High Speed Route Lookup for Variable-Length IP Address

Wanli Zhang, Xiangyang Gong, Ye Tian, Jifan Tang
State Key Laboratory of Networking and Switching Technology
Beijing University of Posts and Telecommunications
Beijing, China
{zhangwanli, xygong, yetian, tangjftom}@bupt.edu.cn

***Abstract*—Since the advent of the Internet, IP addresses have been the core of the Internet. However, with the rapid development of the Internet in recent years, IP addresses are facing more and more problems, such as address exhaustion, low packet efficiency and low flexibility. The reason is that IP addresses use a fixed-length design and lack extensibility. The New IP network architecture and addressing method were born to solve these problems. Based on this architecture, the addressing scheme adopts variable-length and structured addresses. The address space can be smoothly expanded according to the network scale without modifying the old network address configuration. But there are some challenges about New IP, and the greatest one lies in the route lookup of variable-length IP addresses. Content Addressable Memories (CAMs) are widely used in high speed routers to find matching routes for packets in a routing table. They enable the longest prefix matching on fixed-length addresses to be completed in a single clock cycle. However, they can not deal with New IP prefixes with variable lengths directly. In this paper, we propose a mechanism using Binary CAMs (BCAMs) and Ternary CAMs (TCAMs) to efficiently store New IP addresses and complete a route lookup in constant time. Moreover, we combine the hash scheme and CAMs matching scheme to shorten the extremely long New IP addresses and reduce TCAM storage space consumption. The simulation results show that our mechanism can provide high speed route lookup with low power consumption.**

***Index Terms*—New IP, Route Lookup, TCAM, BCAM, Hash**

I. Introduction

The data network has experienced rapid development for more than 40 years. As its core, Internet Protocol (IP) addresses are used to identify hosts and provide locations in the network. However, with the explosive growth of Internet and the emergence of multiple heterogeneous networks, the current IP address system is facing more and more problems.

The biggest problem is the address exhaustion of IPv4 address. By February 2011, the last IPv4 address were assigned and no more IPv4 addresses are available from the Internet Assigned Numbers Authority (IANA) [1]. Although the birth of IPv6 can temporarily solve the problem of address exhaustion, it has poor compatibility with IPv4. Moreover, IPv6 limits the address space, and will also fall into the dilemma that IPv4 once faced. In addition, the demand for heterogeneous network interconnection is increasing rapidly, especially in the Internet of Things (IoT) field. Most IoT devices have limited hardware resources. Short IP addresses are more suitable for them. Therefore, there is an urgent need to break the design constraints of fixed-length, delimited, and ordered network protocols.

978-1-7281-6992-7/20/$31.00 ©2020 IEEE

New IP is a new network protocol suite [2] [3]. Under this architecture, the protocol uses variable-length, structured address design [4]. Network addresses of different lengths will coexist in the data packet. The address space can be smoothly expanded according to the network scale without modifying the old network address configuration. Network interconnection and expansion do not depend on protocol conversion or address mapping gateway devices, making the network formation more flexible.

A New IP address consists of several address segments, as shown in Fig. 1. Each segment is a natural number without an upper limit theoretically. We use dots to separate different segments. For example, 53.17.319.106.228 is a New IP address with five segments.

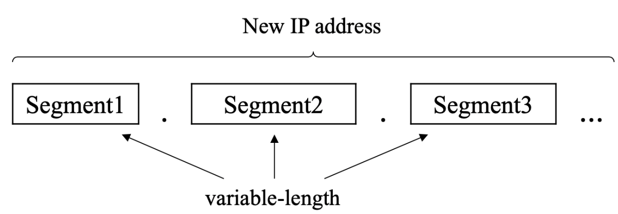

Fig. 1. Structure of the New IP address.

