PAPER *Special Section on Technology and Architecture for Sustainable Growth of the Internet*

Access Load Balancing with Analogy to Thermal Diffusion for Dynamic P2P File-Sharing Environments*

Masanori TAKAOKA[†,††a)], **Masato UCHIDA**[†††], **Kei OHNISHI**[††], ***Members, and*** **Yuji OIE**[††], ***Fellow***

SUMMARY In this paper, we propose a file replication method to achieve load balancing in terms of write access to storage device ("write storage access load balancing" for short) in unstructured peer-to-peer (P2P) file-sharing networks in which the popularity trend of queried files varies dynamically. The proposed method uses a write storage access ratio as a load balance index value in order to stabilize dynamic P2P file-sharing environments adaptively. In the proposed method, each peer autonomously controls the file replication ratio, which is defined as a probability to create the replica of the file in order to uniform write storage access loads in the similar way to thermal diffusion phenomena. Theoretical analysis results show that the behavior of the proposed method actually has an analogy to a thermal diffusion equation. In addition, simulation results reveal that the proposed method has an ability to realize write storage access load balancing in the dynamic P2P file-sharing environments.

key words: *P2P file sharing, file replication, write storage access load balancing, thermal diffusion, dynamic environment*

1. Introduction

Unstructured peer-to-peer (P2P) file-sharing networks [1] require a mechanism to find queried files certainly and quickly, because these networks have no mechanism to manage file locations in the network. A mechanism to enhance file search performance is to make replicas of files so as to increase the total number of files in the network [2], [3]. However, if files are frequently replicated in particular peers that are easily accessed by a lot of peers in order to enhance reliability and response speed to find queried files, storage accesses, i.e., loads, are biased toward these peers [4]. Therefore, unstructured P2P file-sharing networks also require an access load balancing mechanism in order to stabilize themselves.

P2P file-sharing networks are usually dynamic environments. For instance, a network topology of a P2P file-sharing network keeps changing due to departure and participation of peers, query distribution in a P2P file-sharing network also keeps changing due to appearance of new files, and so forth. Therefore, a load balancing mechanism must adaptively stabilize such a dynamic P2P file-sharing environment. Furthermore, P2P file-sharing networks are large-scaled, and therefore, a load balancing mechanism should be not centralized but autonomous and distributed.

In the previous study [5] we proposed a file replication method inspired by thermal diffusion phenomena, which can balance loads of peers in an autonomous and distributed manner. However, we did not considered adaptability of the file replication method to the dynamic environment. In the previous method, the load of a peer is defined as the *storage utilization ratio* that is the ratio of the amount of files to storage capacity. Since the storage utilization ratio is a cumulative value with the upper limit value, the previous method cannot distinguish loads of peers when the storage utilization ratios of the peers reach the upper limit value. Therefore, once the storage is saturated with files, the previous method can no longer adaptively stabilize the dynamic environment.

In the present paper, we focus on adaptability of the replication method to the dynamic P2P file-sharing environment. We propose a file replication method using the *write storage access ratio* as the index of the load of a peer, which is a temporal variation of the frequency of write access to storage device. We focus on write access to storage device because it immediately arises from the consequence of file replications. The proposed file replication method is also inspired by thermal diffusion phenomena as our previously proposed method in [5]. The write storage access ratios of all or most peers cannot hardly become equal to the upper limit value, and therefore, we expect that the write storage access ratio can be used for sensing a change of the dynamic environment.

The present paper is organized as follows. Section 2 briefly describes related research. Section 3 presents the definition of the write storage access ratio as the load of a peer and a file replication method using the write storage access ratio. In addition, an analogy between the proposed replication method and a diffusion equation is analytically shown in Sect. 3. Section 4 shows the results of simulations. Section 5 gives the conclusions.

2. Related Research

In the present paper, we present a method to dynamically balance loads of peers in unstructured P2P file-sharing networks, using an analogy to thermal diffusion phenomena.

Manuscript received September 16, 2009.
Manuscript revised December 28, 2009.

[†]The author is with the NCOS Laboratory, NEC Communication Systems, Ikoma-shi, 630-0101 Japan.
[††]The authors are with the Graduate School of Computer Science and System Engineering, Kyushu Institute of Technology, Iizuka-shi, 820-8502 Japan.
[†††]The author is with the Network Design Research Center, Kyushu Institute of Technology, Tokyo, 100-0011 Japan.

[*]The earlier version of this paper was presented at The Third International Multi-Conference on Computing in the Global Information Technology (ICCGI2008).

a) E-mail: takaoka.msn@ncos.nec.co.jp
DOI: 10.1587/transcom.E93.B.1140

We here describe studies on dynamic load balancing in unstructured P2P file-sharing networks and studies that use a diffusion scheme as methodology.

2.1 Load Balancing for File-Sharing Networks

Various file replication methods in unstructured P2P file-sharing networks have been studied [2], [3], [5]–[7].

The path replication method [2] creates replicas of queried files on all the peers on the successful search path. The path replication method can achieve near optimal search performance. In [6], in order to improve the search performance of the path replication method, a file replication method that creates replicas of files on the peers on the successful search path with probabilities that are determined based on the global information about the network, i.e., the rank of the item's query rate among all data items. Although these methods have good search performance, they do not consider load balancing performance.

The path random replication method creates replicas of queried files on the peers on the successful search path with constant probability. The path random replication method can reduce the overall storage load of the P2P network, which comes from file replications, compared with the path replication method. However, the path random replication method does not have the mechanism to balance the imbalance in load distribution.

On the other hand, we have proposed file replication methods for load balancing in unstructured P2P file-sharing networks in [5] and [7]. These methods can balance loads in networks, where the storage utilization ratio mentioned in Sect. 1 is adopted as the load of a peer. These methods are probabilistic replication schemes, which create replicas of files in peers on a successful search path with a certain probability. This probability is referred to as replication probability. The file replication method presented in [5] controls the replication probability of all peers to obtain the local equilibrium of storage utilization ratios as thermal diffusion phenomena. Since this method uses only local information, the method works in an autonomous and distributed manner. This method can broadly tune a performance trade-off between search and load balancing, however, adaptability to the dynamic environment is not considered.

2.2 Diffusion Scheme as Methodology

In [8], [9] and [10], dynamic load balancing methods based on a diffusion equation have been studied for distributed computing. In [11], the authors have surveyed load balancing methods in grid computing, and address the new challenges in this area. These load balancing methods are all dynamic and distributed, because systems to which they are applied are large-scaled and dynamic.

