Priority based Packet Classification

Vijayalalitha A, Dharmaraj R

Abstract— **In an internet routers packet classification is one of the challenging factors, which involve a multi-dimensional, search that to be performed ad wire speed. Routers classify packets based on which flow they belong to and to determine what service they should provide or receive. This paper describes various packet classification algorithms and proposes a new efficient packet classification algorithm using priority trie approach. This algorithm improves the router performance in terms of search speed and memory size.**

Index Terms— **Trie-based algorithm, Tuple space based algorithm, Cutting based algorithm.**

I. Introduction

In general router provides the best-effort service to all incoming packets. Advanced routers can use packet classification to support higher level functions such as QoS, access control. As an essential prior condition, packets need to be classified into multiple flows from the packet header field compared with predefined rules in rule set. The rule for packet classification consists of a set of fields such as source address, destination address and so on. Each rule in a classification table has a priority which is defined by multiple fields. In a given incoming packets are compared with worth corresponding rule fields and there may be a possibility of multiple rules can match to an incoming packet. The highest priority rule is selected among the matching rules. Most of the previous packet classification algorithms have a trade-off between the required memory size and searching speed. The searching speed is measured by the number of memory accesses since memory access is the most time consuming function in the searching procedure.

Ternary Content Addressable Memory (TCAM) has been widely used in commercial routers. It provides very good search performance [2].However TCAM consumes a lot of power and larger memory space than original memory. To reduce the power consumption and to improve the throughput by make use of large classifiers implementation in TCAM which is impractical.

The basic trie-based algorithms use a source IP prefix and destination IP prefix to build tries .A hierarchical trie(H-trie) [3] constructs the first trie using the source prefix field and each node of the trie hierarchically connects to the second trie constructed with the destination field rules with the same source field ,so that both source and destination fields are searched simultaneously. In H-trie searching for matching rules has to be continued until leaf of the trie is visited. Packet classification speed can be evaluated by the number of memory access. Hierarchical approach is very effective and providing high speed search performance, but existing hierarchical approach has two problems: back-tracking and empty internal nodes. Set-pruning [3] is used to overcome the above problem and to improve the search time of H-trie by coping all possible matching rules in the ancestor nodes are copied into the leaves. It improves the search time but it requires huge amount of memory. The grid-of-trie [4] overcomes the disadvantages of rule duplication by pre-computing best matching rules of each node and storing switch pointer of each nodes.

The HiCuts (hierarchical intelligent cuttings) [5] and HyperCuts[6] algorithms partition a multi-dimensional search space based on heuristics that exploit the structure of classifiers. The decision tree is constructed based on its depth and degree of the node. Local search decision to be made at each node is determined in a pre-processing step based on the structure of the classifier. In decision tree each leaf node includes a pre-determined small number of rules that can be searched linearly. The cutting algorithms have some issues that the search speed is highly depended on the characteristics of the classifiers and excessive pre processing time may be required. In this paper propose a new approach called Priority Trie (PT) is employed to perform match in entire rule fields in a very efficient way. The PT is constructed based on the destination prefix fields of rules.

.

II. RELATED WORK

In this section, we discuss related work on trie-based algorithm and cutting based algorithm.

A. Trie-based algorithms

Hierarchical trie (H-Trie) [3], [4] first builds a source prefix trie and each prefix node of the source trie hierarchically connects to a destination prefix trie with the same source prefix field. For a given input packet the search is performed in the source prefix trie first. If there is a match with the source prefix, then the search control moves to the corresponding destination prefix trie. If there is match in the destination trie, the rule number is stored and the search is further needed then continued from the node of the source trie from where it traversed to the destination trie. The search is continued until there are no match nodes to proceed in the source trie. H-trie has some empty internal nodes which are not associated with a prefix or a rule. While searching, all the destination trie connected to every matched node of the source trie should be visited in order to determine the highest-priority rule, and this kind of search procedure is called back-tracking. The back-tracking causes the excessive number of memory accesses. The search complexity depends

Manuscript received Jan, 2014.
Vijayalalitha A*, Department of Computer Science and Engineering, Sri Vidya College of Engineering and Technology, Virudhunagar, India, +91 9786311795.*
Dharmaraj R*, Research Scholar, M.S.University, Tirunelveli, India.*

on the number of destination tries visited, and the memory requirement depends on the number of nodes. The H-trie includes many empty nodes in the path to prefix node (or a rule node). The empty nodes waste memory space as well as degrade the search performance by causing unnecessary memory accesses.

