

# A Growth-Based Address Allocation Scheme for IPv6

Mei Wang

Stanford University, Stanford, CA 94305, USA  
wmei@stanford.edu

**Abstract.** IP address allocation policies significantly impact the Internet infrastructure, affecting many parties such as router manufacturers, ISPs, and end users. An address allocation policy can also directly affect the performance of the Internet. For example, address fragmentation, a key problem in IPv4, degrades address lookup performance in routers. Thus, a well-designed address allocation policy needs to minimize fragmentation while using the address space efficiently.

This paper attempts to quantify the performance of address allocation policies by modeling key features that lead to fragmentation and inefficient address space usage. Our main contributions are: (i) we identify a drawback of the current IPv6 address allocation policy, which treats all entities uniformly, (ii) we propose a scheme that takes future growth rate into account for allocations, and (iii) an analytical model for measuring the efficiency of allocation schemes, allowing us to quantify the improvement our proposal offers over the current scheme. We believe that a quantitative study of allocation policies is timely since IPv6 address allocation is just beginning in earnest.

## 1 Introduction

IP address lookup is a key element of packet processing in Internet routers. The performance of lookup algorithms is largely impacted by the size and structure of routing tables. Recent research has shown that address allocation policies significantly affect the structure and growth of routing tables and, hence, the performance of lookup algorithms [4, 5, 7]. Therefore, it is natural to ask: How should one design an address allocation scheme that will lead to well-structured routing tables? A timely answer to this question will help the address allocation practice for IPv6, because with its larger address space (due to 128-bit addresses) IPv6 poses challenging problems to the performance of lookup algorithms [1, 2].

Address space fragmentation, a phenomenon that causes one entity (say an ISP) to have multiple non-contiguous address blocks or prefixes instead of a single prefix, is one cause of bad routing table structures. This makes address lookup algorithms, which are longest-matching-prefix based, to perform poorly. While there are several reasons why address fragmentation occurs, one key factor is a poor address allocation policy. This is because addresses are allocated to an

entity on an “as-needed” basis. Thus, an entity whose size experiences a large growth might end up with non-contiguous address blocks.

Avoiding, or minimizing, address space fragmentation is a major goal of an allocation policy. Another major goal is address space conservation [8]; that is not overallocating addresses to an entity which may not use the entire allocation. Given IPv6’s large address space, it is tempting to overallocate so as to minimize the chance of address fragmentation. However, as demonstrated by many examples in the history of computing and networking, it is important to guard against this temptation since a large supply of a resource invariably invites its increased usage (some of which is even hard to foresee). Therefore, we may summarize the goals of a good address allocation policy as the minimization of address fragmentation while ensuring an efficient use of the address space.

This paper provides a quantitative model and an accompanying analysis of allocation schemes, rating them on the degree of address aggregation (as opposed to fragmentation) and conservation they achieve. The current allocation policy suggests a bisection algorithm (described in more detail later) for maximizing address aggregation [15]. However, it does not take into account the potential future growth of an entity while making the initial allocation. Here we propose a scheme that dynamically partitions the address space according to the growth rate of each entity. Using a theoretical model and via simulations we find that, compared to the current bisection method, our growth-based scheme significantly reduces address fragmentation and improves the efficiency of address usage.

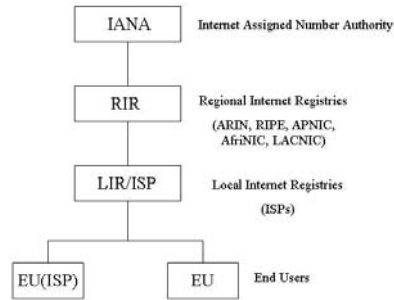
The rest of this paper is organized as follows: Section 2 presents the background information on the current IPv4 and IPv6 allocation policies. Two allocation algorithms, bisection and growth-based schemes, are described in Section 3. Theoretical models and analysis are presented in Section 4 followed by simulation results presented in Section 5. Section 6 presents the conclusion.