In [12], a diffusive scheme is used for a routing mechanism in sensor networks. A data dissemination paradigm is developed for the routing mechanism. The authors refer to this paradigm as directed diffusion. This method forms a gradient field from local information, and then packets are propagated on the gradient field as diffusion phenomena. The goal of this method is to obtain high robustness and achieve low energy consumption. In [13], a diffusive scheme is used for decentralization of traffic in high speed networks. This method autonomously decentralizes packets, which are buffered in nodes, using only local information in order to avoid concentration of packets on some nodes in a high speed network. The authors used the diffusion equation in order to determine the parameters in the diffusive scheme.

In [14], a sensor network topology is created with a concept of reaction-diffusion phenomena. The authors adopt a reaction-diffusion model to a sensor network in order to create a topology for periodic data gathering. The goal of this method is to create a network topology with low energy consumption and low latency. In [15], a P2P file-sharing network is modeled as diffusion phenomena. The authors adopt epidemic model of mathematical biology to a hybrid P2P file-sharing network. And then, the authors attempt to get a file propagation model, using stateful nodes and local information.

The above-mentioned methods and the proposed method are similar in that they are inspired by thermal diffusion phenomena. However, they are different in the observed objects for controlling the network, the controlled objects based on the observation, and the purpose of the control. Therefore, we can not compare these methods with our proposed method without translating these methods each other.

3. File Replication Using Write Storage Access Ratio

In this section, we define the write storage access ratio as a load index and a file replication method using this index.

3.1 Difference between Previous and Present Methods

In the previous study [5] we proposed a file replication method with analogy to thermal diffusion for achieving write storage access load balancing. The replication method creates replicas of queried files on the successful search path, on which queries of files are transferred to retrieve. Our objective in the previous study was not only to achieve good search performance, but also to achieve write storage access load balancing in an autonomous and distributed manner. Here, we adopted the *storage utilization ratio* as the load of a peer. The storage utilization ratio is the ratio of the amount of files to storage capacity. This method can broadly tune the performance trade-off between search and write storage access load balancing. This method is a probabilistic replication scheme, which create replicas of files in peers on the successful search path. Then, loads can be balanced by controlling the replication probabilities in peers, which are probabilities to replicate the transfered file. The replication probability of a peer is determined with storage utilization ratios of neighboring peers to obtain the local equilibrium of storage utilization ratios. System behavior induced by this control mechanism is analogous to thermal diffusion phenomena, which means that equalizing local thermal gradients results in thermal systems achieving uniform temperature.

The previously proposed method dynamically allocates replicas of files using the differences in the write storage access load among peers, and can therefore balance write storage access loads. However, we found that the storage utilization ratio is not suitable for the index of the load. One reason for this is that the storage utilization ratio is a cumulative number of stored replicas, and therefore, would not be an appropriate value when access concentration occurred during a short period of time, which would affect the system largely. Another reason is that the storage utilization ratio has the upper limit value, so there is no difference in the storage utilization ratio among peers with fully occupied storage. Therefore, the replication method can not know the difference in the load among these peers.

In this section, we propose a file replication method for write storage access load balancing to adaptively stabilize the dynamic environment. The proposed method is based on the previous study [5], in which we used an analogy to thermal diffusion phenomena. However, unlike the previous study, in the proposed method, we adopt the *write storage access ratio*, which is a temporal variation of the access frequency, as the load of a peer. Since the write storage access ratio is a temporal variation of the access frequency and has no convergence value, the write storage access ratio is expected to be a good index of loads in the dynamic environment. In addition, the thermal diffusion-based method is expected to be able to balance loads of peers in a dynamic, distributed, and autonomous manner. We then compare the proposed method and our previous method through simulations.

3.2 Write Storage Access Ratio

There are two types of storage loads caused by file replications from a quantitative point of view, namely cumulative load and momentary load. These correspond to utilization of storage and access to storage, respectively. The former load represents consumption of storage capacity. The latter load represents consumption of CPU resource, network communication processing resource, device accessing resource, and so forth. In the previous study [5], we adopted the former, referring to the ratio of the amount of files to storage capacity as the storage utilization ratio. In the present study, we adopt the latter. The storage access load is classified into two types: read access to storage device and write access to storage device. In this paper, we are interested in the storage loads that immediately arise from the consequence of file creations and deployments. Therefore, we focus on write access to storage device. However, since load balancing in terms of write access to storage device would realize uniform deployment of replicas on peers in the network, we can expect that it realizes load balancing in terms of read access to storage at the same time. We herein refer to the frequency of writing accesses to storage per unit time as the write storage access ratio. The write storage access ratio is a temporal variation of access frequency, on the other hand, the storage utilization ratio is cumulative value of access. Although the storage utilization ratio increases monotonically with time up to the upper limit value, the write storage access ratio may not converge. Therefore, the write storage access ratio can be used as an index to control file-sharing networks at any time.

The write storage access ratio is defined with access histories. For simplicity, we regard the time variable t as discrete. A search and file replication are carried out at each discrete time. If we define the write storage access ratio $L(x,t)$ as the simple moving average (SMA) [16] of time-series data of access history $I(x,t)$, each peer requires a memory that can store N access histories. Here, $I(x,t)$ is an access on the x-th peer at t, that is, $I(x,t) = 1$ if accessed and $I(x,t) = 0$ if not accessed, N is the time used to derive the average write storage access ratio. In order to reduce the amount of memory required, we define the write storage access ratio as modified moving average (MMA) [16] of time-series data of access history $I(x,t)$:

$$L(x,t) = \frac{1}{N}(I(x,t-1) + (N-1)L(x,t-1)), \tag{1}$$

where $L(x,t)$ satisfies the inequality $0 \leq L(x,t) \leq 1$. From Eq. (1), it follows that the write storage access ratio at the current time, $L(x,t)$, can be determined using only two values. These values are $I(x,t-1)$, the access history at $t-1$, and $L(x,t-1)$, the write storage access ratio at $t-1$.

3.3 File Replication Method

The file replication method in the present paper is described in the previous study [5]. However, we briefly introduce this method in this subsection. In thermal diffusion, a non-uniform temperature distribution becomes uniform as a result of the integrating actions of local thermal conduction. Similarly, in write storage access load balancing in P2P file-sharing networks, we assume that actions of local interactions are integrated and then the non-uniform distribution of storage loads, i.e., write storage access ratios, becomes uniform.