B. Cutting based algorithms

Cutting based algorithms partitioning a search space into multi-dimensional space composed of each rule field based on the heuristics of rules in a given rule set, and each partitioned space is mapped to a node of a decision tree. For constructing the decision tree, HiCuts uses a local optimized decision at each node in determining the dimension (or field) of the cuts and the number of cuts to be made in the chosen dimension [5]. The criterion is to balance the storage requirement and search speed. While HiCuts algorithm only considers one field at a time in selecting the dimension of cuts, HyperCuts [6] algorithm considers multiple fields at a time. In selecting the dimension of cuts, the ratio of the number of distinct elements to the total number of possible values representing the dimension is considered. When compared with HiCuts decision tree, the decision tree of HyperCuts has smaller depth, as multiple fields are used at the same time in a single node, and thus, the search speed is improved. However, the number of entries is more in HyperCuts, due to the generation of many unnecessary entries.

C. Tuple space based algorithms

In tuple space algorithm [3], each rule in the rule set is specified as a pair of prefixes. The lengths of the prefix pair in the rule is defined as tuple and denoted as (i, j), where ‘i’ is the length of the source prefix and ‘j’ is the length of the destination prefix. A packet arriving at a link is queried with all the non-empty tuple spaces by extracting ‘i’ bits from source IP address and ‘j’ bits from destination IP address.

III. PRIORITY TRIE (PT)

Most of trie based packet classification algorithm has many empty internal nodes which waste memory spaces and increases the number of memory access. In priority trie (PT), a given input first compared with the shortest prefix of the rules and even if a match was found, corresponding rule number stored in that node and the search needs to be continued until there is no match or a leaf is visited. There may be a possibility for highest priority rule match found at a lower level of trie. Our proposed priority trie construct a tree for destination prefix fields based on its priority. We can relocate the highest priority rules; relocated rule is termed as priority rule stored is its highest priority level instead of its original level. PT constructed based on the destination prefix field associated with the information about entire rule fields. On searching a match node is determined if it has matching prefix with the source and destination prefix value or matching with all fields in the corresponding rules. If match found, then searching process is immediately finished without searching complete trie. This property effectively improves the search performance.

A. Building PT

The Priority Trie(PT) can be constructed based on its destination prefixes from its rule set. The destination prefixes with its priorities are used to determine the path of the trie. Initially, first rule and its priority and the complete information about rule is inserted at the root node. If new entry is added, it has highest priority than rule at root node; it replaces the root node information. Rule at root node finds its new position with the help of its destination prefix value, if it starts with ‘0’ then create a new node is insertes at left or if its ‘1’ then inserted at right of the root node relevant rule number also stored in a new node.

On inserting a new rule into the trie, if the destination prefix reaches its own position even if the node had a highest priority than rules are already in a trie, that rule has to be replaced. If a single node has more than one matched rules, that rules are connected by a linked list with its decreasing priority order .Example rule set shown in table I and the corresponding priority trie, which is constructed based on its destination prefix filed, is shown in fig 1. From the fig 1, entire rule fields are stored in each node of priority trie, which are represented in the form of ‘rule number (source prefix, destination prefix). Dark nodes are original nodes and white nodes are priority nodes. An empty internal nodes are replaced by highest priority rule belonging to the sub-trie, is rooted by that empty internal node. Rules are relocated at the same level or the highest level than its original destination prefix length.

B. Searching in PT

The searching procedure in Pt is basically same as the search in binary trie. Search procedure starting from sequential inspection of destination address bits.

