PAPER

A Proposal of Dual Zipfian Model for Describing HTTP Access Trends and Its Application to Address Cache Design

Masaki AIDA[†*], *Member*, Noriyuki TAKAHASHI[†], *Nonmember, and* Tetsuya ABE[†], *Member*

SUMMARY This paper proposes the Dual Zipfian Model addressing how to describe HTTP access trends in large-scale data communication networks, and discusses how to design the capacity of address cache tables in an edge router of the networks. We show that destination addresses of packets can be characterized by two types of Zipf's law. Fundamental concept of the Dual Zipfian Model is in complementary use of these laws, and we can derive the relationship between the number of accesses and the number of destination addresses. Experimental results show that the relation gives a good approximation. Applying this relation, we derive cache hit probabilities of the address cache table that incorporates high-speed address resolution. Using the probabilities, design issues including the capacity of the cache tables and aging algorithms of cache entries are also discussed.
key words: *data network, address resolution, Zipf's law, cache, HTTP, WWW*

1. Introduction

Computer communication networks are attracting attention, and their importance, in terms of expanding the market for data services, is rapidly increasing. Thus, it is necessary to enlarge network scale and improve network speed, at low cost.

In order to meet such network requirements, many data networking architectures using the broadband capability of Asynchronous Transfer Mode (ATM) have been proposed and studied. In these architectures, data packets are processed in a router by the following three processes.

- Packet reassembly process (if necessary).
  At the input port of the router, incoming cells are reassembled into a packet.
- Address resolution process.
  This process finds the destination ATM address and/or the next-hop ATM address from the destination network address, e.g., the IP address. This is necessary for data forwarding through ATM links.
- Packet transfer process.
  The acquired address is used to send the packet to the corresponding output port.

Manuscript received November 20, 1997.
Manuscript revised February 12, 1998.
[†]The authors are with NTT Multimedia Networks Laboratories, Yokosuka-shi, 239–0847 Japan.
[*]Presently, with NTT Advanced Technology Corporation (NTT-AT).

If a buffer has enough capacity for packet reassembling, the degradation of performance in the packet reassembly process is easily avoided. The processes that usually cause degradation of performance are address-resolution and packet-transfer. If address resolution is not fast enough, the throughput of packet transfer will be diminished. In the packet transfer process, packet overflow at the output buffer causes diminished packet throughput. The nature of this performance degradation in the packet transfer process is basically the same in connection-oriented networks. However, in order to analyze the performance degradation, it is necessary to know characteristics of data traffic. Such a traffic model has been reported in [8] for WWW accesses.

The performance degradation in the address resolution process is an issue peculiar to connectionless data networks. This paper focuses on this issue.

Figures 1 and 2 show our network models. When a packet arrives at the edge router, the packet is processed for address resolution. In the case of Fig. 1, if the address cache table at the edge router contains the entry corresponding to the destination address of the packet, address resolution is immediately carried out. Otherwise, the router must request address resolution from the address server, which is a large database that

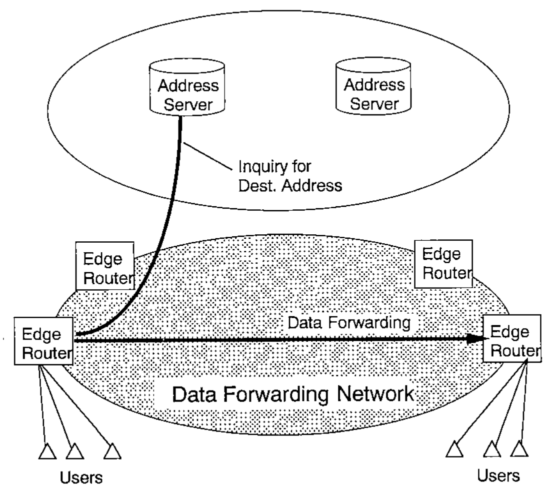


Fig. 1 Network model (address resolution using the address server).

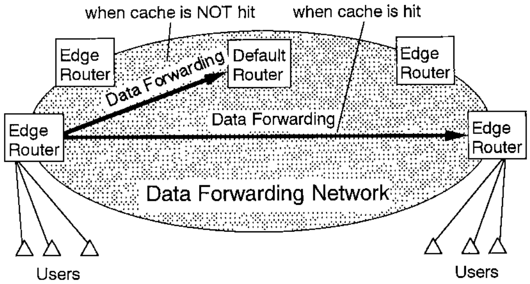


Fig. 2 Network model (address resolution using the default router).

has comprehensive information [6]. This inquiry and retrieval process greatly increases the address resolution time. In the case of Fig. 2, if the address cache table at the edge router does not contain the entry corresponding to the destination address of the packet, the router sends the packet to its default router, which has all address information [7]. The default router then analyzes the address of the packet and transfers it.

In both cases, increasing the cache table capacity is effective in shortening the address resolution time and in reducing the utilization of the default router, but doing so is expensive. In addition, the cache hit probability tends to increase very slowly as the capacity of the cache table is increased.

Raj Jain [5] discusses characteristics of destination addresses and compares several caching schemes, based on a concept of locality (characteristics of access patterns). Since his work assumes simple locality, it is insufficient for the task of cache design in a complicated large-scale network.

In this paper, we show that the distribution of the destination IP addresses can be characterized by two types of Zipf's law [1], [2]. Based on the laws, we propose a Dual Zipfian Model for describing HTTP access trends. Using the Dual Zipfian Model, we derive the relationship between the number of total accesses and the number of the destination IP addresses. Based on the relation, we derive the cache hit probability (CHP) of packet destination addresses. We also discuss how to design the capacity of cache table and the aging algorithm.

This paper is organized as follows: In Sect. 2, we briefly note Zipf's law describing the distribution of the destination URLs. In Sect. 3, we show two types of Zipf's law are applicable to the distribution of the destination IP addresses. In addition, we propose the Dual Zipfian Model based on the idea of applying the laws complementarily, and derive the relationship between the number of accesses and the destination IP addresses. Section 4 shows the generalized treatment of the Dual Zipfian Model. In Sect. 5, based on the relation acquired in Sect. 3, we discuss the CHP for some classical aging algorithms. Using the results in Sect. 5, we show how to design the address cache table with respect to both its capacity and aging algorithm in Sect. 6.