Devices in the same Local Area Network (LAN) differ only in the last segment. The gateway device of a LAN stores the same prefix of the devices in the LAN. Therefore, devices in the same LAN can use very short addresses for communication. Also, a device can communicate with devices in different LANs using a long address with more address segments. As shown in Fig. 2, only the last segment of the complete address is used for communication between devices A and B within the LAN. When the device A communicates with an external server, the destination IP address of the data packet is a complete address, and the source IP address still reserves the last segment. When data packets passes through the gateway device, it supplements the source IP addresses to complete addresses. And the data packets are routed to the server through the backbone network. Similarly, the destination IP addresses of data packets constructed by the

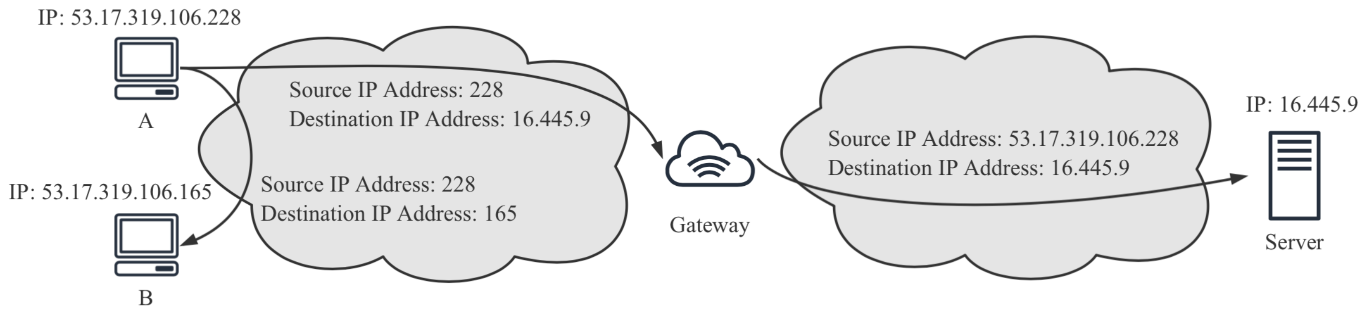

Fig. 2. New IP variable-length addresses communication.

server is complete addresses, and the source IP addresses only retain the last segment.

While New IP brings many conveniences, it also faces some challenges. One of them is a high speed and efficient routing table lookup algorithm. In New IP, the routing table stores New IP address prefixes of different lengths and corresponding next-hop addresses. When a data packet arrives, the router will perform the longest prefix matching (LPM) on the destination address carried in the packet to find the best next hop address. For example, a New IP address 1.2.3.4.5 consists of five address segments. It can match four address prefixes 1.*, 1.2.*, 1.2.3.*, 1.2.3.4.*, where * is a wildcard. The longest prefix 1.2.3.4.* is the result returned by the LPM algorithm. Compared with the route lookup of traditional IP addresses, New IP route lookup has the following two challenges:

- First, the length of the IPv4 address prefix is less than 32 bits, and the key (i.e. the destination IP address) input by the LPM algorithm must be 32 bits. Therefore, the IPv4 LPM algorithm only needs to handle IP addresses with a length of less than 32 bits. But the LPM algorithm of New IP has to deal with structured address with unfixed length.
- Second, New IP has enormous address space and good compatibility, resulting in many times more addresses than existing IP addresses. As the number of addresses increases, the size of the routing table will expand, resulting in a significant increase in router manufacturing and operating costs. Therefore, it is very challenging to design a New IP route lookup algorithm that is both high-speed and cost-saving.

In order to solve the above two challenges, we mainly made the following contributions in this paper:

1) Although the New IP address space is huge, the routing table of each router only stores a small part of it. By analogy with IPv4, the IPv4 address space is $2^{32} \approx 4 * 10^9$, and the routing table size of backbone routers is about $9{*}10^5$ [5]. Considering that the structured design of New IP addresses can make routes aggregated on a larger scale, we can believe the New IP address prefixes in the routing table to be quite sparse. Therefore, we propose a New IP address route lookup mechanism using Binary Content Addressable Memories (BCAMs) and Ternary Content Addressable Memories (TCAMs).
2) Since there is no upper limit for the length of New IP addresses theoretically, there may be some extremely long addresses. The algorithm proposed in 1) requires that the TCAM width should not be less than the length of the longest address, which results in a waste of TCAM storage space. Considering that most addresses are much shorter than the longest address, we can map these long addresses to a short address space through a hash function. Therefore, we propose a long address shortening method combining the CAMs matching scheme and the hash scheme to reduce TCAM storage space consumption.