Table I: Example rule set

Rule Number	**Source Prefix**	**Destination Prefix**
R0	**010***	**10***
R1	**00***	**01***
R2	*****	**10***
R3	*****	**01***
R4	**1***	**1***
R5	**1101***	**001***
R6	*****	**11***
R7	*****	**001***
R8	**111***	**01***
R9	**010***	**10***
R10	**111***	*****

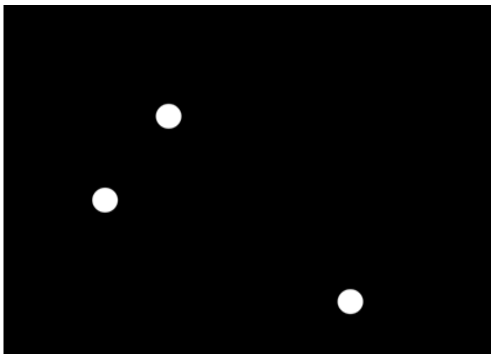

Fig.1. Priority Trie (PT) structure for packet classification.

If there is a match with all fields in a node then corresponding priority number is stored. Search will be finished in case, if a match with its priority rule or reach a leaf, it will always finished with the leaf of trie. For example, assume a given input packet (001011, 011000), then 011000 is taken as an input for search. At a root node input does not matched. Since the first bit of input to be taken, is '0' then searching traverse to the left child of root node. The input is compared with R1 and match with the rule to input packet to be found. Hence currently R1 is considered as the current best match. Then search moves to the right child hence second bit is 1. Since the stored rule R3 and R8 does not match with the input packet and R3 matched with input but it does not have higher priority than the current best match. Now the search is over because it reaches the leaf and the best matched rule of given input is R1. If more than one rule has the same prefix field, those rules are linearly stored in the same node, and linearly searched with the input packet fields.

IV. CONCLUSION

A novel framework for packet classification is proposed in this paper. We present the priority trie algorithm including build process, search process. Compared with previous algorithm, the highest priority rule that is included in the search trie is compared first. In this way, empty nodes are completely removed and hence the required memory size and search space is improved. Hence, the priority trie for packet classification algorithm improves the router performance.

V. FUTURE WORK

While considering an effective classification of packet in router to be considered and priority based algorithm mainly based on rules priority from rule set. After completing that process signature added to each prioritized packets.

REFERENCES

[1] Sarang Dharmapurikar, Praveen Krishnamurthy, and David E. Taylor, "Longest Prefix Matching Using Bloom Filters", EEE/ACM Transactions on Networking,Vol. 14,No.2,April 2006.

[2] Hyesook Lim, Hyeong-gee Kim and Changhoon Yim, "IP Address Lookup for Internet Routers Using Balanced Binary Search with Prefix Vector", IEEE transaction on communication,2009.

[3] P.Gupta and N.Mckeown, Algorithm for packet classification,IEEE Network 15(2):24-32 (2001).

[4] V.Sriniasan,G.Varghese,S.Suri,M.Waldvogel,Fast and scalable layer four switching, ACM SIGCOMM, 1998.

[5] P.Gupta and N.Mckeown ,Classification using hierarchical intelligent cuttings, IEEE Micro,20(1);34-41(2000)

[6] S.Singh ,F.Baboescu ,G.Varghese and J.Wang, Packet classification using multidimensional cutting Proc.SIGCOYM,213-224(2003).

[7] Hyesook Lim, Soohyun Lee, Earl E, Swartzlander Jr, " A new hierarchical packet classification algorithm", Computer Networks,2012.

[8] Hyuntae Park, Hyejeong Hong, Sungho Kang, "An efficient IP address lookup algorithm based on a small balanced tree using entry reduction",Computer Networks,2012.

[9] Chun-Nan Lu, Ying-Dar Lin, Chan-Ying Huang, Yuan-Cheng Lai, "Session level flow classification by packet size distribution and session grouping", Advanced Information Networking and Applications, 2012.

[10] Ashley Thomas, "RAPID: Reputation based Approach for Improving Intrusion Detection Effectiveness" ,2008.

[11] Alex X. Liu, Chad R. Meiners, and Eric Torng, "TCAM Razor: A Systematic Approach Towards Minimizing Packet Classifiers in TCAMs", IEEE/ACM Transactions on Networking, vol. 18, no. 2, april 2010.

[12] Hyesook Lim,So Yeon Kim, "Tuple pruning using Bloom filters for Packet classification", EEE Computer Society, 2010.

[13] Haoyu Song, Jonathan Turner,"Fast Filter Updates for Packet Classification using TCAM",IEEE, 2006.