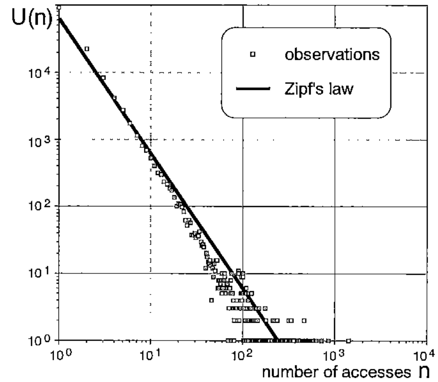


Fig. 3 The relationship between n, the number of hits, and $U(n)$, the number of URLs with n hits.

2. Distributions of URLs and Zipf's Law

Zipf's law is an empirical law and a vast number of various social phenomena are known to be described by it [3]. In general, Zipf's law is expressed as

$$A(n) \cong \frac{C}{n^{\alpha}}, \tag{1}$$

where $A(n)$ is the size of an object belonging to the class specified by n, and n is the rank of a certain object placed according to its size. C and α are constants that are independent of n.

Of course, Zipf's law is not applicable to all n. It is known that Zipf's law describes the region of small n. The most famous form of Zipf's law is for the case of a hyperbolic form $\alpha = 1$, i.e.,

$$A(n) \cong \frac{C}{n}. \tag{2}$$

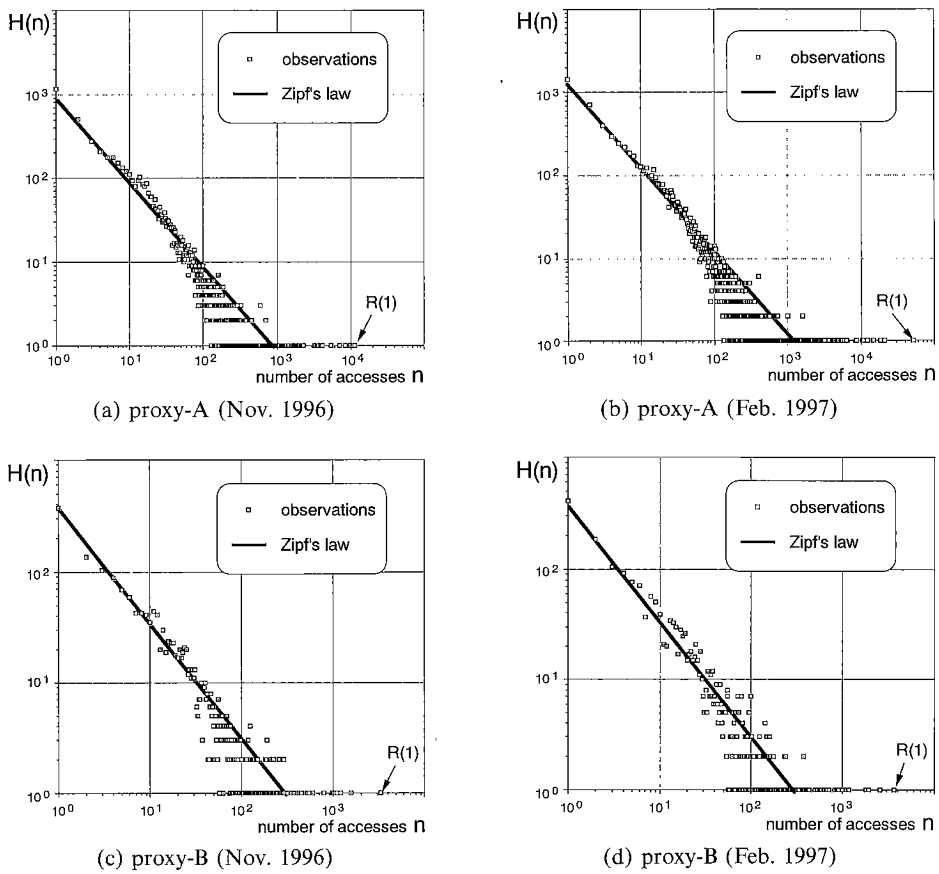
In [4], [9], applying Zipf's law to HTTP access and its validity are discussed by analyzing a WWW access log. Let the number of different URLs be $U(n)$ such that each is accessed just n times in a certain fixed period. From an analysis of WWW access logs, it is reported that the distribution of $U(n)$ is described by Zipf's law with $\alpha = 2$ [4], [9], i.e.,

$$U(n) \cong \frac{C}{n^2}, \tag{3}$$

in the region of small n. Figure 3 shows a distribution for $U(n)$. These data are from logs of a proxy server (proxy-A as stated later). It is clear that the Zipf law (3) is applicable to the distribution in the region of small n.

3. Dual Zipfian Model (DZM)

In general, the structure of a URL is



(a) proxy-A (Nov. 1996)

(b) proxy-A (Feb. 1997)

(c) proxy-B (Nov. 1996)

(d) proxy-B (Feb. 1997)

Fig. 4 The relationship between n, the number of accesses, and $H(n)$, the number of destination IP addresses with n accesses.

$$\texttt{http : //}\underbrace{\texttt{host.in.some.domain}}_{\text{<host>}}\underbrace{\texttt{/FOO/BAR/BUZ.html}}_{\text{<path>}}$$

in the HTTP case. In the above example, the distribution of $U(n)$ is distinguished by each URL. Our interest, however, is only in the address resolution process in the edge router. This process handles only the <host> part of the URL, i.e., IP address. Therefore, it is necessary to investigate the number of accesses for each IP address.

In addition, since Zipf's law is applicable for only small n, it is insufficient for deriving information when n is large.

In the remainder of this section, we propose the Dual Zipfian Model that gives a solution of the above issues and enables the relationship between the number of accesses and the number of the destination IP addresses to be obtained.