The rest of this paper is organized as follow. In Section II we briefly review the traditional IP and name route lookup methods. Section III details the New IP address route lookup algorithm, as well as the hash scheme to shorten long addresses. In section IV, two measurements are employed to evaluate the proposed lookup algorithm from the perspective of time complexity and space consumption. Finally, we conclude this article in Section V.

II. Related Work

There are few researches on New IP route lookup today. Given that the content name in NDN [6] has a similar structure to New IP address, we can refer to the researches of name route lookup. Name route lookup algorithms are mainly divided into three categories: Trie-based algorithms, hash-based algorithms and hardware-based algorithms. The most Trie-based algorithms construct all entries in the routing table into a Trie [7] with component granularity. Some papers have proposed improved methods [8] [9], but the time required to complete a search still depends on the depth of the tree. When the lengths of names increase, the lookup performance decreases significantly.

Hash-based algorithms classify name prefixes by length [10] [11]. Name prefixes with the same number of components are placed in the same hash table. During route lookup, it searches each hash table in decreasing order of length until finding a matching prefix. To improve performance, Wang et al. [12] proposed a greedy name lookup method called Greedy-SPHT, which is combined with the string-oriented perfect hash table. The authors use a greedy strategy to choose the lookup order of name prefixes based on their length distribution. However, the hash-based algorithms will cause hash collisions when the routing table is getting larger, resulting in incorrect packet forwarding. And its time complexity is also related to the length of the name.

Most hardware-based algorithms use CAMs, which are divided into BCAMs and TCAMs. There can be two states in the memory cell of BCAMs: 0 or 1, so that exact match can be performed. In addition to 0 and 1, TCAMs memory cells can also have a third state called "don't care". A memory cell with "don't care" status can match both 0 and 1. Because of the existence of the third state, TCAMs can achieve the longest prefix matching. The architecture of TCAMs used for LPM is shown in Fig. 3.

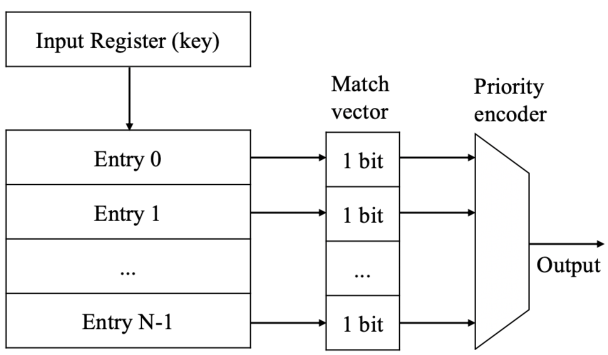

Fig. 3. Architecture of TCAMs used for LPM.

A key (i.e. destination IP address) is stored in the input register and each address prefix is stored in an entry of TCAMs. When matching, the key is compared with all prefixes in parallel and the results are stored in the match vector, where 1 represent the corresponding prefix that matches the key. Then the priority encoder selects the longest one among the prefixes that can be matched. At last, the output signal is used to find the corresponding outgoing port. A TCAM-based routing table takes very little time to perform a lookup because it allows the input key to compare all the entries stored in TCAMs at the same time in one clock cycle.

While the TCAM-based lookup is very fast, TCAMs have a major disadvantage: high power consumption. It results from the circuit complexity of each TCAMs memory cell. A typical TCAMs memory cell requires two SRAM memory cells to store both the value bit and the mask bit, and four transistors for the match logic. A typical SRAM memory cell requires six transistors, meaning that each TCAMs memory cell requires 16 transistors, which is about 2.7 times that of a typical SRAM memory cell.

Therefore, many solutions have been proposed to use TCAMs more efficiently and to reduce the required amount of TCAMs memory cells [13] [14]. Sun et al. [15] proposed a name-based longest prefix matching algorithm for information-centric networks using TCAMs. The algorithm uses a hash function to convert the name prefix into many fixed-length binary strings, which can be stored in fixed-width BCAMs and TCAMs for lookup. However, due to the need for multiple lookups based on the length of the name, the time complexity is difficult to meet the requirements.

