Pseudo-Address Generation Algorithm of Packet Destinations for Internet Performance Simulation

Masaki Aida, Tetsuya Abe

Abstract—**This paper investigates the stochastic property of the packet destinations and proposes an address generation algorithm which is applicable for describing various Internet access patterns. We assume that a stochastic process of Internet access satisfies the stationary condition and derive the fundamental structure of the address generation algorithm. Pseudo IP-address sequence generated from our algorithm gives dependable cache performance and reproduces the results obtained from trace-driven simulation. The proposed algorithm is applicable not only to the destination IP address but also to the destination URLs of packets, and is useful for simulation studies of Internet performance, Web caching, DNS, and so on.**

Keywords—**Internet, Destination address, LRU stack, World Wide Web, Caching.**

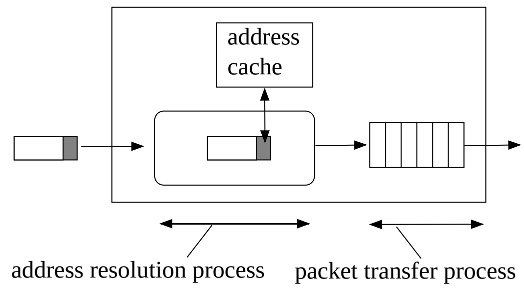


Fig. 1. Packet forwarding in a router.

I. Introduction

IN high-speed data networks, it is necessary to forward packets in a few microseconds or sub-microseconds. Packet forwarding operations include searching for the required address through an address look-up table. Although the similar searching operations appear also in the designs of DNS and Web content caches, the design of routers in high-speed networks requires the most time-sensitive operations.

To improve and enhance the IP-based Internet-type network, various network architectures, such as IP over ATM [1], NHRP [2], and so on, have been proposed and studied. Packet forwarding operations at a router consist of the following two processes.

- **address resolution process**
  This process finds a link address, an ATM address for example, of the destination and/or the next-hop for the destination IP address of the packet.
- **packet transfer process**
  This process transfers the packet by using the link address obtained from the address resolution process.

Packet processing in the routers within the connection-less network specifically includes the address resolution process. This process is indispensable for the packet transfer over physical links such as ATM, and the efficiency of the address resolution process has significant influence on the throughput [3]–[5].

Performance of the address resolution process depends on the number of arriving packets regardless of their lengths in byte. On the other hand, performance of the packet transfer process depends on the amount of bytes to be transferred regardless of the number of arriving packets. Performance of packet forwarding operations at a router, therefore, comes from the combined performance of the two different processes (Fig. 1). In this complicated situation, computer simulations and experimental test-beds probably are powerful approaches to evaluate router performance.

These approaches require a pseudo traffic generator which is based on a traffic model of Internet access behavior. Interarrival distribution of the packets and packet length distribution are a significant part of the traffic model. They have been investigated in [6], [7]. The other significant part of the traffic model is the assignment of the appropriate destination address to a packet generated at a pseudo traffic generator. Since address resolution process includes cache memory look-up of packet address information, performance of this process is strongly influenced from the diversity of the destination addresses of the packets.

This paper shows an address generation algorithm which is applicable to various Internet access patterns, *e.g.*, at access networks, backbone networks, and a WWW server. In addition, the algorithm can be applicable not only to the destination IP address but also to the destination URLs of packets. Therefore, it is expected that the proposed algorithm is useful not only for the design of routers, but also for the designs of DNS servers, contents cache servers, and so on.

This paper is organized as follows.

In Sec. II, we introduce the notion of the inverse stack growth function, used in computer memory reference, for describing the destination address generations.

In Sec. III, we assume that the inverse stack growth function has time-translation invariance in order to represent a *stationary* generation process of the address, and we derive the result that the address generation is described by an LRU stack model. In addition, we show the relationship between the probability of address generation and the inverse stack growth function.

In Sec. IV, we show address generation algorithm by using the results of Sec. III. As well, we show that the address sequence derived from the proposed algorithm has a time-translation invariance and reproduces the inverse stack growth function.

In Sec. V, we investigate the working set size behaviors of actual Internet accesses and show the validation of time-translation

M. Aida is with the Traffic Research Center, NTT Advanced Technology Corporation (NTT-AT), Musashino-shi, 180-0006, Japan. E-mail: aida@m.ieice.org .

T. Abe is with NTT East R&D Center. E-mail: tetsuya.abe@rdc.east.ntt.co.jp .

invariance and an asymptotic power law of the inverse stack growth function. These properties are applicable not only to the destination IP addresses, but also to the destination URLs, at access networks, backbone networks or at a WWW server.

In Sec. VI, we assume the power law of the inverse stack growth function, and investigate the property of address sequence generated from the proposed algorithm.

In Sec. VII, we show the effectiveness of the proposed address generation algorithm through performance evaluations of the address cache.

Section VIII shows conclusions and the residual issues.

II. PRELIMINARY

A. Notations and Definitions

Internet access behaviors include similar issues in computer memory access patterns. We, therefore, prepare the framework for Internet access models, using well-known notions in computer memory access models.

Time t is incremented to $t+1$ when an access occurs, and $t = 0, \pm 1, \pm 2, \ldots$.

