PAPER

Stochastic Model of Internet Access Patterns: Coexistence of Stationarity and Zipf-Type Distributions

Masaki AIDA[†] *and* Tetsuya ABE[††], *Regular Members*

SUMMARY This paper investigates the stochastic property of packet destinations in order to describe Internet access patterns. If we assume a sort of stationary condition for the address generation process, the process is an LRU stack model. Although the LRU stack model gives appropriate descriptions of address generation on a medium/long time-scale, address sequences generated from the LRU stack model do not reproduce Zipf-type distributions, which appear frequently in Internet access patterns. This implies that the address generation behavior on a short time-scale has a strong influence on the shape of the distributions that describe frequency of address appearances. This paper proposes an address generation algorithm that does not meet the stationary condition on the short time-scale, but restores it on the medium/long time-scale, and shows that the proposed algorithm reproduces Zipf-type distributions.

key words: *Internet, destination address, LRU stack, Zipf's law, caching*

1. Introduction

To evaluate the performance and quality of service (QoS) of Internet-type networks, it is necessary to understand the stochastic properties of traffic characteristics and construct an appropriate model. The important nature of Internet traffic patterns is categorized by two types of characteristics [1], [2]. One concerns the volume of traffic. This includes the interarrival distribution of packets and the packet length distribution. They have been investigated in [3], [4], and mainly influence the performance of the queueing mechanisms in the networks. The other is the destinations of the traffic. This includes the destinations of Internet accesses as well as packets. They mainly influence the performance of the caching mechanisms in the networks. This paper focuses on the latter issues.

The fact that computer memory references exhibit *locality* behavior is now well established [5]. Raj Jain reported that frames on computer networks also exhibit locality behavior in the destinations of packets [1]. Since the cache performance is influenced by locality behavior, it is important for evaluating Internet performance. In general, comparisons of the performances of different caching mechanisms are conducted through trace-driven simulations [1], [6], [7]. These use logs of actual Internet accesses as input traffic for the simulation. Hence, we can reproduce a specific network traffic situation by using trace-driven simulation, but cannot obtain information about other types of networks. Aida and Abe [8], [9] investigated the stochastic properties of Internet access patterns and showed the stochastic process that describes Internet access destinations by assuming the process is stationary. More precisely, if we assume a stationary condition for the address generation process, the process is a least recently used (LRU) stack model. Simultaneously, this model can describe locality behavior (especially, temporal locality) in Internet access patterns. Many experimental data support the stationary condition of the process and the process accurately gives cache performance for various Internet access patterns.

On the other hand, it is well known that Internet access destinations exhibit concentrated distributions [2], [7], [10]. For example, only some of the destinations appear very frequently. Such distributions are characterized by Zipf-type distributions and appear frequently in various Internet access patterns.

Unfortunately, the LRU stack model shown in [8], [9] does not reproduce Zipf-type distributions. In other words, the stationary condition and the Zipf-type distributions are not compatible in the present model.

If our purpose is only reproduction of Zipf-type distributions, we can take the independent reference model, which is a simple and primitive model for Internet access patterns. This model assumes accesses are independent. The probability that a new access has address i is determined by 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 recently accessed address do not give any information about the next address, the independent reference model cannot capture temporal locality [1]. So, the model cannot give appropriate cache performance. Although this model may reproduce Zipf-type distributions by using Zipf-type distribution p_i, it is insufficient model that describes Internet access patterns.

This paper investigates the stochastic property of packet destinations in order to allow the coexistence of the stationary condition and Zipf-type distributions. Although the pseudo-address sequences generated from the LRU stack model do not reproduce Zipf-type dis-

Manuscript received June 15, 2001.
Manuscript revised February 26, 2002.
[†]The author is with NTT Information Sharing Platform Laboratories, NTT Corporation, Musashino-shi, 180-8585 Japan.
[††]The author is with NTT EAST R&D Center, Tokyo, 163-8019 Japan.

tributions, we can recognize that the LRU stack model gives appropriate descriptions of address generation on the medium/long time-scale. This property implies that the address generation behavior on the short time-scale has a strong influence on the shape of the distributions that describe the frequency of address appearances. This paper proposes an address generation algorithm that does not meet the stationary condition on the short time-scale, but restores it on the medium/long time-scale, and shows that the proposed algorithm reproduces the Zipf-type distributions.

This paper is organized as follows.

In Sect. 2, we introduce the notions of the working-set size, the inverse stack growth function (ISGF), and the LRU stack model, used in computer memory reference, in order to describe the destination address generations.

In Sect. 3, we investigate Internet accesses patterns with respect to both working-set size behaviors and the distributions of packet destination addresses. This section shows that actual Internet access patterns are characterized by the behavior of ISGF and Zipf-type distributions.

In Sect. 4, we assume that the ISGF is stationary, and investigate the characteristics of the pseudo-address sequence obtained from the stationary address generation process described by an LRU stack model. This section shows that the stationary address generation process cannot reproduce Zipf-type distributions.

In Sect. 5, we introduce a weak stationary condition into the ISGF, and propose a new address generation process that meets the stationary condition on the medium/long time-scale.

In Sect. 6, we show that the proposed address generation process shown in Sect. 5 reproduces the characteristics of actual Internet access patterns described in Sect. 3: both the working-set behaviors and Zipf-type distributions appear in actual Internet access patterns.