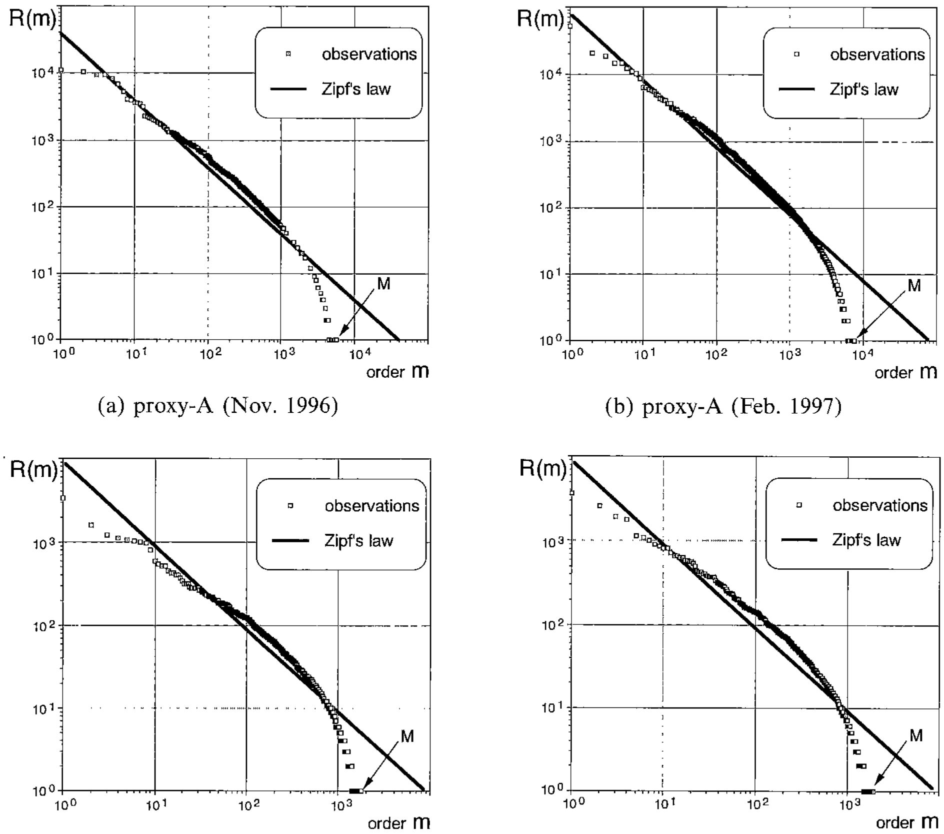
3.1 Destination IP Addresses and Two Types of Zipf's Law

We apply Zipf's law to the distribution of packet destination <host> for describing the HTTP access pattern. Let the number of IP addresses be $H(n)$ such that each is accessed just n times. Figure 4 shows the distribution of $H(n)$ with respect to n. These data are from the logs of two different proxy servers in our laboratories referred to here as proxy-A and B. The logs of proxy-A have 351, 968 and 714, 931 accesses in Nov. '96 and Feb. '97, respectively. The logs of proxy-B have 59, 098 and 73, 621 accesses in the same months. These data are for 15 successive days. From Fig. 4, it is clear that the Zipf law (1) with $\alpha = 1$ is applicable to these cases in the region of small n.

On the other hand, we can consider another type of Zipf's law. We arrange all IP addresses in order of the number of times they are accessed, i.e., we assign $m = 1$ to the most frequently accessed IP address, assign $m = 2$ to the next most frequently accessed IP address, and so on. Then let the number of accesses be $R(m)$ with respect to an IP address at mth order. Figure 5 shows the distribution of $R(m)$ with respect to m. These data are from the logs of proxy-A and B. From Fig. 5, one can also see that the Zipf law (1) with $\alpha = 1$ is applicable to these cases in the region of small m.

3.2 Complementary Use of Two Types of Zipf's Law

We adopt the two types of Zipf's law as the starting point. It is worth noting the important points that large



(a) proxy-A (Nov. 1996)
(b) proxy-A (Feb. 1997)
(c) proxy-B (Nov. 1996)
(d) proxy-B (Feb. 1997)

Fig. 5 The relationship between order m and $R(m)$, the number of accesses to the mth order IP address.

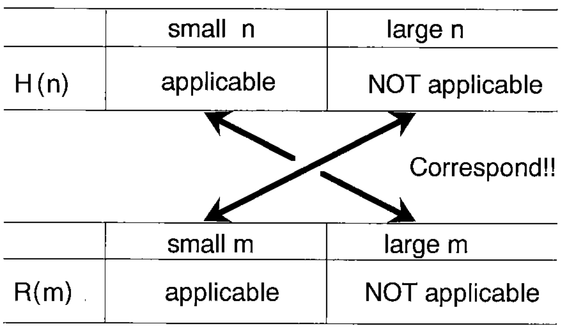


Fig. 6 Duality in two types of Zipf's law.

n corresponds to small m, and, conversely, large m corresponds to small n (Fig. 6). This nature is called duality. Thus, we can expect that the domain where one Zipf's law fails is one in where the other Zipf's law is applicable. Therefore, we use the two types of Zipf's law complementarily. This is the fundamental concept of Dual Zipfian Model.

Let the total number of accesses be N. The total number of destination IP addresses is denoted by M. We consider the two types of Zipf's law with respect to $H(n)$ and $R(m)$. Note that the domain of $R(m)$ is $1 \leqq m \leqq M$, and $R(m) > 0$ for all m in this domain. On the other hand, the domain of $H(n)$ is $1 \leqq n \leqq R(1)$ restricted by the range of $R(m)$. In this case, there are some n such that $H(n) = 0$, although n is in this domain.

We introduce a parameter β such that Zipf's law with respect to $H(n)$ is approximately applicable in domain $1 \leqq n \leqq R(1)^{1/\beta}$. Hence, we have M as

$$\begin{aligned} M &= \sum_{n=1}^{R(1)^{1/\beta}} H(n) + \sum_{m=1}^{R(1)^{(\beta-1)/\beta}} 1 \\ &= H(1) \sum_{n=1}^{R(1)^{1/\beta}} \frac{1}{n} + R(1)^{(\beta-1)/\beta} \\ &\cong H(1) \left[\gamma + \frac{1}{\beta} \ln R(1) + \frac{1}{2R(1)^{1/\beta}} \right] \\ &\quad + R(1)^{(\beta-1)/\beta}, \end{aligned} \tag{4}$$

where γ is the Euler number and $\gamma \cong 0.57721$. The last equality in (4) was obtained using