- working set: $W(t, \tau)$
  The set whose elements are distinct addresses generated during a period $[t-\tau, t)$ (*i.e.*, $\{t-\tau, t-\tau+1, \ldots, t-1\}$). In case for $\tau < 0$, the period is $(t, t-\tau]$.
- working set size: $w(t, \tau)$
  Size of a working set $W(t, \tau)$, *i.e.*, the number of elements of $W(t, \tau)$.
- inverse stack growth function (ISGF) [8]: $f(t)$
  Expectation value of the number of distinct addresses generated during a period $(t, t+\tau]$,
  $$f(t, \tau) := \mathrm{E}[w(t, -\tau)]. \quad (1)$$
  In addition, stack growth function (SGF), $g(t, k)$, denotes the expectation value of the number of accesses such that the number of distinct addresses is k, *i.e.*,
  $$f(t, \tau) = k \quad \leftrightarrow \quad g(t, k) = \tau, \quad \text{or} \quad (2)$$
  $$g = f^{-1}. \quad (3)$$

Note that because ISGF f is defined only on integer, SGF g cannot be defined directly. The consistent way to obtain the relation between ISGF and SGF is shown in [8]. Hereafter, we regard ISGF and SGF as functions defined on the real number with respect to τ and k, respectively.

B. Reference Models for Computer Memory

In computer memory references, it is well known that the concept about *locality* of a reference pattern appears. The locality (especially, temporal locality) implies a high probability of reuse. A similar concept also appears in Internet access [3].

This subsection reviews typical memory reference models arranged for Internet access models [3].

- independent reference model (IRM)
  This model assumes accesses are independent. The probability that a new access has address i is determined by the address i (and the probability is denoted as p_i). Accessed address sequence is generated by i.i.d. and the distribution p_i. Because recent accessed addresses do not give any information about the next accessed address, IRM cannot capture locality.
- working set model (WS model) [9]
  This model assumes that the addresses accessed in the last W accesses are highly likely to be accessed. The WS model, therefore, captures locality. The interval W is called the working set window size. For a given working set window size W, the smaller working set size $w(t, W)$ means stronger locality.
- least recently used (LRU) stack model [10]
  An LRU stack is a list of addresses sorted in order according to the times of their most recent access. So, the most recently accessed address is at the top of the stack and the least recently accessed address is at the bottom. The probability that the newly accessed address is the same as the address at k-th position in the LRU stack is determined by position k, and its probability is denoted as a_k. Usually, a_k decreases with respect to increases in k. Because the most recently accessed address is at the top of the LRU stack, the probability that the next accessed address is the same is high. The LRU stack model, therefore, captures locality.

In performance evaluation of address resolutions, the performance of the address cache is an essential object [3], [5]. Cache performance strongly depends on locality of Internet accesses. Because IRM cannot take locality into account, it is an insufficient model for Internet access modeling.

Since we can evaluate cache performance if ISGF, f, is given [5], ISGF and WS models are important. Hereafter, from natural and basic assumptions on ISGF and investigation of the working set, we show that the address generation probability must obey an LRU stack model. In addition, we show the relationship between the address generation probability a_k and ISGF f.

III. BASIC STRUCTURE OF ADDRESS GENERATION PROBABILITY

In this section, we assume that the ISGF f satisfies natural and basic properties, and investigate the basic structure of an address generation probability.

A. Assumption

We assume the following property of ISGF.

Time-translation invariance:
ISGF $f(t, \tau)$ is independent of the time t which denotes time when we start to measure address generation. For any t, s,
$$f(t, \tau) = f(s, \tau). \quad (4)$$
From this assumption, we denote $f(\tau) := f(t, \tau)$.

Validation of time-translation invariance is discussed in Sec. V through experimental data.

In queueing models, the typical and usual traffic model as an input is described by a stationary stochastic process. On the other hand, the notion of a *stationary process* of address generation is not well-defined yet. However, it is natural to accept that if the address generation process is *stationary*, the ISGF should have a time-translation invariance. Thus, the time-translation

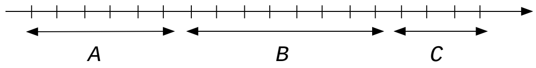


Fig. 2. Time intervals A, B and C.

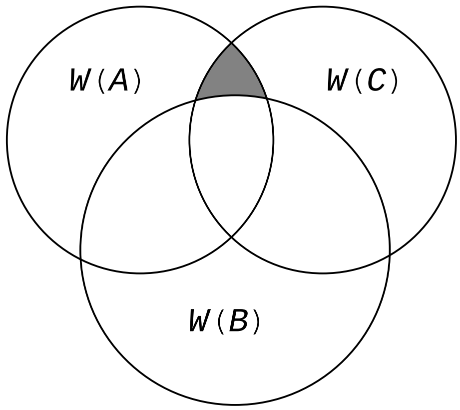


Fig. 3. Venn's diagram of working sets.

invariance should be, at least, a necessary condition of the stationarity. We assume it in order to produce a stationary process of the address generation.

Incidentally, we can consider other possibilities for the stationarity condition. For example, we may adopt the assumption that the working set size distribution [11],

$$p(k,\tau) := \Pr\{w(t,\tau)=k\}, \tag{5}$$

is independent of t. Although this seems to be a stronger stationarity condition than the time-translation invariance of ISGF f, stationarity of (5) is obtained from (4) as a result. This is because the assumption (4) completely determines the fundamental structure of the address generation algorithm, as shown in the following sections.

B. Time-Reversal Invariance

Let us consider $f(-\tau)$. This means the expectation value of the number of distinct addresses generated in the period which is from a time t until the past time $t-\tau$ through the backward derection of time. We have time-reversal invariance for ISGF,

$$\begin{aligned} f(-\tau) &= f(t,-\tau) \\ &= f(t-\tau-1,\tau) \\ &= f(\tau). \end{aligned} \tag{6}$$

C. Address Generation Probability

We consider three adjacent periods A, B, and C (Fig. 2). Let the sets of distinct addresses generated in these periods, *i.e.*, working set, be $W(A)$, $W(B)$, and $W(C)$, respectively. Figure 3 denotes Venn's diagram of these working sets.