First, we introduce an operator D to $L(x,t)$. The operator D derives a new function $DL(x,t)$ from the function $L(x,t)$, where $DL(x,t)$ is defined as the difference between the write storage access ratio of the peer and the average of its neighboring peers. That is, $DL(x,t)$ is given by the following equation:

$$DL(x,t) = \frac{1}{d(x)}\sum_{i=1}^{d(x)} L(x_i,t) - L(x,t), \tag{2}$$

where x_i $(i = 1,\ldots,d(x))$ is a peer adjacent to x, and $d(x)$ is the degree of peer x. From $0 \leq L(x,t) \leq 1$, $DL(x,t)$ satisfies the inequality $-1 \leq DL(x,t) \leq 1$. Equation (2) reduces to

$$DL(x,t) = \frac{1}{d(x)(d(x)-1)} \sum_{i<j} \left\{ (L(x_i,t) - L(x,t)) - \left(L(x,t) - L(x_j,t) \right) \right\}. \quad (3)$$

This equation has an analogy to the second derivative of a function $f(x)$:

$$\frac{d^2 f(x)}{dx^2} = \lim_{\delta x \to 0} \frac{\frac{f(x+\delta x)-f(x)}{\delta x} - \frac{f(x)-f(x-\delta x)}{\delta x}}{\delta x}.$$

Therefore, D on networks corresponds to the Laplacian, $\Delta = \nabla^2$, which is second derivative.

Using $DL(x,t)$, we can determine how much bigger or smaller a storage load of peer x is compared to those of its neighbors. Therefore, we define the file replication probability of peer x at t, $P(x,t)$, as a function of $DL(x,t)$. If $P(x,t)$ is a linear function of $DL(x,t)$, we can naively obtain

$$P(x,t) = \frac{1}{2} + \frac{1}{2} DL(x,t), \quad (4)$$

because $-1 \leq DL(x,t) \leq 1$. However, from a practical point of view, $P(x,t)$ should be more flexible in order to set the sensitivity of $P(x,t)$ to $DL(x,t)$. Therefore, we modify $P(x,t)$ as follows:

$$P(x,t) = \frac{1}{2} + \frac{1}{2} \tanh\left(\mu + \lambda \tanh^{-1} DL(x,t) \right), \quad (5)$$

where λ and μ are parameters. The meaning of these parameters is discussed in Sect. 3.4. Figure 1 shows several instances of $P(x,t)$ for combinations of $\mu = 0$ and several values of λ.

In the following, we show that existing schemes, path replication and path random replication, are involved in the proposed method as its special cases. First, when $\lambda = 0$ in Eq. (5), we obtain

$$P(x,t) = \frac{1}{2} + \frac{1}{2} \tanh(\mu). \quad (6)$$

The file replication probability $P(x,t)$ in Eq. (6) is independent of the write storage access ratio $L(x,t)$. Therefore, our proposed scheme with $\lambda = 0$ is equivalent to path random replication, which creates replicas of queried files on the peers on the successful search path with constant probability. For example, when $\mu = 0$ in Eq. (6), we obtain

$$P(x,t) = \frac{1}{2},$$

which represents the path random replication with replication probability of 0.5. In addition, when $\mu \to \infty$ in Eq. (6), we obtain

$$P(x,t) = 1,$$

which represents the path replication.

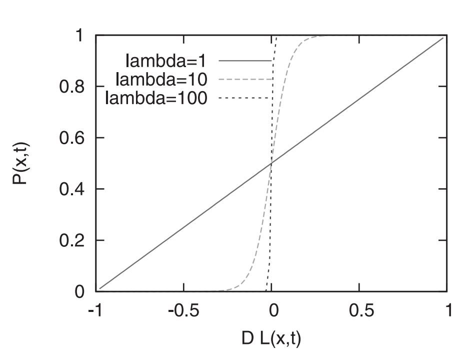


Fig. 1 Examples for $P(x,t)$.

3.4 Analogy to Thermal Diffusion

In this subsection, we show analytically that the proposed method has an analogy to a thermal diffusion equation. This analogy is justified by the fact that the state equation of the write storage access ratio has the diffusion term.

First, Eq. (1) is transformed to

$$L(x,t) - L(x,t-1) = \frac{1}{N}(I(x,t-1) - L(x,t-1)).$$

Here, we consider continuous form of this discrete equation given by

$$\frac{L(x,t+\delta t) - L(x,t)}{\delta t} = \frac{1}{N}(I(x,t) - L(x,t)).$$

Therefore, when $\delta t \to 0$, we obtain

$$\frac{dL(x,t)}{dt} = \frac{1}{N}(I(x,t) - L(x,t)). \quad (7)$$

We regard Eq. (7) as a state equation at time-space point (x,t). In addition, although $I(x,t)$ is a variable of time-space point (x,t), we replace $I(x,t)$ with its expectation value in order to see the characteristics of Eq. (7). The expectation value of $I(x,t)$ is given by the probability of an access which is expressed by the product of $P_{path}(x,t)$ and $P_{replica}(x,t)$ as

$$P_{path}(x,t) \cdot P_{replica}(x,t), \quad (8)$$

where $P_{path}(x,t)$ represents the probability of being on random walk path of query, and $P_{replica}(x,t)$ represents the replication probability. By substituting Eq. (8) into $I(x,t)$ in Eq. (7), we obtain

$$\frac{dL(x,t)}{dt} = \frac{1}{N}(P_{path}(x,t) \cdot P_{replica}(x,t) - L(x,t)). \quad (9)$$

We then analyze mathematically the scheme when replication probability is given by Eq. (5). By applying Maclaurin expansion to the replication probability (5) for $DL(x,t) \sim 0$ and $\mu \sim 0$, we obtain

$$P_{replica}(x,t) \simeq \frac{1}{2} + \frac{1}{2}(\mu + \lambda DL(x,t)). \quad (10)$$

By substituting Eq. (10) into $P_{replica}(x,t)$ in Eq. (9), we obtain

$$\frac{dL(x,t)}{dt} = \frac{1}{N}\left(P_{path}(x,t)\left(\frac{1}{2}(1+\mu) + \frac{1}{2}\lambda DL(x,t)\right) - L(x,t)\right). \quad (11)$$

Finally, if the query is on the search path, i.e., $P_{path}(x,t) = 1$, we obtain

$$\frac{dL(x,t)}{dt} = \frac{1+\mu}{2N} + \frac{\lambda}{2N}DL(x,t) - \frac{1}{N}L(x,t). \quad (12)$$

The left side of Eq. (12) is a time variation of the write storage access ratio, and three terms in the right side are a heat term, a diffusion term and a reaction term, respectively.

