IPv6 Address Allocation -- An Alternative Algorithm for the Sparse Allocation Process

Mei Wang

Advanced Architecture Group
Cisco Systems
mei@cisco.com

Overview

- An analytical model to quantify the effects of address allocation schemes
  - Fragmentation
  - Efficiency
- Propose an alternative allocation scheme:
  - Treat customers differently
  - Utilize more information from customers: i.e. growth-rate
  - Reduce fragmentation and increase efficiency

Overview

- Acknowledgements

Stanford University:	Cisco Systems:
Balaji Prabhakar	**Larry Dunn**
Pankaj Gupta	**Tony Hain**

- Full paper is at: **Networking 2005,** Canada

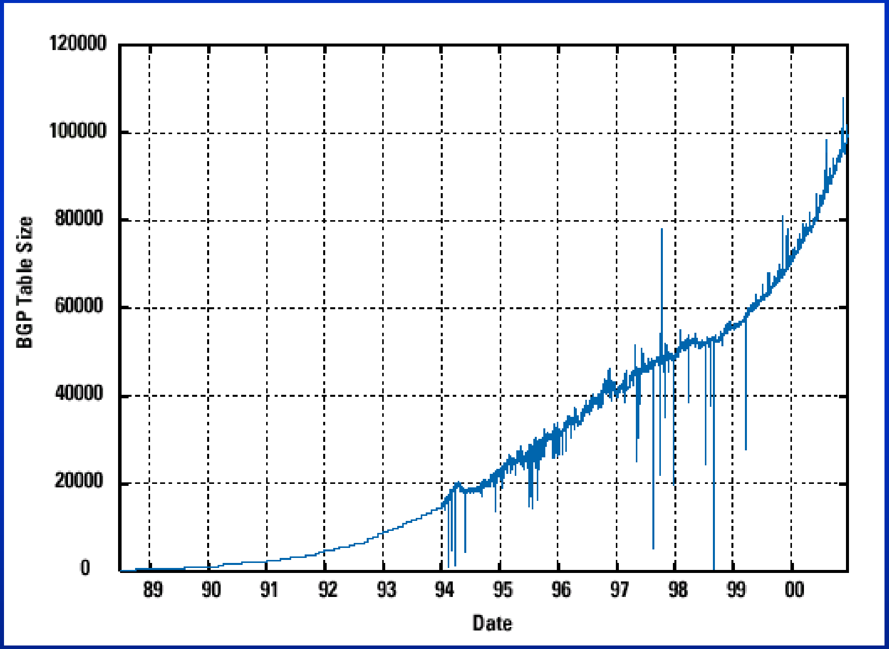
  http://www.cs.uwaterloo.ca/conferences/networking2005/coming/program.html#wed16

Outline

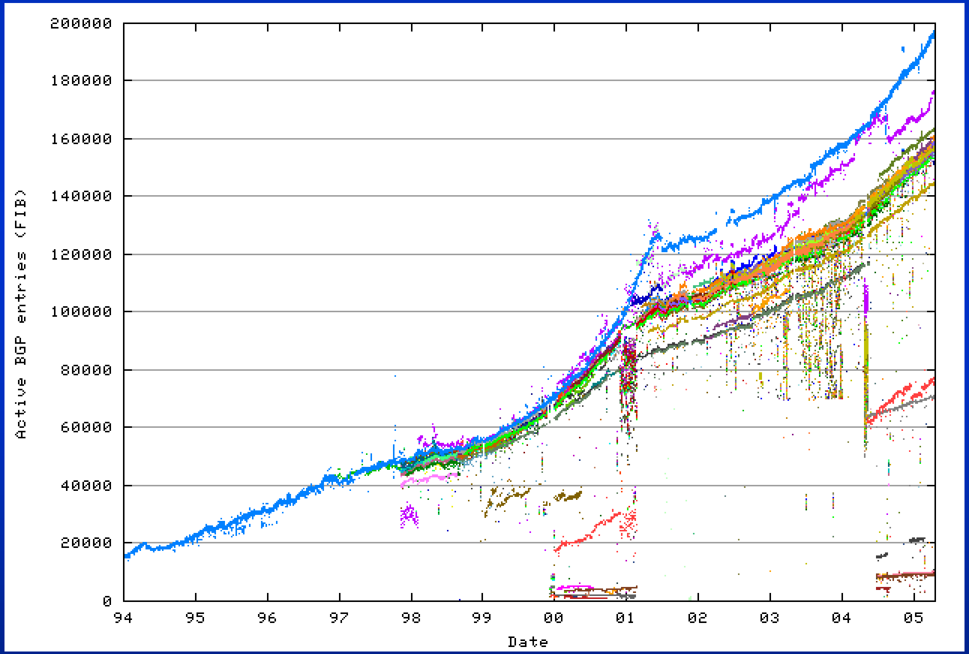
Introduction

Routing Table Growth

1988 -- 2000



1994 -- 2005



http://www.cisco.com/warp/public/759/ipj_4-1/ipj_4-1_bgp.html

http://bgp.potaroo.net

Introduction

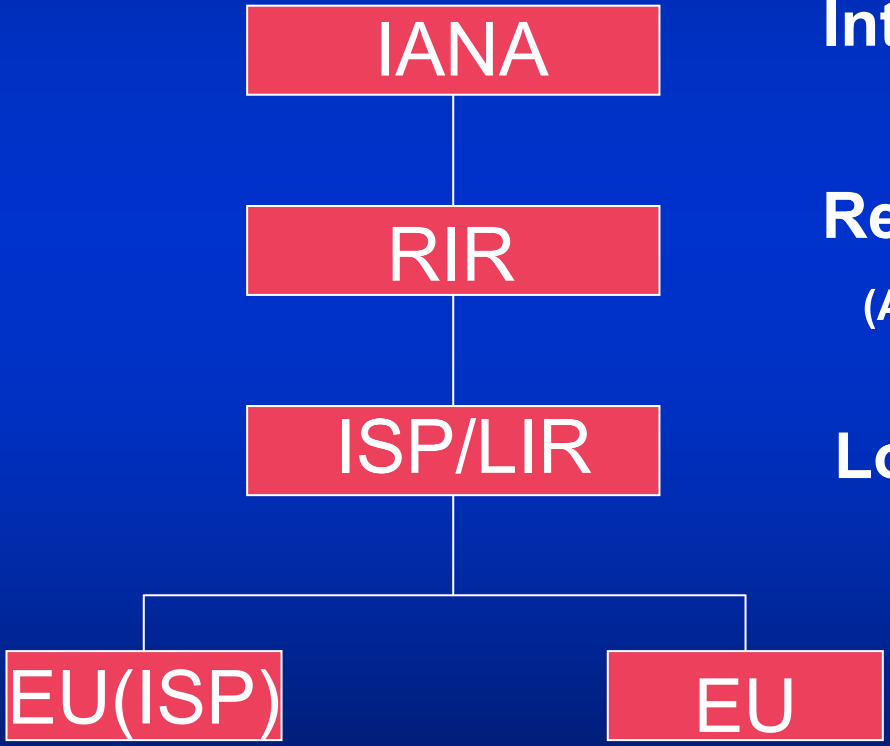
- Why IPv6 allocation policies are important?
  - Impacts on routing: aggregation
    - Reduce address fragmentation, a key problem in IPv4:
      - one ISP has multiple prefixes, discontinuous address space
    - Control routing table size
    - Improve hierarchy, lookup and routing efficiency
  - Efficient usage of address space:
    - IPv6 address space 3.4×10^{38}. Not all usable.
    - Large demands of addresses: devices, appliances, avoid NAT
    - Different usages
  - IPv6 is taking off, now is a critical time
    - Do it right from the beginning, lessons from IPv4
    - IP addresses are critical foundations of ISPs
    - Lots of interested parties can be impacted by the outcome of the policies