Here, we focus on the subset W^* whose elements are also elements of both $W(A)$ and $W(C)$, but are not elements of $W(B)$, *i.e.*, the hatched part in Fig. 3,

$$W^* := \{W(A)\cap W(C)\}\setminus W(B). \tag{7}$$

The size of W is obtained as

$$\begin{aligned} |W^*| &= |\{W(A)\cap W(C)\}\setminus W(B)| \\ &= \{|W(B)\cup W(C)| - |W(B)|\} \\ &\quad - \{|W(A)\cup W(B)\cup W(C)| \\ &\quad - |W(A)\cup W(B)|\}\,. \end{aligned} \tag{8}$$

We can choose m, n and 1 accesses as the periods A, B, and C, respectively (Fig. 4). Then we have

$$\begin{aligned} |W^*| &= \{w(t,n+1) - w(t-1,n)\} \\ &\quad - \{w(t,n+m+1) - w(t-1,n+m)\}. \end{aligned} \tag{9}$$

Applying the time-reversal invariance, we have

$$\begin{aligned} \mathrm{E}\,[|W^*|] &= \{f(-(n+1)) - f(-n)\} \\ &\quad - \{f(-(n+m+1)) - f(-(n+m))\} \\ &= \{f(n+1) - f(n)\} \\ &\quad - \{f(n+m+1) - f(n+m)\}. \end{aligned} \tag{10}$$

The physical meanings of $\mathrm{E}\,[|W^*|]$ are as follows. $|W^*|$ is 1 only when the address X generated at a time (period C) is not in $W(B)$ but in $W(A)$. Otherwise, $|W^*|$ is 0. Thus, $|W^*|$ is a random variable and can be denoted using indicator function $\mathbf{1}\{\cdot\}$ by

$$|W^*| = \mathbf{1}\{X\notin W(B), X\in W(A)\}. \tag{11}$$

Therefore, $\mathrm{E}\,[|W^*|]$ means the probability of $|W^*|=1$,

$$\mathrm{E}\,[|W^*|] = \Pr\{X\notin W(B), X\in W(A)\}. \tag{12}$$

Next, we choose n such as $f(n)=k-1$ ($k=1,2,\ldots$) and choose m such as $f(n+m)=k$. Substituting $n=g(k-1)$ and $n+m=g(k)$, (10) gives

$$\begin{aligned} \mathrm{E}\,[|W^*|] &= \{f(g(k-1)+1) - (k-1)\} \\ &\quad - \{f(g(k)+1) - k\}\,. \end{aligned} \tag{13}$$

Equation (13) means the probability that the newly accessed address, X, is identical with the k-th most recently accessed address.

Address generation probability:

Here, we define a_k as

$$\begin{aligned} a_k &:= \{f(g(k-1)+1) - (k-1)\} \\ &\quad - \{f(g(k)+1) - k\}\,. \end{aligned} \tag{14}$$

This means that the probability that X is identical with the most recent accessed address is a_1, the probability that X is identical with the 2nd most recently accessed address is a_2, …, and the probability that X is identical with the k-th most recently accessed address is a_k. In other words, address generation must obey the LRU stack model whose probability is (14).

If $\{a_k;\ k=1,2,\ldots\}$ is the probability, ISGF f must satisfy $a_k\geq 0$ and

$$\sum_{k=1}^{\infty} a_k = 1, \tag{15}$$

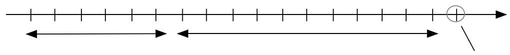


Fig. 4. Time intervals.

that is, $\{f(g(k)+1)-k\}$ is monotonously decreasing with respect to increases in k $(=0,1,2,\ldots)$ and satisfies

$$\lim_{k\to\infty}\{f(g(k)+1)-k\}=0. \qquad (16)$$

IV. Address Generation Algorithm

This section shows an address generation algorithm by applying the results of the above section. The proposed algorithm gives a pseudo-address sequence, a sequence of integer.

A. Address Generation Procedure

We define the generative LRU stack vector $\mathbf{L}(i,m)$ as follows. At the time immediately before i-th $(i=0,1,2,\ldots)$ address generation, we denote the most recently accessed address as x_1, the 2nd most recently accessed address as x_2, ..., and the k-th most recently accessed address as x_k. Then

$$\mathbf{L}(i,m):=\{x_1,x_2,\ldots,x_m\}. \qquad (17)$$

Here, the number of components of $\mathbf{L}(i,m)$, m, is called the depth of $\mathbf{L}(i,m)$. Initially, the depth is chosen as 0, *i.e.*, $m=0$ for $\mathbf{L}(0,0)$. Then m means the number of distinct address generations.

Let X_i denote the address of the i-th access. Address sequence $\{X_i; i=0,1,2,\ldots\}$ is generated by the following procedure.

1. Initially, we choose $X_0=1$, $\mathbf{L}(1,1)=\{1\}$ and $i=1$.
2. Determine the number j as a realization of the i.i.d. random variable J which obeys the distribution
$$\Pr\{J=k\}=a_k \qquad (k=1,2,3,\ldots), \qquad (18)$$
by using (14).
3. For the depth m of $\mathbf{L}(i,m)$, if $m<j$, then assign the address of new access as
$$X_i=m+1, \qquad (19)$$
and update $\mathbf{L}(i+1,m+1)$ and increment $i\leftarrow i+1$. Return to 2.
4. For the depth m of $\mathbf{L}(i,m)$, if $m\geq j$, then assign the address of new access as
$$X_i=x_j, \qquad (20)$$
and update $\mathbf{L}(i+1,m)$ and increment $i\leftarrow i+1$. Return to 2.

