Evaluation of the Number of Destination Hosts for Data Networking and Its Application to Address Cache Design

Masaki Aida Noriyuki Takahashi
NTT Multimedia Networks Laboratories
1-1, Hikarino-oka, Yokosuka-shi, Kanagawa, 239, Japan

Abstract

This paper discusses how to design the capacity of address cache tables for large-scale computer communication networks. We show that destination addresses of packets can be assumed to be characterized by two types of Zipf's law. Based on the complementary use of these laws, we derive the relation 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 the upper/lower bounds for cache hit probability. Using the probabilities, design issues including the capacity of the cache table and aging algorithm of cache entries are also discussed.

1 Introduction

Computer communication networks are attracting attention and their importance in terms of expanding the market for data services is rapidly increasing. Hence, 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 undergo the following three fundamental processes in a router.

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

If we prepare a buffer with enough capacity for packet reassembling, we can easily avoid degradation of quality at the packet reassembling process. The processes that usually cause degradation of quality are the address-resolution and packet-transfer processes. If address resolution is not good enough, the throughput of packet transfer is diminished. In the packet transfer process, packet overflow at the output buffer causes degradation of quality and packet throughput is again diminished.

The nature of the quality degradation in the packet transfer process is, fundamentally, the same in the connection-oriented network. However, in order to analyze the quality degradation, it is necessary to know characteristics of data traffic. Such a traffic model is reported in [7] for WWW access.

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

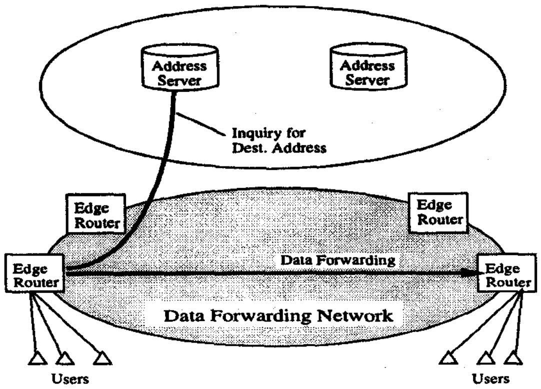
Figures 1 (a) and (b) show our network model. When a packet arrives at the edge router, the packet is processed for address resolution. In the case of Figure 1 (a), if the address cache table at the edge router contains the address corresponding to the destination of the packet, address resolution is immediately carried out. Otherwise, the router must request address resolution from the address server, which is a large database that has comprehensive information [5]. This inquiry and retrieval process greatly increases the address resolution time. In the case of Figure 1 (b), if the address cache table at the edge router does not contain the address corresponding to the destination of the packet, the router sends the packet to the default router, which has all address information [6]. Then the default router analyzes the address of the packet and transfers it.

In both cases, increasing the cache table capacity can be used to shorten the address resolution time or reduce the utilization of the default router, but doing so is both expensive and difficult to implement. In addition, the cache hit probability tends to increase very slowly as the capacity of the cache table is increased.

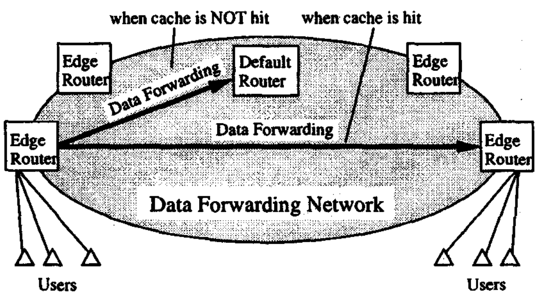
In this paper, we show that the distribution of destination host addresses can be characterized by two types of Zipf's law. Based on the complementary use of these laws, we derive the cache hit probability (CHP) of packet destination addresses and show how to design the capacity of the address cache table. To this end, we derive the relation between the number of total accesses and the number of destination hosts.

This paper is organized as follows: In Section 2, we introduce two types of Zipf's law describing HTTP access patterns. Based on the two laws, we show the idea of applying the laws complementarily and derive the relation between the number of accesses and destination hosts in Section 3. Section 4 shows the generalized treatment of the derivation in Section 3. In Section 5,

1095-2055/97 $10.00 © 1997 IEEE



(a) Address resolution using address server



(b) Address resolution using default router

Figure 1: Network models.

based on the relation acquired in Section 3, we discuss the CHP for some classical aging algorithms. Using the results in Section 5, we show how to design the address cache table with respect to both its capacity and aging algorithm in Section 6.