## 2 Background

IP addresses are allocated hierarchically. The size of the address blocks decreases as the level of the hierarchy increases. With reference to Figure 1, the addresses at the top of the hierarchy are controlled by the Internet Assigned Numbers Authority (IANA). IANA allocates large address blocks to each of the five Regional Internet Registries (RIR) serving the North American, European, Asian, African, and the Latin American and the Caribbean regions. The regional registries divide up these large address blocks into medium blocks to allocate to Local Internet Registries (LIRs), consisting mainly of ISPs. The ISPs further assign small address blocks to their customers, including companies, universities, smaller ISPs, etc.

### 2.1 Allocation Policies

In this paper, we focus on global unicast address allocation which consumes most of today’s address space. Each block of addresses is represented by a prefix which is denoted by `prefix-value/prefix-length`, e.g., `5.0.0.0/8` in IPv4 and `2000::/3`



**Fig. 1.** IP address allocation hierarchy

in IPv6 [3]. The prefix length specifies the number of significant bits in the prefix value. For IPv4 with 32-bit total address length, an allocation of 5.0.0.0/8 represents an address block size of  $2^{24}$  containing addresses all begin with 5 as the first byte in the address values. The smaller the prefix length, the larger the block size.

The current IPv4 policies are [8-11]: IANA allocates to RIRs in the unit of /8. Different RIRs adopt their own policies for allocations to LIR/ISPs with unit sizes varying from /10 to /20. The sizes assigned to end users by each ISP also vary greatly. Due to historical allocation schemes, fragmentation is a common problem in IPv4 [7]: one ISP is often left with multiple prefixes.

For IPv6, although there are 128 bits in the IP address, the last 64 bits are assigned to interface ID [2], e.g., an ethernet's MAC address. Thus, address allocation only considers the top 64 bits. The allocation unit from IANA to RIRs is /23 [13]. RIRs are coordinating to have a common policy for allocation to LIR/ISPs [14]. Currently, the minimum initial allocation unit is /32. There is a proposal [15] from major RIRs recommending a Common Address Pool (CAP) in IANA for all RIRs, instead of each RIR keeping a separate pool of addresses for allocation. To end users, unlike IPv4, IPv6 is generally assigned in fixed amounts (/48).

Allocation policies vary for different registries and different layers of the hierarchy. At present, documents on allocation policies mainly specify the size of prefixes and the criteria for allocation. Few specific procedures exist on allocation algorithms and how the address space should be partitioned. A good allocation algorithm can be applied to any layer in the allocation hierarchy and any registry. It can be combined with the allocation policy to provide efficient usage of address space and preserve address aggregation.

### 3 Allocation Algorithms

For a given address pool with the address length of  $N$  bits, the total address space of this pool is  $2^N$ . This space can be represented by an address line, ranging

from 0 to  $2^N-1$ . Each allocation is a block of addresses and can be shown as a section on the address line. The starting location of the address block is labeled by the prefix value. The size of the address block, also the length of this section on the address line, is  $2^{N-l}$ , where  $l$  is the prefix length.

For an address provider with a prefix length  $/P$  and its customer with prefix length  $/l$ , their address spaces are  $2^{N-P}$  and  $2^{N-l}$ , respectively. For IPv6,  $N=64$  if we use the first 64 bits as the total address space for allocation. An ISP with  $/32$  can have a total of  $2^{(64-32)}/2^{(64-48)}=2^{16}$  customers with  $/48$  if none of them out-grow their address space. If the smallest address block size allocated by a provider with  $/P$  has a prefix length  $/l_{min}$ , we can treat this block size as a base unit. The problem is equivalent to allocating an address space of  $2^{l_{min}-P}$  with each customer occupying a minimum space of 1.

When a customer grows out of its initially allocated address space, the provider can double this customer's space by combining the customer's current address block with the one immediately after it on the address line in order to keep the same prefix value. The prefix length of the customer is reduced by 1. This process can be repeated every time it out-grows its current space. HD ratio [18] can be used for expansion criteria.

A collision occurs when a customer needs to expand its space and a neighboring customer on the address line is already occupying that space or a section of that space. When this occurs, the provider can either find a new location on the address line with a different prefix value or still double the customers space by carving out the section already occupied by the neighbor. Both options lead to fragmentation: non-continuous address blocks for one entity, i.e., more than one prefix representing a single entity in the routing table. Fragmentation reduces lookup and routing efficiency and increases routing table size. Such practice should be avoided whenever it is possible. A third option is to move either one of the two customers to a different address location, which will create much hassle and may not always be viable.