Because the number j can be obtained using a binary search algorithm, high-speed address generation is possible.

B. Time-Translation Invariance and Asymptotic Power Law

When the depth of the generative LRU stack vector $\mathbf{L}(t,m)$ is m at a time t, we consider the probability that the working set size $w(t,n)$ increases, *i.e.*,

$$b_m:=\Pr\{w(t+1,n+1)=m+1|\,w(t,n)=m\}. \qquad (21)$$

This means the generation probability of a brand-new address which does not exist in $\mathbf{L}(t,m)$. So, it is obtained as

$$\begin{aligned} b_m &= \sum_{k=m+1}^{\infty} a_k \\ &= f(g(m)+1)-m. \end{aligned} \qquad (22)$$

Equation (22) is valid for all m $(m=0,1,2,\ldots)$. The probability of a brand-new address generation depends only on the depth of the generative LRU stack vector. Because the same depth gives the same probability, b_m is independent of the time when we start to measure address generations. Thus, the address sequence obtained from our algorithm reproduces the time-translation invariance of ISGF.

Let us consider the time that the depth m of the generative LRU stack vector $\mathbf{L}(t,m)$ is incremented to $m+1$. Because the time obeys a geometrical distribution, the mean time is $1/b_m$. Therefore, SGF g is obtained as

$$g(k)=\sum_{m=0}^{k-1}\frac{1}{b_m}. \qquad (23)$$

We define the slope of f as the difference

$$\begin{aligned} \left.\frac{\Delta f}{\Delta g}\right|_m &:= \frac{f(g(m)+1)-f(g(m))}{(g(m)+1)-g(m)} \\ &= b_m. \end{aligned} \qquad (24)$$

From (23), we have

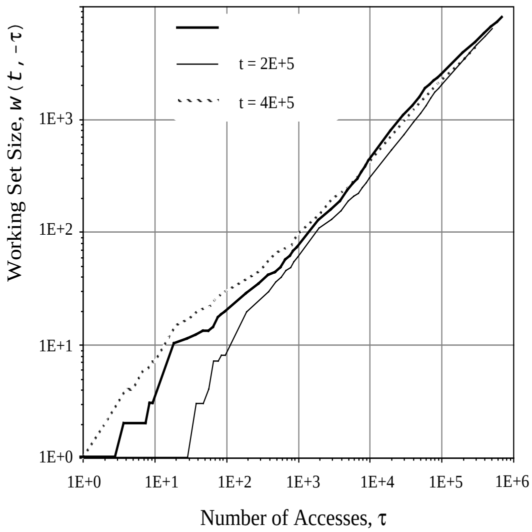
$$g(k)=\sum_{m=0}^{k-1}\left[\left.\frac{\Delta f}{\Delta g}\right|_m\right]^{-1}. \qquad (25)$$

This means the address sequence obtained from the proposed algorithm reproduces ISGF, f.

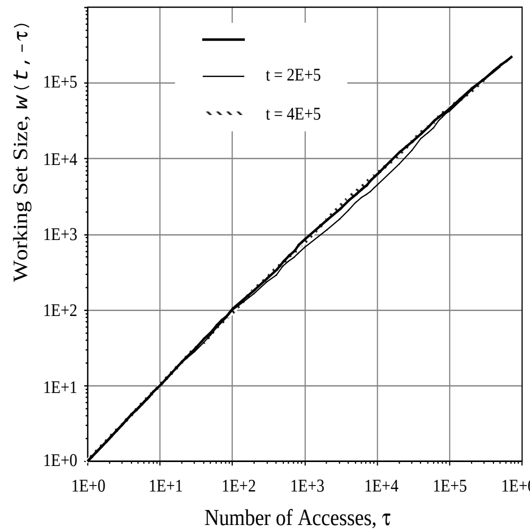
V. Working Set Size Behaviors of Internet Accesses

In order to obtain pseudo-address sequence by using the proposed algorithm in practice, it is necessary to determine ISGF f. In this section, we investigate asymptotic property of working set size behaviors of Internet accesses from experimental data, and show that a power law is widely applicable. Simultaneously, validation of the time-translation invariance is discussed.

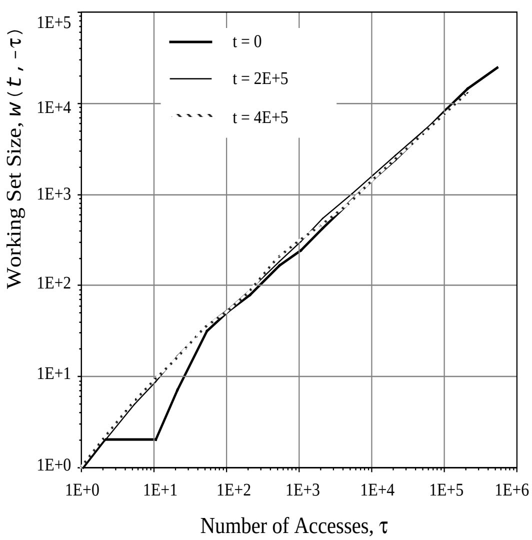
In most cases, it is well known that the destination addresses or URLs are characterized by Zipf-type distributions [5], [12]–[14] (strictly, parts of them are Lotoka-type distributions [15]). Although, in some cases, we can calculate ISGF by using Zipf-type distribution [5], [16], it is not always feasible. Therefore, it is required to determine ISGF from generally applicable property with respect to Internet accesses.