$$\sum_{k=1}^{K} \frac{1}{k} \cong \gamma + \ln K + \frac{1}{2K}.$$

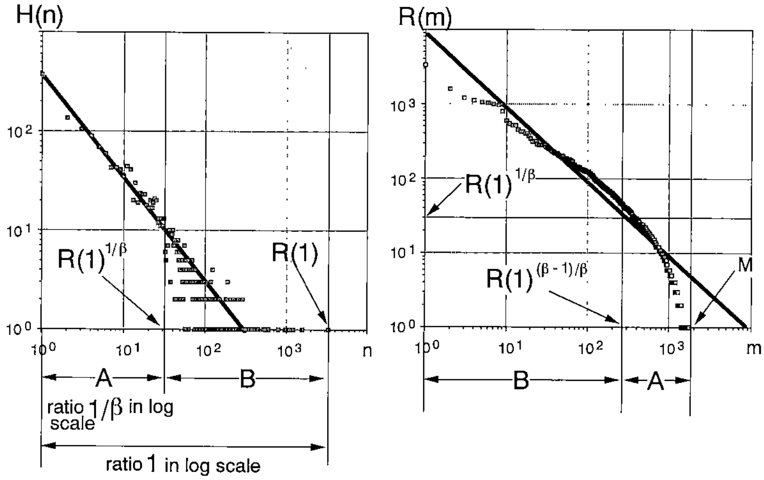


Fig. 7 Example of the Dual Zipfian Model.

Using β, we can denote M from the Zipf laws. The physical meaning of (4) is as follows: Zipf's law with respect to $H(n)$ is approximately applicable in domain $1 \leqq n \leqq R(1)^{1/\beta}$, that is domain A in the left of Fig. 7. Then, $n = R(1)^{1/\beta}$ corresponds to the order $m = R(1)^{(\beta-1)/\beta}$ by using the other Zipf law with respect to $R(m)$. In addition, the number of IP addresses corresponding to $n \geqq R(1)^{1/\beta}$ (domain B in the left of Fig. 7) corresponds to the order $1 \leqq m \leqq R(1)^{(\beta-1)/\beta}$ (domain B in the right of Fig. 7) where Zipf's law with respect to $R(m)$ is applicable.

N is also denoted by

$$\begin{aligned} N &= \sum_{n=1}^{R(1)^{1/\beta}} nH(n) + \sum_{m=1}^{R(1)^{(\beta-1)/\beta}} R(m) \\ &= H(1) \sum_{n=1}^{R(1)^{1/\beta}} 1 + R(1) \sum_{m=1}^{R(1)^{(\beta-1)/\beta}} \frac{1}{m} \\ &\cong H(1)R(1)^{1/\beta} \\ &\quad + R(1)\left[\gamma + \frac{\beta-1}{\beta}\ln R(1) + \frac{1}{2R(1)^{(\beta-1)/\beta}}\right]. \end{aligned} \tag{5}$$

The physical meaning of (5) is similar to that of (4).

Since, for large m, $R(m)$ decreases quickly in a log scale, we assume $R(m)$ to satisfy Zipf's law up to $m \cong M$. Then, N is also expressed as

$$N \cong R(1)\left[\gamma + \ln M + \frac{1}{2M}\right]. \tag{6}$$

We call a set of Eqs. (4)–(6) as a Dual Zipfian Model (DZM). Since the tail of $R(m)$ is characterized by the upper-tail of $H(n)$, we can derive a more faithful model with respect to Eq. (6) without additional arbitrary parameters (see Appendix). Although the faithful model includes additional equations and requires more complicated treatment than the above model, accuracies of both evaluations using the above model and the faithful model are almost the same, at least in HTTP case.

3.3 Self-Consistent Treatment of DZM

From Eqs. (4)–(6), we can determine M, $H(1)$, $R(1)$ for given N. This section shows the self-consistent treatment of Eqs. (4)–(6) in derivation of M.

First, we set

$$M_0 = N, \tag{7}$$

as the initial condition. Then, for $i = 1, 2, 3, \ldots$, we calculate M_i, R_i, and H_i recursively as

$$R_i = \frac{N}{\gamma + \ln M_{i-1} + \frac{1}{2M_{i-1}}}, \tag{8}$$

$$\begin{aligned} H_i = R_i^{(\beta-1)/\beta}\Bigg[&\ln M_{i-1} - \frac{\beta-1}{\beta}\ln R_i \\ &+ \frac{1}{2M_{i-1}} - \frac{1}{2R_i^{(\beta-1)/\beta}}\Bigg], \end{aligned} \tag{9}$$

$$M_i = H_i\left[\gamma + \frac{1}{\beta}\ln R_i + \frac{1}{2R_i^{1/\beta}}\right] + R_i^{(\beta-1)/\beta}. \tag{10}$$