In the state Eq. (12), there is a diffusion term. This suggests that a time variation of the write storage access ratio is diffused spatially. Therefore, we can expect that the write storage access ratio is uniformed spatially.

We discuss the meaning of λ and μ. Since the heat term (the first term of the right side) in the state Eq. (12) has μ, we can expect that the greater μ is, the greater write storage access ratio $L(x,t)$ is. When we set μ to be large, we expect that search performance becomes high and load balancing performance becomes low. On the other hand, since the diffusion term (the second term of the right side) in the state Eq. (12) has λ as coefficient of $DL(x,t)$, we can expect that the greater λ is, the higher sensitivity of a variation of the write storage access ratio $L(x,t)$ is. When we set λ to be large, we expect that sudden deterioration of the unbalance in the write storage access ratio $L(x,t)$ recovers sooner. We will demonstrate our mentioned expectations here through simulations in Sect. 5.

4. Simulation

We examine the adaptability of the replication methods in a dynamic P2P environment through simulations. Correctly, we observe the variation of search and load balancing performances with time in the dynamic P2P environment. We here focus only on change in popularity trends of files as a factor in causing the dynamic P2P environment. It is possible to cause sudden change in write storage access load distribution among peers by the change in popularity trends of files. The P2P file-sharing environment without the change in the popularity trends of files is stable and homogeneous. In this section, we compare the proposed method and the previous method [5]. The previous method uses the storage utilization ratio, rather than the write storage access ratio, as the load, $L(x,t)$.

4.1 Simulation Model

The configuration of the P2P network simulation model is described as follows.

The total number of peers in the network is 10,000, and the total number of links between peers is 20,000. The topology of the P2P network in the simulation model is generated by the algorithm described in [17], where the maximum degree is 625. The topology follows a power law with respect to the distribution of degrees. In Fig. 2, the degree distribution of peers in this network is shown. The present paper does not consider dynamic changes in network topology.

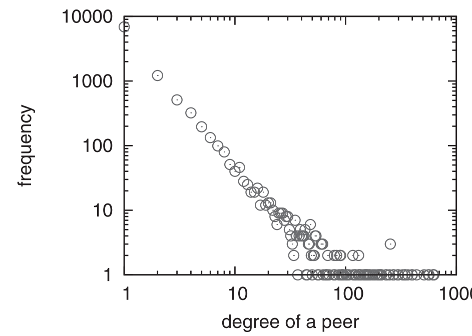


Fig. 2 Degree distribution.

The maximum storage capacity of each peer is 40. Each file consumes one unit of storage, so that the maximum number of files that a peer can hold is 40. When the storage of a peer is full and a request for a replica of a requested file arrives at the peer, the oldest file is replaced by the requested file (FIFO replacement discipline).

The query forwarding method used in the present paper is a 16-walker random walk [3]. All walkers, each with a query, are allowed up to 100 hops in each file search, and it is possible for a walker to revisit the same peer one or more times. In one run of the simulation model, the search for a requested file is repeated in a one-by-one manner 250,000 times, and a peer that makes a query is chosen randomly.

The total number of file types in the network is 10,000, with 10 files initially allocated over the network for each type of file. This initial distribution of 100,000 files is random, but is determined in the same way each time the simulation model is run.

A file type requested by each peer is determined with Zipf's law. This means that a popularity trend of files is represented by Zipf's law $y \propto R^{-1}$, where y is the number of requests for a file type that is ranked R-th. For example, the file request probabilities of the 1st, 2nd, and 3rd ranked types are 7.5%, 4.4%, and 3.1%, respectively.

Ranks of only 100 types of files among all of files vary every 1,000 units of time. For example, the top 100 ranks of file types are removed and then other file types of from 101-th to 10,000-th ranks slide in the direction of first rank. Finally, new 100 types of files are added as the 9,901-th to the 10,000-th ranks of file types. Note that queries for files ranked high have a majority under the condition that the distribution of popularity trend of queried files follows Zipf's law. Therefore, the replacement of files ranked high based on the above model can represent an abrupt change in popularity trend of queried files. Therefore, we can consider that the simulation model adopted in this paper is an approximation of abrupt change in popularity trend observed in the real world or more. The reason we adopt this simulation model is that we want to focus on adaptability of the replication method to the dynamic P2P file-sharing environment. That is, we expect the P2P file-sharing system that can adapt to abrupt change in access frequency can also adapt to gradual change in access frequency. The situation like this might be realized. In a real file-sharing P2P system, for example, there might be a smash hit and a decline of public movies and musics. There might be also a rapid increase and decrease of accesses disaster information. We consider a burst traffic problem is a major issue in a P2P file-sharing system as well as in the Internet.

The simulation run begins from the state that the average storage utilization ratio over all of peers is 0.5. The initial state is set up as follows. Search and replication with a constant replication probability of 0.5 is repeated until the average storage utilization ratio reaches 0.5. Here, a file replication method with constant replication probability is referred to as path random replication (PRR), which was used in our previous study [7].

4.2 Evaluation and Methods

The evaluation criterion for search performance is defined using the number of hops required to find a file. A smaller number of hops represents better search performance. The evaluation criterion for load balancing performance is defined using the standard deviation of $s(d)$, where $s(d)$ is the average storage utilization/access ratio of peers of degree d in the network. A smaller standard deviation represents better load balancing ability.

We test 9 sets of (μ, λ), which makes up the combinations of $\mu = \{-0.5, 0, 0.5\}$ and $\lambda = \{1, 10, 100\}$, with $N = 100$ in Eq. (1). The simulation runs 50 times for each combination. We observe the variation of the search performance index and the load balancing performance index over time. The method for comparison presented in the previous study [5] uses the storage utilization ratio as $L(x, t)$ instead of the write storage access ratio in the method presented in this paper. The simulation also runs with $(\mu, \lambda) = (0, 0)$ and $(\mu, \lambda) = (10, 0)$ for comparison. The former corresponds to the path random replication with replication probability of 0.5 and the latter corresponds to the approximation of the path replication.

We plot the search performance index and the load balancing performance index with respect to time t. The graph is plotted at $t = 0$, 10,000, 20,000, 50,000, 100,000, and 200,000. In order to evaluate the variation of performance indices in detail just after the change of popularity trends, we plot an additional 20 data every 50 units of times in each of the above plots.