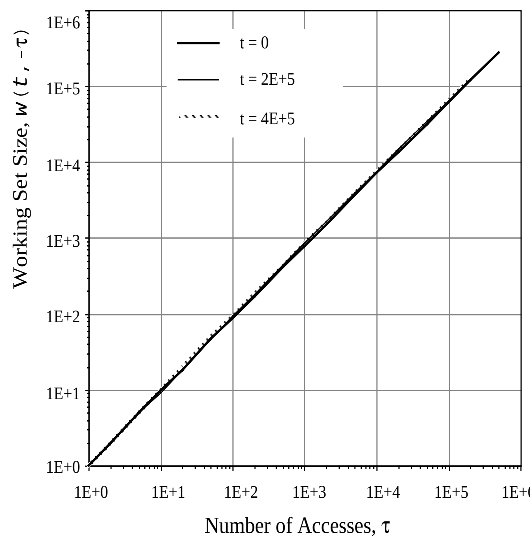
(a) For the destination IP addresses
(NTT labs., Mar., 1997).



(b) For the destination URLs
(NTT labs., Mar., 1997).



(c) For the destination IP addresses
(NLANR, Jul. 30, 1999).



(d) For the destination URLs
(NLANR, Jul. 30, 1999).

Fig. 5. Working set size behaviors of destinations (NTT labs. and NLANR).

A. Working Set Size Behaviors and Time-Translation Invariance of ISGF

A.1 Access Logs from Proxy Server of NTT Laboratories

As an example of access networks of the Internet, we investigate access logs of a proxy server at NTT Laboratories. Logs are for 15 successive days in March 1997.

Figure 5 (a) shows the relationship between working set size, $w(t,-\tau)$, and the number of accesses, τ, for the destination IP addresses in log-log scale. Each line indicates $w(t,-\tau)$ for $t=0$, 2×10^5, and 4×10^5, respectively. Similarly, Fig. 5 (b) shows the relationship between working set size, $w(t,-\tau)$, and the number of accesses, τ, for the destination URLs in log-log scale. Each figure shows $w(t,-\tau)$ is independent of t for large τ. From the law of large number, asymptotic behavior of the working set size can be related to the behavior of ISGF, as

$$w(t,-\tau) \sim f(t,\tau) \qquad (\tau \gg 1). \tag{26}$$

Thus, these figures imply the time-translation invariance of ISGF, at least in the asymptotic region, $\tau \gg 1$.

A.2 Access Logs from Cache Server of NLANR

As an example of backbone networks of the Internet, we investigate access logs of a cache server at NLANR [17]. Logs are for July 30, 1999.

Figure 5 (c) shows the relationship between working set size, $w(t,-\tau)$, and the number of accesses, τ, for the destination IP addresses in log-log scale. Each line indicates $w(t,-\tau)$ for $t=0$, 2×10^5, and 4×10^5, respectively. Similarly, Fig. 5 (d)

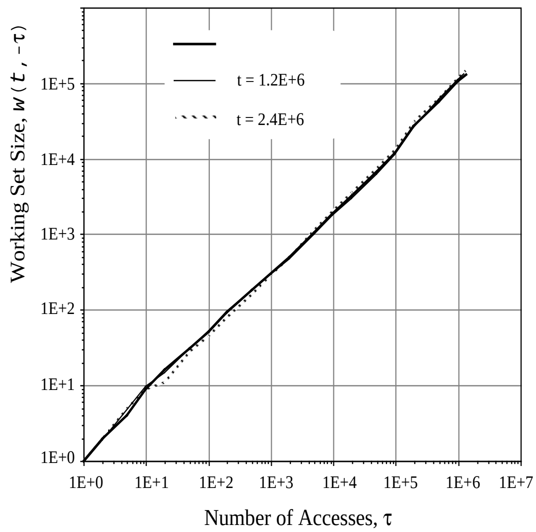


Fig. 6. Working set size behaviors of originations (NTT DIRECTORY, Oct., 1997).

shows the relationship between working set size, $w(t, -\tau)$, and the number of accesses, τ, for the destination URLs in log-log scale. These figures also imply the time-translation invariance of ISGF, at least in the asymptotic region, $\tau \gg 1$.

A.3 Access Logs of NTT DIRECTORY Server

As an example of WWW servers of the Internet, we investigate access logs of a directory site, NTT DIRECTORY [18]. Logs are for 31 successive days in October 1997.

Figure 6 shows the relationship between working set size, $w(t, -\tau)$, and the number of accesses, τ, for the origination IP addresses in log-log scale. Each line indicates $w(t, -\tau)$ for $t = 0$, 1.2×10^6, and 2.4×10^6, respectively. This figure also implies the time-translation invariance of ISGF, at least in the asymptotic region, $\tau \gg 1$.

B. Asymptotic Power Law

From the above graphical observations, we can recognize that $f(\tau)$ satisfies a power law with respect to τ for $\tau \gg 1$,

$$f(\tau) \simeq \tau^{\alpha} \qquad (\tau \gg 1), \tag{27}$$

where α is a constant. A similar property is known in computer memory access [8]. This property is widely applicable, *e.g.*, for access networks, backbone networks, at a WWW server, and for the destination IP addresses and URLs.

VI. Address Generation Algorithm using Power Law

This section investigates pseudo-address generation by applying the power law of ISGF.