We describe two address allocation schemes in this section: the basic bisection scheme suggested by the registries and our proposed scheme based on the growth rate of each allocation.

### 3.1 Bisection Scheme

The current address allocation practice is based on the bisection scheme, proposed by registries of North America, Asia, and Europe. Address blocks are allocated bisectionally in the following way: each new address block is allocated by evenly splitting a section on the address line between the existing customer and the new customer. This leaves maximum possible space for potential growth of both customers. An example using this scheme is illustrated in Figure 2. The first four customers are allocated by dividing the total address space equally into four parts as shown in Figure 2(a). When the fifth customer applies for an address block, it is placed to evenly split the section labeled by the arrow in Figure 2(b). The 6th, 7th, and 8th customer will be placed sequentially in the spaces after the 2nd, 3rd, and 4th customer, respectively. Only after the largest

empty slots have been exhausted, will new allocations be assigned to bisect the smaller slots at the next level. A similar technique can be found for dynamic memory allocation in Operating Systems and is already used in IPv4.

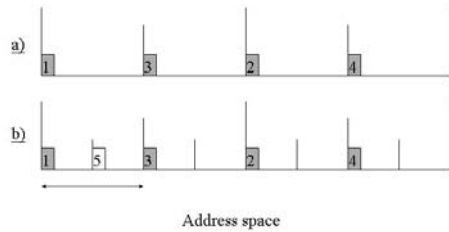


Fig. 2. Bisection scheme

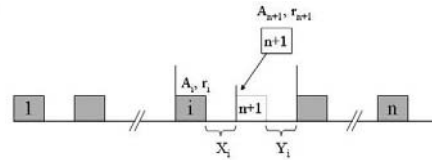


Fig. 3. Procedure to determine address location for the new customer using growth-based scheme

This method uniformly separates the allocated blocks to maximize the spacing between customers. However, different customers are likely to have different growth rates and require different sizes in the address space. Thus this uniform partition may not be the most efficient way to utilize the address space, especially when there are a few very fast-growing customers. These fast-growers can collide with their neighbors and cause address space fragmentation.

### 3.2 Growth-Based Scheme

To utilize the address space more efficiently and reduce collisions, a modified scheme is proposed to make allocations based on the growth of each customer. When there are  $n$  existing customers, there are at most  $n$  possible address location candidates for the  $(n + 1)^{th}$  customer. Instead of treating each customer equally, as in the bisection scheme, the growth-based scheme evaluates all the options and chooses the location based on the available space sizes as well as the growth of the existing customers and the new customer. For example, one can choose the location for the new customer that maximizes the time before the first collision is projected to take place. To maintain the same prefix value for each address block, the newly allocated block has to start at the middle point of the available address space.

The ability to obtain reasonably accurate growth estimation directly impacts the effectiveness of growth-based algorithms. Since each address provider (a registry, an ISP, or a company) can access the information of its customers' utilization, the estimated growth of each customer can be derived from its growth history and current utilization. Price related incentives can be applied to help collect more accurate estimations from customers. This scheme still works even when the estimation of the growth rate is not accurate or even wrong. Because for each address allocation there is still space left for potential growth. Since we check each available block's growth every time inserting a new comer, the growth rate can be adjusted frequently and adaptively according to any changes. The worst case would be a fast growth entry coming in when the whole address

space is about full, then there will be less gain or no gain using this scheme, but this case is rare.

For now, we assume that the growth information is available, known as growth rate. Let

$n$  : the number of existing customers;  $n + 1$  : the new customer;

$A_i$  :  $i=1, \dots, n, n+1$ , allocation for each customer;

$r_i$  :  $i=1, \dots, n, n+1$ , estimated growth rate of each allocation;

$X_i$  :  $i=1, \dots, n$ , empty space behind each allocated block to bisection point;

$Y_i$  :  $i=1, \dots, n$ , the empty space available for newly allocated customer to grow;

as shown in Figure 3. The new location is chosen to maximize the time before the first collision occurs, using:

$$\max\{\min[t(X_i, r_i), t(Y_i, r_{n+1})], i = 1, \dots, n\}. \quad (1)$$

$t(X, r)$  is the time it takes for allocation  $A$  with growth rate  $r$  to fill up the space  $X$ . In other words, the new location is chosen among all available spaces that maximizes the time it takes for either the existing or the new customer to run into the boundary. Function  $t(X, r)$  can be of different forms depending on the definition of growth rate  $r$ .

