# Non-random Generator for IPv6 Tables

Mei Wang, Stephen Deering, Tony Hain, Larry Dunn
Advanced Architecture Group
Cisco Systems
{anitwang, deering, ahain, ldunn}@cisco.com

*Abstract*— **The next generation Internet Protocol, IPv6, has attracted growing attention. The characteristics of future IPv6 routing tables play a key role in router architecture and network design. In order to design and analyze efficient and scalable IP lookup algorithms for IPv6, IPv6 routing tables are needed. Analysis of existing IPv4 tables shows that there is underlying structure that differs greatly from random distributions. Since there are few users on IPv6 at present, current IPv6 table sizes are small and unlikely to reflect future IPv6 network growth. Thus, neither randomly generated tables nor current IPv6 tables are good benchmarks for analysis. More representative IPv6 lookup tables are needed for the development of IPv6 routers. In this paper, from the analysis of the current IPv4 tables, algorithms are proposed for generating IPv6 lookup tables. Tables generated by the methods suggested here exhibit certain features characteristic of real lookup tables, reflecting not only new IPv6 address allocation schemes but also patterns common to IPv4 tables. These tables provide useful research tools by a better representation of future lookup tables as IPv6 becomes more widely deployed.**

## I. Introduction

Due to the fast growth of the Internet, the shortage of IP addresses is becoming a more pressing issue. This has brought growing interest in the next generation internet protocol, known as IPv6 [1]. With 128-bit address, IPv6 provides $3.4\times10^{38}$ addresses theoretically. It has been gaining wider acceptance [21] to replace its predecessor, IPv4. IPv6 has already emerged out of the testing phase [6] and is seeing early deployment in Europe, Asia, and North America [22].

The longer address length and larger address space in IPv6 pose a challenge to existing internet routers. Address lookup schemes with good scalability in memory consumption as well as lookup and update performances need to be developed. In order to evaluate lookup algorithms, IPv6 routing tables are needed. Router vendors today need a good model of the structure of IPv6 tables. Since IPv6 is not yet widely deployed, existing IPv6 tables are small [5] and unlikely to reflect future IPv6 network growth. Currently, tables generated randomly are often used for IPv6 research and development. However, analysis of existing IPv4 tables shows that they are far from random distributions [4]. Thus, neither current IPv6 tables nor randomly generated tables are good benchmarks for analysis. More representative IPv6 lookup tables are needed for the development of IPv6.

The size and structure of internet routing tables are the result of the address allocation schemes and routing practices. They directly impact the performance of lookup algorithms implemented in internet routers. Analysis of existing IPv4 routing tables reveal many of the features that can help guide address allocation policies for more efficient utilization of address space. The development of better lookup algorithms can also benefit from identifying these features. This paper presents new measures, such as bit entropy, to study the hierarchical structure of IPv4 routing tables.

Although IPv6 allows better addressing structure and provides enhancements over IPv4, many of the features observed in IPv4 routing tables are expected to emerge in IPv6 routing tables. This is due to three main reasons: allocation policies, routing practices, and the evolution of the Internet. These are the three key areas that affect the formation of the IP table structure. First of all, the allocation policies for IPv6 follow the similar fundamentals of IPv4 allocation, despite the differences in detailed rules. Secondly, the overall network topological distribution, which affects routing, is expected to be intact through IPv4 to IPv6 migration. Thirdly, the natural evolution of the Internet will continue. The business relationship among IP prefix providers and the customers remain the same: the same companies that offer IPv4 services will provide IPv6 services. Thus, we expect that similar structures will be developed according to IPv6 allocation schemes for the IPv6 Internet.

The overall structure of a routing table can be represented by three major aspects: the overall table size, prefix value distribution, and prefix length distribution. In this paper, several different IPv6 table generation schemes are proposed based on existing IPv4 tables to inherit the key hierarchical properties that are the results of years of evolution. Although all these schemes are based on mapping IPv4 prefixes of up to 32 bits to IPv6 prefixes of up to the first 64 bits, only a few preserve the original hierarchical structure. The structure is reflected in the randomness of prefix values, the parent-children relationship, and the sparseness of the address tree.