Background

- Non-technical factors:
  - Economic, social
  - Hot debate topic
- Technical issues
  - Focus of this work, quantitative analysis
  - Similar to dynamic memory allocation in Operating Systems
- Goals of allocation
  - Uniqueness
  - **Aggregation**
  - **Conservation**

Background

Allocation Hierarchy



Internet Assigned Number Authority

Regional Internet Registries

(ARIN, RIPE, APNIC, AfriNIC, LACNIC)

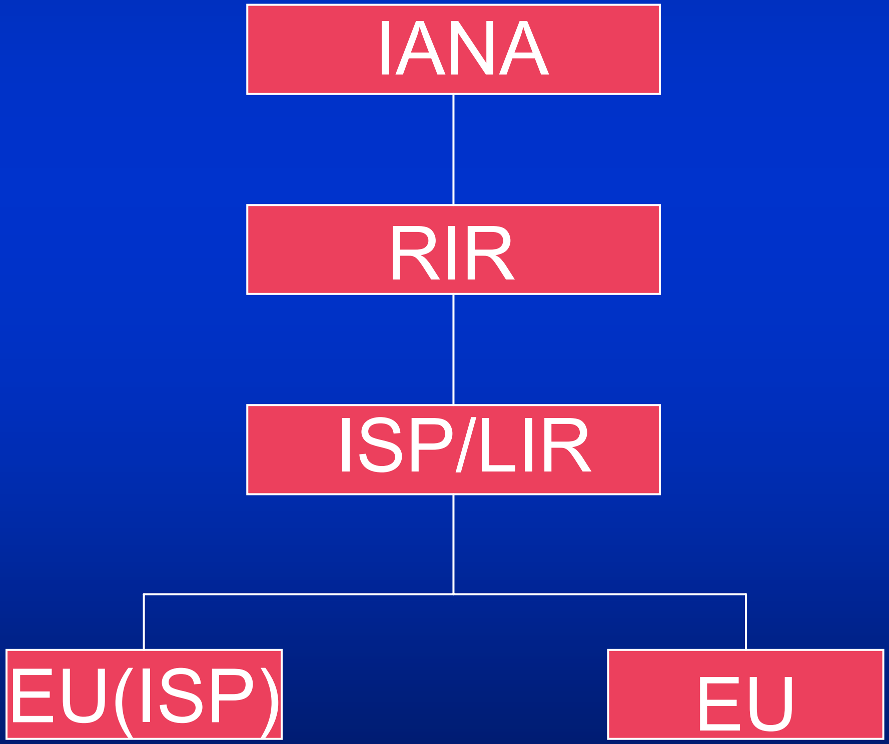
Local Internet Registries

(ISP’s)