2 Distributions of Destination Hosts 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

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

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 $\alpha = 1$, that is,

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

In [4, 8], applying Zipf's law to HTTP access and its validity are discussed by analyzing a WWW access

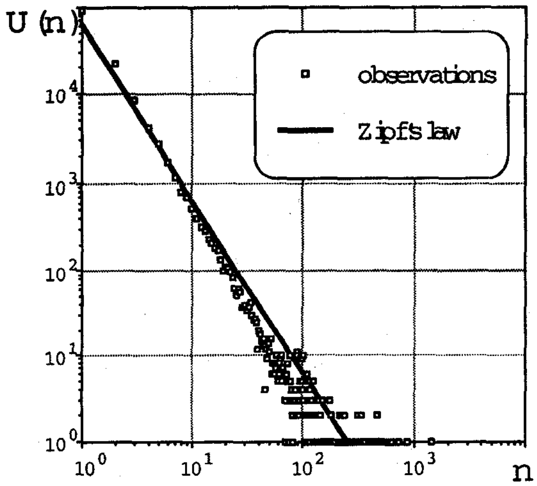


Figure 2: The relation between n, the number of hits, and $U(n)$, the number of URLs with n hits.

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, 8], that is,

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

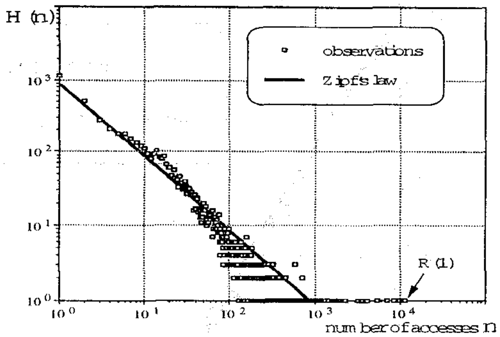
in the region of small n. Figure 2 shows a distribution for $U(n)$. These data are from logs of a proxy server (proxy-A as stated later). It is clear that the Zipf law (3) is applicable to the distribution in the region of small n.

In general, the structure of a URL is

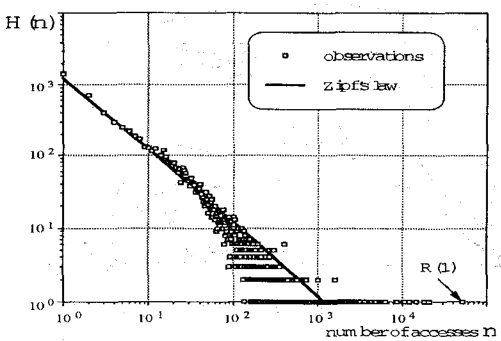
$$\texttt{http://}\underbrace{\texttt{host.in.some.domain}}_{\text{<host>}}\underbrace{\texttt{/FOO/BAR/BUZ.html}}_{\text{<path>}}$$

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 is only on the <host> part of the URL. Therefore, it is necessary to investigate the number of accesses for each <host>.

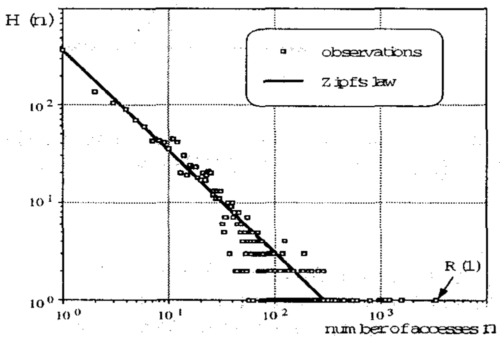
We apply Zipf's law to the distribution of packet destination <host> for describing the HTTP access pattern. Let the number of hosts be $H(n)$ such that each is accessed just n times. Figure 3 shows the distribution of $H(n)$ with respect to n. These data are from the logs of two different proxy servers 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. 3, it is clear that the Zipf law (1) with $\alpha = 1$ is applicable to these cases in the region of small n.