Overall, the Trie-based and hash-based algorithms in the content name route lookup can be used in New IP routers that do not require high forwarding speed due to their low cost. Hardware-based algorithms is not suitable for New IP routing because the name is more complex and diverse than New IP address. In order to meet the high-speed and low-energy forwarding requirements of backbone routers, we need to design a dedicated route lookup algorithm for New IP.

III. New IP Lookup and Long Address Shorten

In this section, we first introduce the New IP address prefix storage and lookup algorithm that combines BCAMs and TCAMs. After that, we will present the long addresses shortening method using the hash function.

A. Storage and Lookup

Considering the structure of New IP addresses, we can not handle the complete address uniformly to achieve LPM. Therefore, we first split the New IP address into segments and set up a separate BCAM for each segment. For a BCAM, we assign a binary digital label for each entry from small to large. In this way, we rename a label for each segment address so that a long segment address can be replaced by a short label. After replacing each segment with a label, we splice these labels in the order of the segments. Finally, a complete binary digital label address is obtained and can replace the original New IP address unambiguously.

A problem needs to be pointed out is that there are obvious separators "." between the segments of a New IP address, and the label addresses are all represented by binary numbers. We need to separate the labels of different segments to avoid ambiguity. For BCAMs, we set the length of each label in the same segment to the same. But the label length of different segments can be different, depending on the number of stored entries in the BCAM. As the routing table is updated and the number of entries changes, the label length of each segment will also change to accommodate a larger number of entries or shorten the width of CAMs.

In this way, each segment can be separated in the label address without the need for a separator. Through BCAMs, we replace the hierarchical New IP address with a flat label address. We can specify the mask length of a label address prefix as the length of it. By storing label address prefixes in the TCAM in order of their mask lengths, the storage of New IP address prefixes is completed as shown in Fig. 4. Here is an example to further illustrate the prefix storage method.

Suppose there are three New IP address prefixes in the routing table: 23.61.147.*, 49.33.*, 82.*. So there are three entries in the BCAM storing the first segment address: 23, 49, 82, and the corresponding labels are 00, 01, 10, respectively. There are two entries in the BCAM storing the second segment

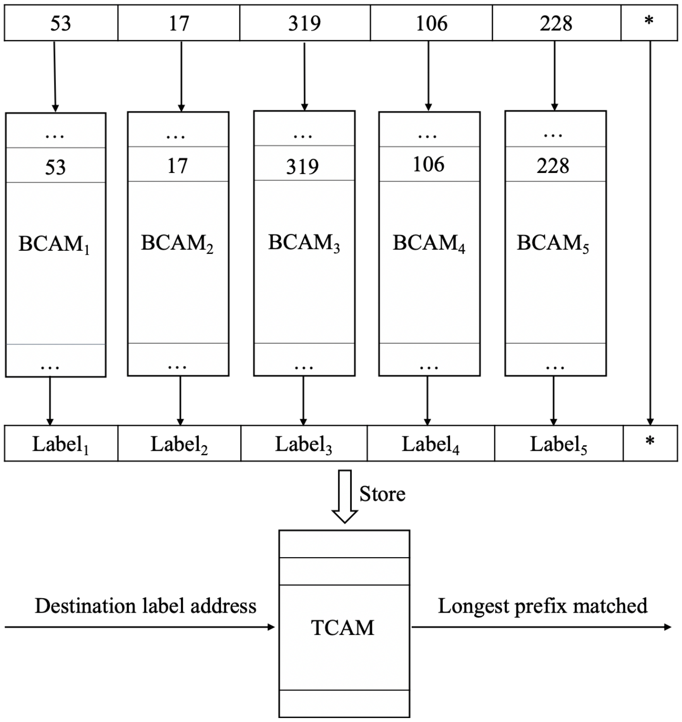

Fig. 4. New IP address prefixes storage and lookup.

address: 61, 33, corresponding to labels 0 and 1, respectively. There is one entry in the BCAM storing the third segment address: 147, corresponding to label 0. The New IP address prefix 23.61.147.* is replaced by the label 0000/4 through BCAMs, where /4 represents the mask length. The prefix 49.33.* is replaced by label 011/3, and the prefix 82.* is replaced by label 10/2. After that, we can store these label address prefixes in the TCAM.

