Binary Decision Diagrams for Efficient Hardware Implementation of Fast IP Routing Lookups *

Rama Sangireddy and Arun K. Somani

Dependable Computing & Networking Laboratory
Department of Electrical and Computer Engineering
Iowa State University, Ames, IA 50011
e-mail: {sangired, arun}@iastate.edu

Abstract

With an immense continuous growth in the Internet traffic, the demand for routers that perform IP routing at high speed and throughput is ever increasing. The key issue in the router performance is the IP routing lookup mechanism based on the longest prefix matching scheme. Earlier works on fast IPv4 routing table lookup are based on Content Addressable Memory (CAM), memory lookups and the CPU caching. These schemes depend on the memory access technology which limits their performance. Besides, these address lookup schemes designed for the IPv4 32-bit address mostly are not extensible to adapt to the forthcoming IPv6 where the IP address is 128 bits long. The paper presents a Binary Decision Diagrams based optimized combinational logic for an efficient implementation of fast address lookup scheme in reconfigurable hardware. The experimental results show that, for the 32-bit IP address large MAE-east routing table, the number of redundant nodes is more than 99.99% in constructing the binary decision tree. With the binary encoding of the output port, an additional 36% reduction is obtained in the number of effective nodes. Besides the performance of the scheme, routing table update and the scalability to IPv6 issues are discussed.

1 Introduction

With the doubling of Internet traffic every few months, the fast handling of communication packets has become a major issue. The router uses the destination IP address of the incoming packet to lookup the next hop router to which the packet has to be forwarded. Since the introduction of Classless Inter Domain Routing (CIDR) in 1993, the IP address lookup has become a major contention. The problem involves two steps, first to obtain all the matching prefixes from the routing database for the particular destination IP, and second to retrieve the next hop port for the longest matched prefix. The other major problem is the frequent update of the routing table [1]. As the topology of the network changes, new routing information is distributed among the routers resulting in changes in routing tables. Consequently, one or more entries must be added, updated, or deleted from the table. This updating of the table can, in the worst case scenario, be of the order of a few hundred updates per second. The IP address lookup schemes, both hardware and software, developed so far, do not address the above two problems with equal efficiency. Though the previous related IP lookup address schemes have their positive points, most of them emphasize on the table lookup at the expense of the table update.

In this paper, we propose a reconfigurable hardware solution, using the well received concept of Binary Decision Diagrams (BDDs), that provides an efficient IP address lookup along with providing a better scheme for updating the routing table. Binary Decision Diagrams (BDDs) are one of the biggest breakthroughs in CAD in the last decade. BDDs are a canonical and efficient way to represent and manipulate Boolean functions and have been successfully used in numerous CAD applications. Although the basic idea has been around for more than 30 years [2], it was Bryant who described a canonical BDD representation [3] and efficient implementation algorithms [4].

The rest of the paper is organized as follows. In Section 2, we describe the longest prefix matching problem. In Section 3, we discuss the related work done and address the associated problems to support the scheme being proposed in this paper, followed by a brief overview on Binary Decision Diagrams and the motivation for the proposed scheme. Section 4 gives the details of the proposed scheme and the implementation issues. It also discusses in detail, the routing table update scheme and the associated observations. The performance analysis of the scheme is given in Section 5. Finally, Section 6 concludes the discussion.

2 Longest Prefix Matching

The routing of the communication packets in the IP domain is done on the Next-Hop basis, i.e., the router takes the responsibility of sending any incoming packet till the next hop only. Consequently, the packet reaches its final destination in multiple hops. The next hop for a packet is determined by the router based on its destination IP address. Each router has a database, in the form of a routing table, of the prefixes of varying length and the corresponding next hop port (NHP) for each prefix. A typical routing table is shown in Table 1.

The length of the prefixes can vary from 0 to 32 bits. For

---

*The research reported in this paper is partially funded by the grants from National Science Foundation, ANI9973102 and CCR9900601.

0-7803-7128-3/01/$10.00 (C) 2001 IEEE

Table 1: A sample routing table.