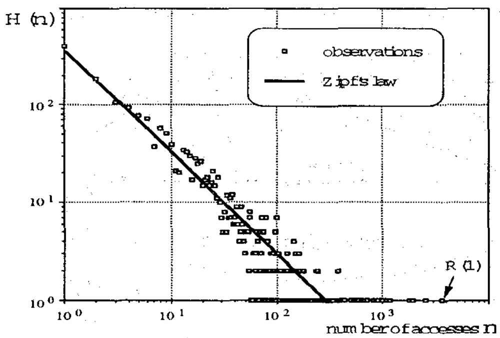
(a) proxy-A (Nov. 1996)



(b) proxy-A (Feb. 1997)

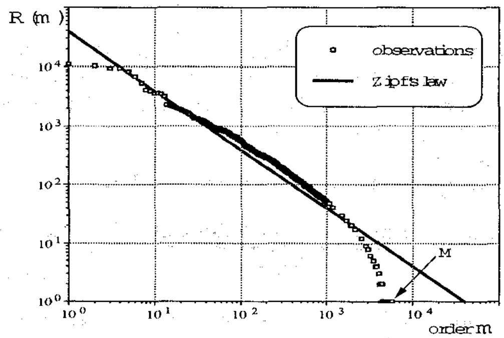


(c) proxy-B (Nov. 1996)

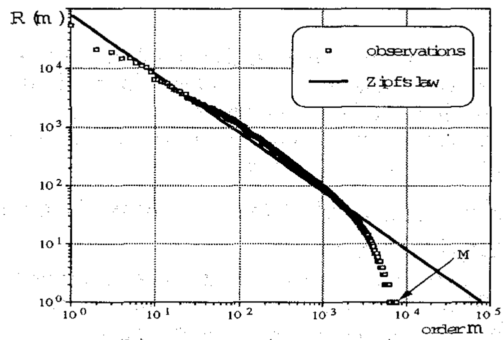


(d) proxy-B (Feb. 1997)

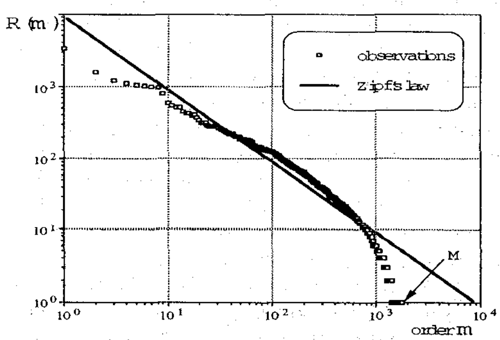
Figure 3: The relations between n, the number of accesses, and $H(n)$, the number of destination hosts with n accesses.



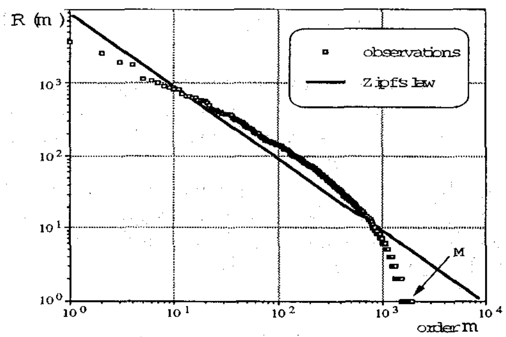
(a) proxy-A (Nov. 1996)



(b) proxy-A (Feb. 1997)



(d) proxy-B (Nov. 1996)



(c) proxy-B (Feb. 1997)

Figure 4: The relations between order m and $R(m)$, the number of accesses to the m^{th} order host.

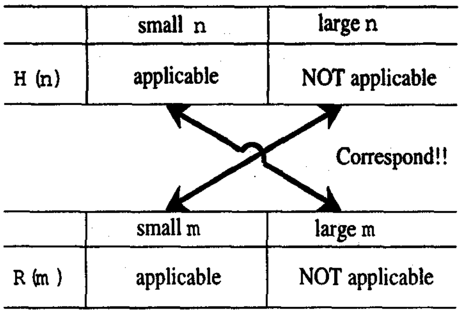


Figure 5: Complementary use of two types of Zipf's law.