A. Address Generation Probability

We assume the asymptotic power law for $f(n)$ is applicable for all n, *i.e.*,

$$f(n) = n^{\alpha}. \tag{28}$$

Then, address generation probability (14) is obtained as

$$a_k = \left\{((k-1)^{1/\alpha}+1)^{\alpha} - (k-1)\right\} - \left\{(k^{1/\alpha}+1)^{\alpha} - k\right\}. \tag{29}$$

Because $a_k > 0$ and

$$\sum_{k=1}^{\infty} a_k = 1 - \lim_{k\to\infty} \left\{(k^{1/\alpha}+1)^{\alpha} - k\right\} = 1 \tag{30}$$

for $0 < \alpha < 1$, $\{a_k;\, k = 1, 2, \ldots\}$ is probability. The asymptotic behavior of a_k is expressed as

$$a_k = (1-\alpha)\, k^{-1/\alpha} \qquad (k \gg 1). \tag{31}$$

B. Time-Translation Invariance and Asymptotic Power Law

Using power law (28), the probability that the working set size increases, (22), is expressed as

$$b_m = \sum_{k=m+1}^{\infty} a_k = \left\{(m^{1/\alpha}+1)^{\alpha} - m\right\}. \tag{32}$$

Using the asymptotic behavior of b_m,

$$b_m = \alpha\, m^{-(1/\alpha-1)} + O(m^{-(2/\alpha-1)}), \tag{33}$$

and (23), we have

$$g(k) = \sum_{m=0}^{k-1} \left\{\frac{1}{\alpha} m^{1/\alpha-1} + O(m^{-1})\right\}. \tag{34}$$

Therefore,

$$\lim_{k\to\infty} \frac{\log g(k)}{\log k} = \frac{1}{\alpha}. \tag{35}$$

This means the address sequence obtained from our algorithm reproduces the asymptotic power law of ISGF,

$$f(n) = n^{\alpha} \qquad (n \gg 1). \tag{36}$$

VII. Cache Miss Ratio

A. Theoretical Values for Cache Miss Ratio

In performance evaluation for address resolution, performance of address cache is an essential object. Figure 7 shows an address processing model in a router. The model includes address cache with k of capacity (k address entries). If the address cache contains the entry corresponding to the destination address of the packet, the address resolution is immediately carried out. Otherwise, in the case of cache miss, the router must request an address resolution from a database that has comprehensive information. New address information is written over an entry of the address cache by using aging algorithm.

If we have ISGF f (or SGF g), we can evaluate the cache miss ratio for the following two classical aging algorithms [5]:

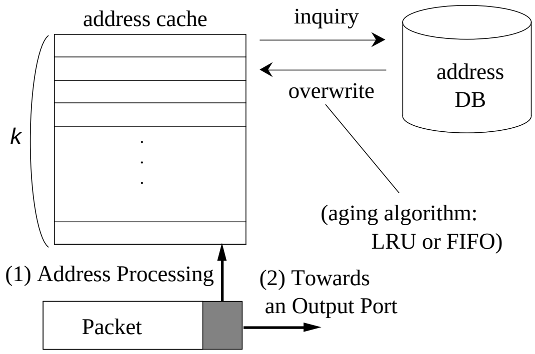


Fig. 7. Address processing model in a router.

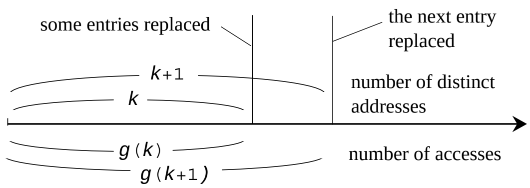


Fig. 8. Interval of entry replacements.

- Least Recently Used (LRU)
  When cache miss occurs, the new address data is written over the least recently used entry.
- First In First Out (FIFO)
  When cache miss occurs, the new address data is written over the current oldest entry.

For LRU, the address cache always has k different IP addresses which are the most recently accessed. The number of accesses that generate k different IP addresses is $g(k)$. Similarly, the number of accesses that generate $k+1$ different IP addresses is $g(k+1)$. Thus the number of accesses between two successive replaced entries, is $g(k+1) - g(k)$ (Fig. 8). Cache miss ratio, P_L, is obtained as

$$P_L = \frac{1}{g(k+1) - g(k)}. \tag{37}$$

Applying the power law (28), we have

$$P_L = \frac{1}{(k+1)^{1/\alpha} - k^{1/\alpha}}. \tag{38}$$

Incidentally, for $k \gg 1$, (38) is evaluated as

$$P_L = \alpha\, k^{1-1/\alpha} \qquad (k \gg 1). \tag{39}$$

This means the behavior of the cache miss ratio obeys a power law, asymptotically. The same behavior is well known in computer memory reference, and is called Chow's empirical power law [19], [20].

For FIFO, unlike LRU, the address cache does not always have k different IP addresses that are most recently accessed. Instead, let the expectation value of the number of accesses in the period, such that all cache entries are replaced, be N_x. In addition, we call the period generating N_x accesses the N_x-period.

Let us consider that the time axis is divided into N_x-periods as shown in Fig. 9. Then, the cache entries at a certain time and

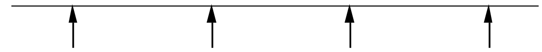


Fig. 9. N_x-periods.

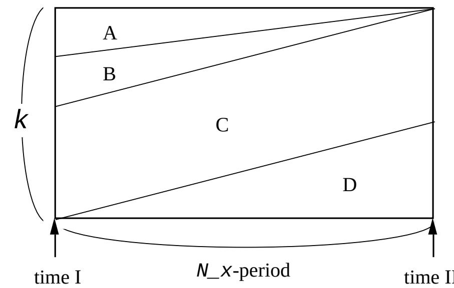


Fig. 10. Types of cache entries.

at N_x-period later can be categorized into the following four types (Fig. 10).

A: IP addresses which have corresponding cache entries in the cache at time I, but are not accessed during the following N_x-period, and their entries expiring at time II.

B: IP addresses which have corresponding cache entries in the cache at time I, with access during the following N_x-period, but their entries expiring at time II.