End Users

Background

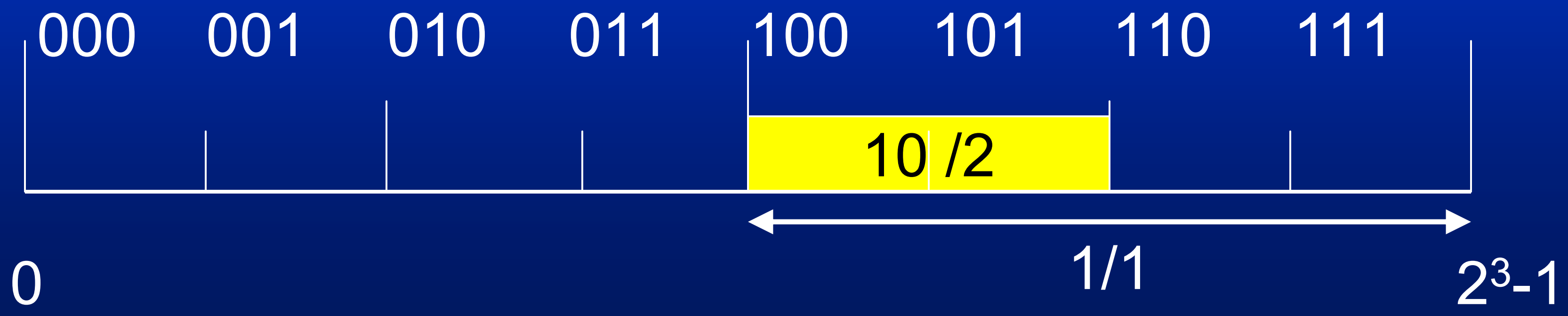
Current Allocation Policies



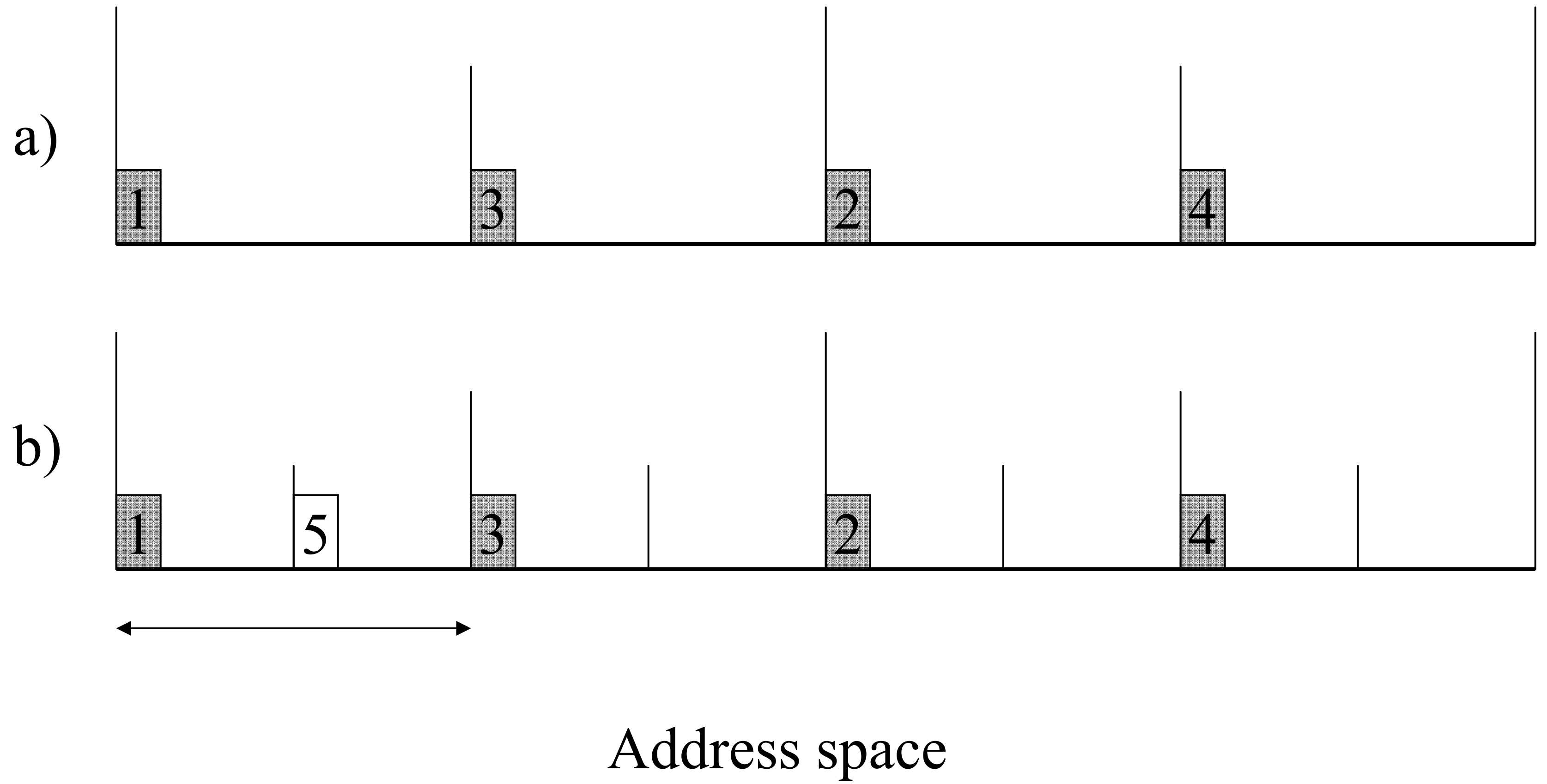
IPv4 (32-bit)	**IPv6** (128-bit)
/8	**/23**
/10 -- /20	**/32**
various	**/48**

Allocation Algorithms -- Model Settings

- Address provider: an address line
- Customer: a block on the line
  - A block of addresses is represented by a prefix, i.e., 10/2
    10 is the prefix value in binary, 2 is the prefix length.
  - Example: 5.0.0.0/8 (IPv4), 2000::/3 (IPv6)
- Growth: double the block size to the right
  - For the correct aggregation
- Collision: run out of space to grow => fragmentation



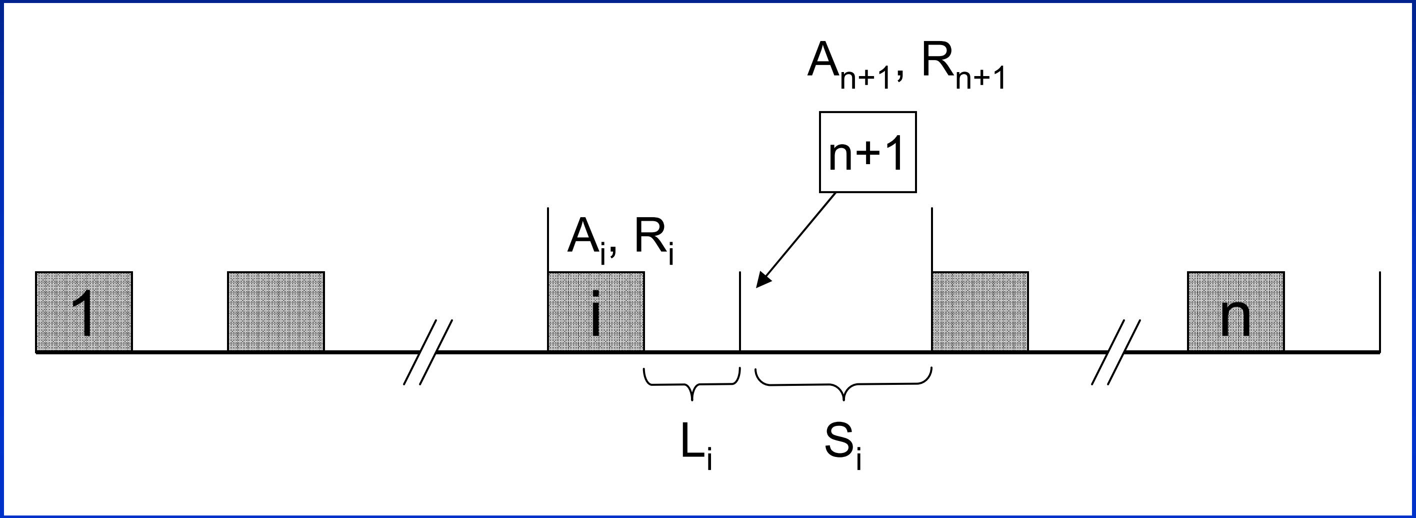
Current Allocation Algorithm -- Bisection



Algorithm We Propose -- Growth-based

- Scheme 1:
  - Take growth rate into consideration
  - Growth rate can be in any form
  - Fixed initial prefix length allocation
- Scheme 2:
  - Variable initial prefix length allocation
- Scheme 3:
  - Optimization of variable length allocation

