Routing Lookup by Using Enhanced Patricia Tree

Dai Akashi, Hiroaki Nishi *Member, IEEE*

***Abstract*—DRAM-based routing lookup function has merits of low-power consumption and large scale capacity. In DRAM-based routing lookup, routing table is structured as a patricia tree. This paper proposes a new structure of routing table tree, EPT (Enhanced Patricia Tree). It can reduces number of nodes by pruning redundant branches by having the results in all nodes. EPT enables to reduce the number of memory access and is useful for future large routing table in IPv6 environment.**

I. Introduction

Currently, dissemination of broadband network enlarges the size of Internet. Also ubiquitous network, "anytime and anywhere network", increases the number of hosts connected to the internet. These actions are accelerating by developing more attractive services that receive benefits of high throughput network.

However, this advancement also raises the problem of increasing number of nodes and routes that is managed by the Internet. Routing lookup is a basic function of Internet router for distributing packets to their appropriate destination according to exchanged routing information. This routing process should be done with keeping wire-rate packet processing. Since routing-table lookup is a comparatively heavy task, high-throughput routing lookup are strongly recommended.

In this paper, EPT (Enhanced Patricia Tree) is proposed to address these requirements. EPT enables to resources and lookup latency.

II. Routing Lookup

Routing table is a memory that consists of IP address, prefix length, and the address of output port. The table is stored in TCAM [1] ordinarily for support Longest Prefix Match (LPM). LPM is a selection rule of a routing table entry, which has the longest prefix length. It can be the only result of the selection when some entries match.

Although TCAM can attain high-speed routing lookup, TCAM has to use comparators on each entry. These comparators require high-power consumption and large hardware resources. In near future, there is a possibility that TCAM is not a suitable device to manage a growing routing table, i.e. IPv6.

D. Akashi is with School of Integrated Design Engineering, Graduate School of Science and Technology, Keio University, 3-14-1 Hiyoshi, Kohokuku, Yokohama 223-8522 Japan (E-mail:akashi@west.sd.keio.ac.jp)

H. Nishi are with Department of System Design, Faculty of Science and Technology, Keio University, 3-14-1 Hiyoshi, Kohokuku, Yokohama 223-8522 Japan (E-mail: west@sd.keio.ac.jp).

A. RAM based LPM search

Compared with TCAM, RAM has the merits of low-power consumption and large memory capacity. Meanwhile, RAM requires multiple memory accesses because tree-based lookup algorithm is needed for supporting LPM. LPM lookup under tree-based algorithm is achieved by succeeding access of entries from the root to the leaf of the search tree. This feature becomes a bottleneck of routing lookup.

In order to reduce the multiple accesses, cache memory architecture for routing table lookup by utilizing IP traffic locality has been studied widely [2]. Although utilization of the cache can reduce multiple memory accesses, the cache cannot hit all packets. Thus, this cache miss still causes multiple memory accesses to routing table. It is necessary that the forms of routing table or search tree are changed to be a suitable structure for reducing the accesses.

B. Tree based routing table lookup

Ordinarily, routing table is formed as a patricia tree [3]. Patricia tree is a reduced binary tree by compressing omissible nodes. As similar tree, Trie [3] is also proposed. Trie is a reduced binary tree which has results of lookup in not only terminal nodes but some internal nodes. However, the searching procedure of Trie cannot stop until the terminal node and a memory is indispensable for keeping all result of internal nodes found in the searching procedure.

III. Enhanced Patricia Tree

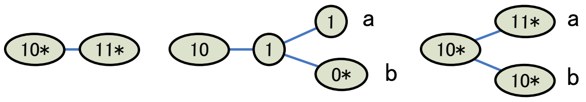
Enhanced patricia tree (EPT) can eliminate the omissible nodes of existing tree and its characteristics of structure are as follows.

1. All nodes have default result of routing lookup
2. New entry is registered on empty leaf as possible even if the node has duplicated 0* or 1* nodes.

Although patricia tree and Trie are regarded as suitable structure for routing lookup, they cannot stop searching until it reaches terminal nodes. If all nodes have the information of a result, searching procedure can stop when it reaches a node which satisfies LPM. This can reduce counts of access for routing lookup.

The other feature of EPT is its node-addition method, characteristic 2. The new entry is added to empty leaf as long as its inclusion relation with the concerning node and two branch nodes is kept. The registration method is attained by using structuring algorithm described later. To explain the merits of this method, examples are shown in Fig.1.

In this case, 1011* is added on 10* first, and after that a



(a) Original (b) Patricia tree (c) EPT

Fig.1 Structuring process for constructing balanced tree

new entry 1010* is added too. In patricia tree (b), a new intermediate node 101 is added. In EPT(c), 1010* is registered on empty nodes under 10* directly. In this building process, the tree can be constructed symmetrically by adding new entry on empty leaf efficiently. This method can reduce omissible intermediate nodes and construct near balanced tree.