When a New IP data packet needs to be forwarded, the destination address is also split by segment. The corresponding BCAMs are queried for each segment address to get label address. And the label addresses of each segment are spliced into the complete label address in order. Finally, the LPM result can be obtained by entering the complete label address into the TCAM.

However, if a certain segment address fails to match in the BCAM, we can use a specific unassigned label to represent the segment without matching (called the unmatched label). This method not only ensures the correctness of the LPM result, but also saves the storage space of BCAMs and TCAMs.

For example, for a BCAM located in the k-th segment of the New IP address with a label length of three, we set 111 as unmatched label. For a destination address with the number of segments $m(m>k)$, if the k-th segment address does not match in the BCAM, there is no entry in the routing table with the same k-th segment as the destination address. Therefore, the number of segments of the prefix that the destination address can match successfully in the routing table must be less than k. When the destination address is converted to a label address, the label in the k-th segment is 111. The k-th segment of all label address prefixes with the number of segments greater than or equal to k stored in the TCAM is not 111. Therefore, the destination label address used as the TCAM input key can not match these label address prefixes whose number of segments is greater than or equal to k. For those label address prefixes with the number of segments less than k in the TCAM, the memory cells of the k-th segment address are “don’t care”, which can match the destination label address normally. We can conclude that the LPM result output by the TCAM is a prefix whose number of segments is less than k.

There is a problem that needs to be pointed out. Since the third state “don’t care” in TCAMs memory cells can match any input, if there is no input entered into the “don’t care” memory cell, TCAMs will determine that the match is successful. According to the LPM rule, the longest prefix that the address 157.643.72 should be matched of is 157.643.*. But if there is a prefix 157.643.72.* in the routing table, the LPM result output by TCAMs is 157.643.72.* incorrectly. Our solution is to delete the last segment of the destination address before lookup in BCAMs. After obtaining the label address, we fill in the unmatched label of the next segment after the last segment. In this way, the address 157.643.72 can only match the prefixes of no longer than two segments.

In summary, we have achieved the route lookup of New IP addresses through the above method of combining BCAMs and TCAMs. But TCAMs consume a lot of power, which is proportional to the number of memory cells. We need to reduce the consumption of TCAM storage space as much as possible to save costs.

B. Long Address Shorten

Since the lengths of the New IP addresses have a wide range, using TCAMs that can accommodate the maximum length of the New IP address prefix to store all the entries will result in a great waste of TCAM storage space. Our initial idea is to use two TCAMs. If the length of the binary label address converted by a New IP address prefix does not exceed the width of the first TCAM (called $TCAM_1$), it will be stored directly in $TCAM_1$. If the length exceeds the width of $TCAM_1$, we can use another wider TCAM (called $TCAM_2$) to store these long prefix entries separately. When the length of the destination label address is not greater than the width of $TCAM_1$, only $TCAM_1$ needs to be queried. Otherwise, both of the them need to be queried to ensure that the destination address can match the prefix entry with any length in the routing table. Obviously, the length of all prefixes in $TCAM_2$ is larger than that of $TCAM_1$. Therefore, if there is a matching prefix in $TCAM_2$, the prefix is the LPM result. Otherwise, the matching prefix of $TCAM_1$ is output.

This method reduces the use of TCAM storage space to a certain extent, but it can be further optimized for $TCAM_2$. Considering that the number of these long addresses will not be very large, we can use an appropriate hash function to calculate the portion of the label address that does not exceed the width of $TCAM_1$ into a short value. Then this value is concatenated as the previous part with the excess part of the label address and stored in $TCAM_2$.

When the destination address in a data packet is converted into a binary label address and the length does not exceed the width of $TCAM_1$, only $TCAM_1$ needs to be queried. Otherwise, we split out the part of the destination label address that does not exceed the width of $TCAM_1$ to query $TCAM_1$. At the same time, the destination label address is processed based on the same method as the long address prefix, and is queried in $TCAM_2$. If $TCAM_2$ has no matching result, the LPM result is the matching result of $TCAM_1$. Otherwise it is the matching result of $TCAM_2$.