The rest of the paper is organized as follows. In section II, existing IPv4 tables are analyzed to extract common features. IPv6 table generation schemes are introduced and examined in section III. The conclusions are given in section IV.

## II. IPv4 Table Analysis

We propose several different ways of analyzing existing IPv4 tables to capture the structures that might emerge in IPv6. The study of a series of IPv4 tables [8] over the past few years shows that, despite the growth in size with time, the recent tables share certain common characteristics. In this section, we present the results from one table to illustrate these

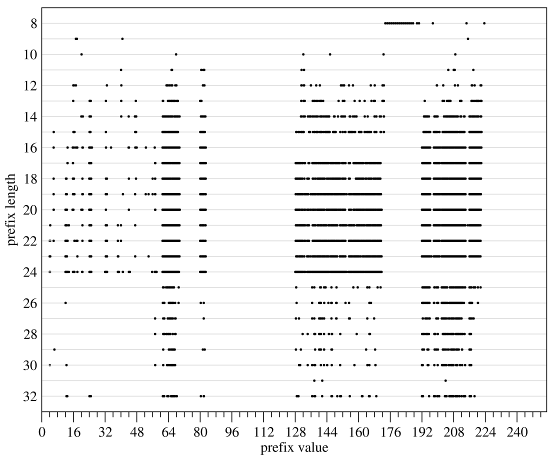


Fig. 1. Prefix distribution in an IPv4 table from November 2003. The value of the first byte is labeled on the x-axis.

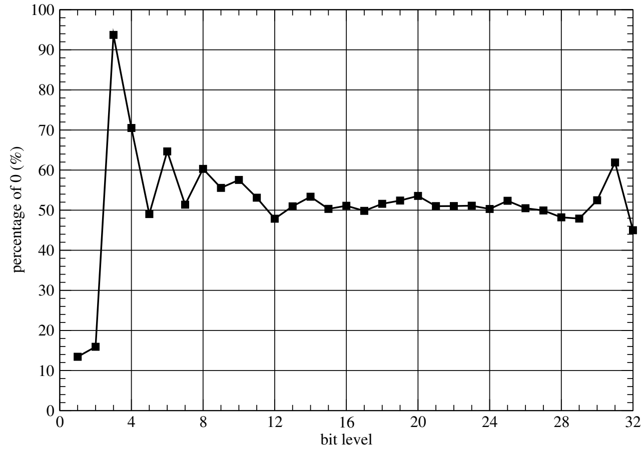


Fig. 2. IPv4 table prefix value distribution of each bit. For each bit position in the prefixes, the percentage of the value being 0 is plotted as the y-value. From this probability, the entropy of each bit can be calculated, which gives the randomness of the prefix value.

common features. The majority of the data analyzed here is obtained from a November 2003 table [8].

A tree structure formed by the prefixes is an effective means to illustrate and analyze the distribution and relationship of the prefixes. There are numerous tree-based lookup algorithms that construct trees with specific configurations to achieve efficient performance. In this paper, we focus on the intrinsic characteristics of the tables that are independent of lookup algorithms. Thus, the tree discussed in this work is the original binary tree resulting from the relationship of the prefixes in the address space. This tree is referred to as the address tree throughout the paper.

In this section, we first analyze the overall distribution of prefixes. Next we examine two of the three major aspects pointed out in section I: prefix length and prefix value. There have been many presentations of data and charts on the growth of IPv4 table size over the years [12]. We are not going to discuss it in detail here. The size of IPv6 tables will be discussed in section III.B.

### *A. Overall prefix distribution*

The three major aspects all contribute to the overall prefix distribution. Each single prefix consists of prefix length $l$ – the number of significant bits to match while searching (the rest of the bits are "don't care bits") and prefix value – the value of these first $l$ bits. An address space can be represented by an address line. For example, the 32-bit IPv4 address space can be denoted by an address line ranging from 0 to $2^{32}$-1. A prefix covers a range in the address space with a starting and ending point on the address line. How are IPv4 prefixes distributed in the IPv4 address space? Figure 1 gives an example of a real IPv4 table. X-axis represents the full 32-bit address space with only the value of the first byte being labeled. Y-axis labels the prefix length, which illustrates the distribution of prefix lengths for a given address value.