Finally, we compare the proposed method with the previous method proposed in [5], which can work without global knowledges as well, through simulation examinations. In addition, we use the path replication and the path random replication, which can also work without global knowledges, for comparison. However, we do not compare with the file replication methods that can not work without global knowledges, e.g., the file replication method proposed in [6], because the assumption about the available information for controlling the network adopted in these methods is different from that in the proposed method.

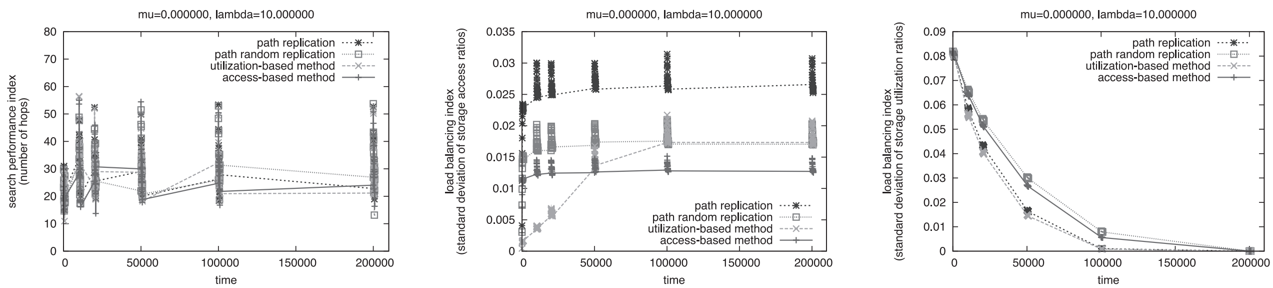
5. Results and Discussion

Figures 3(a), 3(b), and 3(c) show the variation of the search performance, the load balancing performance with respect to the write storage access ratio, and the load balancing performance with respect to the storage utilization ratio over time for $\mu = 0$ and $\lambda = 10$, respectively. The method presented in this paper is referred to as access-based method and the previously proposed method for comparison is referred to as utilization-based method hereinafter. These figures also include the result of path replication and path random replication with replication probability of 0.5 for comparison. In this section, We mainly discuss the result of $\mu = 0$ and $\lambda = 10$. However, in order to investigate the effect of μ and λ, we compare graphs of several combination of μ and λ. Figures 4 and 5 are the graphs for $(\mu, \lambda) = (0, 100)$ and $(\mu, \lambda) = (0.5, 10)$.

5.1 Load Balancing and Search Performance

Figure 3(b) shows that the utilization-based method and access-based method exhibit quite different load balancing performance with respect to the write storage access ratio. That is, the load balancing performance with respect to the write storage access ratio of the utilization-based method is better than that of the access-based method in the initial phase of the simulation, and becomes worse as the simulation progresses, while that of the access-based method remains constant. This would be due to the fact that the storage utilization ratio is the cumulative value of access with upper limit value, whereas the write storage access ratio is the temporal variation of the access frequency. In order to investigate the origin of these phenomena, we observe the variation of the storage utilization ratio and the write storage access ratio over time for peers of high and low degrees for $\mu = 0$ and $\lambda = 10$. In this section, a peer with over 100 degrees is referred to as a high-degree peer, and a peer with less than 100 degrees is referred to as a low-degree peer. The simulation is run 50 times. In order to observe typical values of the storage utilization ratio and the write storage access ratio, we first take average of these ratios among the high-degree peers and the low-degree peers, respectively. We then take average of these average values over 50 runs. The simulation results are shown in Fig. 6.

Figure 6(a) shows that the overall trends over time in write storage access ratios of high- and low-degree peers remain essentially unchanged. In other words, the graph includes spikes but is unchanged as a whole. In addition, the difference value of write storage access ratios between high- and low-degree peers is kept constant. This is because frequent accesses in high-degree peers caused by frequent arrivals of search queries are diffused over the network using a thermal diffusive mechanism. That is, the access dis-

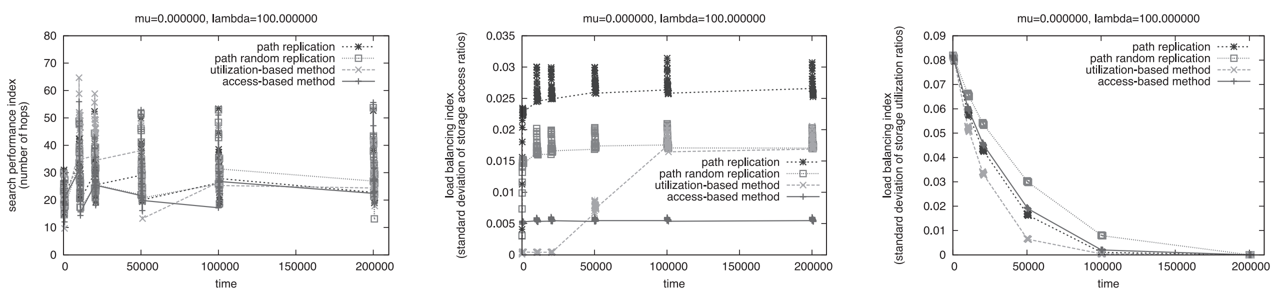


(a) Search performance.

(b) Load balancing performance with respect to write storage access ratio.

(c) Load balancing performance with respect to storage utilization ratio.

Fig. 3 Variation of performance indices over time for $\mu = 0$ and $\lambda = 10$.

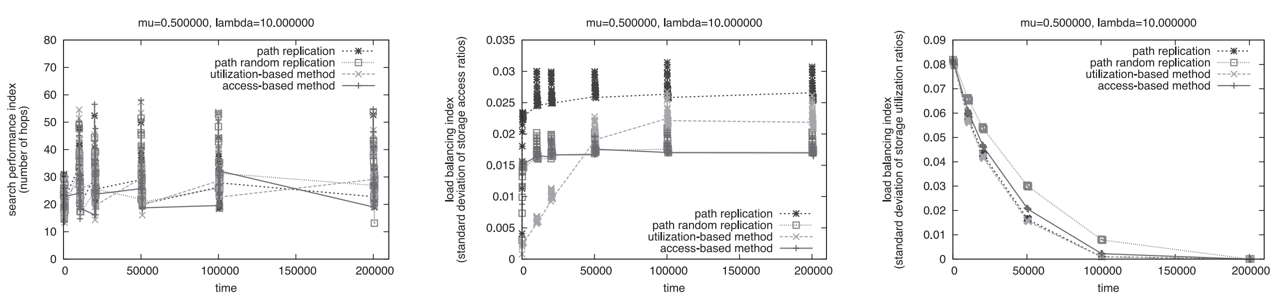


(a) Search performance.