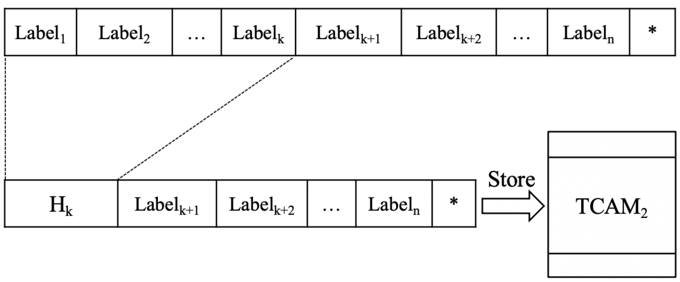

Fig. 5. Long address prefixes storage.

Suppose the width of $TCAM_1$ is represented by W_1, and the width of $TCAM_2$ is represented by W_2. The i-th segment of the label address is represented by $Label_i$, and the length of $Label_i$ is represented by L_i. As shown in Fig. 5, for a n-segment prefix that has been converted into a label address, assume that the sum of the lengths of the first $k(k<n)$ segments is less than or equal to W_1 and the sum of the lengths of the first $k+1$ segments is greater than W_1. Formulated as:

$$\sum_{i=1}^{k} L_i \leq W_1, \quad \sum_{i=1}^{k+1} L_i > W_1$$

We hash the entire first k segments label address into a shorter binary label H_k, splice $Label_{k+1}$ to $Label_n$ in the latter part, and store it in $TCAM_2$. $TCAM_1$ only stores prefix entries with a number of segments less than or equal to k.

For a destination label address with $m(m>k)$ segments as shown in Fig. 6, it can also be determined that the sum of the lengths of the first k segments is less than W_1 because the length of each segment label address is fixed. We hash the first k segments into H_k in the same way as above, and splice $Label_{k+1}$ to $Label_m$ behind H_k to be queried in $TCAM_2$. In this way, the destination address can be compared with all prefixes in the routing table whose number of segments is greater than k. Simultaneously, we split out the label address $Label_1$ to $Label_k$. And it is queried in $TCAM_1$ to be compared with all prefixes whose number of segments is less than or equal to k. Obviously, the number of prefix segments stored in $TCAM_2$ must be greater than that of $TCAM_1$. If there is a matching result in $TCAM_2$, it is output as the LPM result. Otherwise the result of $TCAM_1$ is output.

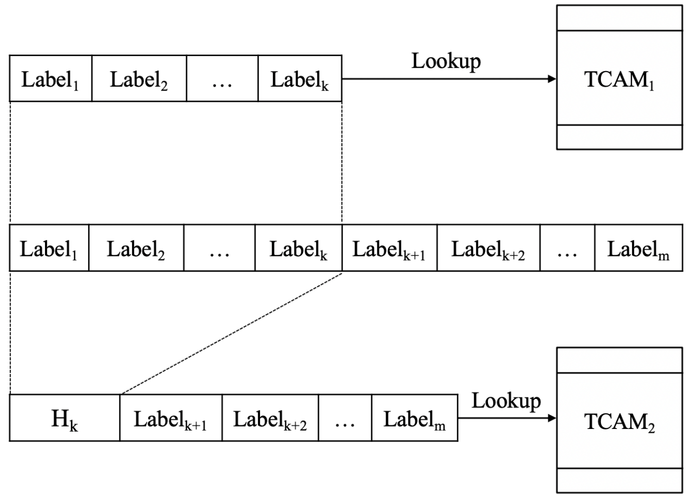

Fig. 6. Long addresses lookup.

IV. Evaluation

In this section, we evaluate the performance of the proposed New IP route lookup mechanism from two perspectives: lookup latency and TCAM storage space consumption.

A. Lookup Latency

The basic unit of lookup latency here is the clock cycle, which is the duration from the beginning of entering a key into a CAM to the end by returning a matching result. For normal addresses lookup, it takes one clock cycle to lookup in BCAMs getting the label address, and another one for lookup in the TCAM to obtain the LPM result. The lookup latency should be equal to two clock cycles. It is worth mentioning that our lookup method can apply pipeline technology. Therefore, the New IP route lookup mechanism can obtain a LPM result every single clock cycle.