There are clear gaps with no prefixes around the same prefix values in all the tables we analyzed. For example, no prefixes exist in the range of 83-127, which means that, in the routing table, there are no prefixes that start with the first byte value between 83 and 127. In the recent tables, new prefixes are slowly emerging in these blank areas thereby narrowing the gaps. The prefixes around the first byte value of 80 are fairly recent additions to the tables. There are also several /8 prefixes between the first byte values of 173-190. However, there are no prefixes with longer prefix lengths underneath them. These prefixes are new additions as well. The cause of the gaps is that these address areas are either inactive or never allocated in the Internet. This can also be due to administrative issues or special reservations of address registries. The blank gaps also suggest imbalance in various parts of the address tree. The current topology of the IPv4 internet and the structure of IPv4 tables are the result of allocation policies, routing practices, and natural evolution. For various historical reasons, there is inefficient usage of the address space, which is partially reflected in the unbalanced address tree. This can be improved in the new allocation schemes and address management policies for IPv6 to achieve more efficient utilization of the address space.

### *B. Prefix value and entropy*

A prefix value consists of the value of each bit within the prefix length. At each bit (up to the prefix length), counting the number of 0's of all prefixes in a routing table yields the value distribution of that bit. The probability of each bit being 0, $p$, is displayed in Figure 2. With this probability, the entropy of each bit is calculated in Table I with the following definition:

$$H(p) = -p \log p - (1-p)\log(1-p). \tag{1}$$

Entropy provides a measure of the randomness of prefix values. It reaches its maximum when the bit value is equally distributed between 0 and 1, i.e., $p$=1/2. There are very few distinct first byte values among all the prefixes, all the branches

TABLE I

ENTROPY OF EACH BIT (IPV4).

| Bit Number | Bit Entropy | Bit Number | Bit Entropy | Bit Number | Bit Entropy | Bit Number | Bit Entropy |
|---|---|---|---|---|---|---|---|
| 1 | 0.569265 | 9 | 0.991091 | 17 | 0.999991 | 25 | 0.998418 |
| 2 | 0.63251 | 10 | 0.983544 | 18 | 0.99927 | 26 | .999933 |
| 3 | 0.33859 | 11 | 0.997208 | 19 | 0.998346 | 27 | 0.999998 |
| 4 | 0.875105 | 12 | 0.998687 | 20 | 0.996342 | 28 | 0.999074 |
| 5 | 0.999726 | 13 | 0.999733 | 21 | 0.999707 | 29 | 0.998735 |
| 6 | 0.937066 | 14 | 0.996751 | 22 | 0.999698 | 30 | 0.998237 |
| 7 | 0.999462 | 15 | 0.99997 | 23 | 0.999643 | 31 | 0.958712 |
| 8 | 0.969196 | 16 | 0.999651 | 24 | 0.999975 | 32 | 0.992774 |

are spawned off these values. That is why there is higher probability that the distribution in the first few bits is skewed. The bits with $p$ around 50% are more evenly distributed. For example, there are a large number of prefixes that contribute to distinct values at bit level 24. Thus it is expected that the values are well evened out to approach maximum entropy.

The skew of the first few bits also shows the inefficiency of address allocation and utilization. The address tree is unbalanced with more weight on one side. This is consistent with our observation from Figure 1. A more balanced distribution of 0 and 1, especially for the first few bits, is important to achieve more efficient usage of the address space.

### C. Prefix length and parent-children relationship

The address allocation schemes determine the number of prefixes to be allocated for each length. Another key contribution to the prefix length distribution in a routing table is how these allocated prefixes are utilized in the network. An allocated prefix can exist in a routing table in its original form, or as a less specific prefix (with a shorter prefix length) aggregated with other prefixes, or as a split into multiple more specific prefixes (of longer prefix lengths).