## 4 Theory

To quantitatively study the effects of the allocation schemes, we model the system where new customers keep being allocated in the address space and continue growing in size with various rates. Analytical expressions are derived in this section and are compared to simulations in the next section. In order to obtain closed-form expressions, we treat the address space and the space needed by individual customers as continuous variables in this section. The total allocated address space for IPv4 has been growing exponentially during the last decade, i.e., proportional to the total size of the network [17]. For the rest of this paper, we use the same functional form to represent growth, i.e., exponential increase for both the number of customers and the space needed by individual customers. The methodology developed here can be easily adapted for other functional forms of growth. To compare the performance of different allocation schemes, we compare the total number of customers served and the total address space utilized before the first segmentation takes place.

### 4.1 Bisection Scheme

We assume that the number of customers grows proportional to the total number of existing customers, i.e., exponential increase with time

$$n(t) = n_0 f(t), \quad f(t) = 2^{v \times t}, \quad (2)$$

where  $n_0$  is the initial number of customers at  $t=0$  and  $f(t)$  is the growth function of the number of customers.  $v$  is the growth rate. Different forms of growth functions can be used in place of expression  $2^{v \times t}$  for other growth patterns.

The bisection scheme attempts to partition the total address space  $S_{tot}$  equally among all customers. The amount of space to grow for each customer,  $s(t)$ , decreases with time as more customers are added to the address space:

$$s(t) = S_{tot}/n(t) = S_{tot} \times 2^{-v \times t}/n_0. \quad (3)$$

As time progresses, customers can request additional space as they themselves grow in size. Customers can be grouped into different classes according to their growth rates  $r_i$  and initial address space  $l_i^0$ , where  $i$  is the class index ( $1 \leq i \leq m$ ,  $m$  is the total number of classes). Within each class, the customers are labeled by index  $j$  which corresponds to the order of which they are added into the address space. Depending on the time of its entry,  $t_{i,j}^0$ , the size of the address space a customer requires at a later time  $t$  is

$$l_{i,j}(t) = l_i^0 \times 2^{r_i \times (t - t_{i,j}^0)}. \quad (4)$$

The sequence of incoming customers are considered to be random. The entrance time  $t_{i,j}^0$  of the  $j^{th}$  customer of class  $i$  can be estimated from the following: when the total number of customers at this time multiplied by the probability  $p_i$  (the probability that a customer picked at random belongs to class  $i$ ) equals  $j$ , i.e.,  $n(t_{i,j}^0) \times p_i = j$ ,

$$t_{i,j}^0 = \frac{1}{v} \times \log_2\left(\frac{j}{p_i \times n_0}\right). \quad (5)$$

A collision takes place when the space needed by a customer becomes greater than the space available for its growth. Since it is the first customer of each class that has the largest size at a given time, the problem of finding the time at which first collision occurs reduces to  $t_{min}^c = \min\{t_{i,1}^c, i = 1, \dots, m\}$ . The time of collision of the first customer of class  $i$ ,  $t_{i,1}^c$ , is obtained by equating the average space for growth of each customer [Eq.(3)] and the space needed [Eq.(4)]:

$$t_{i,1}^c = \frac{1}{v + r_i} \times \left[ -\frac{r_i}{v} \log_2(n_0 \times p_i) + \log_2\left(\frac{S_{tot}}{n_0 l_{i,1}^0}\right) \right]. \quad (6)$$

The total number of customers served at the time of the first collision follows from Eq.(2),  $n(t_{min}^c) = n_0 \times 2^{v \times t_{min}^c}$ .