As for long addresses lookup, the time of hash calculation needs to be added on the basis of two clock cycles. The hash calculation can be realized by hardware, so it will not take much time [16]. Since the long address is only a small part, it can be ignored.

B. TCAM Storage Space Consumption

We estimate the relationship between the size of the routing table and the width of TCAMs. The previous work introducing New IP proposed that the number of address prefix segments of a subnet represents the level of the network domain where the subnet is located [4]. The network domain can be determined according to the geographical administrative level. For example, there are 8 levels based on continent, country, province, city, county (district), street (town), building, and room. The first segment of a New IP address indicates the number of a continent. The second segment indicates the number of a country within the continent represented by the first segment, and so on. Therefore, we can assume that the number of segments of most New IP addresses is less than 8. We do not exclude that there are some addresses with more than 8 segments, but we think these long addresses are very few. Taking into account the actual situation, the expansion factors from the first level to the eighth level are 8, 64, 32, 16, 16, 32, 1024, and 512 respectively. The expansion factor represents the number of network domains of the next level expanded from a network domain. The number of subnets that can be allocated is $8*64*32*16*16*32*1024*512+8*64*32*16*16*32*1024+...+8 \approx 2^{46}$. The number of IPs that can be allocated is $2^{46}*\lambda \approx 2^{50}$ (λ is the average number of devices in each subnet), far exceeding the IP demand today.

According to the data released by CIDR Report [5], the number of BGP routing table entries for IPv4 in the backbone router is approximately 900,000, and the number of entries for IPv6 is approximately 100,000. We randomly generate 1,000,000 New IP address prefixes with different number of segments as routing table entries. Each segment of a prefix is randomly generated from 0 to the expansion factor of the segment. As the generated prefixes gradually increase, we calculate the width of the $TCAM_1$ required to store them. The storage space of $TCAM_2$ is negligible due to its small number of entries. In this way, an image between the number of routing table entries and the width of the $TCAM_1$ is drawn. The TCAM width used by the algorithm in [15] is constant at 64, and the width required for IPv6 route lookup is always 128, which is not shown in the figure.

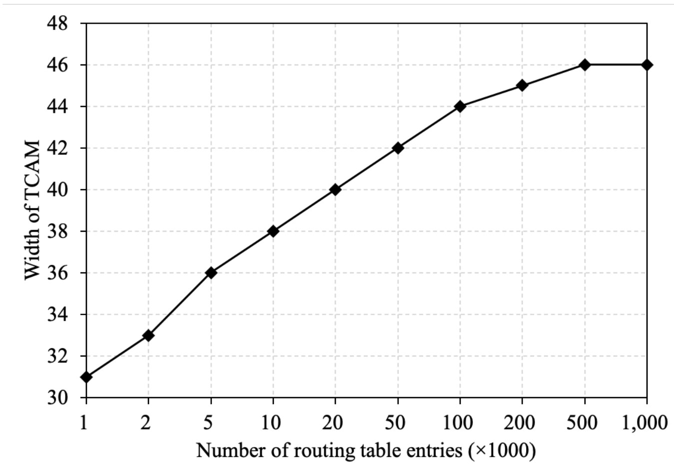

Fig. 7. Storage space consumption of TCAMs.

From the Fig. 7, we can conclude that the router can choose appropriate TCAM width based on the size of its routing table. In addition, the New IP routing table can be aggregated on a much larger scale than IPv4 because the New IP address adopts a structured design and the address allocation is on the basis of the geographical location.

V. Conclusion

In this paper, we propose a high speed New IP route lookup mechanism to achieve LPM using a hybrid configuration of BCAMs and TCAMs. BCAMs are used to convert structured New IP addresses into flat label addresses. TCAMs can perform the LPM of label addresses efficiently. We combine the hash scheme and CAMs matching scheme to shorten the extremely long address and reduce TCAM storage space consumption. Our lookup mechanism can be pipelined so that only one clock cycle is needed to complete a LPM. Analysis and the experiment result show that when the number of New IP that can be allocated far exceeds the current IP demand, only a small TCAM storage space is needed to store the routing table of the backbone router.