(b) Load balancing performance with respect to write storage access ratio.

(c) Load balancing performance with respect to storage utilization ratio.

Fig. 4 Variation of performance indices over time for $\mu = 0$ and $\lambda = 100$.



(a) Search performance.

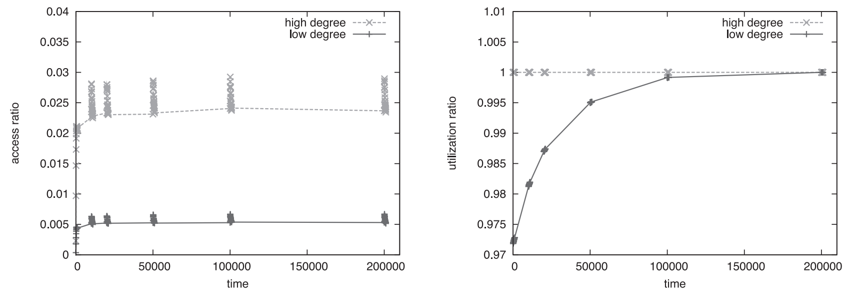
(b) Load balancing performance with respect to write storage access ratio.

(c) Load balancing performance with respect to storage utilization ratio.

Fig. 5 Variation of performance indices over time for $\mu = 0.5$ and $\lambda = 10$.

tribution among peers converges to an equilibrium due to a continuous thermal diffusive process and constant diffusive speed. This means that load balancing can be stably achieved. Taking this into account, we can explain the fact that load balancing performance of the write storage access ratio in the access-based method is constant in Fig. 3(b).

Figure 6(b) shows that the storage utilization ratio of high-degree peers has already reached the upper limit value from the beginning of the simulation (i.e., time 0), while that of low-degree peers gradually increases up to the upper limit value with time. The reason for this result is that the path random replication (PRR) for setting up the initial state of the simulation created replicas of files more frequently in the high-degree peers than the low-degree ones before starting the simulation. Therefore, in the initial phase of the simulation, the utilization-based method has a better load balancing performance because the difference of the utilization ratio between high- and low-degree peers is very large. However, the load balancing performance of the utilization-based method is degraded as the simulation progresses, and the load balancing of the utilization-based method eventually ceases to work, because storage utilization ratios of not only high-degree peers but also low-degree peers reach the upper limit value. In this case, the utilization-based method becomes equivalent to PRR method with constant replication probability, $1/2 + 1/2\tanh(\mu)$, because $DL(x, t)$ in Eq. (5) is



(a) Write storage access ratio. (b) Storage utilization ratio.

Fig. 6 Variation for high and low degree peers with time for $\mu = 0$ and $\lambda = 10$.

zero, and its load balancing performance becomes worse. Therefore, the load balancing performance with respect to the write storage access ratio for the utilization-based method is superior to that for the access-based method at the beginning, but then becomes worse than the access-based method, and finally becomes almost constant, as shown in Fig. 3(b).

On the other hand, in Fig. 3(a), there is no marked difference among the search performance of the access-based method, that of the utilization-based method, that of the path replication method and that of the path random replication method. Figure 3(c) shows that the access-based method is inferior to the utilization-based method and path replication method in terms of the load balancing performance with respect to the storage utilization ratio, but that difference looks small. However, Figs. 3(a) and 3(c) show that the search performance and the load balancing performance with respect to the storage utilization ratio become better and better with time and reach the best in all methods, which are the utilization-based method, the access-based method, the path replication method and the path random replication method, except for burst fluctuations caused by change of the popularity trends (This point is discussed in the next subsection). These observations mean that the file-sharing network can be controlled by all of the file replication methods in the case that load balancing performance is measured by the storage utilization ratio. In other words, all methods, which are the access-based method, the utilization-based method, the path replication method and the path random replication method, can control and stabilize the dynamic P2P file-sharing environment from the viewpoint of the search performance and the load balancing performance with respect to the storage utilization ratio.

Figure 3(b) shows that the load balancing performance with respect to the write storage access ratio becomes worse and worse with time and reaches the worst in the utilization-based method. On the other hand, those for the access-based method, the path replication method and the path random replication method remains essentially unchanged as a whole. This means that the file-sharing network can be controlled only by the access-based method, the path replication method and the path random replication method in the case that the storage load performance is measured by the write storage access ratio. However, the converged value of the load balancing performance of access-based method is superior to those of the path replication and the path random replication. Therefore, the access-based method has better load balancing performance than the path replication method and the path random replication method.

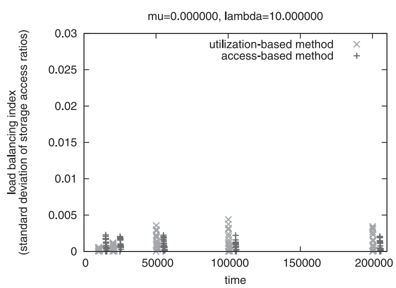
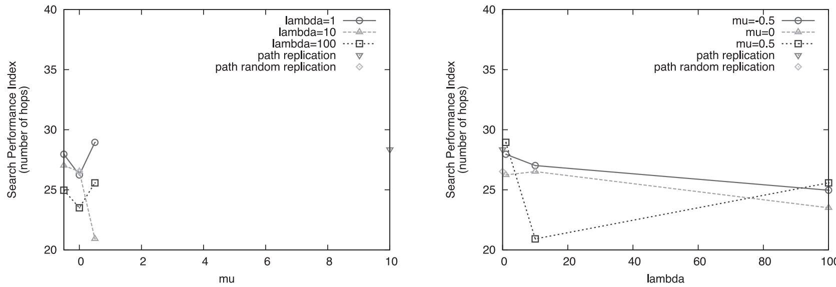


Fig. 7 Spikes of load balancing index with respect to write storage access ratio for $\mu = 0$ and $\lambda = 10$.