To obtain the total space used at the time of the first collision, one can sum over the space occupied by existing customers of each class. This can be done by finding the number of customers of each class at  $t_{min}^c$ , the time of entry of each customer, and the space needed by each customer at  $t_{min}^c$ . To obtain an analytic expression, we approximate the summation by an integral over time,

$$S(t_{min}^c) = \sum_{i=1}^m \int_0^{t_{min}^c} l_i^0 2^{r_i(t_{min}^c - t)} p_i \frac{dn(t)}{dt} dt = \sum_{i=1}^m n_0 l_i^0 p_i v \frac{2^{v t_{min}^c} - 2^{r_i t_{min}^c}}{v - r_i}. \quad (7)$$

Eq.(7) can be understood as follows:  $[dn(t)/dt] dt$  is the number of new customers added between time  $t$  and  $t + dt$ . Among these,  $p_i \times [dn(t)/dt] dt$  of them belong to class  $i$  that grow to size  $l_{i,1}^0 \times 2^{r_i \times (t_{min}^c - t)}$  at  $t_{min}^c$ .

The results from the analytic method is shown in Figure 4 as a function of the total address space. These results compare well with a simulation to be described in Section 5. The small deviation of the analytical method from the simulation results comes from the fact that the address space is treated as continuous in the analytic method as opposed to integers in the simulation. The same can be said about the results given in Figure 5 for the growth-based scheme.

#### 4.2 Growth-Based Scheme

The growth-based scheme dynamically optimizes the partition in the address space for each incoming customer. We find that it is possible to obtain a closed-form expression for a specific case where there are only two classes of customers: (1) ones with a fixed size without growth and (2) ones that grow and grow at the same rate. In reality, we can categorize most customers into these two groups: slow growth and fast growth. The probabilities of finding a customer belonging to classes 1 and 2 are  $p_1$  and  $p_2$ , respectively, with  $p_1 + p_2 = 1$ . For simplicity, we assume that customers of both classes require an initial size of 1.

When the first customer of class 2 enters into the address space, there are, on average, a total of  $1/p_2$  customers in the address space. The available space for this customer to grow would be roughly  $s_{2,1} = S_{tot} p_2$ . We make an assumption that the optimum arrangement of the address space would be letting each customer of class 2 (i.e., with growth) keep all its potential growth space obtained at its initial allocation. We will show at the end of this section that this assumption is justified under certain conditions. Following this assumption, when the second customer of class 2 enters, there are  $2/p_2 - 1$  customers partitioning the rest of the space,

$$s_{2,2} = (S_{tot} - s_{2,1}) / (2/p_2 - 1) = S_{tot} (p_2(1 - p_2)) / (2(1 - p_2/2)). \quad (8)$$

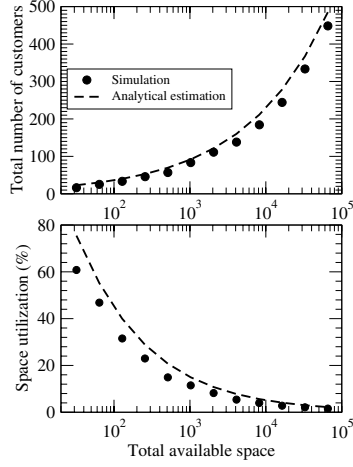
Analogously, the  $j^{th}$  customer of class 2 gets a growth-space of size

$$s_{2,j} = S_{tot} p_2 (1 - p_2)^{j-1} / \left[ j \prod_{k=1}^j \left( 1 - \frac{k-1}{k} p_2 \right) \right]. \quad (9)$$

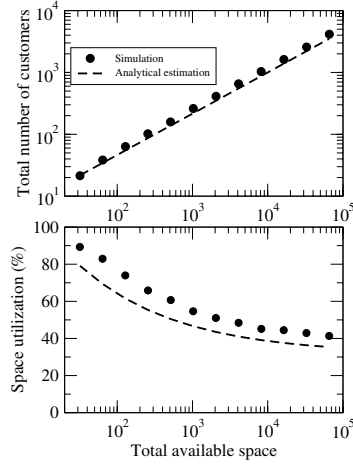
Since customers of class 1 keep a fixed size, we only need to consider the collisions from customers of class 2. The space that the  $j^{th}$  customer of class 2 requires at a later time  $t$  is given by  $l_{2,j}(t) = 2^{r_2 \times (t - t_{2,j}^0)}$ , where  $t_{2,j}^0$  is the time of its entry and can be obtained using Eq.(5). The time that the  $j^{th}$  customer of class 2 exhausts its growth space is when  $l_{2,j}(t) = s_{2,j}$ :

$$t_{2,j}^c = \frac{1}{v} \times \log_2 \frac{j}{n_0 p_2} + \frac{1}{r_2} \times \log_2 \left[ S_{tot} \frac{p_2 (1 - p_2)^{j-1}}{j \prod_{k=1}^j \left( 1 - \frac{k-1}{k} p_2 \right)} \right]. \quad (10)$$