2. Notation and Definitions

Internet access behaviors involve similar issues to computer memory reference patterns. We, therefore, base the framework for Internet access models, on 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., $\{t-\tau, t-\tau+1, \ldots, t-1\}$). 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 $W(t,\tau)$,

$$w(t,\tau) := |W(t,\tau)|. \tag{1}$$

- Inverse Stack Growth Function (ISGF) [5]: $f(t,\tau)$
  The expectation value of the number of distinct addresses generated during a period $(t, t+\tau]$,

$$f(t,\tau) := \mathrm{E}[w(t,-\tau)]. \tag{2}$$

  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.,

$$f(t,\tau) = k \quad \leftrightarrow \quad g(t,k) = \tau, \quad \text{or} \tag{3}$$

$$g = f^{-1}. \tag{4}$$

  Note that because ISGF f is defined only for integers, SGF g cannot be defined directly. The consistent way to obtain the relationship between ISGF and SGF is shown in [5]. Hereafter, we regard ISGF and SGF as functions defined for real numbers with respect to τ and k, respectively.
- Least Recently Used (LRU) stack model [11]
  In computer memory references, it is well known that the concept about the locality of a reference pattern appears. The locality (especially, temporal locality) implies a high probability of reuse. The LRU stack model is one computer memory reference model that can describe temporal locality.
  An LRU stack is a list of addresses sorted in order according to the times of their most recent access. 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 the k-th position in the LRU stack is determined by position k, and its probability is denoted as a_k. Usually, a_k decreases as k increases. Because the most recently accessed address is at the top of the LRU stack, there is a high probability that the next address to be accessed will be the same.

3. Characteristics of Internet Access Patterns

This section gives a brief review of the characteristics of Internet access patterns: the working-set behavior and destination address distributions. We investigated access logs of a proxy server at NTT Laboratories. They covered 15 successive days in March 1997 when there were 351,968 accesses.

3.1 Working Set Size Behaviors and Time-Translation Invariance of ISGF

The characteristics of the working-set size behaviors for Internet access patterns have been investigated in [8], [9].

Figure 1 shows the relationship between working set size, $w(t,-\tau)$, and the number of accesses, τ, for the destination IP addresses on a log-log scale. Lines

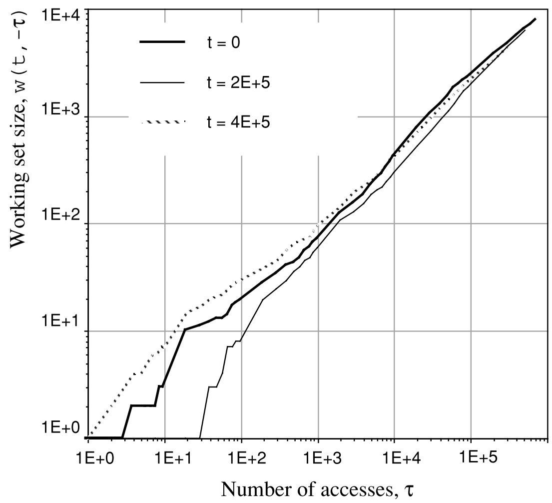

Fig. 1 Working set size behaviors of destination IP addresses (NTT Labs., March 1997).

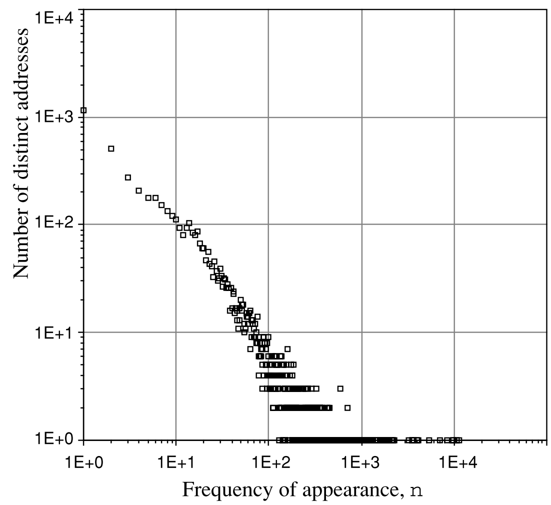

Fig. 2 Lotoka-type plot of the destination IP addresses.

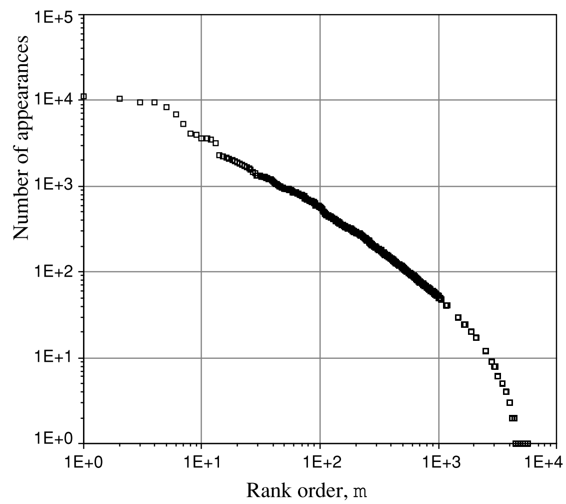

Fig. 3 Zipf-type plot of the destination IP addresses.

indicate $w(t, -\tau)$ for $t = 0$, 2×10^5, and 4×10^5, respectively.

This figure reveals two characteristics:

- $w(t, -\tau)$ is independent of t for large τ.
- $w(t, -\tau)$ exhibits a power law behavior with respect to τ for large τ.