Due to these aggregations and splits, prefixes in a table are not independent of each other. Some of them are supersets of others. This is called parent-children relationship. A child covers a subset of the range that its parent spans on the address line. Thus, for any address point within the range of the child, both the child and its parent have matching prefixes. To determine which prefix to take when there are multiple matches in a routing table, longest prefix matching (LPM) [11] is needed, i.e., choosing the prefix with the longest prefix length among all the matching entries. In other words, if both the parent and its child match the address of an incoming packet, the child is the LPM. In terms of the parent-children hierarchy, the parent is at level 0 in the address tree and the child is called the first level child of that parent. Similarly this first level child can have its own children, which would be grandchildren of the original parent. They are called the second level children of the original parent, and so on. Prefixes without any parent or children are called stand alone prefixes.

For each parent-children hierarchy level, the percentage of prefixes distributed at different prefix lengths is shown in Figure 3. For example, 38.8% of the total number of prefixes in the table are stand alone prefixes. Their lengths distribution is shown in Figure 3(a). The rest of the prefixes are mainly concentrated at level 0 to level 2 of the parent-

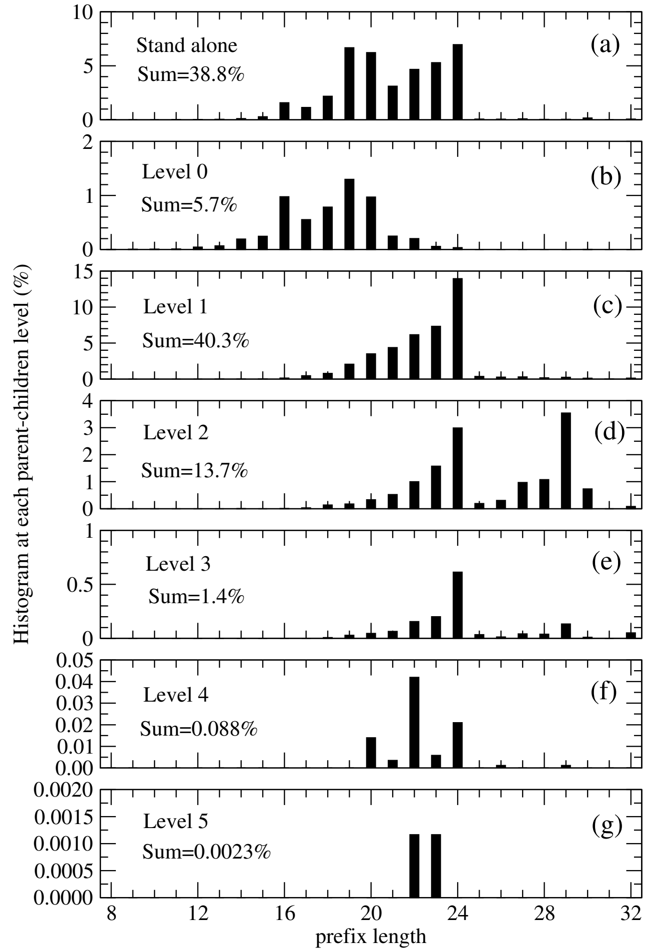


Fig. 3. Prefix length distribution of different parent-children levels.

children hierarchy. There are very few prefixes at level 4 and level 5, as shown in Figure 3(f)(g). From level 0 to level 2, the profile of the prefix length histogram shifts towards the right, i.e., to higher prefix lengths. For example, the number of prefixes at parent-children level 0 (parents) peaks at prefix length 19 in Figure 3(b). In comparison, the number of prefixes at level 2 peaks at lengths 24 and 29, extending to 32 in Figure 3(d).

Another interesting observation about the prefix length distribution of the current IPv4 tables is that only about 1/4 of prefixes have odd prefix lengths. The domination of even prefix lengths is probably due to the fact that the original allocation is mainly on even numbers and that odd lengths are largely created by aggregations and splits.

## III. IPV6 TABLE GENERATION

Allocation policies directly affect the structure of IP lookup tables. In this section, the current policies for IPv6 are described. Next we propose several schemes for IPv6 table generation with comparisons. Existing IPv6 tables are also discussed.