Acknowledgment

This work was supported in part by the National Key R&D Program of China under Grant 2019YFB1802603 and in part by the National Natural Science Foundation of China under Grant 61802024.

References

[1] 2011. Free Pool of IPv4 Address Space Depleted. Available:https://www.nro.net/ipv4-free-pool-depleted/.

[2] X. Zheng, S. Jiang, and C. Wang, “NewIP:new connectivity and capabilities of upgrading future data network,” Telecommunications Science, 2019, 35(9): 2-11.

[3] Z. Chen, C. Wang, G. Li, Z. Lou, S. Jiang and A. Galis, “NEW IP Framework and Protocol for Future Applications,” NOMS 2020 - 2020 IEEE/IFIP Network Operations and Management Symposium, Budapest, Hungary, 2020, pp. 1-5.

[4] J. Tang, W. Zhang, X. Gong, G. Li, D. Yu, Y. Tian, B. Liu, and L. Zhao, “A flexible hierarchical network architecture with Variable-Length IP address,” in 2020 IEEE INFOCOM WKSHPS: New IP: The Next Step, Toronto, Canada, Jul. 2020.

[5] CIDR-Report. Available:https://www.cidr-report.org/as2.0/.

[6] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, K. Claffy, P. Crowley, C. Papadopoulos, L. Wang, and B. Zhang, “Named data networking,” ACM SIGCOMM Computer Communication Review, vol. 44, no. 3, pp. 66-73, 2014.

[7] Y. Wang, H. Dai, J. Jiang, K. He, W. Meng and B. Liu, “Parallel Name Lookup for Named Data Networking,” 2011 IEEE Global Telecommunications Conference (GLOBECOM), Houston, TX, USA, 2011, pp. 1-5.

[8] D. Li, J. Li and Z. Du, “An improved trie-based name lookup scheme for Named Data Networking,” 2016 IEEE Symposium on Computers and Communication (ISCC), Messina, 2016, pp. 1294-1296.

[9] S. Ren, D. Yu, G. Li, S. Hu, Y. Tian, X. Gong, and R. Moskowitz, “Routing and addressing with length variable ip address,” in Proceedings of the ACM SIGCOMM 2019 Workshop on Networking for Emerging Applications and Technologies. ACM, 2019, pp. 43–48.

[10] W. So, A. Narayanan and D. Oran, “Named data networking on a router: Fast and DoS-resistant forwarding with hash tables,” Architectures for Networking and Communications Systems, San Jose, CA, 2013, pp. 215-225.

[11] H. Yuan and P. Crowley, “Reliably scalable name prefix lookup,” 2015 ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS), Oakland, CA, 2015, pp. 111-121.

[12] Y. Wang, B. Xu, D. Tai, J. Lu, T. Zhang, H. Dai, B. Zhang, and B. Liu, “Fast name lookup for Named Data Networking,” 2014 IEEE 22nd International Symposium of Quality of Service (IWQoS), Hong Kong, 2014, pp. 198-207.

[13] H. Noda et al., “A cost-efficient high-performance dynamic TCAM with pipelined hierarchical searching and shift redundancy architecture,” in IEEE Journal of Solid-State Circuits, vol. 40, no. 1, pp. 245-253, Jan. 2005.

[14] Y. Sun and M. S. Kim, “A Hybrid Approach to CAM-Based Longest Prefix Matching for IP Route Lookup,” 2010 IEEE Global Telecommunications Conference (GLOBECOM), Miami, FL, 2010, pp. 1-5.

[15] Y. Sun, N. Egi, G. Shi, and J. Wu, “Content-based route lookup using CAMs,” 2012 IEEE Global Communications Conference (GLOBECOM), Anaheim, CA, 2012, pp. 2677-2682.

[16] F. Yamaguchi and H. Nishi, “Hardware-based hash functions for network applications,” 2013 19th IEEE International Conference on Networks (ICON), Singapore, 2013, pp. 1-6.