From the law of large numbers, asymptotic behavior of the working set size can be related to the behavior of ISGF, as

$$w(t, -\tau) \sim f(t, \tau) \qquad (\tau \gg 1), \tag{5}$$

where $F(x) \sim G(x)$ means $\lim_{x\to\infty} F(x)/G(x) = 1$. Thus, the first characteristic implies the time-translation invariance of ISGF, for any t and s,

$$f(t, \tau) = f(s, \tau) \qquad (\tau \gg 1), \tag{6}$$

at least in the asymptotic region, $\tau \gg 1$. Similarly, the second characteristic implies that $f(t, \tau)$ satisfies a power law with respect to τ for $\tau \gg 1$,

$$f(t, \tau) \simeq \tau^{\alpha} \qquad (\tau \gg 1), \tag{7}$$

where α is a constant. A power law behavior of ISGF is known in computer memory access [5].

These properties are widely applicable, *e.g.*, for access networks, backbone networks, at a WWW server, and for the destination IP addresses and URLs [8], [9].

3.2 Destination Address Distributions

It is well known that the destination addresses or URLs are characterized by Zipf-type distributions [2], [3], [6], [7], [10]. (strictly, some of them are Lotoka-type distributions [12]).

Figures 2 and 3 show Zipf-type plots of the destination IP addresses.

In Fig. 2, the horizontal axis denotes the frequency n of appearances (the number of accesses). The vertical axis denotes the number of addresses that appeared n times. This figure exhibits linearity on the log-log scale for small n. This is called Lotoka's law or Zipf's 2nd law and means that many addresses have a low frequency of appearances.

In Fig. 3, the horizontal axis denotes the rank order m of addresses where all addresses are sorted in descending order according to the frequencies of appearances. The vertical axis denotes the frequency of the corresponding address appearances (the number of accesses). This figure also exhibits linearity on the log-log scale for small m. This is called Zipf's law and means that only some of the addresses appear very frequently.

In this paper, both Lotoka's and Zipf's laws are called Zipf-type distributions.

4. Address Generation Process and Stationarity

In this section, we show the stationary process of address generation whose ISGF f satisfies the time-translation invariance, and investigate the characteristics of the generated pseudo-address sequence.

4.1 Assumption

From the discussion in Sect. 3 based on experimental data, it is natural to assume the following property of ISGF for all τ.

Time-translation invariance:

ISGF $f(t, \tau)$ is independent of the time t that denotes the time when we start to measure address generation. For any t, s,

$$f(t, \tau) = f(s, \tau). \tag{8}$$

From this assumption, we denote $f(\tau) := f(t, \tau)$ and $g(k) := g(t, k)$.

We assume this in order to produce a stationary process of address generation. Although (6) has the restriction on τ, (8) is assumed for all τ.

4.2 Stationary Address Generation Process

First, we introduce the probability a_k as

$$\begin{aligned} a_k := &\{f(g(k-1)+1) - (k-1)\} \\ &- \{f(g(k)+1) - k\}. \end{aligned} \tag{9}$$

Our previous studies [8], [9] showed that the address generation process is described by the LRU stack model that has probability a_k for selecting the k-th position on the LRU stack, if and only if the ISGF is time-translation invariant (8). This means that the probability that a newly accessed address X is identical with the most recently 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. The derivation of (9) is described in Appendix A.

4.3 Address Generation Algorithm

This subsection shows an address generation algorithm based on the LRU stack model mentioned above. This algorithm gives a pseudo-address sequence, a sequence of integers [8], [9].

We define the generative LRU stack vector $\mathbf{L}(i, m)$ as follows. At the time immediately before the 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

$$\mathbf{L}(i, m) := \{x_1, x_2, \ldots, x_m\}. \tag{10}$$

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 that obeys the distribution
$$\Pr\{J = k\} = a_k \qquad (k = 1, 2, 3, \ldots), \tag{11}$$
by using (9).
3. For the depth m of $\mathbf{L}(i, m)$, if $m < j$, then assign the address of new access as
$$X_i = m + 1, \tag{12}$$
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 the new access as
$$X_i = x_j, \tag{13}$$
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 (see Appendix B).

4.4 Destination Address Distributions

This subsection shows the distributions of the pseudo-address sequence obtained from the above mentioned LRU stack model that satisfies the time-translation invariance of ISGF (8). We use the specific expression of ISGF as

$$f(\tau) = \tau^{\alpha}, \tag{14}$$

for all τ and set $\alpha = 2/3$ from graphical observation of Fig. 1. Then, the form of the probability (9) is expressed as

$$\begin{aligned} a_k = &\left\{((k-1)^{1/\alpha} + 1)^{\alpha} - (k-1)\right\} \\ &- \left\{(k^{1/\alpha} + 1)^{\alpha} - k\right\}. \end{aligned} \tag{15}$$

Figures 4 and 5 show Lotoka-type and Zipf-type plots for the pseudo-address sequence, respectively. The length of the pseudo-address sequence is 350,000. They are drawn to the same scale as Figs. 2 and 3 for ease of comparison.

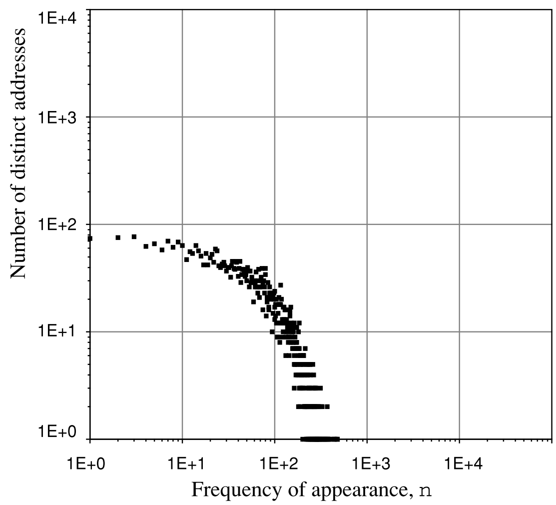

Fig. 4 Lotoka-type plot of the pseudo-address sequence satisfying the time-translation invariance (8).

