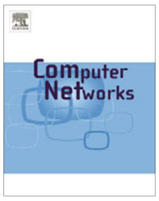
Computer Networks 56 (2012) 3010–3022

Contents lists available at SciVerse ScienceDirect

Computer Networks

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



A new hierarchical packet classification algorithm

Hyesook Lim [a,*], Soohyun Lee [a], Earl E. Swartzlander Jr. [b]

[a] *Department of Electronics Engineering, Ewha W. University, Seoul, Republic of Korea*
[b] *Department of Electrical and Computer Engineering, The University of Texas at Austin, Austin, TX, United States*

A R T I C L E I N F O

Article history:
Received 8 September 2011
Received in revised form 9 March 2012
Accepted 17 April 2012
Available online 3 May 2012

Keywords:
Packet classification
Set-pruning
Binary search tree
Back-tracking

A B S T R A C T

Packet classification is one of the most challenging functions in Internet routers since it involves a multi-dimensional search that should be performed at wire-speed. Hierarchical packet classification is an effective solution which reduces the search space significantly whenever a field search is completed. However, the hierarchical approach using binary tries has two intrinsic problems: back-tracking and empty internal nodes. To avoid back-tracking, the hierarchical set-pruning trie applies rule copy, and the grid-of-tries uses pre-computed switch pointers. However, none of the known hierarchical algorithms simultaneously avoids empty internal nodes and back-tracking. This paper describes various packet classification algorithms and proposes a new efficient packet classification algorithm using the hierarchical approach. In the proposed algorithm, a hierarchical binary search tree, which does not involve empty internal nodes, is constructed for the pruned set of rules. Hence, both back-tracking and empty internal nodes are avoided in the proposed algorithm. Two refinement techniques are also proposed; one for reducing the rule copy caused by the set-pruning and the other for avoiding rule copy. Simulation results show that the proposed algorithm provides an improvement in search performance without increasing the memory requirement compared with other existing hierarchical algorithms.

1. Introduction

One of the challenges in building an Internet router is in the design of an efficient packet forwarding engine [1]. The major functional blocks of a packet forwarding engine that require wire-speed processing are the IP address lookup and the packet classification. IP address lookup determines an output port using the destination IP address of an incoming packet [1–7]. Packet classification classifies an incoming packet based on predefined rules so that the packet can be provided with the service defined for the class to which it belongs [8–22]. For Internet routers to provide different service qualities for each packet flow, packet classification is an essential function. The time allocated to each packet is less than a few nanoseconds for packets arriving at rates of giga bits or tera bits per second. Therefore, the packet processing speed is of primary importance.

The use of application-specific integrated circuits (ASICs) with off-chip ternary content addressable memories (TCAMs) has been the best solution in providing wire-speed packet forwarding [23–25]. However, TCAMs consume 150 times more power per bit than SRAMs. TCAMs also cost about 30 times more per bit of storage than DDR SRAMs. Another issue is the rule duplication problem related to the port number fields. The port number fields can be a fixed number or a range specified by low and/or high boundaries. If a rule has the field specified by a range, the range has to be converted into multiple prefixes, and the rules with each converted prefix are then stored into a TCAM entry. Since a port range can be converted into a maximum of 30 prefixes, if two port fields are specified by ranges, a maximum of 900 TCAM entries are required. Therefore, efficient packet classification

* Corresponding author. Tel.: +82 2 3277 3403; fax: +82 2 3277 3491.
E-mail address: hlim@ewha.ac.kr (H. Lim).

1389-1286/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.comnet.2012.04.014

algorithms using ordinary memories such as SRAMs are required.

Packet classification speed can be evaluated by the number of memory accesses in an algorithmic approach since memory lookup is the slowest operation in the search process. Hierarchical approaches are very effective in providing high-speed search performance since the search space is significantly reduced whenever a one-dimensional search is completed. However, the existing hierarchical approach has two challenges: back-tracking and empty internal nodes. In an attempt to avoid back-tracking, the hierarchical set-pruning trie [11] uses a set-pruning, in which rules included in the ancestor nodes are copied into all the descendant nodes. The grid-of-tries [14] computes switch pointers in advance. However, there is no known algorithm that simultaneously avoids back-tracking and empty internal nodes.