When M_i is sufficiently close to M_{i-1} at $i = j$, we stop the iteration and

$$M = M_j, \tag{11}$$

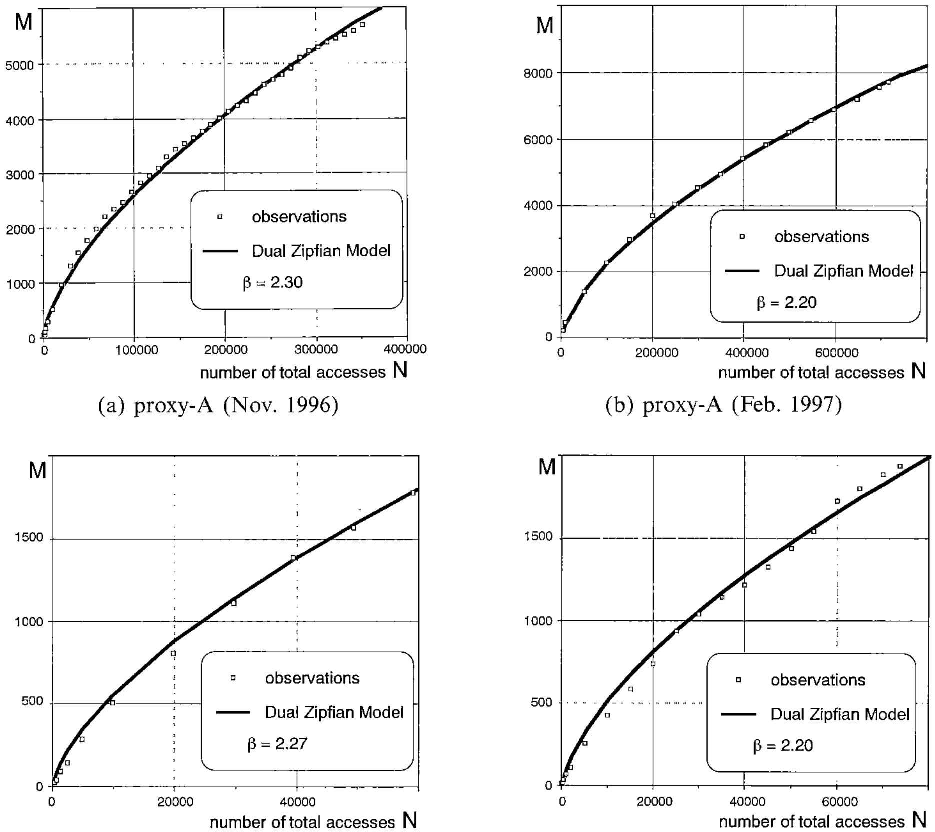
$$R(1) = R_j, \tag{12}$$

$$H(1) = H_j. \tag{13}$$

Fortunately, this iteration converges fast; it needs only about 10 iterations.

3.4 Evaluation of Calculated M

Here we show the accuracy of the calculated M, i.e.,



(a) proxy-A (Nov. 1996)

(b) proxy-A (Feb. 1997)

(c) proxy-B (Nov. 1996)

(d) proxy-B (Feb. 1997)

Fig. 8 Accuracy of evaluated M.

the number of destination IP addresses when the total number of accesses N are given.

Figure 8 shows accuracy of evaluated M of proxy-A and B with respect to N. For proxy-A, we set $\beta = 2.30$ for Nov. '96, and $\beta = 2.20$ for Feb. '97. For proxy-B, we set $\beta = 2.27$ for Nov. '96, and $\beta = 2.20$ for Feb. '97. These figures show calculated M is sufficiently close to the real number of destination IP addresses.

Here, there are two remarkable points:

- the value of β is independent of the number of accesses N in each case.
- the values of β are almost the same in these cases.

The first feature implies that if we only know a pair of M and N, then we can evaluate β and can evaluate M for any N using the evaluated β. This means, to acquire M for arbitrary N, we have to measure M only for small N and evaluate the value of β.

The second feature appears to imply that the value of β is fairly universal and that β characterizes the nature of HTTP access trends. However, at present we can not determine whether or not the value of β has true universality. Further study using many data from different proxies is required.

The authors have been measuring the experimental M since November 1996. So far, almost all βs are in $2.2 \leq \beta \leq 2.3$.

4. Generalized Self-Consistent Treatment of DZM

In the HTTP case, the Zipf law (1) with $\alpha = 1$ is utilized to describe $H(n)$ and $R(m)$ as shown in Sect. 3. However, in general, it is not guaranteed that the distributions of destination IP addresses are described by the same Zipf law.

This section shows the generalized self-consistent treatment of DZM, which is applicable to cases of $\alpha \neq 1$. Let α of Zipf's law (1) with respect to $H(n)$ be $\alpha = \alpha_h$, and that with respect to $R(m)$ be $\alpha = \alpha_r$. Then Eqs. (4)–(6) are generalized as follows:

$$\begin{aligned} M &= \sum_{n=1}^{\lfloor R(1)^{1/\beta} \rfloor} H(n) + \left[R(1)^{(\beta-1)/\beta} \right]^{1/\alpha_r} \\ &= H(1) \sum_{n=1}^{\lfloor R(1)^{1/\beta} \rfloor} \frac{1}{n^{\alpha_h}} + \left[R(1)^{(\beta-1)/\beta} \right]^{1/\alpha_r}, \end{aligned} \tag{14}$$

$$N = \sum_{n=1}^{\lfloor R(1)^{1/\beta} \rfloor} n H(n) + \sum_{m=1}^{\lfloor R(1)^{(\beta-1)/(\beta\alpha_r)} \rfloor} R(m)$$

$$= H(1) \sum_{n=1}^{\lfloor R(1)^{1/\beta} \rfloor} \frac{1}{n^{\alpha_h - 1}}$$

$$+ R(1) \sum_{m=1}^{\lfloor R(1)^{(\beta-1)/(\beta\alpha_r)} \rfloor} \frac{1}{m^{\alpha_r}}, \quad (15)$$

$$N = R(1) \sum_{m=1}^{\lfloor M \rfloor} \frac{1}{m^{\alpha_r}}, \quad (16)$$

where $\lfloor x \rfloor$ denotes the maximum integer less than or equal to x.

For given N, we derive M recursively using the following procedure: First, we set

$$M_0 = N, \quad (17)$$

as the initial condition. Then, for $i = 1, 2, 3, \ldots$, we calculate M_i, R_i, and H_i recursively as

$$R_i = \frac{N}{\sum_{m=1}^{\lfloor M_{i-1} \rfloor} m^{-\alpha_r}}, \quad (18)$$

$$H_i = \frac{N - R_i \sum_{m=1}^{\lfloor R_i^{(\beta-1)/(\beta\alpha_r)} \rfloor} m^{-\alpha_r}}{\sum_{n=1}^{\lfloor R_i^{1/\beta} \rfloor} n^{-(\alpha_h - 1)}}, \quad (19)$$

$$M_i = H_i \sum_{n=1}^{\lfloor R_i^{1/\beta} \rfloor} \frac{1}{n^{\alpha_h}} + \left[R_i^{(\beta-1)/\beta} \right]^{1/\alpha_r}. \quad (20)$$

When M_i is sufficiently close to M_{i-1} at $i = j$, we stop the iteration and

$$M = M_j, \quad (21)$$

$$R(1) = R_j, \quad (22)$$

$$H(1) = H_j. \quad (23)$$

It is worth noting when the above generalization is valid. The above generalization assumes $R(m)$ to satisfy Zipf's law up to $m \cong M$. Therefore, $R(m)$ must decrease quickly in a log scale. In addition, the generalization also assumes that the upper-tails of both $H(n)$ and $R(m)$ where Zipf's laws are applicable have the overlapping domain, i.e., $R(m)$ satisfies Zipf's law up to $m \geqq R(1)^{(\beta-1)/(\beta\alpha_r)}$.