Prefix	*length*	*NHP*
*	0	0
0*	1	1
01*	2	3
10*	2	2
001*	3	1
101*	3	2

an incoming packet, its destination address is compared with all the current prefixes in the routing table and the next hop port (NHP) associated with the longest matching prefix is determined to be the output port for the packet. For example, a destination IP address 129.186.200.205 matches three prefixes 129/8 (prefix/length), 129.186/16 and 129.186.192/20 in which case, the longest matched prefix 129.186.192/20 is considered to be the best match and the packet is routed to the output port associated with that particular prefix. In other words, routing based on the longest prefix matching is equivalent to routing the packet to the nearest possible IP address. If none of the prefixes match with the destination IP address, the packet is sent to a default port, which is associated with a prefix of length zero.

The metrics taken into consideration, in general, while designing the IP lookup algorithms are *Preprocessing time*, *Storage requirements*, *Lookup rate* and *Update time*. However the lookup rate is the most significant parameter that needs to be addressed followed by the update time. With the latest advancements in the network technology, the communication speed is leaping from Ethernet of 10Mbps to Fiber Distributed-Data Interface (FDDI) of 100Mbps to gigabit Ethernet. With the OC192c Line (Line-rate 10Gbps), 31.25 million packets (average size of 40Bytes) have to be processed each second, while for the OC768c (Line-rate 40Gbps), the processing rate required is 125 million packets per second. Besides, the largest IP routing table known today is MAE-east [1] which contains around 45000 entries. Consequently, the need for enormous amount of data processing at phenomenal speeds, makes the longest prefix matching problem more complicated. Further, the IP address, which is 32 bits long in IPv4, would be 128 bits long, when IPv6 is introduced, making the problem of IP routing even more complex.

3 Previous Work

The IP routing lookup schemes so far introduced can be broadly classified into two categories, viz., *software* and *hardware* approaches. Software solutions include the various tree based approaches [5, 6, 7, 8] and other binary search methods like the *multiway and multicolumn search* [9]. Besides, several schemes based on binary trie approach [10, 11, 12, 13, 14] had been proposed. The Patricia trie scheme is a popular software implementation of the search task in the routing tables. However, the Patricia trie is a complex structure for hardware implementation and it requires a large amount of memory as it stores the complete key (32 bits with IPv4 and 128 bits with IPv6) within each entry. Similarly, the LC-trie scheme gives a reasonable lookup speeds but requires high insertion times.

Simultaneously, various hardware schemes like Content Addressable Memories (CAMs) [15], memory lookup based [16, 17, 18] and CPU caching [19] have been proposed. The Content Addressable Memory can search all of its entries in parallel given a destination IP address. The scheme in [15] uses a separate CAM for each possible prefix length and hence, in worst case might require 32 CAMs for IPv4 and 128 CAMs for IPv6 resulting in an expensive solution. Besides, CAM might not be able to keep pace with the fast developing optical fiber technology as it depends and is limited by the IC process technology. Memory lookup schemes are based on SRAM indirect indexing, and hence require an additional ASIC to corporate with the memory. Besides the memory access speed might not be able to cope up with the advent of new optical link rates and hence limits its performance. In the past, caching has not worked well in backbone routers because of the need to cache full address. This potentially dilutes the cache with hundreds of addresses that map to the same prefix. Besides, typical backbone routers may expect to have hundreds of thousands of flows to different addresses. The Wilder study [20] reports up to 240,000 concurrent flows, with less than 20 packets per flow. Short web transfers are likely reason for this behavior. Some studies have shown cache hit ratios of around 50 to 70 percent[21]. Thus, caching can help but does not avoid the need for fast lookups.

3.1 *Binary Decision Diagrams*

As is well-known, a Boolean function f: $B^n \rightarrow$ B can be represented by a Binary Decision Diagram (BDD), a directed acyclic graph obtained by applying an ordering constraint over the input variables and reduction operations on a binary decision tree, as proposed by Bryant [3]. The binary decision tree and the diagram for the function $f = x_0x_1 + x_1x_2 + x_2x_0$ are as shown in Fig. 1.

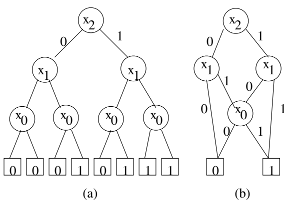

Figure 1: Function $f = x_0x_1 + x_1x_2 + x_2x_0$ represented as (a) Binary decision tree and (b) BDD.