On the other hand, we can consider another type of Zipf's law. We arrange all hosts in order of the number of times they are accessed, *i.e.*, we assign $m = 1$ to the most frequently accessed host, assign $m = 2$ to the next most frequently accessed host, and so on. Then let the number of accesses be $R(m)$ with respect to a host at m^{th} order. Figure 4 shows the distribution of $R(m)$ with respect to m. These data are from the logs of proxy-A and B. From Fig. 4, one can also see that the Zipf law (1) with $\alpha = 1$ is applicable to these cases in the region of small m.

3 Evaluation of the Number of Destination Hosts

3.1 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 n corresponds to small m, and, conversely, large m corresponds to small n (Fig. 5). Thus, it is natural to expect that the domain in which one Zipf law fails is one in which the other Zipf law is applicable. Therefore, we use the two types of Zipf's law complementarily.

Let the total number of accesses be N. The total number of destination hosts is denoted by M. We consider the two types of Zipf's law with respect to $H(n)$ and $R(m)$. From the numerical example of the proxy logs, the Zipf law with respect to $R(m)$ is approximately applicable up to $m \simeq M$. Note that the domain of m of $R(m)$ is $1 \leq m \leq M$, and $R(m) > 0$ for all m in this domain. On the other hand, the domain of n of $H(n)$ is $1 \leq n \leq R(1)$ restricted by the range of $R(m)$. In this case, although n is in this domain, there are n such that $H(n) = 0$.

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

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

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

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

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 $1 \leq n \leq R(1)^{1/\beta}$. Hence, $n = R(1)^{1/\beta}$ corresponds to $m = R(1)^{(\beta-1)/\beta}$ by using the other Zipf law with respect to $R(m)$. In addition, the number of hosts corresponding to $n \geq R(1)^{1/\beta}$ is equal to the order $m = R(1)^{(\beta-1)/\beta}$ of the other Zipf law with respect to $R(m)$.

N is also denoted by

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

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

In addition, N is also expressed as

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

The special case of evaluating M using (4) with $\beta = 2$ is discussed in [1].

3.2 Self-Consistent Treatment of the Two Types of Zipf's Law

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

$$M_0 = N \qquad (7)$$

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

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

$$H_i = R_i^{(\beta-1)/\beta}\left[\ln M_{i-1} - \frac{\beta-1}{\beta}\ln R_i + \frac{1}{2M_{i-1}} - \frac{1}{2R_i^{(\beta-1)/\beta}}\right], \quad (9)$$

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

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

$$M = M_j, \quad (11)$$
$$R(1) = R_j, \quad (12)$$
$$H(1) = H_j. \quad (13)$$

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

3.3 Comparison Evaluated and Experimental M

Here we show the accuracy of the calculated M, that is, the number of destination hosts when the total number of accesses are given.

Figure 6 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 hosts.

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 a Bayesian estimation [9] of M is possible. If we only know M for small N, then we can evaluate β and can evaluate M for large 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 universality. This will require further study using many data from different proxies. So far, all βs derived from our current data are in $2.2 \leq \beta \leq 2.45$.

4 Generalized Self-Consistent Treatment

In the HTTP case, the Zipf law (1) with $\alpha = 1$ is utilized to describe $H(n)$ and $R(m)$ as shown in Sec. 3. However, in general, it is not guaranteed that the distributions of destination hosts are described by the same Zipf law. It is known that the range of α is restricted to $1 \leq \alpha \leq 2$ for most social phenomena.