Growth-based Scheme



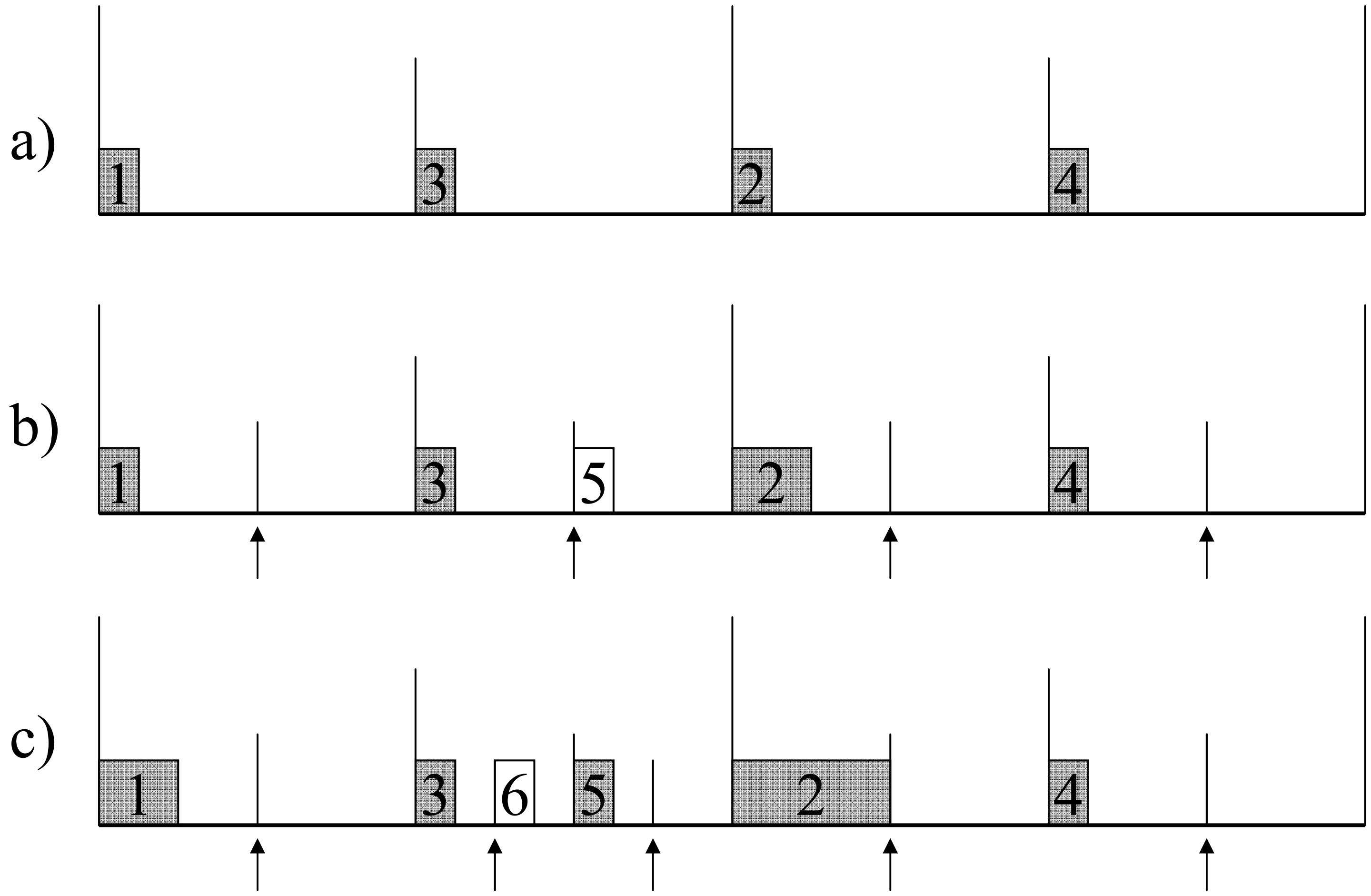
$$\max\left\{\min\left[t(L_i, R_i),\ t(S_i, R_{n+1})\right],\ \ i = 1,\ldots,n\right\}$$

- For the new coming customer with growth rate R_{n+1}, find a slot that maximizes the time it takes before a collision.

$\max\{\min[t(L_i, R_i),\ t(S_i, R_{n+1})],\ \ i = 1,\ldots,n\}$

Growth-based Scheme -- Example

Growth rates: $R_2 > R_1 > R_4 = R_5 >> R_3 > R_6$

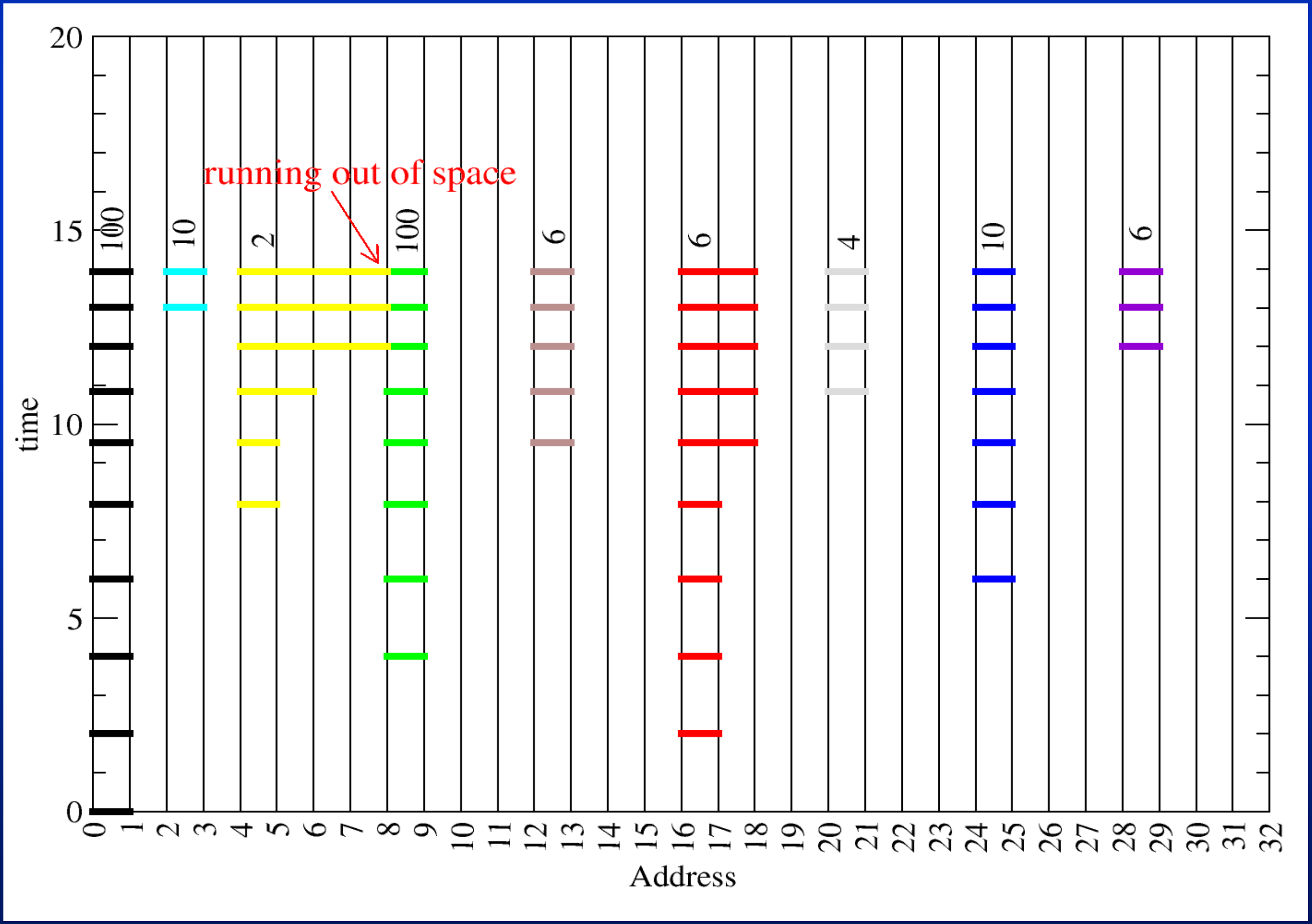


Simulations

- Simulate customer requests and allocation decisions
- Profile of customers:
  - Can take any form of growth, i.e., exponential growth
  - Growth rates are Gaussian distribution
  - Random sequence of incoming customers
- Measurement metrics:
  - Before any collision (without collision)
    - Number of customers served without fragmentation
    - Percentage of total address space utilized without fragmentation
  - Allowing collision: (with collision)
    - percentage of fragmentation