**Fig. 4.** Bisection scheme: comparison of analytic method with simulation



**Fig. 5.** Growth-based scheme: comparison of analytic method with simulation

To determine which customer of class 2 is the first to exhaust its growth space, we compare the time-to-collision between the  $(j + 1)^{th}$  and  $j^{th}$  customers:

$$\begin{aligned}
 t_{2,j+1}^c - t_{2,j}^c &= \frac{1}{v} \times \log_2 \frac{j+1}{j} + \frac{1}{r_2} \times \log_2 \left[ \frac{j(1-p_2)}{(j+1) \left(1 - \frac{j}{j+1} p_2\right)} \right] \\
 &\approx (r_2 - v)/(v \times r_2) \times \log_2 (1 + 1/j) \quad (\text{if } p_2 \ll 1) \\
 &> 0 \quad (\text{if } r_2 > v),
 \end{aligned}
 \tag{11}$$

where the approximate sign holds when the probability of fast-growers (i.e., class 2) is much less than 1. Eq.(11) shows that the time-to-collision increases with the customer number under the condition that the space growth-rate of the growing customers ( $r_2$ ) is faster than the rate of increase in the total number of customers ( $v$ ). Our starting assumption, that each customer of class 2 keeps all its potential growth space, is therefore justified as being the optimum choice for the placement of incoming customers under these conditions. Therefore, the first customer of class 2 is the first to hit its boundary of potential growth:

$$t_{min}^c = -1/v \times \log_2 (n_0 p_2) + 1/r_2 \times \log_2 (S_{tot} p_2). \tag{12}$$

The total number of customers when the first collision takes place can be obtained from Eqs.(12) and (2),  $n(t_{min}^c) = p_2^{\frac{v-r_2}{r_2}} \times S_{tot}^{\frac{v}{r_2}}$ . The total space utilized follows from Eqs.(12) and (7):

$$S(t_{min}^c) = \frac{v}{r_2 - v} n_0^{1-\frac{r_2}{v}} \times p_2^{2-\frac{r_2}{v}} \times S_{tot} + \left(1 - \frac{p_2 r_2}{r_2 - v}\right) \times p_2^{\frac{v-r_2}{r_2}} \times S_{tot}^{\frac{v}{r_2}}. \tag{13}$$

Figure 5 shows that the results using the analytic method agree with those obtained by a simulation technique described in the next section. These results are obtained using the following parameters:  $p_2=10\%$ ,  $r_2=1/4$ ,  $v=1/6$ ,  $n_0=4$ .

## 5 Simulation Results

The projected growth rate is essential for the application of the growth-based algorithm. The growth rate can be represented by various functional forms, such as the total projected address space, the probability that extra bits will be needed in the future, the projected size increase for each year, etc. The basis of the theoretical analysis and simulation techniques put forth in this paper can be adapted for the different functional forms or their combinations.

The performance differences on address space conservation and fragmentation between the bisection and growth-based schemes are compared using simulations. Two metrics are used to measure the performance on conservation: the total number of customers served and the percentage of address space allocated before the first collision takes place. This implies that all the customers served have one continuous address block without fragmentation, i.e., every customer has one prefix only. For performance on fragmentation, the percentage of fragmented addresses is measured for different address utilizations.

In the simulation we assume that customers with different growth rates enter the address space at random order. The identical sequence of customers is fed to both schemes for a one-to-one comparison. A variety of different distributions in the customers' growth-rates have been explored with qualitatively similar results. The results for a Gaussian distribution are presented in this paper, which serves as a representative. In these simulations, most of the customers have medium growth rates but there are a few with very fast or very slow growth rates.