This paper describes several hierarchical packet classification algorithms and proposes a new algorithm. The proposed algorithm constructs a hierarchical binary search tree, which does not include empty internal nodes, for the pruned set of rules. In other words, the proposed algorithm uses set-pruning to avoid back-tracking and uses a binary search tree to avoid empty internal nodes. Hence the proposed algorithm improves the search performance as well as eliminates the memory overhead of empty nodes.

However, because of the set-pruning applied to our proposed algorithm, rules are heavily replicated. Two refinement techniques are suggested for reducing or completely removing the memory overhead caused by the set-pruning. The first proposed refinement technique is to use controlled rule copy. Since rules with a wildcard in the first field are copied into every second-field tree by the set-pruning and rules with a wildcard usually have low priorities, by building a separate tree for rules with a wildcard, the controlled rule copy technique significantly reduces the number of replicated rules.

The second proposed refinement technique is to use ancestor pointers without the set-pruning. Based on the nesting relationship of the first field, the hierarchy of second-field trees is identified. An ancestor pointer points to the second-field tree of its direct ancestor. Since rules with more specific prefixes generally have higher priorities, if a matching rule with the most specific prefixes is identified first, it is more likely that the rule is the best matching rule, and hence the tree search can be avoided for most of the inputs by just comparing with the highest priority number of each second-field tree.

This paper is organized as follows. In Section 2, the packet classification problem is formally defined. Section 3 describes previous hierarchical packet classification algorithms and their characteristics. Section 4 describes the proposed hierarchical set-pruning binary search tree (HSP-BST). Section 5 suggests two refinement techniques of our proposed algorithm. Section 6 shows the simulation results of the proposed algorithms and provides a comparison with other existing algorithms. A brief conclusion is provided in Section 7.

2. The packet classification problem

The header fields used in packet classification are usually the source IP address, the destination IP address, the source port number, the destination port number, and the protocol field. One of the difficulties in packet classification is due to the fact that the search methods are different depending on the header type. The IP address fields require a prefix match, the port numbers require a range match, and the protocol field requires an exact match. An input packet can match multiple rules. All of the matching rules are returned when using multi-match packet classification; the best matching rule (BMR) with the highest priority is returned when using single-match packet classification.

Packet classification problem can be formally defined as follows [11]. Let $\mathcal{R} = \{R_1, R_2, \ldots, R_N\}$ be a given set of rules, for $i = 1, \ldots, N$, where each rule R_i consists of three entities: (1) A regular expression $R_i[j]$ for $j = 1, \ldots, D$, on each of D header fields, (2) A priority number, $p(R_i)$, indicating the priority of each rule in the set, where a smaller number indicates a higher priority, and (3) An action, referred to as action(R_i). An incoming packet P matches rule R_i if all the header fields $P[j]$, for $j = 1, \ldots, D$, of the packet match the corresponding fields of $R_i[j]$. Let $\mathcal{M}(P)$ be the set of rules in $\mathcal{R}$ that P matches, then $\mathcal{M}(P) = \{R_i \in \mathcal{R} | \ P \text{ matches } R_i\}$. The D-dimensional packet classification problem is to find the best matching rule (BMR) R_k in $\mathcal{M}(P)$ such that $p(R_k) < p(R_i)$ for all R_i, $k \in \mathcal{M}(P)$, $i \neq k$. In other words, the packet classification problem is to find the BMR, which is the matching rule with the highest priority. Once the BMR R_k is determined, the input packet is treated as specified by the action(R_k).

Table 1 shows an example rule set with 15 rules that is referred to as the class-bench database [26]. A smaller rule number is assumed to indicate a higher priority. The first two fields are related to source and destination prefixes, and they require prefix match operation. The next two fields are related to source and destination port ranges (or numbers), which require range match, representing the start and the end of the range. The last field is related to a protocol type, and it requires exact match. For example, an input packet of (110100, 110000, 53, 25, 4) has the rule R_4 as the BMR, and the packet is treated according to the action specified by R_4. The action field is omitted in Table 1 for simplicity. This example set will be consistently used in describing each hierarchical algorithm including our proposed algorithm.

3. Related works

3.1. The hierarchical trie (H-trie)

The effectiveness of the hierarchical approach is a result of the diversity of flows related to IP prefix fields is much greater than that of other fields [12]. The hierarchical trie [11] firstly builds a source prefix trie, where each node of the source trie hierarchically connects to the destination prefix trie, which is composed of rules with the same