5. Evaluations of Cache Hit Probability

Using the DZM, we derive cache hit probability (CHP), a destination address making a hit in the address cache table, for classical aging algorithms.

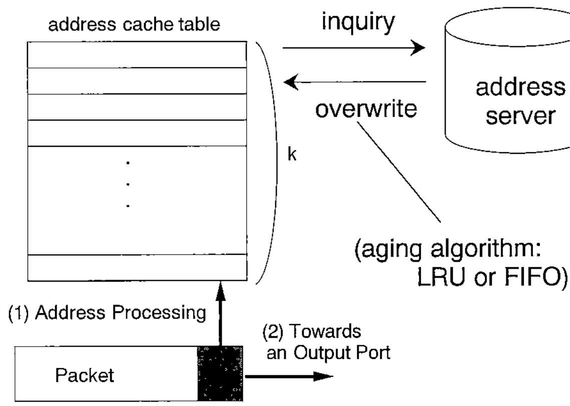


Fig. 9 Address processing in an edge router.

5.1 Notations

- k: Capacity of the cache table (the number of address entries, Fig. 9).
- f: Mapping from the total number of accesses N to the number of IP addresses M obtained from the DZM (4)–(6), i.e.,

$$M = f(N). \quad (24)$$

5.2 Aging Algorithms for Cache Entries

Let us focus on the following two classical algorithms.

- LRU algorithm
  When the cache does not hit, new address data is written over the least recently used entry.
- FIFO algorithm
  When the cache does not hit, new address data is written over the current oldest entry.

5.3 Cache Hit Probability for LRU

For LRU, the address cache always has k different IP addresses which are most recently accessed. The number of HTTP accesses that generate k different IP addresses is $f^{-1}(k)$. Similarly, the number of HTTP accesses that generate $k+1$ different IP addresses is $f^{-1}(k+1)$. Thus the number of HTTP accesses between two successive replacements, is $f^{-1}(k+1) - f^{-1}(k)$ (Fig. 10). CHP for LRU P_L is given as

$$P_L = 1 - \frac{1}{f^{-1}(k+1) - f^{-1}(k)}. \quad (25)$$

5.4 Cache Hit Probability for FIFO

For FIFO, unlike LRU, the address cache does not always have k different IP addresses that are most recently

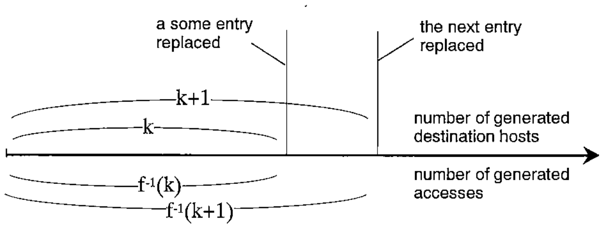


Fig. 10 Interval of entry replacements.

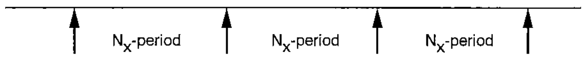


Fig. 11 N_x-periods.

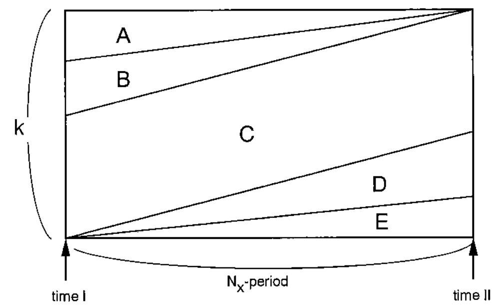


Fig. 12 Types of cache entries.

accessed. Here, let 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 HTTP accesses the N_x-period. From the characteristics of FIFO, we have $f^{-1}(k) < N_x < f^{-1}(2\,k)$.

The number of the destination IP addresses generated during an N_x-period is $f(N_x)$, and the number of the destination IP addresses generated during consecutive $2\,N_x$-periods is $f(2\,N_x)$. Obviously $f(N_x) < f(2\,N_x) < 2\,f(N_x)$.

Let us consider that the time axis is divided into N_x-periods as shown in Fig. 11. Then the cache entries at a certain time and at N_x-periods later can be categorized into the following five types (Fig. 12).

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 expire at time II.

B: IP addresses which have corresponding cache entries in the cache at time I, and are accessed during the following N_x-period, but their entries expire 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 last N_x-period and where also accessed during the next N_x-period and have cache entries at time II.

E: IP addresses which have no corresponding cache entries in the cache at time I, and were not accessed during the last N_x-period, but were accessed during the next N_x-period and have cache entries at time II.

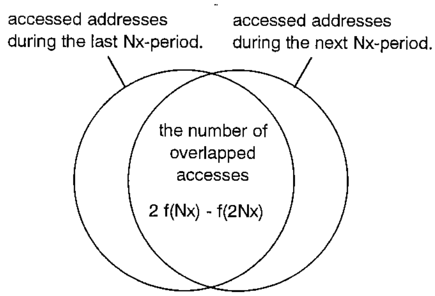


Fig. 13 Destination IP addresses generated in adjacent N_x-periods.

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. 13). In FIFO case, whether a cache entry survives or not is independent of frequency of access, and all entries have 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 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 miss-hit 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

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

Let us now 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 entries at time I, none of the entries 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 the N_x-period is the lifetime of a cache entry, the number of the restored entries at time II is a half of the 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:

$$4\,f(N_x) - 3\,k - \sqrt{9\,k^2 - 8\,k\,\{f(2\,N_x) - f(N_x)\}} = 0. \tag{26}$$

Using (26), N_x can be determined. Thus, we can write CHP for FIFO as

$$P_F = 1 - \frac{k}{N_x}. \tag{27}$$

5.5 Upper-Bound for Cache Hit Probability

For simplicity, we assume the access trends are repeated day by day according to the human cycle. Larger cycles over days, weeks, and months are neglected.

Let N_d be the average number of accesses for a day by all users accommodated by the edge router. Then, for both LRU and FIFO, the upper-bound P_{sup} for CHP is given as

$$P_{\text{sup}} = 1 - \frac{f(N_d) + k - f\left(f^{-1}(2\,k) - f^{-1}(k)\right)}{N_d}. \tag{28}$$

The quantity $k - f\left(f^{-1}(2\,k) - f^{-1}(k)\right)$ in (28) is a correction of the number of cache hit for the initial cache entries.