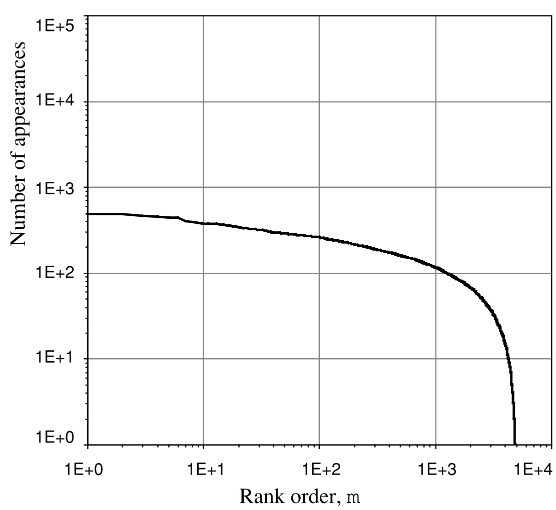

Fig. 5 Zipf-type plot of the pseudo-address sequence satisfying the time-translation invariance (8).

Unlike Figs. 2 and 3, Figs. 4 and 5 do not exhibit the concentration of appearance frequencies. This is because the LRU stack model 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. Therefore, the shape of the Zipf-type distributions are not reproduced from the LRU stack model.

There are probably two reasons why the LRU stack model mentioned above cannot reproduce Zipf-type distributions. They are related to:

- Assuming the time-translation invariance (8) of ISGF.
- Assuming (14) in (15).

Equations (8) and (14) are only asymptotic properties. The time-translation invariance seems to be a natural assumption. Without this 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. In particular, Fig. 1 gives the validation of (6) which is the time-translation invariance of ISGF at least in the asymptotic region $\tau \gg 1$. On the other hand, it is preferable to have the power law adopted for ISGF replaced by a stricter relationship. However, the behavior of ISGF is probably not essential to reproduce Zipf-type distributions.

Hereafter, we introduce violations of (8) and (14) into our model, and try to support the coexistence of stationarity and Zipf-type distributions.

5. Address Generation Process with Asymptotic Time-Translation Invariance

5.1 Approach

Let us consider the partial violation of the time-translation invariance (8) for small τ. Figure 1 does not validate (8) for small τ. So we use the asymptotic time-translation invariance (6), and try to modify the address generation process in order to reproduce a Zipf-type distribution.

The strict time-translation invariance is valid if and only if the generation process is the LRU stack model with probability (9). Therefore, if we use anything except the LRU stack model, the time-translation invariance (8) is naturally violated. This means that we have many possibilities for constructing the extended address generation process. Hereafter, we impose the following three requirements, and introduce the violation of the time-translation invariance into our address generation model.

- On the medium/long time-scale, the model behaves the same as the LRU stack model described in Sect. 4.2.
- On the short time-scale, the model has different behavior from the LRU stack model, but its structure can be adjusted to the LRU stack model described in Sect. 4.2.
- Only some of the addresses appear very frequently.

We take these three requirements into consideration and construct the extended model.

First, the probability (9) that the k-th stack position is selected is rewritten as

$$a_k = \{f(n(k)) - f(n(k)-1)\} \\ \quad - \{f(n(k+1)) - f(n(k+1)-1)\}, \qquad (16)$$

where $n(k)$ denotes that there have been $n(k)$ accesses since the time when the address at the k-th position of the LRU stack most recently appeared, and $n(1) = 1$. Because, for $k \gg 1$,

$$n(k) \simeq g(k-1) + 1 \qquad (k \gg 1), \qquad (17)$$

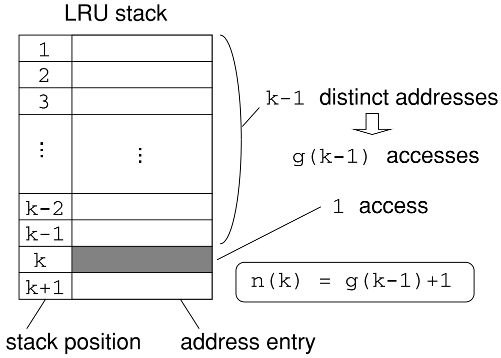

Fig. 6 Relationship between $n(k)$ and $g(k-1)$.

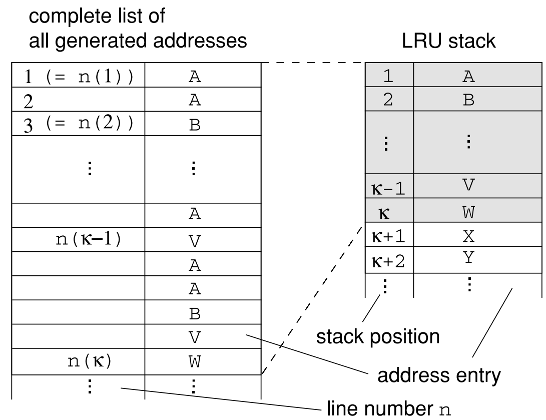

Fig. 7 Relationship between the complete list of all generated addresses and the LRU stack.