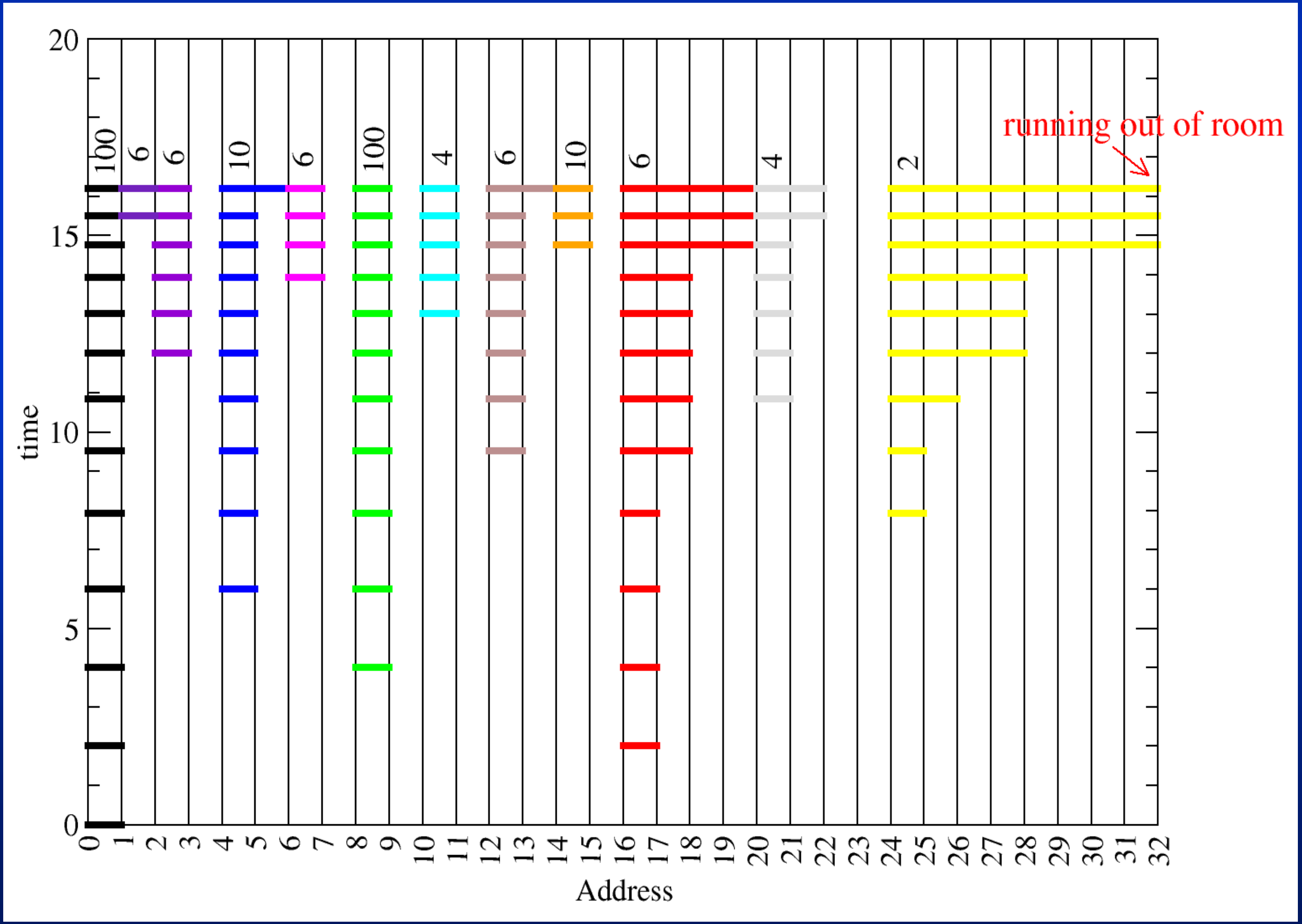
An Illustration: Bisection Scheme

- Address occupation of customers with various growth rates joining at different time.