IV. STRUCTURING ALGORITHM

Patricia tree only checks whether next bit of prefix is “1” or “0” when adding a new node. Contrary to patricia tree, EPT has a novel structuring algorithm that can grow its tree structure by comparing prefix of a new node and nodes which have added. Fig.2 shows whole algorithm of EPT as a virtual program.

```
While (New entry exists){
    New entry is fetched;
   If (Target node does not have both leaves){
      Register to left;
   }else if(Target node has only left leaf){
      if(New entry is not included by left leaf)
         Register to right;
      }else
         Move pointer to left;
   }else if(Target node has both leaves){
      if(New entry is included by left leaf){
         Move pointer to left;
      }else if(New entry is included by right leaf){
         Move pointer to right;
      }else if(Both leaves have inclusion relation){
         Reconfigure and register;
      }else if(Both leaves have no inclusion relation){
         If(New entry includes at least one leaf){
           Reconfigure and register;
         }else if(New entry does not include both leaves){
            Add new intermediate node;
            Reconfigure and register;
}  }  }  }
```

Fig.2 Structuring algorithm

V. EVALUATION OF EPT

Tree building algorithm and search algorithm of both EPT and patricia tree are implemented by using C language, real BGP table and network trace captured in WIDE without an anonymous filter. I evaluated (1) total number of nodes (2) Total number of entries at each depth (stage) (3) Average counts of memory access of routing lookup.

As a result of (1), EPT can reduce number of nodes by 2,500 at smallest table and 19,000 at biggest table. This result proves that EPT can reduce omissible nodes at any size of BGP table. As a result of (2), the peak of the distribution of EPT is shifted to lower depth than patricia tree as shown in Fig.3. Fig.3 shows the number of entries at each depth in both trees. This shift results in reducing the counts of memory accesses. As a result of (3), average number of memory access of EPT is 0.47 counts smaller than that of patricia tree. This difference will become larger when it is used in future IPv6 environment. Now, the number of IPv6 entries is almost only 800, and EPT cannot be efficient yet. However, EPT has a possibility to improve memory access costs in the future IPv6 environment which has a number of table entries in huge address space sparsely.

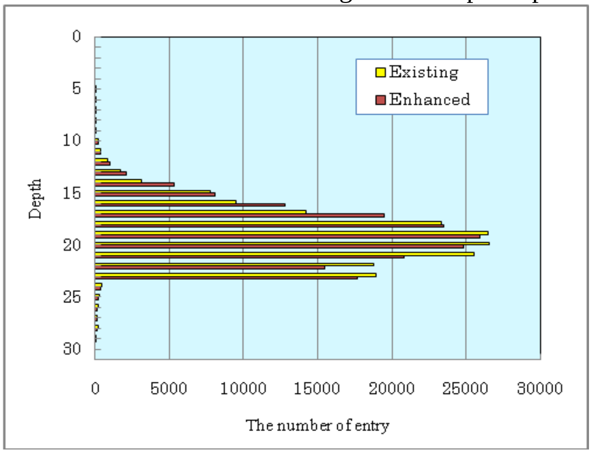


Fig.3 The number of nodes at each layer

A. *Hardware size*

Searching logic of EPT was implemented by using verilog hardware description language and cell based library of 0.18 micro-meter technology. It is simulated by using Cadence NC-Verilog simulator version ldv51 and synthesized by Synopsys Design Compiler 2005.09.As a result, total logic size of EPT search hardware equals to the size of 1937 NAND gates. It can be implemented at small cost.

VI. CONCLUSION AND FUTURE WORKS

EPT improves structure of patricia tree and enables efficient routing lookup by reducing the number of memory-access count. Hardware of EPT is comparatively small and it can be used as a hardware element of parallel search for high speed routing table lookup.

ACKNOWLEDGMENT

This work was partially supported by National Institute of Information and Communications Technology in Japan and Joint Research Fund “Network Systems using End-to-End Hosts”, National Institute of Informatics (NII), Japan.

REFERENCES

[1] V.C. Ravikumar, Rabi N. Mahapatra, and Laxmi Narayan Bhuyan, ” EaseCAM: An Energy and Storage Efficient TCAM-Based Router Architecture for IP Lookup ”, IEEE TRANSACTIONS ON COMPUTERS, Vol. 54. No.5, pp. 521-533, 2005

[2] I.L. Chvets and M.H. MacGregor. Multi-zone caches for accelerating IP routing table lookups. In *High Performance Switching and Routing (HPSR) 2002*, 2002.

[3] M. A. Ruiz-Sbnches, E. W. Biersack, W.Dabbous, “Survey and Taxonomy of IP Address Lookup Algorithms”, IEEE Network, March/April 2001