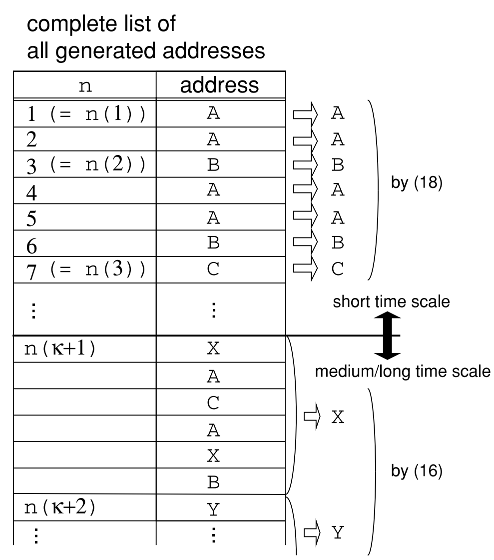

Fig. 8 Example of address generation for the proposed model.

(16) is equivalent to (9) for $k \gg 1$ (see Fig. 6), and the LRU stack models using (16) and (9) show the same behavior for $k \gg 1$. The expression of the probability (16) in $k \gg 1$ is for an easy connection to the probability on a short time-scale, as shown below.

Next, we introduce a threshold parameter κ to separate the time scale into short and medium/long. If the stack position k selected by the probability (16) is $k > \kappa$, the address generation obeys the ordinary LRU stack model. If $k \le \kappa$, let us consider the probability $\widetilde{a}_n$, which denotes the probability that a new access has the same address as appeared n access ago. By natural rewriting of (16), we assume

$$\widetilde{a}_n = \{f(n) - f(n-1)\} - \{f(n+1) - f(n)\}. \tag{18}$$

The relationship between (16) and (18) can be written as follows. For $k \le \kappa$,

$$a_k = \sum_{n=n(k)}^{n(k+1)-1} \widetilde{a}_n. \tag{19}$$

For example,

$$a_1 = \widetilde{a}_1 + \widetilde{a}_2 + \cdots + \widetilde{a}_{n(2)-1}, \tag{20}$$

$$a_2 = \widetilde{a}_{n(2)} + \widetilde{a}_{n(2)+1} + \cdots + \widetilde{a}_{n(3)-1}, \tag{21}$$

and so on. The subscript k of a_k denotes a stack position of the LRU stack, but the subscript n of $\widetilde{a}_n$ denotes the number of accesses that shows a new access has the same address as appeared n access ago. Since probability (18) is not in the form of (9), the address generation process by (18) is not an LRU stack model and it is not time translation invariant. From another viewpoint, (18) is the probability of selecting the n-th stack position in an LRU stack, if all generated addresses are mutually different and thus the LRU stack has the complete list of all generated addresses. Figure 7 shows the relationship between the complete list of all generated addresses and the LRU stack. The complete list includes the same address but the LRU stack does not. Figure 8 shows an example of the generated address according to (16) or (18). For $k \le \kappa$, i.e., $n \le n(\kappa+1)-1$, (20) and (21) are

$$a_1 = \widetilde{a}_1 + \widetilde{a}_2, \tag{22}$$

$$a_2 = \widetilde{a}_3 + \widetilde{a}_4 + \widetilde{a}_5 + \widetilde{a}_6, \tag{23}$$

in an example of Fig. 8. On the short time-scale, multiple appearances of the addresses cause strong autocorrelation. This structure allows frequently accessed addresses to have an extremely high frequency.

5.2 Separation of Time Scale

This section shows a way to determine the value of κ, which separates the time scale into short and medium/long. To examine ISGF behaviors in detail, we utilize the cache performance for the LRU cache. This is because the cache miss ratio for the LRU cache is closely related to the ISGF behavior.

As shown in [2], [8], the theoretical value of cache miss ratio for LRU is obtained through the following formula. For LRU cache, the cache miss ratio, P_L, is given by

$$P_L = \frac{1}{g(k+1) - g(k)}, \tag{24}$$

where k denotes the capacity of the cache (the number of address entries). Therefore, ISGF (or SGF) completely determines the cache miss ratio. Applying the power law (14), we have

$$P_L = \frac{1}{(k+1)^{1/\alpha} - k^{1/\alpha}}. \tag{25}$$

We show the cache miss ratio obtained from simulation with the pseudo-address sequence generated by the stationary address generation algorithm as input, and compare it with the results of trace-driven simulation and theoretical values. The pseudo-address sequence and the theoretical value are obtained by using the power law (14) with $\alpha = 2/3$ from graphical observation of Fig. 1. The data used in the trace-driven simulation was the same access logs as used in Sect. 3.

Figure 9 shows the relationship between cache miss ratio and the LRU cache capacity. The horizontal axis denotes the cache capacity and the vertical axis denotes the cache miss ratio. The black plots denote results from trace-driven simulation and the white ones denote results from the LRU stack model. The solid line denotes theoretical values. This figure shows that the stationary address generation algorithm gives a dependable evaluation in particular for large cache capacity $k \gg 1$. This means that the asymptotic time-translation invariance (6) is re-validated. On the other hand, for $k \simeq 1$, there is a difference between the values obtained from the trace-driven simulation and those obtained from the pseudo-address sequence. This difference implies the violation of the time-translation invariance of ISGF on the short time-scale. The value of κ ranges from about 1000 to a few hundred.