Further more, the complexity of the BDD is dependent on its size, measured in the number of nodes. Hence, since a long time, one of the main research focuses has been to reduce the number of nodes in BDD representation. Reduction operations consist in eliminating redundant nodes from the binary tree. A node can be eliminated if

a. both the child nodes are equivalent.

b. there exists an other node at the same level in the decision tree and with equivalent high and low child nodes, respectively.

3.2 *Motivation*

The proposed scheme is motivated from two observations, first being that even at the largest Network Access Point, the number of next hop ports (NHPs) is generally not greater than 256. Hence, a next hop port (NHP) associated with any prefix in the routing table can be encoded using a 8-bit binary code. For example, any next hop port (NHP) in the MAE-east [1] routing table can be safely represented by a 6-bit binary code. Every bit of the output port can be computed by a combinational logic circuit whose optimal minimization is obtained with the help of Binary Decision Diagrams. The second observation is that the number of *effective nodes*, defined as the minimal number of nodes required to construct a binary decision tree in order to cover all the prefixes in the routing table is significantly smaller as compared to the upper bound on the theoretically required number of nodes. It is shown in the results section that, for the 32-bit IP address with the biggest available routing table of MAE-east [1], the number of redundant nodes is more than 99.99%. Thus constructing the binary decision tree with a fewer nodes and without any redundant nodes makes it very attractive for the application of Binary Decision Diagrams to optimize the logic. Besides, it is shown in the next sections that, while the upper bound on nodes increases exponentially with the IP address size, the number of *effective nodes* do not, making it an advantageous fact in view of the future implementation of IPv6 with the 128-bit IP address.

4 Details of the Scheme

For further understanding, please consider the routing table given in Table 1. The binary decision tree representation for the routing table is shown in Fig. 2. A node is assigned with the associated next hop port (NHP) if the path taken till that node from the root node forms a valid prefix. Please note that the root node is assigned with a default output port (in this case 0) as the length of the prefix at that node is zero. However, as mentioned earlier, the partial construction of the binary decision tree is sufficient to cover all the prefixes in the routing table resulting in only eight *effective nodes*. The redundant nodes, which can be conveniently ignored in the binary decision tree representation, are shown in dotted lines.

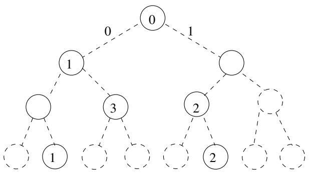

Figure 2: Binary decision tree for the sample routing table. Dotted nodes are redundant.

Now, the number of distinct next hop ports in this case being four, each next hop port (NHP) is encoded with a 2-bit binary code, NHP_1 and NHP_0 being the most significant (MSB) and the least significant (LSB) bits, respectively. When the ports are identified with the binary code, the binary decision tree representations for the NHP_0 and NHP_1 bits are as shown in Fig. 3. It can be observed that a further reduction in the number of *effective nodes* is obtained, the process of which is explained in detail in the subsequent subsection. Please note that any effective node without an output bit assigned to it, would inherit the output of its parent node.

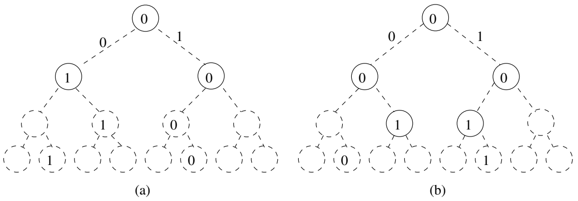

Figure 3: Binary decision tree for (a) NHP_0 (b) NHP_1, with all effective nodes assigned with output. Dotted nodes are redundant.

For the sake of convenience, let's assume that the n-bit IP address is represented by the binary variables x_{n-1}, x_{n-2}, ..., x_0 where x_{n-1} represents the MSB of the IP address. Now, applying the BDD algorithm on the binary decision trees of each bit of the output port, the BDDs for the functions are obtained to be as shown in Fig. 4.