C: IP addresses which have corresponding cache entries in the cache at time I, and have cache entries at time II.

D: IP addresses which have no corresponding cache entries in the cache at time I, but were accessed during the next N_x-period and have cache entries at time II.

The number of IP addresses which are accessed during the last N_x-period and are not accessed during the next N_x-period is $f(2\,N_x) - f(N_x)$ (Fig. 11). In case of FIFO, whether a cache entry survives or not is independent of the frequency of access, and all entries have the same expiration rule. Therefore, the number of cache entries of the type A is

$$\frac{k}{f(N_x)}\,\{f(2\,N_x) - f(N_x)\}\,.$$

In addition, the number of IP addresses which have cache entries at time I and are accessed during the next N_x-period, is equal to the total number of the cache entries belonging to types B or C at time I. This is calculated as

$$k - \frac{k\,\{f(2\,N_x) - f(N_x)\}}{f(N_x)}.$$

Since all cache entries at time II were the missed addresses during the N_x-period from time I to II, the number of cache entries belonging to type B at time I is

$$f(N_x) - k.$$

Therefore, the number of cache entries belonging to type C (same at both time I and II) is

$$2\,k - f(N_x) - \frac{k\,\{f(2\,N_x) - f(N_x)\}}{f(N_x)}.$$

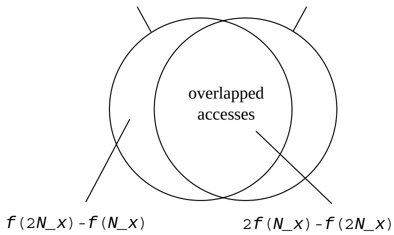


Fig. 11. Destination IP addresses generated in adjacent N_x-periods.

Let us consider the IP addresses that have cache entries at time I and are accessed during the next N_x-period, *i.e.*, the cache entries belonging to types B or C. Although they have cache entry at time I, all the entries do not make hits during the next N_x-period. Entries for IP addresses, which fail during the next N_x-period are restored as cache entries at time II. Since all entries expire under the same condition in FIFO, and N_x-period is the lifetime of a cache entry, the number of restored entries at time II is half the number of original entries at time I. In other words, the number of entries belonging to type C at time II is half that belonging to types B or C at time I. Therefore, N_x must satisfy the following condition:

$$4\,f(N_x) - 3\,k - \sqrt{9\,k^2 - 8\,k\,\{f(2\,N_x) - f(N_x)\}} \quad = \quad 0. \tag{40}$$

Using (40), N_x can be determined. Cache miss ratio for FIFO is obtained using N_x as

$$P_F = \frac{k}{N_x}. \tag{41}$$

Applying the asymptotic power law, (40) is written as

$$4\,N_x^{\alpha} - 3\,k - \sqrt{9\,k^2 - 8\,k\,\{(2\,N_x)^{\alpha} - N_x^{\alpha}\}} = 0. \tag{42}$$

B. Experimental Results

In order to verify the effectiveness of our algorithm described in Sec. IV, we compare the cache miss ratios. We show the cache miss ratio which is obtained from simulation with address sequences generated by our algorithm as input, and compare it with the results of trace-driven simulation and theoretical values derived from (38) and (41) using (42). The data used in the trace-driven simulation is access logs of a proxy server at NTT Laboratories (March, 1997). We focus on the destination IP addresses in the logs and, in this case, the value of α for the asymptotic power law is $2/3$ from graphical observation of Fig. 5 (a).

Figures 12 and 13 show the relationship between cache miss ratio and the capacity of address cache (the number of address entries), for LRU and FIFO, respectively. The horizontal axis denotes the capacity of cache and the vertical axis denotes the cache miss ratio. The black plot denotes the results from our algorithm and the white plot denotes the results from trace-driven simulations. The solid line denotes theoretical values. These figures show that our algorithm gives a dependable evaluation.

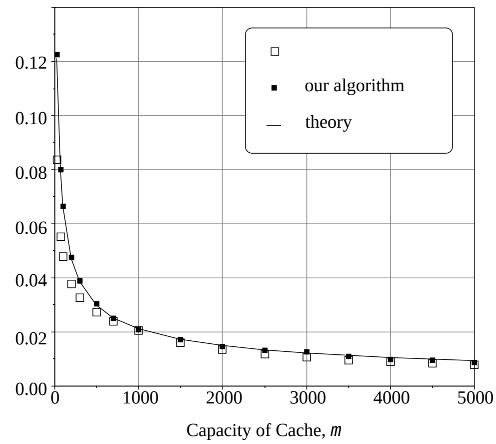


Fig. 12. Relations between the cache miss ratio and the capacity of the cache (LRU).

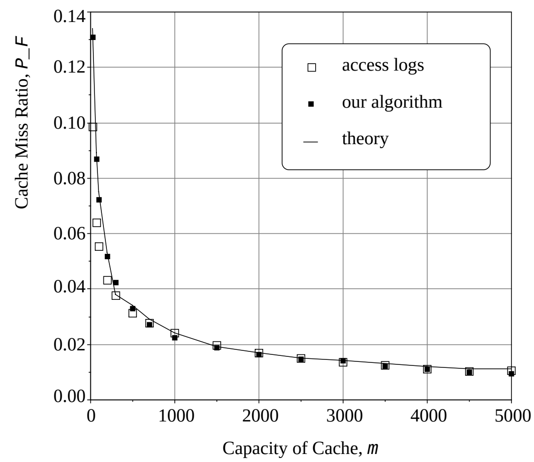


Fig. 13. Relations between the cache miss ratio and the capacity of the cache (FIFO).