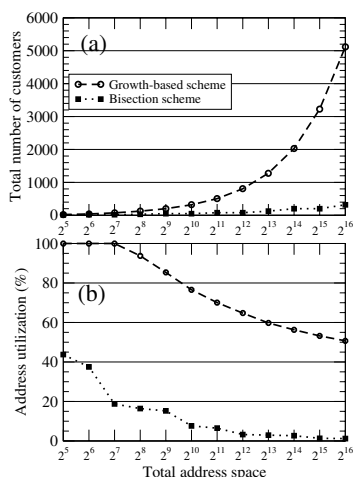
Without loss of generality, we denote the minimum block size allocated by a provider as one base unit in the address space. Simulations of  $2^N$  total address space sizes are considered, with  $N$  ranging from 5 to 16. The case  $N=16$  corresponds to an ISP with prefix length /32 making allocations with the smallest block sizes with prefix length /48. This is exactly the practice in current IPv6 address allocation.

### 5.1 Address Space Conservation

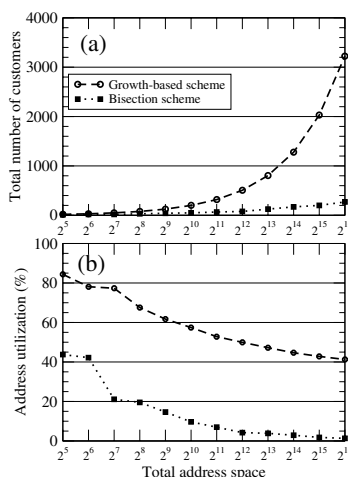
**Fixed Initial Prefix Length Allocation.** The first group of simulations is carried out on fixed initial prefix lengths. According to the current IPv6 allocation policies, the initial assignment is, in almost all cases, of the same initial length for each new customer. It is only when a customer has achieved a certain utilization of the initial address space that it is given a larger block of address space. This policy applies to each layer of the allocation hierarchy with official registries.

Figure 6(a) shows that the growth-based scheme can serve many more customers than the bisection scheme before the first collision. This means more addresses are allocated by the time the first collision happens, thus, the address

utilization is higher. The larger the size of the total address space, the more significant the gain is with the growth-based scheme given the identical profile of customers. The same can be said about the percentage of space utilized, as shown in Figure 6(b). For the case with a total address space of  $2^{16}$ , the growth-based scheme gives a 16-fold improvement over the bisection scheme for the number of customers served and a 42-fold increase in terms of space occupancy.



**Fig. 6.** Simulation results on address space conservation for both bisection and growth-based schemes with fixed initial prefix length allocation



**Fig. 7.** Simulation results on address space conservation for both bisection and growth-based schemes with variable initial prefix length allocation

The amount of gain of the growth-based scheme over the bisection method depends on the input profile. One factor in the input profile is the order that different customers enter into the address space. Another factor is the growth rate of each customer relative to the rate of increase in the total number of customers. From a large number of experiments we explored over different input configurations, the gains of the growth-based scheme are all very substantial but vary in quantity. In one case the gain is not obvious: if a customer with a very fast growth rate comes in when the space is relatively full and the size of the requested block is of the same order as the size of the empty slot. In this particular case, there is not much time left before this new customer reaches its space limit for growth. This very limited amount of time is sometimes not enough to show a dramatic benefit for the growth-based scheme over the bisection one.

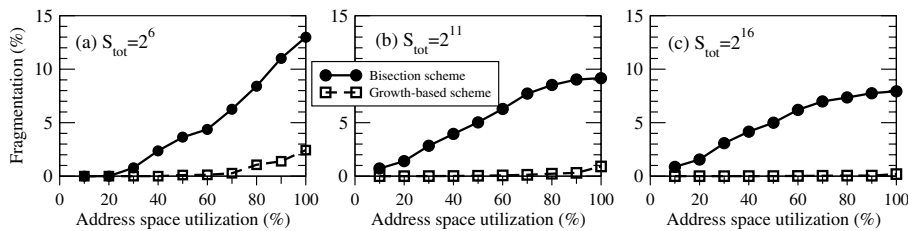
**Variable Initial Prefix Length Allocation.** Even though the current policies require giving a fixed initial prefix length for each new allocation, it is possible that such policy will be relaxed in the future to allow variable initial allocation

lengths. Will the gain of the growth-based scheme still hold? Figure 7 shows that, under the new conditions, the gain is still substantial, although slightly smaller than the previous case with fixed initial length. This is because some customers require large initial lengths. When the space is relatively full, there may not be an empty location of sufficient size available to fit in the new customer. This will also cause fragmentation.