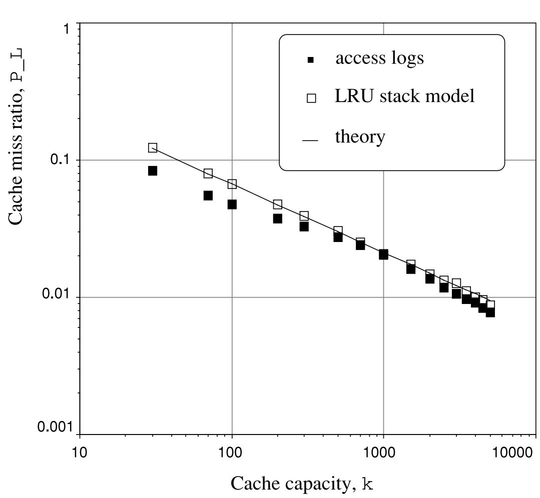

Fig. 9 Relationships between the cache miss ratio and the LRU cache capacity.

5.3 Proposed Address Generation Algorithm

This subsection shows an extension of the stationary address generation algorithm. The extension is carried out according to the three requirements shown in Sect. 5.1.

We define the generative LRU stack vector $\mathbf{L}(i,m)$ as follows. At the time immediately before the 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

$$\mathbf{L}(i,m) := \{x_1, x_2, \ldots, x_m\}. \tag{26}$$

In addition, we define another vector $\mathbf{N}(i,m)$, whose k-th component denotes that there have been $n(k)$ accesses since the time when the address at the k-th component of the generative LRU stack vector $\mathbf{L}(i,m)$ was last accessed.

$$\mathbf{N}(i,m) := \{n(1), n(2), \ldots, n(m)\} \tag{27}$$

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

Let X_i denote the address of the i-th access. We define the generative address vector $\mathbf{X}(i)$ as follows. At the time immediately before the i-th ($i = 0, 1, 2, \ldots$) address generation, $\mathbf{X}(i)$ has the generated addresses in the past as its components,

$$\mathbf{X}(i) := \{X_0, X_1, \ldots, X_{i-1}\}. \tag{28}$$

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 that obeys the distribution
$$\Pr\{J = k\} = a_k \qquad (k = 1, 2, 3, \ldots), \tag{29}$$
by using (16).
3. For the depth m of $\mathbf{L}(i,m)$, if $m < j$, then assign the address of the new access as
$$X_i = m + 1, \tag{30}$$
and update both $\mathbf{L}(i+1, m+1)$ and $\mathbf{N}(i+1, m+1)$, and increment $i \leftarrow i + 1$. Return to 2.
4. For the depth m of $\mathbf{L}(i,m)$, if $m \geq j$ and $j > \kappa$, then assign the address of the new access as
$$X_i = x_j, \tag{31}$$
and update both $\mathbf{L}(i+1, m+1)$ and $\mathbf{N}(i+1, m+1)$, and increment $i \leftarrow i + 1$. Return to 2.

5. For the depth m of $\mathbf{L}(i, m)$, if $m \geq j$ and $j \leq \kappa$, then determine the number l as a realization of the i.i.d. random variable L that obeys the distribution

$$\Pr\{L = n\} = \widetilde{a}_n \qquad (n = 1, 2, 3, \ldots), \tag{32}$$

by using (18). Next, assign the address of the new access as

$$X_i = X_{i-l}, \tag{33}$$

(where X_{i-l} is a component of $\mathbf{X}$(i)) and update both $\mathbf{L}(i+1, m+1)$ and $\mathbf{N}(i+1, m+1)$, and increment $i \leftarrow i + 1$. Return to 2.

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

6. Experimental Results

First, we investigate the working-set size behaviors of the pseudo-address sequence obtained from the generation algorithm described in Sect. 5.3. Here, f in (16) and (18) is assumed to be the power law (14) with the parameter $\alpha = 2/3$. In addition, from the result of Sect. 5.2, we set the threshold parameter $\kappa = 1000$. Figure 10 shows the relationship between working set size, $w(t, -\tau)$, and the number of accesses, τ, for the destination IP addresses on a log-log scale. Lines indicate $w(t, -\tau)$ for $t = 4 \times 10^6$, 8×10^6, and 1.2×10^7, respectively.

This figure has the same characteristics as Fig. 1. From the law of large numbers (5), the asymptotic behavior of the ISGF has the properties of

- Time-translation invariance and
- Power law behavior.

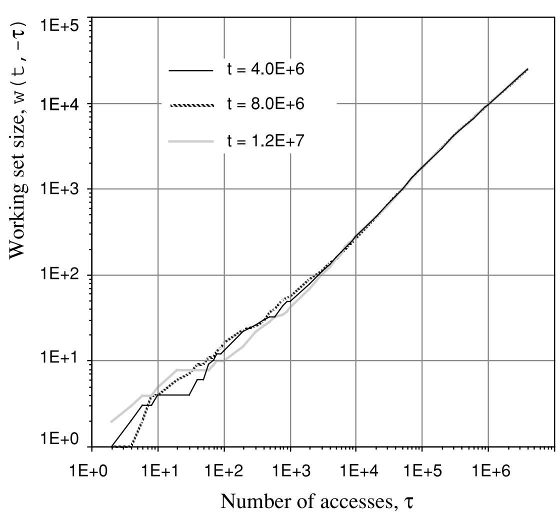

Fig. 10 Working-set behaviors of the pseudo-address sequence obtained from the proposed algorithm.

