For rulesets with 10 K rules, the memory cost of our method is between 16 and 24.5 bytes per rule, which is close to the basic storage cost of 18.5 bytes per rule. Fixed-length linear pipelines are used in our method regardless of the ruleset properties. Hence, our method can guarantee one classification per cycle. If dual-port memories are available, e.g. block RAMs in FPGA, the system allows 2 concurrent classifications per cycle. For proof-of-concept, we implement our method on a Virtex-6 FPGA. The device can operate at 170 MHz and the classification rate is 340 million packets per second (MPPS). The device consumes about 1.43 W of power according to the Xilinx design tool.

The organization of the remaining parts of this paper is as follows. In Section 2, we shall highlight the properties of the rulesets. Knowledge of the ruleset properties will help us to understand the performances of the previously published algorithmic methods. A review of related work will be given in Section 3. Details of our method will be presented in Section 4. Section 5 is devoted to performance evaluation and comparison. Conclusion can be found in Section 6.

2. Ruleset properties

Fig. 1 shows an example set of artificial 5-tuple classification rules. The address fields are reduced to 8 bits in the example to enhance readability. An input value can match more than one rule in the ruleset. For example, the input packet header (DA = 11001010, SA = 00111000, PT = TCP, DP = 80, SP = 2200) matches rules R_a and R_c. In the event of multiple matches, the action of the highest priority (smallest priority number) matching rule is applied. The rule priority is defined by the network administrator. In the current convention of FW and ACL rulesets, the rule priority is implied by the order of the rules in the linear list.

Real-life rulesets contain confidential information, and are not available in the public domain. Hence, most researchers use synthetic rulesets in their studies. In this study we use Classbench [10] to generate 12 synthetic rulesets with the given seed files, and each synthetic ruleset has about 10 K rules. The 12 synthetic rulesets have fairly different properties as shown in Fig. 2. For example, the number of distinct DA prefixes in a ruleset varies from under 300 (ACL3) to over 8000 (FW3, IPC2), the average address prefix length varies from 20 bits (ACL2, IPC1) to 30 bits (FW1, FW5, IPC2), and the number of distinct non-wildcard DP ranges varies from 0 (FW2) to over 230 (ACL4). In 9 out of the 12 synthetic rulesets the number of distinct DA prefixes is greater than the number of distinct SA prefixes. TCP, UDP and ICMP are the most common PT values in real-life rulesets. The maximum number of distinct PT values found in the synthetic rulesets is equal to 11. The distribution of rules in the five partitions is depicted in Fig. 3. We can see that the FW and IPC rulesets contain higher percentages of rules in partitions 3 and 4, i.e. they contain more rules having either the DA or SA field equal to wildcard.

3. Related work and hardware implementation issues

Good survey papers on this subject can be found in [2,3]. In this section we shall mainly review some selected important prior

Rule	DA	SA	PT	DP	SP	Priority	Action
R_a	110*	001*	TCP	80	*	1	*Action*$_a$
R_b	110*	0011*	UDP	>1023	*	2	*Action*$_b$
R_c	1100*	00111*	*	*	>2000	3	*Action*$_c$

Fig. 1. Example classification rules.