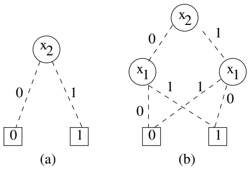

Figure 4: BDDs for (a)NHP_0. (b)NHP_1

4.1 *Reducing Effective Nodes*

When the output port is assigned to each of the node on the binary decision tree, with the further analysis, it is observed that the binary encoding of the output ports has given a further scope for the reduction in the number of nodes. For example, consider a situation where two leaf nodes, with a common parent, are assigned with output ports of 3 and 11, respectively. Suppose the parent node is assigned with an output port of 2. When the next hop port is encoded with a 4-bit binary code, it can be observed, as shown in Fig. 5, that a child node with the same output bit as its parent becomes redundant. The redundant nodes are shown in dotted lines in the figure.

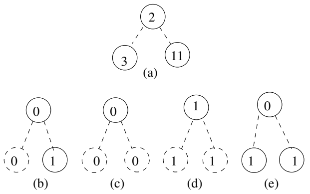

Figure 5: (a) Nodes with assigned NHP ports. (b),(c),(d),(e) Output bits assigned to each of the nodes in 4-bit binary encoding of NHP. Dotted nodes are redundant.

With the above procedure, it is shown that in each of the output bit representation, for the biggest available routing table of MAE-east [1] with 32-bit IP address, an additional 36% reduction is obtained in the number of effective nodes. This significant reduction in the number of effective nodes makes the application of Binary Decision Diagram approach, for obtaining the optimized logic, even more effective.

4.2 *Implementation Issues*

As mentioned earlier, the output interface ports at any router can be identified by at most an 8-bit binary code. Hence, for the above proposed scheme, the combinational logic design has to be done for eight output bits and hence that would give eight Binary Decision Diagrams to be processed. Each of the synthesized logic can be mapped into one or more Configurable Logic Blocks (CLBs) in an FPGA as shown in Fig. 6.

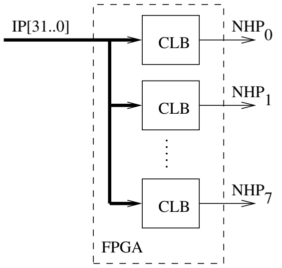

Figure 6: CLB mapping in FPGA.

4.2.1 *Timing Optimization*

Since the logic design obtained for the IP routing table is a combinational circuit, the timing optimization can be achieved using the *pipelining* and *retiming* techniques. Pipelining involves the insertion of delay elements at specific points of a circuit and retiming is the process of moving delays around a circuit such that the overall computation is unaltered. It aims to move a computation in an attempt to reduce the critical path, the path with the longest computation time without delays. By pipelining the computational data path, the throughput in terms of number of address lookups per unit time can be increased with a little or no additional cost in the overall area and latency.

4.3 *Routing Table Update*

As discussed in the earlier sections, the routing table update time is one of the important metrics to be considered for a scheme attempting the IP address lookup problem. In this scheme, we show that when there is an update in the routing table, then in most cases, not all of the logic blocks have to be recomputed, thus reducing the computational complexity. For example if a prefix 11* is inserted with an associated next hop port to be 1 into the routing table shown in Table 1, then the new routing table would be as shown in Table 2.

The binary decision tree for the modified routing table would be as shown in Fig. 7(a). It is obvious that there is no change in the BDD representation for the output bit NHP_1 while the slightly modified BDD representation for output bit NHP_0 is as shown in Fig. 7(b).

Table 2: Modified routing table.

$Prefix$	$length$	NHP
*	0	0
0*	1	1
01*	2	3
10*	2	2
11*	2	1
001*	3	1
101*	3	2

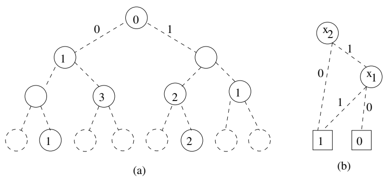

Figure 7: (a) Binary decision tree representation of the modified routing table. Dotted nodes are redundant. (b) Modified BDD for NHP_0.

However, this update of the logic may not be that simple for the 32-bit IP MAE-east routing table, but is also not as complex as assumed in general. To demonstrate this simplicity in updating the routing table, we have considered two consecutive snapshots of the MAE-east routing tables from [1] on April 15^{th} and 16^{th} 2001, with the number of prefixes 19477 and 19525, respectively. The analysis for the updating of the table is done in terms of the number of nodes at each level, in the binary decision tree for the latter routing table that differ in the output as compared to the corresponding nodes in the binary decision tree for the former routing table. Encouraging results have been obtained during this analysis and the results are as shown in the Table 3.