The above upper-bound is reached when many packets that have identical destinations are densely packed together. It is worth noting that the upper-bound P_{sup} is valid not only for LRU and FIFO algorithms, but also for any other aging algorithms [2].

5.6 Numerical Examples

Figures 14 and 15 show numerical examples and their comparison with experimental values with respect to both LRU and FIFO algorithms. The vertical axes of the both figures show (1−CHP) and the horizontal axes show the capacity k of the cache table. The experimental data are obtained using an access log of an HTTP proxy server that has about 60,000 accesses a day. Cache miss events are counted except for compulsory misses. In other words, they are counted after the cache becomes full.

These figures show that CHP evaluations (25) and (27) give sufficiently accurate values.

6. Address Cache Design

Although the above sections focus only on HTTP accesses, the cache table of the edge router, shown in Fig. 1 or 2, maintains all IP addresses for packet forwarding. Therefore, in order to achieve address cache design based on DZM, it is required that DZM for HTTP is also applicable to mixed packet stream.

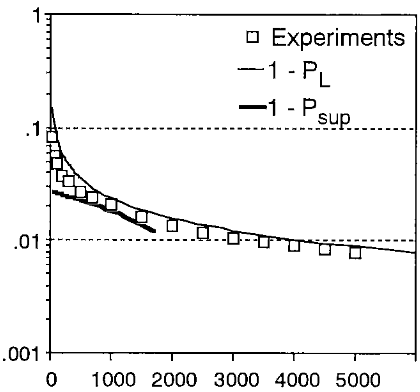


Fig. 14 Relationship between 1-CHP and capacity of the cache table (LRU).

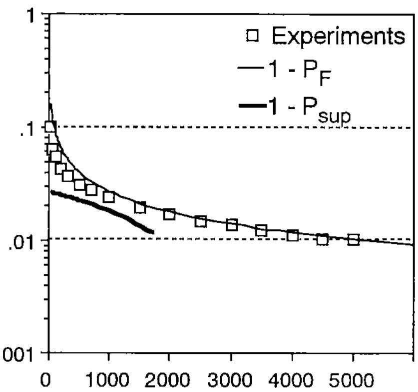


Fig. 15 Relationship between 1-CHP and capacity of the cache table (FIFO).

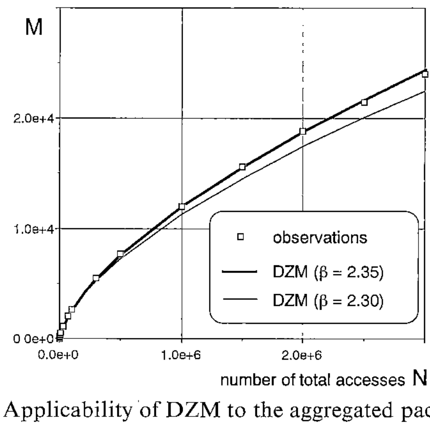


Fig. 16 Applicability of DZM to the aggregated packet stream.

We arrange the logs of both a proxy server (proxy-A) and an SMTP server in time order. The SMTP server accommodates the same users as the proxy-A. Both logs are for the same successive 30 days in Jan. '98. The arranged log includes HTTP, FTP and E-mail. Figure 16 shows the relationship between the number of accesses

N and the number of destination addresses M. Experimental data, denoted by dots, are obtained from the arranged log. Solid lines are calculated using DZM with respect to $\beta = 2.30$ and 2.35. Destination addresses of the arranged data are slightly diverse, but they can be described in the framework of DZM. Therefore, the evaluations of CHP, shown in Sect. 5, are also applicable to our models of Figs. 1 and 2, although the value of β may be different.

Figures 14 and 15 suggest the design of an address cache table can be accomplished as follows.

- Since the upper-bound P_{sup} is valid for any aging algorithm, we cannot expect that there is another complicated aging algorithm with remarkably higher CHP.
- For the same capacity of cache table, LRU is better than FIFO, but the difference is small.

Therefore, since FIFO is much simpler to implement than LRU, it is preferable, given enough cache capacity.

In evaluation of CHP, the only parameter β is required to determine. Its value can be determined by measuring a pair of the numbers of accesses and the destination addresses. The pair is easily obtained from access log data. In actual application, it is a strength of our model. Complicated measurements, e.g., correlation or higher order moments of address generations, are not required.

7. Conclusion

We have shown that the distribution of destination IP addresses can be characterized by two types of Zipf's law and have verified these laws using actual data from logs in proxy servers. Based on these Zipf's laws, we propose DZM and derive the relationship between the number of total accesses and the number of destination IP addresses. Using the derived relation, we derive the CHPs of the packet destination addresses and show guidelines determining both the capacity of an address cache table and the aging algorithm.

Applicability of our results is not limited to the network models shown as Figs. 1 and 2. Other application is design of a DNS server, for example.

A previous study in [2] gives the upper/lower bounds for cache hit probability. It was, however, only a preliminary evaluation and required the number of accesses in a day. Our new approach gives sufficiently dependable evaluations with respect to actual designing issues. In addition, since the number of accesses in a day is not required as the additional input, the application of our approach is not restricted by the capacity of the cache table.

The generalized treatment of DZM shown in Sect. 4 can describe other data networking services, e.g., E-mail, FTP, and so on. In these cases, in general, $\alpha_h \neq 1$ and/or $\alpha_r \neq 1$. Measuring these values and applying the generalized treatment to other types of data networking services remain for further study.

It is also expected that the concept of the DZM can be applied in other fields, for example, in marketing to show the relationship between the number of customers and the types of goods that customers buy.

References

[1] M. Aida and N. Takahashi, "Design of address cache table for data networking based on complementary use of the two types of Zipf's law," Proc. APSITT'97, Hanoi, Vietnam, 1997.
[2] M. Aida and N. Takahashi, "Evaluation of the number of destination hosts for data networking and its application to address cache design," Proc. ICCCN'97, pp.342–349, Las Vegas, NV, 1997.
[3] A.I. Chernyi, "Distributions of Zipfian type: An attempt at content interpretation," Int. Forum Inf. Doc., vol.19, no.1, 1994.
[4] S. Glassman, "A caching relay for world wide web," Comput. Networks and ISDN Systems, vol.27, pp.165–173, 1994.
[5] R. Jain, "Characteristics of destination address locality in computer networks: A comparison of caching schemes," DEC Tech. Rep., DEC-TR-592, 1990.
[6] M. Laubach, "Classical IP and ARP over ATM," RFC 1577, 1994.
[7] J.V. Luciani, D. Katz, D. Piscitello, and B. Cole, "NBMA Next Hop Resolution Protocol (NHRP)," IETF INTERNET-DRAFT, <draft-ietf-rolc-nhrp-11.txt>.
[8] M. Nabe, K. Baba, M. Murata, and H. Miyahara, "Analysis and modeling of WWW traffic for designing internet access networks," IEICE Trans., vol.J80-B-I, no.6, pp.428–437, June 1997.
[9] 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.