Figures 11 and 12 show Lotoka-type and Zipf-type plots, respectively, for the pseudo-address sequence obtained from the proposed algorithm. The length of the pseudo-address sequence is 350,000. They are drawn to the same scale as Figs. 2 and 3 for ease of comparison. Like Figs. 2 and 3, Figs. 11 and 12 exhibit a concentration of the appearance frequencies and reproduce the Zipf-type distributions.

7. Conclusions

In this paper, we have investigated an address generation process that can reproduce Zipf-type distributions.

Both stationarity and Zipf-type distributions frequently appear in Internet access patterns. If we assume the stationary condition as the time-translation invariance of ISGF, the address generation process

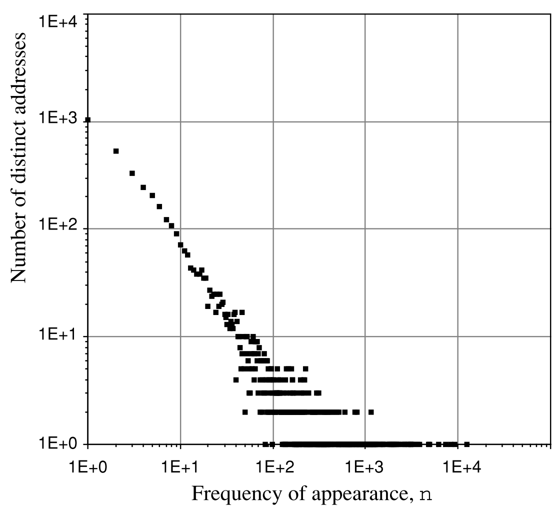

Fig. 11 Lotoka-type plot of the pseudo-address sequence obtained from the proposed algorithm.

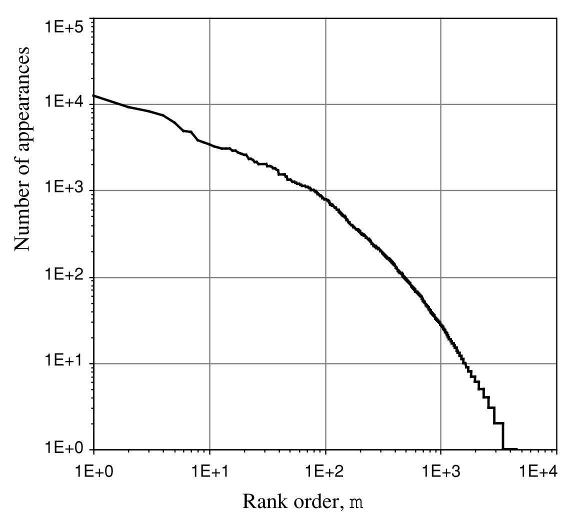

Fig. 12 Zipf-type plot of the pseudo-address sequence obtained from the proposed algorithm.

cannot reproduce Zipf-type distributions. From the cache performance, the time-translation invariance on the medium/long time-scale is a reasonable assumption. So, we introduce two structures into our model: one is violation of the invariance relationship on the short time-scale and the other is its restoration on the medium/long time-scale. Based on these structures, we propose a new address generation algorithm.

The time-translation invariance of the proposed algorithm is restored on the medium/long time-scale, and Zipf-type distributions appear. Since the way of violating the time-translation invariance is not unique, the proposed algorithm is one of several possibilities. However, we note the following two significant points.

- Symmetry breaking on a short time-scale strongly affects the distribution of the long-time sequence.
- Zipf-type distributions are compatible with stationality on the medium/long time-scale.

References

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

[2] 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, July 1998.

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

[4] M. Nabe, K. Baba, M. Murata, and H. Miyahara, "Analysis and modeling of WWW traffic for designing Internet access networks," IEICE Trans. Commun. (Japanese Edition), vol.J80-B-I, no.6, pp.428–437, June 1997.

[5] M. Kobayashi and M.H. MacDougall, "The stack growth function: Cache line reference models," IEEE Trans. Comput., vol.38, no.6, pp.798–805, 1989.

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

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

[8] M. Aida and T. Abe, "Pseudo-address generation algorithm of packet destinations for Internet performance simulation," Proc. IEEE INFOCOM 2001, pp.1425–1433, April 2001.

[9] M. Aida and T. Abe, "Stochastic model of Internet access patterns," IEICE Trans. Commun., vol.E84-B, no.8, pp.2142–2150, Aug. 2001.

[10] L. Breslau, P. Cao, Li Fan, G. Phillips, and S. Shenker, "Web caching and Zipf-like distributions: Evidence and implications," Proc. IEEE INFOCOM'99, pp.126–134, March 1999.

[11] J.E. Shemer and G.A. Shippey, "Statistical analysis of paged and segmented computersystems," IEEE Trans. Electron. Comput., vol.EC-15, no.6, pp.855–863, 1966.

[12] 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.

Appendix A: Address Generation Probability

We consider three adjacent periods A, B, and C (Fig. A·1). Let the sets of distinct addresses generated in these periods, i.e., working set, be $W(A)$, $W(B)$, and $W(C)$, respectively. Figure A·2 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. A·2,

$$W^* := \{W(A) \cap W(C)\} \setminus W(B). \tag{A·1}$$

The size of W^* is obtained as

$$\begin{aligned}|W^*| &= |\{W(A) \cap W(C)\} \setminus W(B)| \\ &= \{|W(B) \cup W(C)| - |W(B)|\} \\ &\quad - \{|W(A) \cup W(B) \cup W(C)| \\ &\quad - |W(A) \cup W(B)|\}.\end{aligned} \tag{A·2}$$