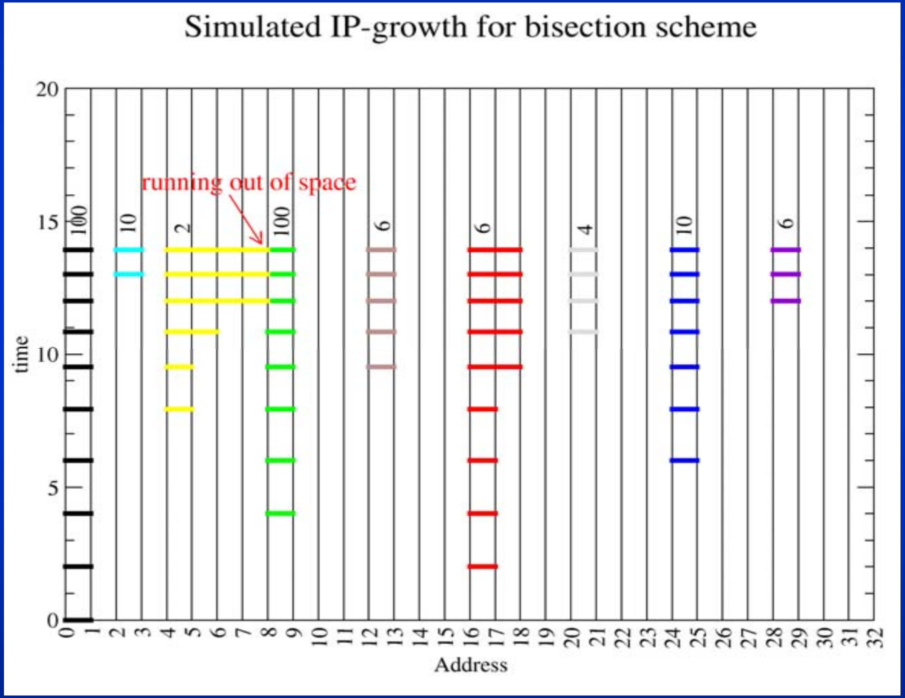


An Illustration: Growth-based Scheme

- More customers are served without collision

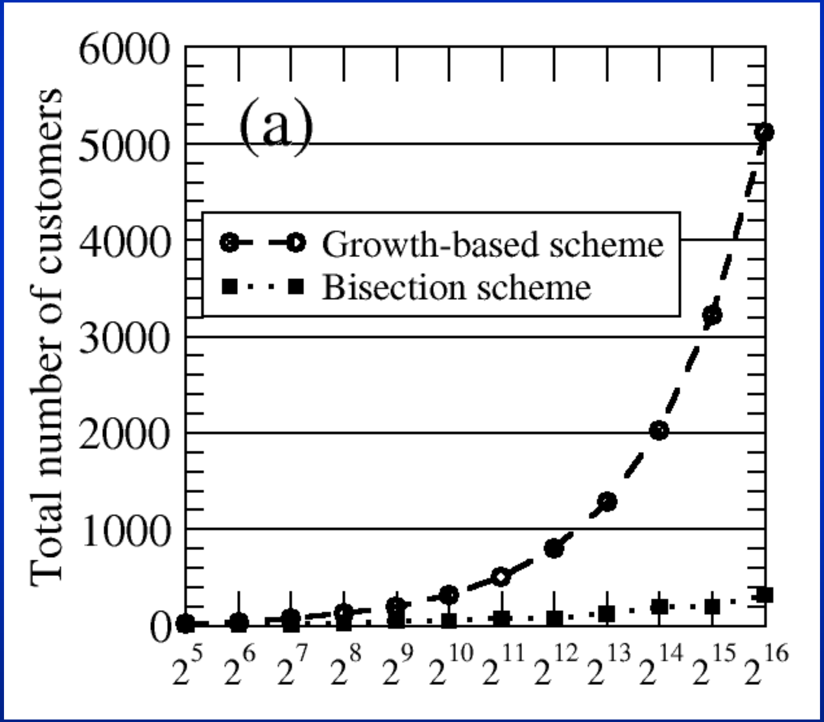


Comparisons: bisection vs. growth-based

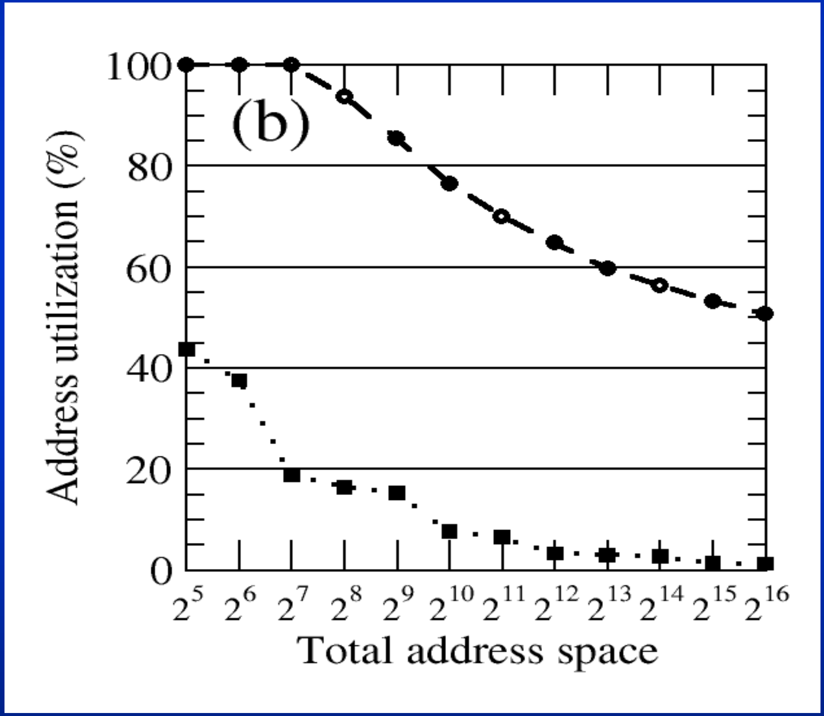


Simulation Results -- without collisions

Number of customers served



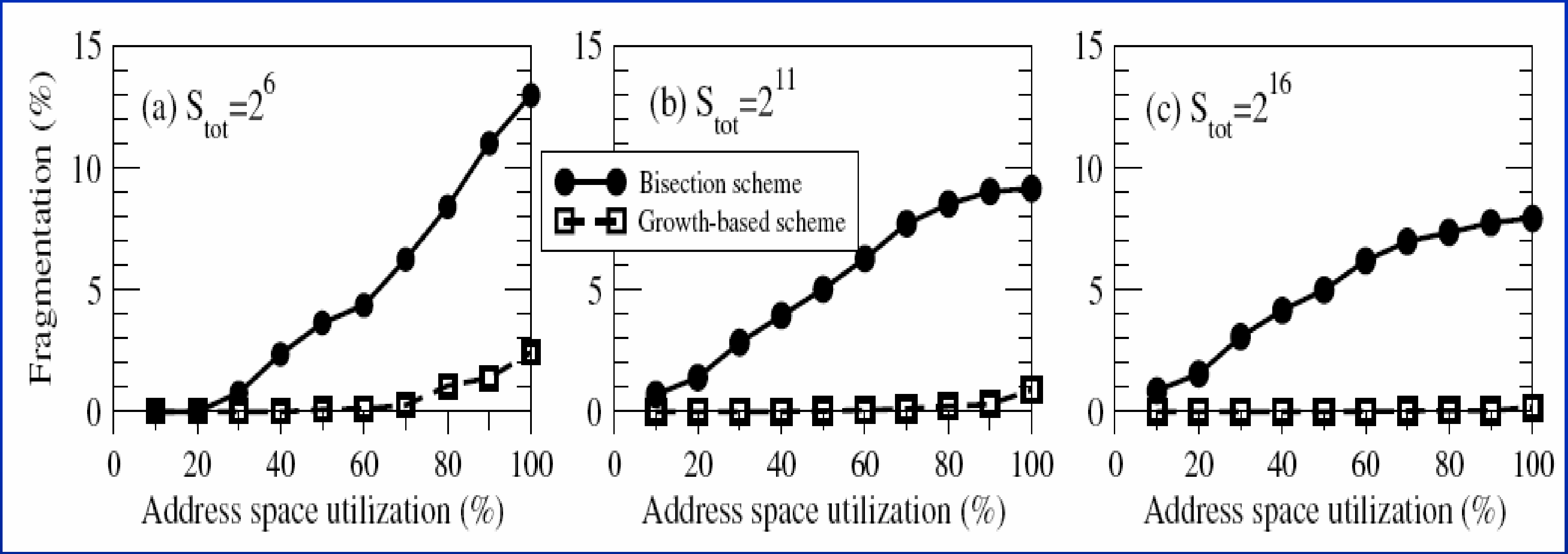
Address space occupied



- The more number of customers served and the more address space is allocated without fragmentation, the better the allocation algorithm is.