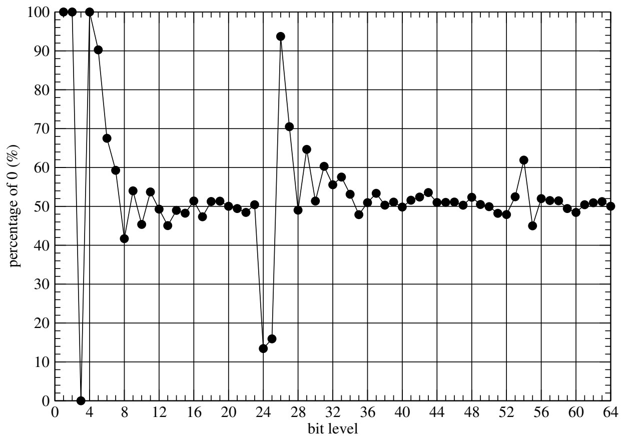


Fig. 4. Synthesized IPv6 table prefix value distribution at each bit.

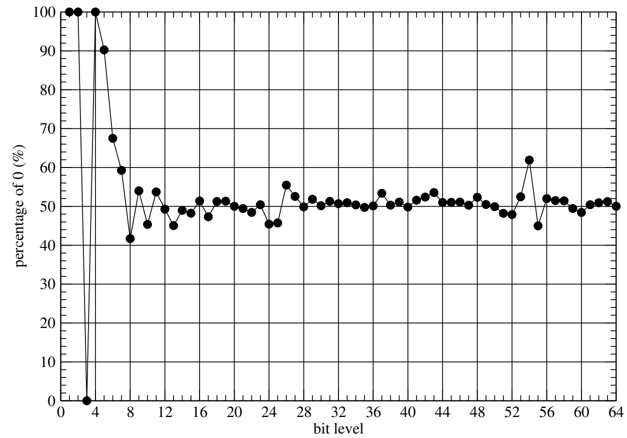


Fig. 5. Synthesized IPv6 table prefix value distribution at each bit after modification.

### A. Current IPv6 address allocation schemes

Because of the large size of the available address space in IPv6, a hierarchical structure of the allocation space is necessary to permit the aggregation of routing information and also to limit the expansion of Internet routing tables.

Under the current IPv6 Address Allocation and Assignment Policy [18–20], the Regional Internet Registries (RIR), receive /23's from the Internet Assigned Numbers Authority (IANA). In turn, the RIRs assign addresses to Local Internet Registries (LIR) or ISPs with a minimum allocation size of /32. In Asia, an extra layer of National Internet Registry (NIR) exists between the RIR and the LIRs. In the general case, end users, such as different organizations and small ISPs, are assigned /48's from the LIR/ISPs. /64 is assigned when it is known that one and only one subnet is needed and /128 is assigned when it is absolutely known that one and only one device is connecting. When a LIR/ISP achieves sufficient address utilization [13], subsequent allocation for additional address space will be provided.

The lower half of the 128-bit IPv6 address is assigned the interface ID. The same allocation policy specifies using the MAC address for this field. Thus, for most cases, only the first 64 bits of IPv6 address are used for routing in the network. In this paper, we focus on global unicast addresses all of which start with the first three bits 001 [3].

### B. IPv6 table generation schemes

IPv6 is to assume the responsibilities of IPv4. As discussed in section I, IPv6 table structure is likely to eventually share certain characteristics common to those of IPv4's. Thus, the goal for the table generation schemes is to generate IPv6 tables that inherit suitable features present in IPv4 tables while adhering to the IPv6 allocation policies.

Different schemes are described in this section in terms of the three major aspects (prefix value, prefix length, and table size) of an IP table to synthesize IPv6 tables based on existing IPv4 tables. We make no claim that the tables synthesized here will be identical to the future real IPv6 tables, or that there are no other effective ways to predict future data. We propose several ideas for best-effort estimation based on what is known today. These schemes and results can serve as a reference for the current IPv6 research and development. The synthesis methods can be adjusted according to changes in IPv6 address allocation and IPv6 deployment.

*1) Prefix value generation:* To generate prefix values, we experiment with two different groups of schemes. We will only briefly describe group A schemes and focus more on group B since group B turns out to be the preferred method.

Schemes in group A insert bits into IPv4 address prefixes. We experiment with various methods to stretch IPv4 prefixes, while at the same time trying to maintain several intrinsic features observed from the existing tables. Depending on the locations in the prefix and how the bits are inserted, one can achieve different degrees of uncertainty and distance by which the new tables deviate from the original IPv4 tables. Schemes A1 and A2 insert extra random bytes in the middle of the IPv4 prefixes; A3 inserts 0 or 1 after each IPv4 bit; A4 and A5 repeat byte by byte or bit by bit of IPv4 prefixes; A6 adds 0 and A7 adds 1 in front of each IPv4 bit.