We can choose m, n and 1 accesses as the periods A, B, and C, respectively (Fig. A·3). Then we have

$$\begin{aligned}|W^*| &= \{w(t, n+1) - w(t-1, n)\} \\ &\quad - \{w(t, n+m+1) - w(t-1, n+m)\}.\end{aligned} \tag{A·3}$$

Applying the time-translation invariance (8), we have

$$\begin{aligned}\mathrm{E}\left[|W^*|\right] &= \{f(-(n+1)) - f(-n)\} \\ &\quad - \{f(-(n+m+1)) - f(-(n+m))\} \\ &= \{f(n+1) - f(n)\} \\ &\quad - \{f(n+m+1) - f(n+m)\}.\end{aligned} \tag{A·4}$$

The physical meanings of $\mathrm{E}\left[|W^*|\right]$ are as follows. $|W^*|$

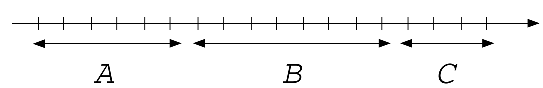

Fig. A·1 Time intervals A, B and C.

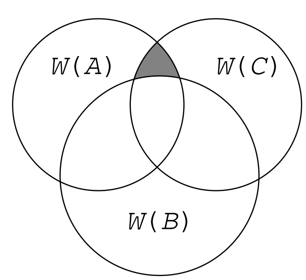

Fig. A·2 Venn's diagram of working sets.

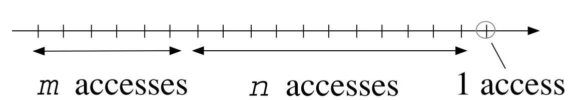

Fig. A·3 Time intervals.

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

$$|W^*| = \mathbf{1}\{X \notin W(B), X \in W(A)\}. \qquad (\mathrm{A\cdot 5})$$

Therefore, $\mathrm{E}[|W^*|]$ means the probability of $|W^*| = 1$ (the expectation of an indicator function is a probability),

$$\mathrm{E}[|W^*|] = \Pr\{X \notin W(B), X \in W(A)\}. \qquad (\mathrm{A\cdot 6})$$

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 $n+m = g(k)$, (A·4) gives

$$\mathrm{E}[|W^*|] = \{f(g(k-1)+1) - (k-1)\} - \{f(g(k)+1) - k\}. \qquad (\mathrm{A\cdot 7})$$

Equation (A·7) 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

$$a_k := \{f(g(k-1)+1) - (k-1)\} - \{f(g(k)+1) - k\}. \qquad (\mathrm{A\cdot 8})$$

This means that the probability that X is identical with the most recently 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 (A·8).

Appendix B: Determination of the Number j

This section shows a way to determine the number j as a realization of the i.i.d. random variable J that obeys the distribution

$$\Pr\{J = k\} = \{f(g(k-1)+1) - (k-1)\} - \{f(g(k)+1) - k\} \quad (k = 1, 2, 3, \ldots). \qquad (\mathrm{A\cdot 9})$$

Because $f(g(k)+1) - k$ decreases with respect to k and

$$f(g(0)+1) - 0 = 1, \qquad (\mathrm{A\cdot 10})$$

$$\lim_{k\to\infty} f(g(k)+1) - k = 0, \qquad (\mathrm{A\cdot 11})$$

the following simple procedure is applicable. Let ξ be a realization of the i.i.d. random variable that is uniformly distributed on [0,1]. If a number k satisfies

$$f(g(k)+1) - k < \xi \leq f(g(k-1)+1) - (k-1), \quad (\mathrm{A\cdot 12})$$

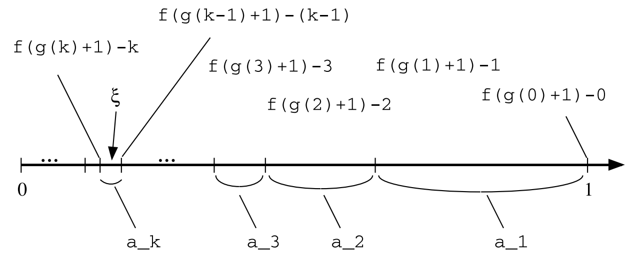

Fig. A·4 Searching for the number k satisfies (A·12).

then we can determine $j = k$ (see Fig. A·4). Thus, we can determine the number j by using a binary search algorithm comparing ξ with $f(g(k)+1) - k$ for appropriate ks.

Masaki Aida received his B.Sc. and M.Sc. degrees in Theoretical Physics from St. Paul's University, Tokyo, Japan, in 1987 and 1989, and received the Ph.D. degree in Telecommunication Engineering from the University of Tokyo, Japan, in 1999. Since joining NTT Laboratories in 1989, he has been mainly engaged in research on traffic issues in ATM networks and computer communication networks. From March 1998 to March 2001, he was a manager at Traffic Research Center, NTT Advanced Technology Corporation (NTT-AT). He is currently a Senior Research Engineer in NTT Information Sharing Platform Laboratories. His current interests include traffic issues in communication systems. He received the Young Investigators Award of IEICE in 1996. Dr. Aida is a member of the IEEE and the Operations Research Society of Japan.

Tetsuya Abe received his B.Sc. degree in Physics from Keio University, Yokohama, Japan, in 1988. He joined NTT Laboratories in 1988, where he has been conducting research on process-device modeling and development on on-chip RISC controller and designing of high-speed network-nodes. Since 2000, he has been an associate manager at NTT EAST R&D Center. His current interests include security of services on broadband networks. He received the Best Industrial ASIC Award of ED&TC in 1996.