This section shows the generalization of the self-consistent treatment, which is applicable to cases of $\alpha \neq 1$. Let α of Zipf 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:

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

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

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

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

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

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

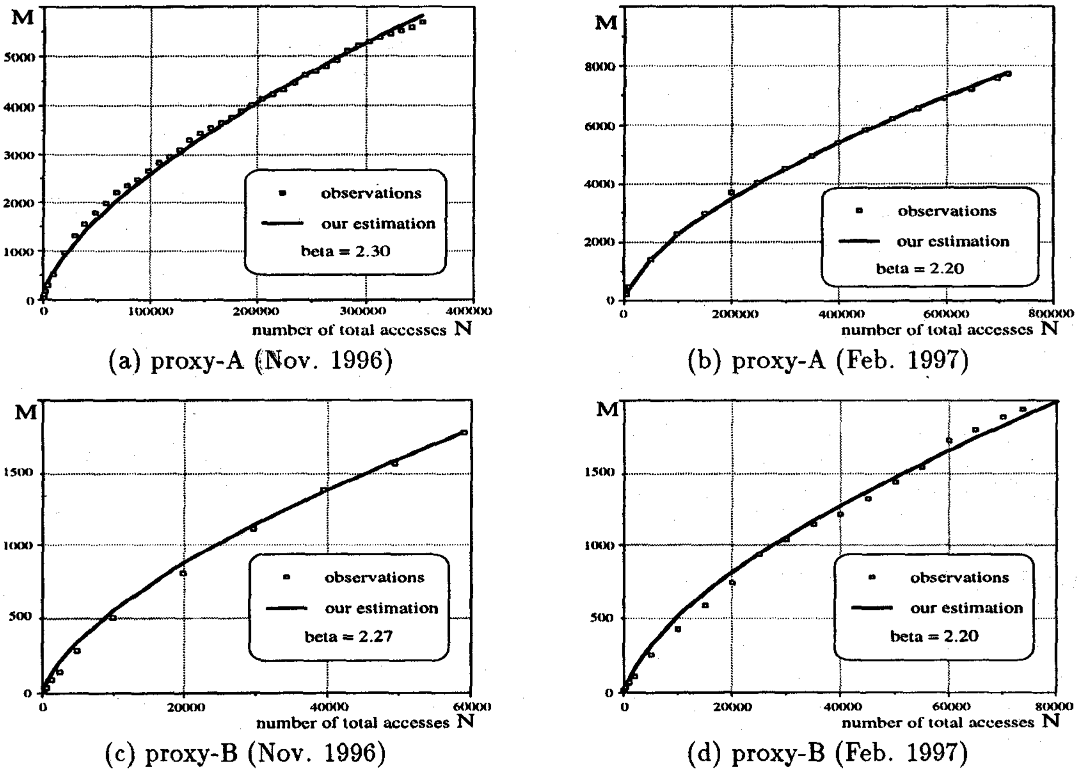
$$M_i = H_i\sum_{n=1}^{\lfloor R_i^{1/\beta}\rfloor} \frac{1}{n^{\alpha_h}} + \left[R_i^{(\beta-1)/\beta}\right]^{1/\alpha_r}. \quad (20)$$

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

$$M = M_j, \quad (21)$$
$$R(1) = R_j, \quad (22)$$
$$H(1) = H_j. \quad (23)$$

5 Evaluations of Cache Hit Probability

Using three classical aging algorithms, we consider CHPs of a destination address making a hit in the



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

Figure 6: Accuracy of evaluated M.

address cache table. For simplicity, we assume the access trends are repeated day by day, and changes in the trends by the day, month, etc. are neglected.

Let m_0 be the capacity of the address cache table at an input port in an edge router as shown in Fig. 7 and N be the average number of accesses for a day by all users accommodated by the input port.

5.1 Aging Algorithms for Cache Entries

Let us focus on the following three classical algorithms.

- Fixed entry
  Initially, a cache table holds fixed entries, and they are not changed, even when the cache is not hit.
- FIFO algorithm
  When the cache is not hit, new address data is written over the oldest present cache entry.
- LRU algorithm
  When the cache is not hit, new address data is written over the least recently used cache entry.

5.1.1 Upper-Bound Cache Hit Probability for Fixed Entry

In order to evaluate the upper-bound for CHP with respect to the fixed entry scheme, we assume that the cache table has top m_0 addresses that are the most frequently accessed.

Using Zipf's law with respect to $R(m)$, the number of cache hits is $\sum_{m=1}^{m_0} R(m)$. Then the upper-bound for CHP P_{fixed} is given by

$$P_{\text{fixed}} = \frac{\sum_{m=1}^{m_0} R(m)}{N} \simeq \frac{\gamma + \ln m_0 + \frac{1}{2m_0}}{\gamma + \ln M + \frac{1}{2M}}. \qquad (24)$$

Since the upper-bound is realized only when the above assumption is valid, it is necessary to accurately predict the frequently accessed destinations. Since, in general, the prediction is almost impossible, the actual CHP is greatly diminished. We, therefore, regard the fixed entry scheme as not applicable to actual implementation for cache table design, and it is out of consideration.