To further improve the methods, schemes in group B use Autonomous System (AS) numbers combined with IPv4 prefixes. Since AS numbers are the identifiers of Autonomous Systems (a group of IP networks), they represent a good aggregation of IP prefixes. AS numbers, which are 16 bits in length, are added in front of each 32-bit IPv4 address. Then we fill the rest of the bits with random numbers to form the 64 bits of the IPv6 address. Thus, a synthesized IPv6 prefix consists of AS number + IPv4 prefix + random number (if prefix length is greater than /48). Based on the current IPv6 allocation policies described in section III.A, very few prefixes longer than /48 are expected to appear in the core routers' routing tables. Therefore the random number part should be rarely needed.

Figure 4 shows the bit value distribution in an IPv6 table generated from a scheme in group B based on the same IPv4 table used for Figure 2. A similar skewed distribution is observed at the top bits, without counting the first 3 bits

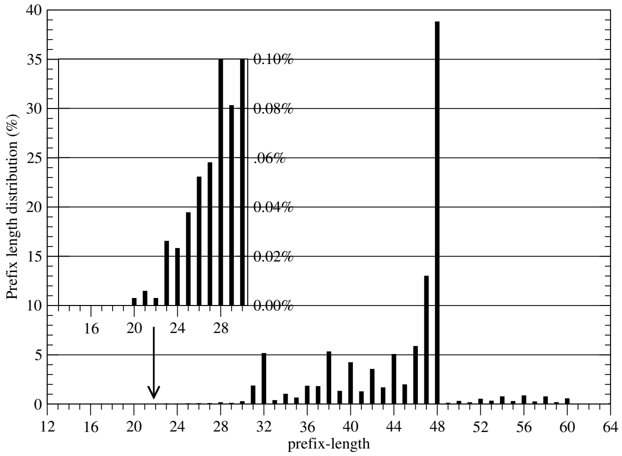


Fig. 6. Prefix length distribution of a synthesized IPv6 table.

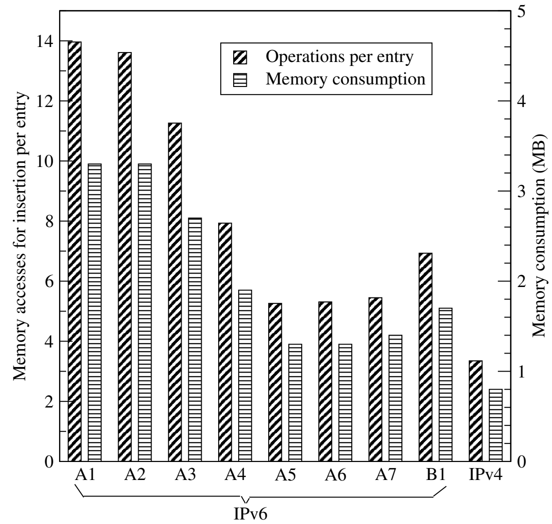


Fig. 7. Performance comparison of IPv6 tables generated by different schemes from an IPv4 table using the same lookup algorithm.

that take on the fixed values of 001. The original IPv4 distribution pattern is replicated in the middle of the bit levels. Randomizing bits 24 to 26 yields a more balanced distribution, illustrated in Figure 5. With well-designed allocation and practice schemes, an even more balanced distribution can be achieved, which will increase the efficiency of address space utilization.

*2) Prefix length distribution:* As mentioned in section III.A, the second 64 bits of an IPv6 address use the MAC address. If this rule is followed, there should not be any prefix lengths between 65 and 127 in routing tables. Thus the hierarchical structure in IPv4 tables inherited by IPv6 tables should be reflected in the first 64 bits of IPv6 prefixes. The goal of generating IPv6 tables based on IPv4 data reduces to stretching IPv4 prefixes of up to 32 bits into IPv6 prefixes of up to 64 bits.