It is interesting to note that there is none or a significantly smaller variation in the output between the consecutive routing tables at the higher levels (level 0 to 15) and the lower levels (level 25 to 31). As is commonly known, the change in the logic would be minimal when the changes are minimum at higher levels in the binary decision tree, and we can observe that the same is the case in the current scenario. Further more, it can be observed that the number of nodes, in the levels 16 to 24, that differ in their outputs, are significantly smaller. Based on the observations, it can be safely concluded that, when the routing table is updated, there would only be a partial change in the combinational logic for each of the output bit. Thus the reconfiguration of only those logic segments, that need to have the updated logic, can be done. The commercial availability of partially reconfigurable FPGAs makes this update scheme even more attractive, where in, only those CLBs that have a modified design can be reconfigured leaving the remaining CLBs unaltered.

5 Results and Analysis

The number of effective nodes are obtained during the construction of the binary decision tree for a few sample routing tables with IP address lengths of 3, 5, 8 and 16 and for the real-time 32-bit IP MAE-east routing table. It is

Table 3: Number of corresponding nodes in each level of binary decision trees that differ in their output. The two binary decision trees compared are for adjacent snapshots of real-time MAE-east routing table.

Level	**Number of nodes**					
	NHP $_5$	*NHP* $_4$	*NHP* $_3$	*NHP* $_2$	*NHP* $_1$	*NHP* $_0$
0	0	0	0	0	0	0
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0
7	0	0	0	0	0	0
8	1	1	0	2	2	0
9	0	0	2	1	1	1
10	0	0	0	1	1	0
11	0	0	0	1	1	0
12	0	1	1	3	3	1
13	0	2	2	4	4	2
14	0	4	4	6	6	4
15	0	8	7	8	8	8
16	1	16	14	14	15	17
17	2	24	23	25	23	24
18	6	35	32	33	33	37
19	20	36	37	40	38	35
20	9	8	10	6	11	5
21	5	4	7	5	6	3
22	7	3	5	4	4	6
23	8	11	10	10	11	7
24	31	42	34	34	39	31
25	0	0	0	0	0	0
26	0	0	0	0	0	0
27	0	0	0	0	0	0
28	0	0	0	0	0	0
29	0	0	0	0	0	0
30	0	0	0	0	0	0
31	0	0	0	0	0	0

observed that the construction of the binary decision tree for the MAE-east routing table required only 91925 effective nodes, which is largely insignificant as compared to the theoretical upper bound of more than eight billion nodes. Further, when the output port is encoded with a 6-bit binary code and the reduction procedure is applied on each of the trees for individual binary output bits, the number of effective nodes obtained were only around 64% of the actual effective nodes. The summary of results is shown in Table 4. The recent research in logic optimization [22, 23] using BDDs has proved that the logic implementation, with a binary decision tree size of more than 100,000 nodes is done in less than a second. Subsequently, it is an encouraging factor when the routing table, with only around 50,000 effective nodes on average, is implemented as a combinational logic optimized using BDDs. Besides, another main advantage with the proposed scheme is that, for an n-bit binary encoding 2^n number of output ports can be represented and hence with an increase by one bit in the binary code, twice the current number of output ports can be represented. Thus, the proposed scheme proves to be more beneficial in the scenario that the number of physical ports in a router would increase continuously.

The combinational circuit obtained, prior to optmization using BDDs, from a partial MAE-east routing table, has been implemented in a device from the family of Altera FLEX10K FPGA. The subsequent observations indicate that an optimized combinational logic obtained from the routing table can be implemented in a single FPGA. Further more, the results from [24] show that the whole MAE-east routing table can be implemented in an FPGA with a circuit complexity of 51K gate counts. Thus, when the routing table is implemented in an FPGA, we can conclude that the IP address lookup rate is bounded only by the CLB delay. The maximum clock period bound for processing the IP address lookup would be the sum of 1 CLB delay and the maximum net delay. Previous hardware schemes have the lookup rate bounded by the RAM access speed. Further, in this scheme the resources required are utmost one FPGA while the other schemes require an ASIC and 3 or 4-bank RAM. Besides, those schemes do not provide an optimal solution for the routing table update.