5.1.2 Upper-Bound Cache Hit Probability for FIFO/LRU Algorithms

In both FIFO and LRU algorithms, the upper-bound for CHP is given by the following identical form:

$$P_{\text{sup}} = \frac{N - M + m_\delta - m_0}{N}, \qquad (25)$$

where N_m and N_{2m} are numbers of accesses such that m_0 and $2m_0$ different destinations, respectively, are accessed. In addition, when N_m and N_{2m} total accesses occur, let the numbers of accesses to the most frequently accessed destinations be R_m and R_{2m}, respectively, and we define

$$N_\delta := N_{2m} - N_m, \quad (26)$$
$$R_\delta := R_{2m} - R_m. \quad (27)$$

Moreover, m_δ denotes the number of destination addresses when N_δ total accesses occur. These values, that is, N_m, N_{2m}, R_m, R_{2m}, and m_δ are easily obtained by using relations (4)–(6).

The above upper-bound is realized when many packets having an identical destination are densely packed together. It is worth noting that the upper-bound P_{sup} is valid not only for FIFO and LRU algorithms, but also for any other aging algorithms.

5.1.3 Lower-Bound Cache Hit Probability for FIFO/LRU Algorithms

In contrast with the above case, when packets having an identical destination are generated in a scattered manner with respect to their average interval, CHPs become their lower-bounds. Let the lower-bounds with respect to FIFO and LRU algorithms be P_{finf} and P_{linf}, respectively. Then they are expressed as

$$P_{\text{finf}} = 1 - \frac{1}{N}\left\{\sum_{i=1}^{\lfloor M \rfloor} \min\left[\max\left[\frac{R}{R_\delta}, 1\right], \max\left[\frac{R}{i}, 1\right]\right] - m_\delta + m_0\right\}, \quad (28)$$

$$P_{\text{linf}} = 1 - \frac{1}{N}\left\{\sum_{i=1}^{\lfloor M \rfloor} \min\left[\max\left[\frac{R_\delta N - R N_\delta}{R_\delta N_\delta}, 1\right], \max\left[\frac{R}{i}, 1\right]\right] - m_\delta + m_0\right\}. \quad (29)$$

6 Design of Address Cache Table

Figure 8 shows numerical examples of the above upper/lower bounds and their comparison with experimental values with respect to both FIFO and LRU algorithms. The experimental data are obtained using access logs of an HTTP proxy server that has about 60,000 accesses a day. The vertical axis of Fig. 8 shows (1 − CHP) and the horizontal shows the capacity of the cache table.

From Fig. 8, the experimental values are close to the optimal ones, that is, $(1 - P_{\text{sup}})$. This means that accesses to identical destinations are generated with a strong correlation. If we can describe the correlation by an appropriate model, more accurate evaluation for experimental values may be possible. Since the correlation, however, is not a quantity controlled by networks, the model must contain additional assumptions about the correlation. Thus we can consider only the upper/lower-bound evaluations and design the capacity of the cache table for the worst case.