It is worth pointing out that the growth-based scheme reduces to the bisection scheme when the growth rates of all customers are much slower than the rate of increase in the number of customers. In this case, the growth-based scheme can be viewed as equivalent to the bisection scheme in the limit of zero growth rate.

## 5.2 Address Fragmentation

As the address space becomes more crowded, the probability of address fragmentation increases. The growth-based scheme can significantly reduce the amount of the fragmentation and delay the onset of fragmentation until a large percentage of the address space is utilized, as shown in Figure 8.



**Fig. 8.** Percentage of fragmented addresses for both bisection and growth-based schemes as a function of address space utilization for different sizes of total address space: (a)  $S_{tot}=2^6$ , (b)  $S_{tot}=2^{11}$ , and (c)  $S_{tot}=2^{16}$

Using the bisection scheme, as the space utilization increases, the percentage of fragmented addresses increases steadily for all three sizes of total address space considered. Given the parameter set we used in this simulation, close to 8% of the addresses are fragmented when the utilization is over 70% given a total address space of  $2^{11}$  (Figure 8(b)) or  $2^{16}$  (Figure 8(c)). In addition, fragmentation takes place at small utilization percentages because of a few fast-growing customers.

In contrast, the fragmentation percentage remains to be small even at 100% space utilization for the growth-based scheme. This is especially true when the total address space is large, e.g.,  $2^{16}$ , giving the address provider enough time adjust for the fast-growing customers. Furthermore, the onset of fragmentation does not take place until  $\sim 40\%$  of space utilization with the growth-based scheme. With this level of utilization, an ISP is qualified to obtain new address blocks [16]. Therefore, for the cases simulated, address fragmentation caused by allocation can be eliminated by using the growth-based scheme.

## 6 Conclusions

The current algorithm suggested by address allocation registries uses a bisection scheme for IPv6 address allocation. An improved version of this algorithm is proposed in this paper to find better address locations for the customers according to their growth-rates. Quantitative analysis of the performances of the two algorithms are conducted based on both theoretical models and simulations. Our studies show that the growth-based algorithm offers significant advantages over the bisection algorithm for both address aggregation and conservation. The benefits of the growth-based algorithm can be realized at all levels of the address allocation hierarchy.

## Acknowledgments

We gratefully acknowledge Balaji Prabhakar, Pankaj Gupta, Larry Dunn, and Tony Hain for their valuable discussions and support.

## 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. Z. Xu, X. Meng, C. J. Wittbrodt, S. Lu, L. Zhang. IPv4 Address Allocation and the Evolution of the BGP Routing Table. UCLA Computer Science Department, 2003.
5. H. Narayan, R. Govindan, G. Varghese. The Impact of Address Allocation and Routing on the Structure and Implementation of Routing Tables. SIGCOMM 2003.
6. <http://6bone.net/>
7. T. Bu, L. Gao and D. Towsley. On Characterizing Routing table growth. Proceedings of Global Internet 2002.
8. APNIC, RIPE, ARIN. IPv6 Address Allocation and Assignment Policy, June 2002.
9. G. Huston. Allocations vs. Announcements, The ISP Column, May 2004.
10. IAB, IESG. RFC 3177 IAB/IESG Recommendations on IPv6 address Allocations to Sites, September 2001.
11. P. Wilson, R. Plzak, A. Pawlik. IPv6 Address Space Management, October 2002.
12. M. Kuhne, P. Rendek, S. Wilmot, L. Vegoda. IPv4 Address Allocation and Assignment Policies for the RIPE NCC Service Region, October 2003.
13. ARIN. IPv4 Policies. <http://www.arin.net/policy/ipv4.html>
14. Policies for IPv4 Address Space Management in Asia Pacific Region, March 2004.
15. K. Hubbard, M. Koster, D. Conrad, K. Karrenberg, J. Postel. RFC 2050 Internet Registry IP Allocation Guidelines, November 1996.
16. ARIN. IPv6 Policies. [http://www.arin.net/policy/ipv6\\_policy.html](http://www.arin.net/policy/ipv6_policy.html)
17. IPv4 Address Space Report, <http://bgp.potaroo.net/ipv4/>
18. A. Durand. RFC 3194 The HD Ratio for Address Assignment Efficiency