Simulation Results -- with collisions

Fragmentation Percentage



- Growth-based scheme dramatically reduces fragmentation compared to bisection scheme for any address space utilization.

Theoretical Analysis: Bisection

number of customers : $n(t) = n_0 2^{t/\tau_n};$

space needed for customer *i* at time *t* : $L_i(t) = L_{0,i} 2^{(t-t_{0,i})/\tau_i};$

ave space available for each customer : $S(t) = \frac{S_{tot}}{n(t)} = \frac{S_{tot}}{n_0} 2^{-t/\tau_n};$

n_0 : initial number of customers;

$1/\tau_n$: the increase rate of number of customers;

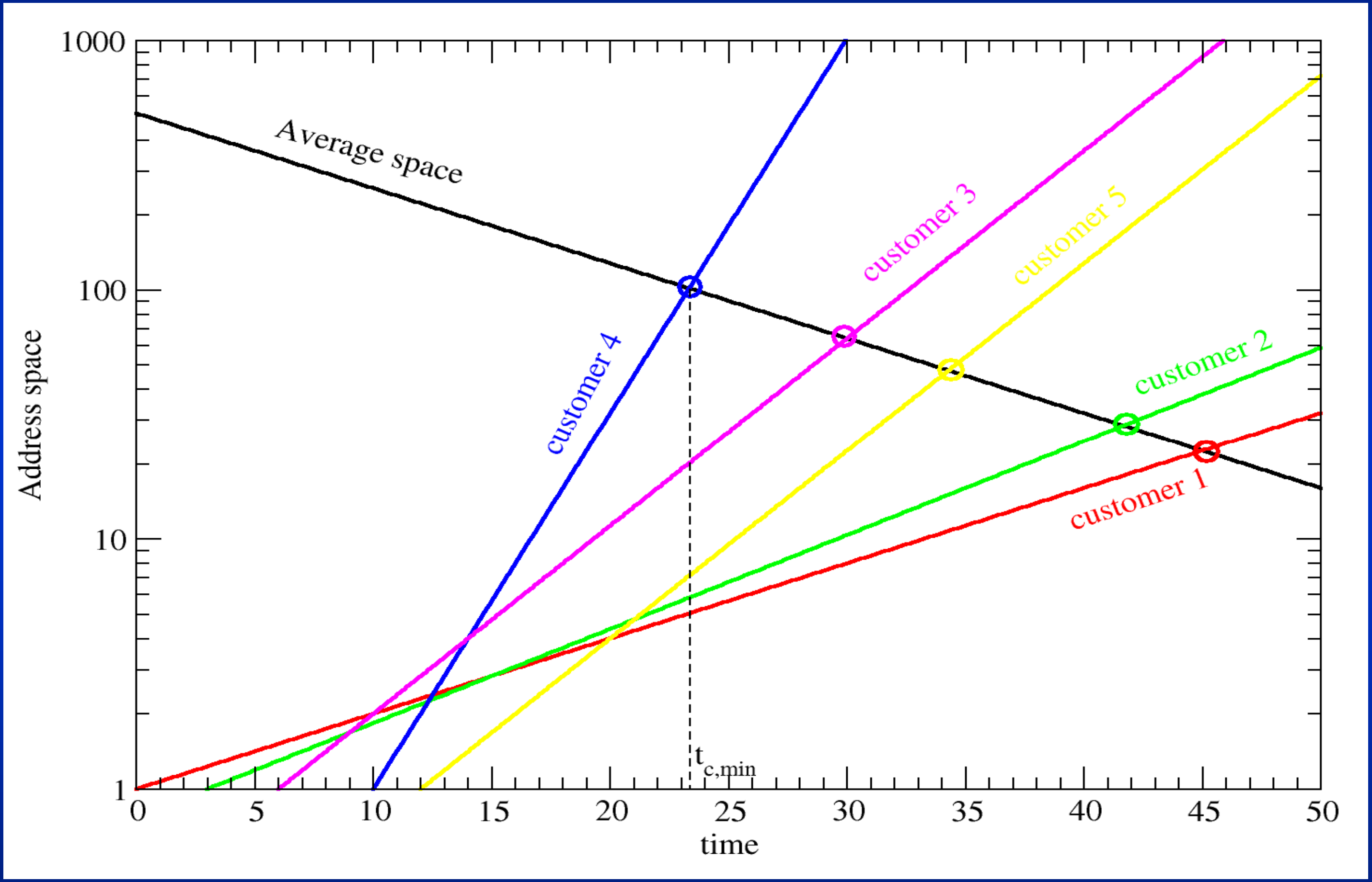
S_{tot} : total address space;

$L_{0,i}$: initial space required for customer *i*;

$t_{0,i}$: the first time customer *i* enters the address space;

$1/\tau_i$: the growth rate of customer *i*;

Theoretical Analysis: Bisection



- For customers with various growth rates and incoming time, the first collision time ($t_{c,min}$) is used to calculate number of customers served and address space occupied without fragmentation.

Bisection

- collision occurs when $L_i(t) = S(t)$;

$$t_{c,i} = \frac{\tau_i \tau_n}{\tau_i + \tau_n}\left[-\frac{\tau_n}{\tau_i}\log_2(n_0 \times P_i) + \log_2\left(\frac{S_{tot}}{n_0 L_{0,i}}\right)\right];$$

- time it takes before the first collision occurs:

$$t_{c,\min} = \min\{t_{c,i}, i = 1,\dots,n\};$$

- when the first collision occurs:

number of customers served: $n(t_{c,\min}) = n_0 2^{t_{c,\min}/\tau_n}$;

total address space used:

$$S_{used} = \sum_i \int_0^{t_{c,\min}} L_i(t_{c,\min} - t) P_i \frac{dn(t)}{dt} dt = \sum_i n_0 L_{0,i} P_i \tau_i \frac{2^{\frac{t_{c,\min}}{\tau_n}} - 2^{\frac{t_{c,\min}}{\tau_i}}}{\tau_i - \tau_n}.$$

Theoretical Analysis: Growth-based

number of customers : $n(t) = n_0 2^{t/\tau_n}$;

space needed for the i^{th} class A customer : $L_i(t) = 2^{(t-t_{0,i})/\tau_l}$;

space available for the i^{th} class A customer : S_i;

n_0 : initial number of customers;

$1/\tau_n$: the increase rate of number of customers;

S_{tot} : total address space;

$t_{0,i}$: the time the i^{th} class A customer enters the address space;

$1/\tau_l$: the growth rate of all class A customers;