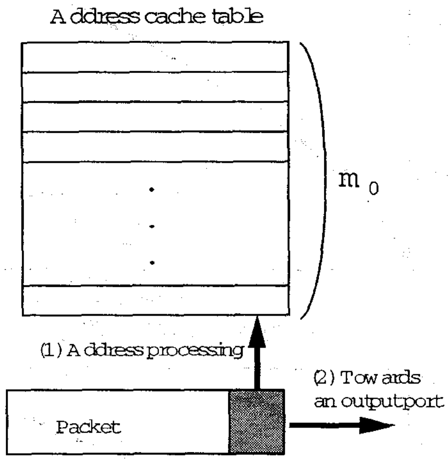


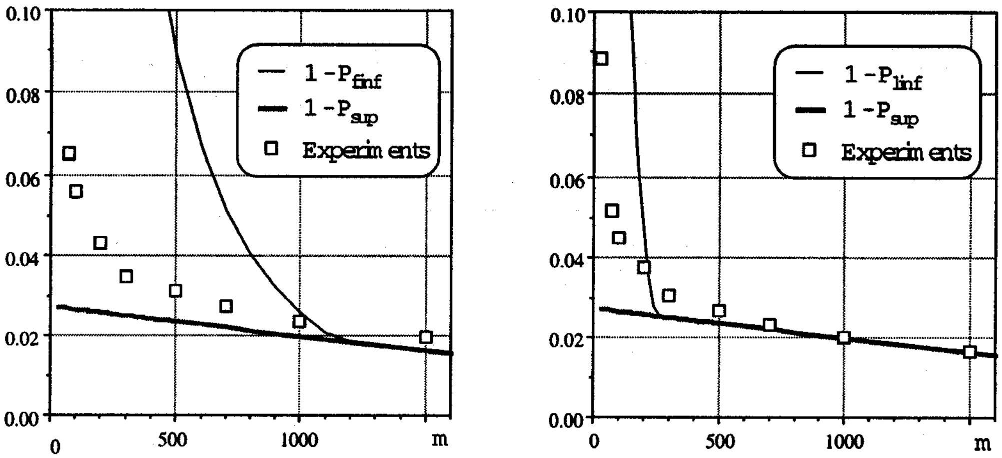
Figure 7: Address processing in an edge router.

In order to obtain an objective CHP, we determine the capacity of the cache table by using (28) or (29). Then, from Fig. 8, we have general guidelines determining the capacity as follows:

- When the upper and lower-bounds are close to each other, it means that CHP can be controlled in such a narrow rage. Thereby the capacity of the cache table should be designed such that the upper/lower-bounds give close values. We call such a region of the capacity a controllable region.
- Since the upper-bound P_{sup} is determined independently of the difference of aging algorithms, it is not necessary to consider more complicated aging algorithms in the controllable region.

In addition, from a comparison of the two aging algorithms, we have the following guidelines:

- The case when the cache table has small capacity, *e.g.*, $300 \leq m_0 \leq 1200$, as in Fig. 8.
Since the LRU algorithm gives close upper/lower bounds with in a smaller capacity of the cache table than does the FIFO algorithm, the LRU algorithm is preferable in terms of controlling the CHP.
- The case when the cache table has a large capacity, *e.g.*, $1200 < m_0$, as in Fig. 8.
Both LRU and FIFO algorithms give close upper/lower bounds; the FIFO algorithm is preferable in terms of simplicity of implementation.



FIFO algorithm LRU algorithm

Figure 8: Relations between CHP and capacity of the cache table.

7 Conclusion

We have shown that the distribution of destination host 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 the complementary use of these Zipf's laws, we derive the upper/lower-bounds for CHPs of the packet destination addresses and show guidelines determining both the capacity of an address cache table and the aging algorithm. In these processes, we derive the relation between the number of total accesses and the number of destination hosts.

The generalized treatment shown in Sec. 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 treatment can be applied in other fields, for example, in marketing to show the relation between the number of customers and the types of goods that customers buy.

References

[1] M. Aida, and N. Takahashi, "Evaluation of the Number of Destination Hosts for Data Networking and Its Application," *IEICE Tech. Rep.*, IN97-26 (1997-04), 1997 (in Japanese).

[2] 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. of APSITT '97*, Hanoi, Vietnam, 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] H. Kitazume, T. Ikegawa, and T. Kubo, "An Architecture for Multiprotocol Connectionless Services over ATM Networks," *Proc. of APSITT '93*, Bangkok, Thailand, 1993.

[6] 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>.

[7] M. Nabe, K. Baba, M. Murata, and H. Miyahara, "Analysis and Synthesis of World-Wide-Web Traffic," *IEICE Tech. Rep.*, SSE96-90 (1996-06), 1996 (in Japanese).

[8] 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 (in Japanese).

[9] H. Saito, "Bayesian Performance Estimation Driven by Performance Monitoring and Its Application," *IEICE Trans. Commun.*, vol. E79-B, no. 1, 1996.