VIII. Conclusion and Discussion

This paper has shown the structure of the address generation probability from the assumption of time-translation invariance of ISGF. The structure is an LRU stack model and has a simple relation between address generation probability and ISGF. In particular, we have revealed the following equivalence:

address generation process whose ISGF has a time-translation invariance.
$$\Updownarrow$$
LRU stack model whose probability is (14).

This paper also has shown the general property of ISGF, that is, the asymptotic power law is applicable to ISGF in a wide number of Internet access cases. In addition, by using these results, we have studied properties of the pseudo-address sequence obtained from our algorithm. This address sequence gives dependable evaluations of the cache miss ratio.

The proposed algorithm is democratic for all addresses in the sense that the probability of an address generation is determined only by the present stack position at the LRU stack. In actual Internet usage, some addresses are accessed with extremely high frequencies. This nature is often characterized by Zipf-type distributions. The address sequence generated from the proposed algorithm, however, does not reproduce these distributions. Let us consider an aging algorithm which writes the new address over the least frequently accessed entry. If we choose this aging algorithm for a cache, the address sequence obtained from our algorithm may not give a dependable evaluation. Of course, because this aging algorithm does not take locality into consideration, we can choose IRM for an address generation algorithm.

There are probably two reasons why our approach cannot reproduce Zipf-type distributions. Issues we should study are listed as follows:

- Validation of the time-translation invariance of ISGF.
- Validation of (28) assumed in (29). Equation (28) is only an asymptotic property.

The time-translation invariance seems to be a natural assumption. If we denied the assumption, the structure of the address generations loses the Markovian property whose state is described by the generative LRU stack vector. This may cause a situation where complicated approaches are required. On the other hand, it is preferable to have the power law adopted for ISGF replaced by a more strict relationship. However, the behavior of ISGF is probably inessential to reproduce Zipf-type distributions. In order that both time-translation invariance and Zipf-type distributions are compatible, an example of extension for the proposed algorithm has been discussed in [21]. These issues are for further study.

References

[1] M. Laubach and J. Halpern, “Classical IP and ARP over ATM,” *RFC 2225*, 1998.

[2] J. V. Luciani, D. Katz, D. Piscitello, B. Cole, and N. Doraswamy, “NBMA next hop resolution protocol (NHRP),” *RFC 2332*, 1998.

[3] R. Jain, “Characteristics of destination address locality in computer networks: A comparison of caching schemes,” *DEC Tech. Rep.*, DEC-TR-592, 1990.

TMWS99-11, 1999.

[5] M. Aida, N. Takahashi, and T. Abe, “A proposal of dual Zipfian model describing HTTP access trends and its application to address cache design,” *IEICE Trans. Commun.*, vol. E81-B, no. 7, pp. 1475–1485, 1998.

[6] C.R. Cunha, A. Bestavros, and M.E. Crovella, “Characteristics of WWW client-based traces,” Tech. Rep. BU-CS-95-010, Boston Univ., 1995.

[7] M. Nabe, K. Baba, M. Murata, and H. Miyahara, “Analysis and modeling of WWW traffic for designing Internet access networks,” *IEICE Trans. Commun.*, vol. J80-B-I, no. 6, pp. 428–437, 1997.

[8] M. Kobayashi and M.H. MacDougall, “The stack growth function: Cache line reference models,” *IEEE Trans. Comput.*, vol. 38, no. 6, 1989.

[9] P.J. Denning and S.C. Schwartz, “Properties of the working-set model,” *Communications of the ACM*, vol. 15, no. 3, 1972.

[10] J.E. Shemer and G.A. Shippey, “Statistical analysis of paged and segmented computer systems,” *IEEE Trans. Electronic Computers*, vol. EC-15, no. 6, 1966.

[11] R. Turner and B. Strecker, “Use of the LRU stack depth distribution for simulation of paging behavior,” *Communications of the ACM*, vol. 20, no. 11, 1977.

[12] S. Glassman, “A caching relay for World Wide Web,” *Comput. Networks and ISDN Systems*, vol. 27, pp. 165–173, 1994.

[13] N. Osawa, F. Hayano, T. Yuba, and K. Hakozaki, “Evaluation of replacement policies in WWW cache based on logs from proxy server,” *Tech. Rep. of IPSJ*, DPS74-33, 1996.

[14] L. Breslau, P. Cao, Li Fan, G. Phillips, and S. Shenker, “Web caching and Zipf-like distributions: Evidence and implications,” proc. *IEEE INFOCOM’99*, March 1999.

[15] R. Rousseau and S. Rousseau, “Informetric distributions: A tutorial review / Les distributions informétriques: Un aperçu,” *Canadian Journal of Information and Library Science*, vol. 18, no. 2, pp.51–63, 1993.

[16] M. Aida, N. Takahashi, and M. Matsuda, “Evaluation of compulsory miss ratio for address cache and replacement policies for restoring packet reachability,” *IEICE Trans. Commun.*, vol. E83-B, no. 7, pp. 1400–1408, 2000.

[17] National laboratory for applied network research (NLANR)/ Measurement and operations analysis team (MOAT), `http://moat.nlanr.net`.

[18] NTT DIRECTORY, `http://navi.ocn.ne.jp`.

[19] C.K. Chow, “Determination of cache’s capacity and its matching storage hierarchy,” *IEEE Trans. Comput.*, vol. c-25, no. 2, 1976.

[20] D. Thiébaut, “On the fractal dimension of computer programs and its application to the prediction of the cache miss ratio,” *IEEE Trans. Comput.*, vol. 38, no. 7, 1989.

[21] M. Aida and T. Abe, “Stochastic process of packet destination addresses describing Internet access patterns,” *IEICE Tech. Rep.*, IN2000-74(2000-09), 2000.