5.2 Adaptability to Sudden Changes

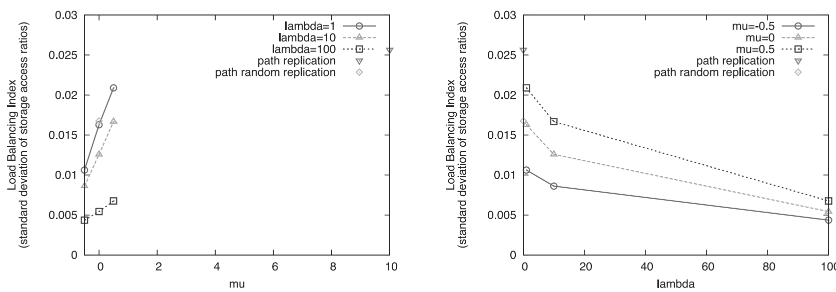
Spikes are observed in the graph of the load balancing index in Fig. 3. These sudden deteriorations are accompanied by changes of popularity trends of queried files. Immediate vanishing of the spikes is also observed in the graphs in Fig. 3(a) and Fig. 3(b) and this indicates that both the access-based and the utilization-based methods can control the file-sharing network adaptively in such a dynamic P2P environment. Next, we compare the heights of spikes of performance indices for access-based method with that for utilization-based method in Fig. 3(b). Figure 7 shows only spikes. Here, since spikes of access-based method and utilization-based method are overlapped, they are arranged alongside for easy comparison.

In Fig. 7, spikes in the graph of the write storage access ratio for the access-based method are shorter than those for the utilization-based method. Furthermore, the spikes of the utilization-based method become longer with the increasing time, however, those in the access-based method are constant. Therefore, the access-based method is superior to the utilization-based method with respect to adaptability to sudden change in access frequency to the peers caused by the



(a) μ.   (b) λ.

Fig. 8 Dependency of search performance on parameters.



(a) μ.   (b) λ.

Fig. 9 Dependency of load balancing performance with respect to write storage access ratio on parameters.

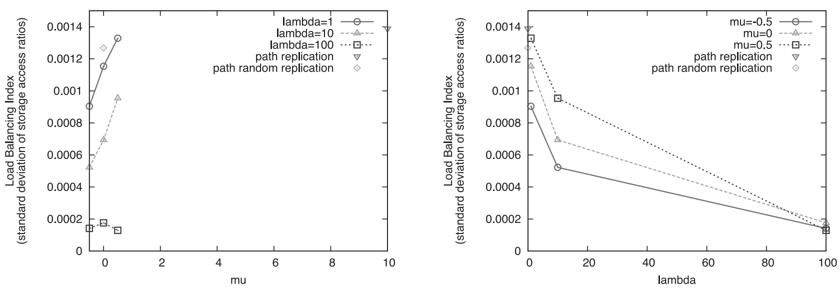
change of popularity trends.

Note that spikes are observed not in Fig. 6(b) but in Fig. 6(a). These results indicate that the write storage access ratio can be used for sensing changes of access frequency, but the storage utilization ratio cannot be used for sensing those because the storage utilization ratio has no reflections of changes of access frequency in Fig. 6(b). The reason for the difference in sensitivity between the storage access and the storage utilization ratios is that while the write storage access ratio is a temporal variation of the access frequency, the storage utilization ratio is a cumulative value of the access frequency. In the access-based method, the diffusive process of the write storage access ratio can effectively work to balance loads among the peers. However, the mechanism to adapt to changes of access frequency in the utilization-based method cannot be explained on the basis of the diffusive process of the storage utilization ratio. Although the utilization-based method cannot control the network with diffusive load balancing after the storage utilization ratio of all peers becomes almost the upper limit value, all the spikes vanish immediately in Fig. 3(b). These facts indicate that the utilization-based method does not control the network in adapting to changes of access frequency. In other words, the mechanism of the utilization-based method is probably not the major cause of adaptation to changes of access frequency. The most likely cause is distribution of access because replicas multiply widely over the network.

5.3 Meaning of Parameters μ and λ

Figure 8 show the relationship between search performance and parameters μ and λ. Figure 9 show the relationship between storage load balancing performance with respect to the write storage access ratio and parameters μ and λ. In each figure, y-axis represents average value of performance indices over the whole simulation time, where we exclude spikes for the calculation of the average in order to see outline of performance. In Figs. 8(a) and 9(a), the x-axis represents μ. In Figs. 8(b) and 9(b), the x-axis represents λ. Furthermore, Fig. 10 represents the relationship between the heights of spikes of the load balancing performance with respect to the write storage access ratio and parameters μ and λ. In this figure, y-axis represents the average value of the heights of spikes over the whole simulation time.

First, we discuss the meaning of μ. In Fig. 8(a) there is no significant relation between the search performance and μ. On the other hand, in Fig. 9(a) the load balancing performance is degraded as μ becomes larger. In Sect. 3.4, we show that μ is in the heat term of the state Eq. (12) and the write storage access ratio $L(x, t)$ becomes larger as μ becomes larger, namely many replicas are generated. These simulation results show that the storage load balancing performance degrades as the value of μ becomes larger because many replicas are placed over the network. However, it is not clear the relationship between the search performance



(a) μ. (b) λ.

Fig. 10 Dependency of spike of load balancing performance with respect to write storage access ratio on parameters.

and the value of μ. Therefore, in the future work, we need further investigation in order to show that μ is the parameter that can tune the performance trade-off between search performance and storage load balancing performance.

Next, we discuss the meaning of λ. In Fig. 8(b), there is no significant relation between the search performance and the value of λ. On the other hand, in Fig. 9(b), the storage load balancing performance with respect to the write storage access ratio is improved as the value of λ becomes larger. However, in Fig. 10(b), the height of spikes of the load balancing performance with respect to the write storage access ratio becomes shorter as the value of λ becomes larger. On the other hand, in Fig. 10(a), the height of spikes of the load balancing performance with respect to the write storage access ratio is independent of μ. In Sect. 3.4, we show that λ is in the diffusion term of the state Eq. (12), and the sensitivity to a variation of the write storage access ratio $L(x,t)$ becomes higher as λ becomes larger, namely the access-based method makes the write storage access ratio uniform quickly, however, search performance becomes worse. Thus, it is suggested that λ is the parameter that can tune the sensitivity to environment variation.

6. Conclusion

We proposed a file replication method using the write storage access ratio as the load of a peer to achieve write storage access load balancing in dynamic P2P file-sharing environments. The simulation results showed that the proposed method can achieve write storage access load balancing in the dynamic environments. The good point of the proposed method was that the performances of search and load balancing are stable at any time due to the use of the write storage access ratio. In addition, theoretical analysis led to the state equation, which has an analogy to a thermal diffusion equation, and discussed the meaning of parameters of the method using the simulation results.