5.1 *Scalability to IPv6*

To demonstrate that an optimized combinational logic can be obtained for a mapping between 128-bit long IP address of IPv6 and the binary encoded next hop port, it would be appropriate to show that the number of effective nodes that need to be processed by the BDD reduction techniques is significantly smaller than the theoretical upper bound. Fig. 8 plots the *effective ratio* with the varying length of IP address. The *effective ratio* is defined as the natural logarithm of the ratio of the number of effective nodes over the upper bound. It can be inferred that the number of effective nodes for the 128-bit long IP address is going to be well under the scope of combinational logic optimization process using the BDD techniques.

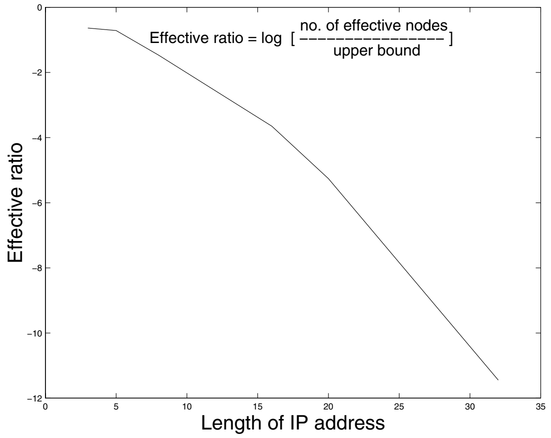

Figure 8: Effective ratio vs. IP address length.

6 Conclusion and Future Work

With the advancements in the communication link technologies the IP address lookup is becoming a major bottleneck in router technologies. We propose a reconfigurable hardware solution, using the well received concept of Binary Decision Diagrams(BDDs), that provides an efficient IP address

Table 4: Effective nodes for sample routing tables and the real-time 32-bit IP MAE-east routing table.

IP address length	Upper bound on nodes	Effective nodes before reduction	Effective nodes after reduction					
			NHP_5	NHP_4	NHP_3	NHP_2	NHP_1	NHP_0
3	15	8	-	-	-	-	5	3
5	63	31	-	-	-	20	18	20
8	511	116	-	-	85	87	81	85
16	$131 * 10^3$	3408	-	2098	2172	2181	2219	2164
32 (MAE-east)	$\approx$ 8.5Billion	91925	57955	58781	58577	58409	58387	58712

lookup along with providing a better scheme for updating the routing table. The argument, to support the adoption of BDD techniques for obtaining an optimized combinational logic, has been put forward by showing the fact that the number of effective nodes required to represent a 32-bit IP address routing table is significantly smaller than the theoretically required number of nodes. Besides, it has been shown that this number of effective nodes can be further reduced when the next hop port is represented with a binary code and a tree representation is obtained for each of the output bits.

Following the justification of the approach and proof that the processing time for the logic optimization is well under limit due to the emphatically smaller number of effective nodes, the next step is to obtain an implementation of the scheme in a partially reconfigurable FPGA and measure the lookup time and analyze the processing time to reconfigure the modified CLBs whenever the routing table is updated.

Acknowledgment

We would like to thank Yana Ong, Senior student, Department of Electrical and Computer Engineering, Iowa State University, for her simulation work as a support to our analytical study on IP routing tables.

References

[1] *Routing-Analysis, Internet performance measurement and analysis (IPMA) project* [online]. URL: http://www.merit.edu/ipma

[2] S. B. Akers, "Binary Decision Diagrams", *IEEE Transactions on Computers*, Vol. c-27, No. 6, June 1978, pp. 509-516.

[3] R. E. Bryant, "Graph-Based Algorithms for Boolean Function Manipulation", *IEEE Transactions on Computers*, Vol. c-35, No. 8, August 1986, pp. 677-691.

[4] K. S. Brace, R. L. Rudell, and R. E. Bryant, "Efficient implementation of a BDD package", *Proc. 27th IEEE/ACM Design Automation Conference*, 1990, pp. 40-45.

[5] Henry Hong-Yi Tzeng and Tony Przygienda, "On fast address-lookup algorithms," *IEEE Journal On Selected Areas In Communications*, Vol. 17, No. 6, June 1999, pp. 1067-1082.