Growth-based

collision occurs when $L_i(t) = S_i;$

$$t_{c,i} = \tau_n \log_2 \frac{i}{n_0 P} + \tau_l \log_2 \left[S_{tot} \frac{P}{i} \prod_{k=1}^{i-1} (1 - \frac{P}{k}) \right].$$

which customer collides first :

$$t_{c,i} - t_{c,i-1} = \tau_n \log_2 \frac{i}{i-1} + \tau_l \log_2 \left[\frac{i-1}{i} (1 - \frac{P}{i-1}) \right]$$

$$\approx (\tau_n - \tau_l) \log_2 \frac{i}{i-1}, \qquad P << 1.$$

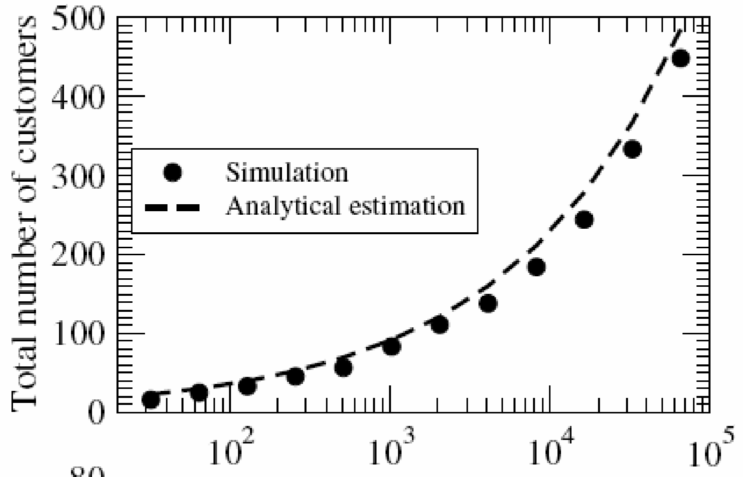
for $\tau_n > \tau_l, \quad t_{c,i} > t_{c,i-1}, \Rightarrow$ the 1^{st} one hits boundry first.

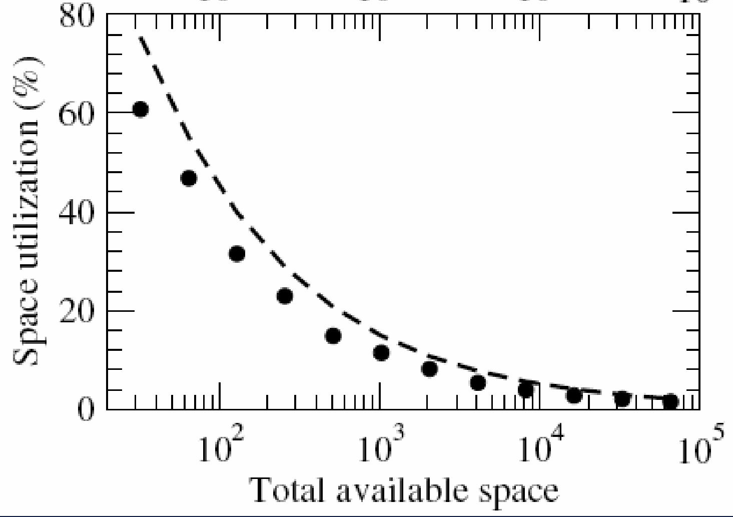
$$t_{c,\min} = t_{c,1} = -\tau_n \log_2 (n_0 P) + \tau_l \log_2 \left(S_{tot} P \right).$$

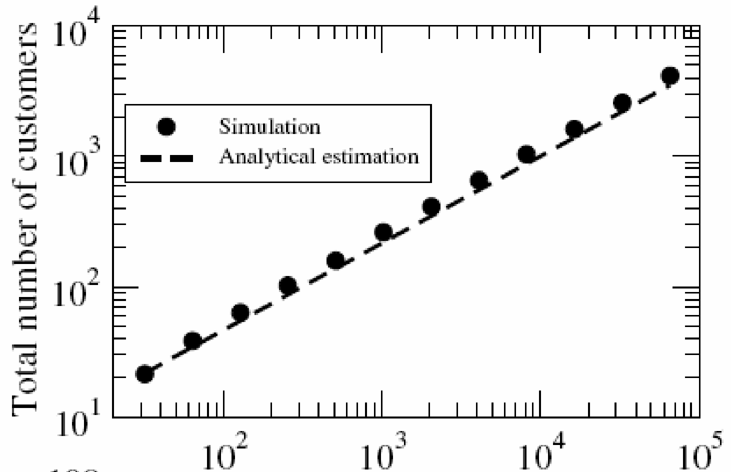
Theoretical Analysis vs. Simulations

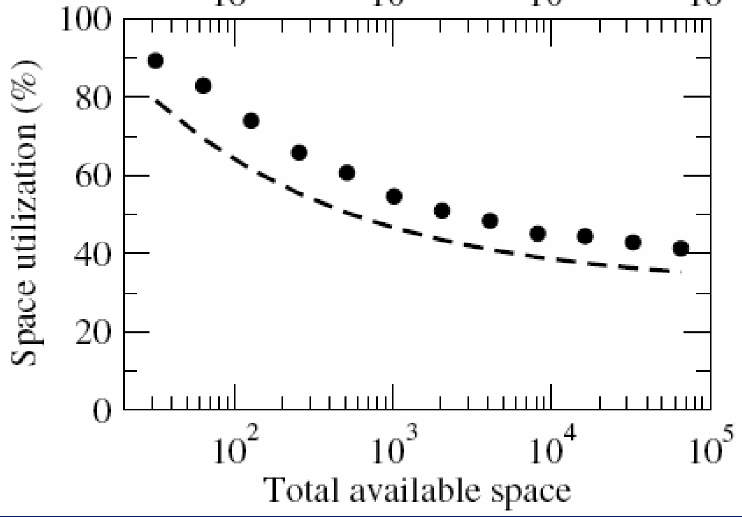
Bisection

Growth-based









Discussions

- Growth rate estimation
  - Can be in any form:
    - the percentile of current occupancy, the probability of potential growth, the total projected address space, the probability of extra bits needed, the projected growth rate (in any functional forms of time)
    - draft-huston-ip6-allocation-unit-00.html: linear growth rate
- Better estimation leads to more gain, the worst cases are:
  - Wrong estimations:
    - Don’t have to be exact
    - Dynamic optimization based on given info
    - Still space to grow
  - Fast growth comes in too late:
    - Less gain or no gain

Conclusions

- Quantitative modeling of address allocation schemes:
  - Simulations
  - Theoretical analysis
- A growth-based IPv6 address allocation scheme:
  - Utilize customer information
  - Reduce address fragmentation, control routing table size, increase lookup and routing efficiency
  - Increase efficiency of address space usage

Thank You

mei@cisco.com