In the future work, in addition to change in query distribution, we will consider other factors that cause dynamic P2P environments as change in topology. In addition, in the present scheme state equation, Eq. (12), has the heat term and the reaction term, which is not complete diffusion equation. We will adopt the scheme that has only diffusion term and examine the new scheme in the future work.

Acknowledgments

This work was supported in part by the Japan Society for the Promotion of Science through a Grant-in-Aid for Scientific Research (S)(18100001).

References

[1] E.K. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim, "A survey and comparison of peer-to-peer overlay network schemes," IEEE Communications Surveys & Tutorials, vol.7, no.2, pp.72–93, Second Quarter 2005.
[2] E. Cohen and S. Shenker, "Replication strategies in unstructured peer-to-peer networks," Proc. ACM SIGCOMM 2002 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, Pittsburgh, PA, USA, Aug. 2002.
[3] Q. Lv, P. Cao, E. Cohen, S. Li, and K. Shenker, "Search and replication in unstructured peer-to-peer networks," Proc. 16th International Conference on Supercomputing, pp.84–95, New York, USA, June 2002.
[4] S. Saroiu, P. Gummadi, and S. Gribble, "A measurement study of peer-to-peer file sharing systems," Tech. Rep. UW-CSE-01-06-02, Department of Computer Science and Engineering, University of Washington, 2001.
[5] M. Uchida, K. Ohnishi, K. Ichikawa, M. Tsuru, and Y. Oie, "Dynamic storage load balancing with analogy to thermal diffusion for P2P file sharing," Interperf'06: Proc. from the 2006 Workshop on Interdisciplinary Systems Approach in Performance Evaluation and Design of Computer & Communications Systems, p.6, 2006.
[6] T. Hara, Y. Kido, and S. Nishio, "Probabilistic replication based on access frequencies in unstructured peer-to-peer networks," Lecture Notes Comput. Sci., vol.4080, p.791, 2006.
[7] H. Yamamoto, D. Maruta, and Y. Oie, "Replication method for load balancing on distributed storages in P2P networks," IEICE Trans. Inf. & Syst., vol.E89-D, no.1, pp.171–180, Jan. 2006.
[8] G. Cybenko, "Dynamic load balancing for distributed memory multiprocessors," Parallel and Distributed Computing, vol.7, no.2, pp.279–301, Oct. 1989.
[9] J. Boillat, "Load balancing and poisson equation in a graph," Concurrency Practice and Experience, vol.2, no.4, pp.289–313, Dec. 1990.
[10] A. Corradi, L. Leonardi, and F. Zambonelli, "Diffusive load balancing policies for dynamic applications," IEEE Concurrency, vol.7, no.1, pp.22–31, 1999.
[11] Y. Li and Z. Lan, "A survey of load balancing in grid computing,"

First International Symposium, Computational and Information Science 2004, pp.280–285, Springer Berlin/Heidelberg, 2005.
[12] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: A scalable and robust communication paradigm for sensor networks," MobiCom'00: Proc. 6th Annual International Conference on Mobile Computing and Networking, pp.56–67, New York, NY, USA, 2000.
[13] C. Takano, K. Sugiyama, and M. Aida, "Parameter design for diffusion-type autonomous decentralized flow control," IEICE Trans. Commun., vol.E91-B, no.9, pp.2828–2837, Sept. 2008.
[14] N. Wakamiya, K. Hyodo, and M. Murata, "Reaction-diffusion based topology self-organization for periodic data gathering in wireless sensor networks," SASO'08: Proc. 2008 Second IEEE International Conference on Self-Adaptive and Self-Organizing Systems, pp.351–360, IEEE Computer Society, Washington, DC, USA, 2008.
[15] K. Leibnitz, T. Hossfeld, N. Wakamiya, and M. Murata, "Modeling of epidemic diffusion in peer-to-peer file-sharing networks," Biologically Inspired Approaches to Advanced Information Technology, vol.3853/2006, pp.322–329, 2006.
[16] Various Authors, "Moving average." http://en.wikipedia.org/wiki/Moving_average
[17] T. Bu and D. Towsley, "On distinguishing between internet power law topology generators," Proc. IEEE Infocom 2002, pp.638–647, New York, NY, USA, June 2003.

Masanori Takaoka received B.S. degree in physics from Osaka City University, Osaka, Japan in 1996 and M.S. degree in physics from Osaka University, Osaka, Japan in 1998. Since 1998, he has been with NEC Communication Systems. From 2007, he has been a Ph.D. candidate at the Graduate School of Computer Science and System Engineering, Kyushu Institute of Technology, Kyushu, Japan. His research interests include computer networks. He is a member of the IPSJ.

Masato Uchida received the B.E., M.E. and D.E. degrees from Hokkaido University, Hokkaido, Japan in 1999, 2001, and 2005, respectively. In 2001, he joined NTT Service Integration Laboratories, Tokyo, Japan. Since August 2005, he has been an Associate Professor in Network Design Research Center, Kyushu Institute of Technology, Kyushu, Japan. His research fields include computer communication networks and information-based learning theory. He is a member of the ACM.

Kei Ohnishi received the B.E., M.E. and D.E. degrees from Kyushu Institute of Design, Japan in 1998, 2000, and 2003, respectively. He worked as a postdoctoral researcher for University of Illinois at Urbana-Champaign, Kyushu Institute of Technology, and Human Media Creation Center/Kyushu. Since October 2007, he has been an Associate Professor at Kyushu Institute of Technology. His research interests include P2P networks and soft computing techniques. He is a member of IPSJ and SOFT.

Yuji Oie received B.E., M.E. and D.E. degrees from Kyoto University, Kyoto, Japan in 1978, 1980 and 1987, respectively. From 1980 to 1983, he worked at Nippon Denso Company Ltd., Kariya. From 1983 to 1990, he was with the Department of Electrical Engineering, Sasebo College of Technology, Sasebo. From 1990 to 1995, he was an Associate Professor in the Department of Computer Science and Electronics, Faculty of Computer Science and Systems Engineering, Kyushu Institute of Technology, Iikuza. From 1995 to 1997, he was a Professor in the Information Technology Center, Nara Institute of Science and Technology. Since April 1997, he has been a Professor in the Department of Computer Science and Electronics, Faculty of Computer Science and System Engineering, Kyushu Institute of Technology. His research interests include performance evaluation of computer communication networks, high speed networks, and queueing systems. He is a fellow of the IPSJ and a member of the IEEE.