Allocation schemes are a key contributor to the prefix length distribution, which affects the sparseness of the address tree. To preserve the characteristics of IPv4 tables, we propose the following scheme to generate IPv6 prefix lengths: IPv6 prefix length = IPv4 prefix length$\times$2. This will generate prefix lengths that are all even in value. To maintain the 1 to 3 ratio of odd to even prefix lengths in the current IPv4 tables, the prefix length of one out of four entries is converted to an odd number by either adding or subtracting 1 from value generated by the above equation.

To account for the existing IPv6 allocation policies[18–20], we convert all the prefix lengths of /8 in IPv4 to /23 in IPv6, since /23 is the equivalent length in IPv6 that IANA assigned to each regional registry. Figure 6 shows the prefix length distribution resulting from this scheme. As expected, the most heavily populated prefix length /24 in IPv4 is scaled to /48 in IPv6. Although peaks are expected at both /32 and /48 based on the current allocation policies, a large number of /48 prefixes will be assigned when IPv6 is widely deployed. Thus, there should be many more /48 prefixes than /32's as indicated in the figure, even though that might not be the case at the beginning of the prefix allocation of IPv6. Table size

The growth of the Internet's IPv4 table size is a popular discussion topic. It is not clear how IPv6 table size will evolve compared with the current IPv4 table size. On one hand, IPv6 is designed to provide more aggregation through proper allocation schemes to prevent the explosion of routing table size if not reducing it. On the other hand, multi-homing is contributing 20-30% of the prefixes in current IPv4 routing tables [23]. Multi-homing is expected to increase in IPv6, which makes controlling the routing table size difficult. It is a challenging task to design a lookup algorithm with good performance that scales with both longer prefix lengths and larger table sizes. It will take time for IPv6 table size to catch up with the current IPv4 table size. For studies within the near future, it is reasonable to use a table size for IPv6 that lies between 1$\times$ and 2$\times$ of today's IPv4 table size.

### C. IPv6 table generation scheme comparison

There are no standard benchmarks or one parameter to compare IPv6 tables. The few parameters used to analyze tables in this paper can be utilized to compare different tables. In addition, different tables behave differently under the same lookup algorithm. This, to a certain degree, can reveal valuable insights into the tables and the corresponding synthesis methods.

An advanced tree-based lookup algorithm[14] is used to measure the performance and cost of different tables. The insertion cost is more associated with the table itself, while the lookup cost also depends on the traffic pattern. Thus, we chose the insertion cost to compare these tables. Figure 7 presents two performance measures for each scheme: the update performance in terms of the number of memory accesses per entry and total memory consumption. Figure 7 shows that the table structure has a significant effect on this algorithm. Each number in the figure is the average performance of multiple tables generated by the same scheme.

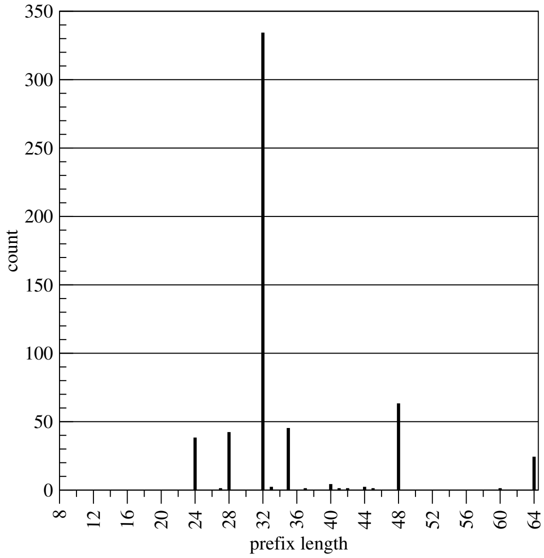


Fig. 8. Prefix length histogram of a current IPv6 table.

Performance wise, schemes A1 and A2 deviate the most from IPv4. This is due to the large amount of new information inserted toward the more significant bits in the original prefixes. This introduction of additional randomness preserves less of the IPv4 characteristics. In contrast, schemes A5, A6, and A7 add in highly predictable patterns without additional randomness. Their performance and cost are closest to IPv4's. However, this may not be a realistic representation of future scenarios. The table generated by scheme B1, using AS numbers, behaves in between the two extremes. In addition, the parent-children relationship is maintained in the tables generated with this scheme. Thus, from both analysis and simulation data, scheme B generates tables that retain some non-random properties from IPv4 and may better mimic future IPv6 tables.