Appendix: Another Formulation of DZM

Another formulation of DZM Eqs. (4)–(6) is possible. Although, for large m, $R(m)$ decreases quickly in a log scale, $R(m)$ does not strictly satisfy Zipf's law up to $m \cong M$. Since the tail of $R(m)$ is characterized by the upper-tail of $H(n)$, we can derive a faithful model with respect to Eq. (6) without additional arbitrary parameters. We consider an HTTP case, i.e., $a_r = a_h = 1$. The new DZM consists of the following six equations:

$$\alpha'_r = -\frac{\ln 2}{\ln \frac{H(1)}{M}}, \tag{A·1}$$

$$m_0 = \exp \frac{\alpha'_r \ln M - \ln R(1)}{\alpha'_r - 1}, \tag{A·2}$$

$$\alpha'_r = \frac{\ln R'}{\ln M}, \tag{A·3}$$

$$M = H(1)\left(\gamma + \frac{1}{\beta}\ln R(1) + \frac{1}{2R(1)^{1/\beta}}\right) + \min\left(R(1)^{(\beta-1)/\beta}, R'\,R(1)^{-1/\beta}\right), \tag{A·4}$$

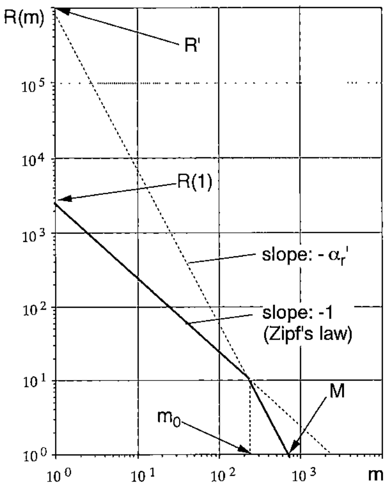


Fig. A· 1 Parameters in the another DZM.

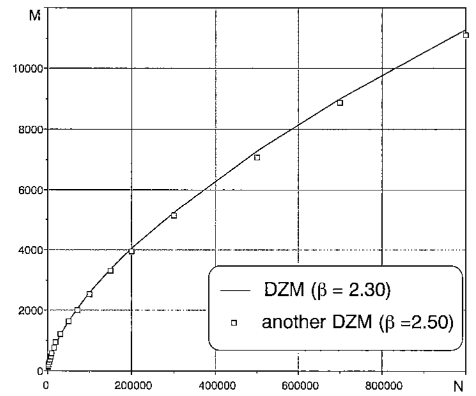


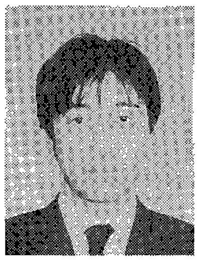
Fig. A· 2 Evaluation of M using the another DZM.

$$\begin{aligned} N &= H(1)R(1)^{1/\beta} \\ &\quad + R(1)\left[\gamma + \ln\min\left(m_0, R(1)^{(\beta-1)/\beta}\right)\right. \\ &\quad \left. + \frac{1}{2\min\left(m_0, R(1)^{(\beta-1)/\beta}\right)}\right] \\ &\quad + \left[R(1)^{(\beta-1)/\beta} - m_0\right]^+ R' \sum_{m=m_0+1}^{R(1)^{(\beta-1)/\beta}} \frac{1}{m^{\alpha_r'}}, \end{aligned} \tag{A· 5}$$

$$N = R(1)\left(\gamma + \ln m_0 + \frac{1}{m_0}\right) + R' \sum_{m=m_0+1}^{M} \frac{1}{m^{\alpha_r'}}, \tag{A· 6}$$

where $[x]^+$ denotes $\max(x, 0)$. The physical meanings of parameters in Eqs. (A· 1)–(A· 6) are shown in Fig. A· 1.

Figure A· 2 shows the comparison between the evaluations of M with respect to the total accesses N. Solid line denotes the evaluation of M calculated by the original DZM (4)–(6) with $\beta = 2.30$. The white boxes denote the evaluation of M calculated by the another DZM (A· 1)–(A· 6) with $\beta = 2.50$. Although the new DZM includes additional equations and requires more complicated treatment than the original DZM, accuracies of both evaluations using the above model and the faithful model are almost the same, in at least HTTP case.



Masaki Aida received the B.S. and M.S. degrees in Theoretical Physics from St. Paul's University, Tokyo, Japan, in 1987 and 1989, respectively. Since he joined NTT Laboratories in 1989, he had been mainly engaged in research on traffic issues in ATM networks and computer communication networks until March 1998. He is currently a manager at NTT Advanced Technology Corporation (NTT-AT) and has interests in network designing issues. He received the IEICE Young Engineer Award in 1996. Mr. Aida is a member of the Operation Research Society of Japan.

Noriyuki Takahashi received his B.S. and M.S. degrees in Information Science from Kyoto University, Japan, in 1990 and 1992, respectively. He joined NTT LSI Laboratories in 1992, where he has been conducting research on computer-aided design/testing for LSI. Since 1996, he has been a research engineer at NTT Multimedia Networks Laboratories. His current interests include high-speed wide-area networking, multimedia communication, network security. He is a member of IPSJ.

Tetsuya Abe received his B.S. degree in Physics from Keio University, Yokohama, Japan, in 1987. He joined NTT Laboratories in 1988, where he has been conducting research on process-device modeling and development on on-chip RISC controller. Since 1996, he has been a research engineer at NTT Multimedia Networks Laboratories. His current interests include high-speed protocol processing. He received the Best Industrial ASIC Award of ED&TC in 1996.