[6] A. Przygienda, R. Shekhar, and R. Coltun, "Fast IP address lookup using multi-resolution trees," Fore Systems, Tech. Rep., Jan. 1997.

[7] T. Kijkanjanarat, H. J. Chao, "Fast IP lookups using a two-trie data structure " *Proc. Global Telecommunications Conference, GLOBECOM'99*, Vol. 2, 1999, pp. 1570-1575.

[8] N. Yazdani, P. S. Min, "Fast and scalable schemes for the IP address lookup problem" *Proc. IEEE Conference on High Performance Switching and Routing*, 2000, pp. 83-92.

[9] B. Lampson, V. Srinivasan, and G. Varghese, "IP Lookups using multiway and multicolumn search," *Proc. IEEE INFOCOM'98*, San Francisco, CA, 1998, pp. 1248-1256.

[10] A. Donnelly and T. Deegan, "IP route lookups as string matching" *Proc. 25th Annual IEEE Conference on Local Computer Networks, LCN*, 2000, pp. 589-595.

[11] W. Doeringer, G. Karjoth, and M. Nassehi, "Routing on longest-matching prefixes," *IEEE/ACM Trans. Networking*, Vol. 4, Feb. 1996, pp. 86-97.

[12] E. Filippi, V. Innocenti, and V. Vercellone, "Address lookup solutions for gigabit switch/router," CSELT, Tech. Rep., 1998.

[13] V. Srinivasan and G. Varghese, "Fast address lookups using controlled prefix expansion," *Proc. ACM Sigmetrics'98*, Madison, WI, 1998, pp. 1-11.

[14] S. Nilsson and G. Karlsson, "IP address lookup using LC-Tries", *IEEE Journal on Selected Areas in Communications*, Vol. 17, No. 6, June 1999, pp. 1083-1092.

[15] A. McAuley, P. Tsuchiya, and D. Wilson. "Fast multilevel hierarchical routing table using content-addressable memory." U.S. Patent serial number 034444, 1995.

[16] P. Gupta, S. Lin and N. McKeown, "Routing lookups in hardware at memory access speeds", *Proc. IEEE INFOCOM'98*, San Francisco, USA, Vol. 3, 1998, pp. 1240-1247.

[17] N. Huang, S. Zhao, J. Pan and C. Su, "A fast IP routing lookup scheme for gigabit switch routers", *Proc. IEEE INFOCOM'99*, New York, USA, Vol. 3, 1999, pp. 1429-1436.

[18] Pi-Chung Wang, Chia-Tai Chan, and Yaw-Chung Chen, "A fast IP routing lookup scheme", *Proc. IEEE International Conference on Communications, ICC 2000*, Vol. 2, 2000, pp. 1140-1144.

[19] T. Chiueh and P. Pradhan, "High-performance IP routing table lookup using CPU caching ", *Proc. IEEE INFOCOM'99*, New York, USA, 1999, Vol. 3, pp. 1421-1428.

[20] K. Thompson, G. J. Miller, and R. Wilder, "Wide-area Internet traffic patterns and characteristics." *IEEE Network*, Vol. 11, No. 6, Nov.-Dec. 1997, pp. 10-23.

[21] P. Newman, G. Minshall, and L. Huston, "IP switching and gigabit routers." *IEEE Communications Magazine*, Vol. 35, No. 1, Jan. 1997, pp. 64-69.

[22] R. Drechsler and W. Gunther, "Optimization of sequential verification by history-based dynamic minimization of BDDs." *Proc. IEEE International Symposium on Circuits and Systems, ISCAS 2000*, Geneva, Vol. 4, 2000, pp. 737-740.

[23] V. Bertacco and M. Damiani, "The disjunctive decomposition of logic functions." *Proc. IEEE/ACM International Conference on Computer-Aided Design, ICCAD'97*, 1997, pp. 78-82.

[24] Ilion Yi-Liang Hsiao, and Chein-Wei Jen, "A new hardware design and FPGA implementation for Internet routing towards IP over WDM and terabit routers." *Proc. IEEE International Symposium on Circuits and Systems, ISCAS 2000*, Geneva, Vol. 1, 2000, pp. 387-390.