### D. Existing IPv6 tables

The IPv6 routing table has grown from less than 100 entries in 2001 to over 550 (as of March 2004) with most of the users in Europe, followed by Asia and North America. The distribution of existing prefix lengths is shown in Figure 8. At this early stage of IPv6 deployment, the largest number of allocations is at /32. This is expected to change as discussed in the previous session. Some LIR/ISPs were assigned /35 from an earlier IPv6 address policy (they are entitled to expand to /32). Users are currently migrating from an experimental network 6bone (3FFE:) to RIR (2001:). Most prefixes assigned have the first byte values of 2001, 3FFE, or 2002.

## IV. CONCLUSIONS

We have introduced new ways to analyze the structure of IPv4 tables. Several non-random table generation schemes for IPv6 are proposed based on the three major aspects of a routing table: prefix value, prefix length, and table size. One such method, combining AS numbers with IPv4 prefixes, produces IPv6 tables that are consistent with several key common characteristics observed in IPv4 tables and, at the same time, reflects IPv6 allocation schemes.

## REFERENCES


[1] S. Deering, R. Hinden, RFC2460 Internet Protocol, Version 6 (IPv6) Specification,

[2] R. Hinden, S. Deering, RFC2373 IP Version 6 Addressing Architecture

[3] R. Hinden, S. Deering, RFC2374 IPv6 Aggregatable global unicast address format,

[4] H. Narayan, R. Govindan, G. Varghese, The Impact of Address Allocation and Routing on the Structure and Implementation of Routing Tables. SIGCOMM 2003.

[5] Active IPv6 BGP entries, http://bgp.potaroo.net/v6/as1221/index.html.

[6] 6bone: (phasing out of the IPv6 testbed), http://6bone.net.

[7] T. Bu, L. Gao, D. Towsley, On Characterizing Routing table growth. Proceedings of Global Internet 2002.

[8] Route Views Project, http://www.routeviews.org.

[9] http://www.merit.edu.

[10] O. Maennel, A. Feldmann, Realistic BGP Traffic for Test Labs.Proceedings of SIGCOMM 2002.

[11] M. Ruiz-Sanchez, E. Biersack, W. Dabbous, Survey and Taxonomy of IP Lookup Algorithms. IEEE Network. Vol. 15. Issue 2. 2001.

[12] http://bgp.potaroo.net.

[13] A. Durand, C. Huitema, RFC 3194 The H-Density Ratio for Address Assignment Efficiency: An Update on the H ratio.

[14] Cisco Internal Documents and Programs.

[15] G. Huston, Analyzing the Internet's BGP Routing Table. Internet Protocol Journal. Volume 4, Number1, March 2001.

[16] E. Kohler, J. Li, V. Paxson, S. Shenker, Observed structure of addresses in IP traffic. Proceedings of 2nd Internet Measurement Workshop, November 2002.

[17] Z. Xu, X. Meng, C. Wittbrodt, S. Lu, L. Zhang, Address Allocation and the Evolution of the BGP routing table. Technical Report CSD-TR03009, UCLA Computer Science Department, 2003, http://www.cs.ucla.edu/wing/pdfdocs/address_tr.ps.

[18] IPv6 Address Allocation and Assignment Policy (RIPE), http://www.ripe.net/ripe/docs/ipv6policy.html.

[19] IPv6 Address Allocation and Assignment Policy (ARIN), http://www.arin.net/policy/ipv6_policy.html.

[20] IPv6 Address Allocation and Assignment Policy (APNIC), http://www.apnic.net/docs/policy/ipv6-address-policy.html.

[21] DoD IPv6 plans, http://www.usipv6.com/2003arlington/presents/Mariyn_Kraus.pdf.

[22] IPv6 Forum, http://www.ipv6forum.com.

[23] T. Bu, L. Gao, D. Towsley, On Characterizing Routing table growth. Proceedings of GlobalInternet 2002.