Regular Paper

Evaluation of the Integration Effect of Content Location and Request Routing in Content Distribution Networks

Hirokazu Miura[†1] and Miki Yamamoto[†2]

Recently the content distribution networks (CDNs) have been highlighted as the new network paradigm which can improve latency for Web access. In CDNs, the content location strategy and request routing techniques are important technical issues. Both of them should be used in an integrated manner in general, but CDN performance applying both these technologies has not been evaluated in detail. In this paper, we investigate the effect of integration of these techniques. For request routing, we focus on a request routing technique applied active network technology, Active Anycast, which improves both network delay and server processing delay. For content distribution technology, we propose a new strategy, *Popularity-Probability*, whose aim corresponds with that of Active Anycast. Performance evaluation results show that integration of Active Anycast and *Popularity-Probability* can hold stable delay characteristics.

1. Introduction

On the Internet, several types of services use replicated servers which are geographically dispersed across the whole network. One typical example of this type of service is content distribution network (CDN)[1]. CDN distributes content by placing it on content servers which are located near the users and user requests to those servers. The aim of this approach is of prevent too many accesses from concentrating at a particular server, which causes degradation of response time of a server itself and congestion in the network around that server. However, if the content cannot be placed on an adequate server or the request cannot be forwarded to an adequate server, user response time would be degraded because of server overloading and network congestion. To avoid degradation of response time, CDN performance should be maintained. In content distribution networks, the request routing[2),3)] and content location techniques[4)–7)] are important technical problems. Both technologies should be used in an integrated manner in general, but CDN performance applying both of these technologies has not been evaluated in detail. In this paper, we investigate the effect of integration of these techniques.

When a client would like to select a good (replication) server to obtain an object, one transparent way is making use of DNS[8),9)]. In this approach, a DNS server has a list of servers and returns a selected server's IP address. Round robin selection is generally used, which cannot take account of the server's location and load. An Anycast server selection is a more sophisticated way of guiding a client's request to one of many hosts[10)–12)]. A packet destined for an Anycast address will be delivered to one of the hosts with an Anycast address, ideally the closest one to the client. This Anycast technology only takes the distance between client and server into consideration. To select the optimal server which gives the smallest response time, server load is also an important factor to be considered. As one possible way to resolve these server selection problems, we have proposed "Active Anycast"[13)]. In Active Anycast, when a user request arrives at an active router, this active router selects an adequate server and directs this request to the selected server.

In content location strategy, the optimization problem is defined as replicating objects so that the average number of hops traversed is minimized when clients fetch objects from the nearest content server containing the requested object. This optimization problem is NP-complete[4)]. Kangasharju, et al.[4)] propose three heuristics (Popularity, Greedy-Single, Greedy-Global) for this optimization problem. These algorithms are designed for the object to be replicated so that the average number of hops traversed is minimized in the basic assumption that Anycast is used for request routing.

In this paper, we claim that there is a sig-

[†1] Fuculty of Systems Engineering, Wakayama University

[†2] Faculty of Engineering Science, Kansai University

nificant difference between the aims of content location strategy and request routing. And we claim that these aims should correspond. When Active Anycast is used for request routing, a user request has a tendency to be guided so that servers inside a network are effectively used. Thus, when request routing guides a user request intelligently so that the load of servers is balanced, content location strategy should work well together with this strategy. From these observations, we propose a new content location strategy, *Popularity-Probability*. In *Popularity-Probability*, objects are randomly located in replicated servers inside a network according to its popularity. It has quite a simple operation, i.e., a specific object is located in a content server with the probability which is given from its relative popularity. By this simple operation, objects are randomly located inside a network and an object with high popularity has a larger number of copies inside a network than lower-popularity objects. With Active Anycast strategy which can select the adequate server from these servers, this content location strategy will provide good performance to CDNs. Performance evaluation in the paper will show that our proposed integration of request routing and content location strategy in CDN can hold stable delay characteristics.

The remainder of the paper is structured as follows. Section 2 describes request routing technology and introduces Anycast and Active Anycast. Section 3 explains previously published content location strategies in detail. Section 4 claims the necessity for robust CDN and proposes a new integration of request routing and content location strategy, i.e. Active Anycast and *Popularity-Probability*. Section 5 investigates the effectiveness of our proposed integration. Section 6 concludes the paper.

2. Request Routing

In the content distribution networks, to effectively respond to requests in a reasonable amount of time, the load must be distributed across multiple servers. Request routing is the technique which directs user requests to an adequate server from the standpoint of improving latency in obtaining objects. URL approach is the simplest one and some modifications of them have been proposed[3]. This approach assumes the request routing decision is made at the client side, so it can be categorized into an end-to-end approach. In the paper, we focus on the network support approach which makes use of active network technology. In the network support approach, higher operation than network layer can be processed at a router inside a network. As a network support approach, we explain Anycast and Active Anycast, in detail.

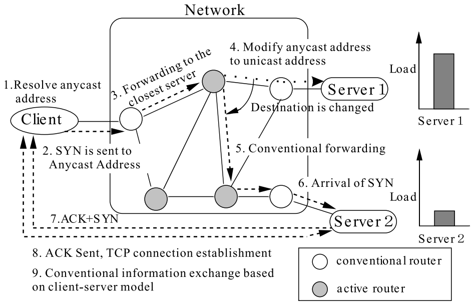

Network
1.Resolve anycast address
Client
2. SYN is sent to Anycast Address
3. Forwarding to the closest server
4. Modify anycast address to unicast address
Destination is changed
Server 1
5. Conventional forwarding
6. Arrival of SYN
Server 2
7.ACK+SYN
8. ACK Sent, TCP connection establishment
9. Conventional information exchange based on client-server model
Load
Server 1
Load
Server 2
○ conventional router
● active router

Fig. 1 Active Anycast.

2.1 Anycast

In Anycast technology, an Anycast address can indicate a group of servers offering the same service[10)–12)]. A router which receives an IP datagram whose destination address field includes an Anycast address forwards this datagram to an output link on the path to the nearest server. The Anycast technology can be used for selecting the closest server without an end-user knowing where it is.

2.2 Active Anycast

As a request routing, we have proposed Active Anycast[13)]. In Active Anycast, a router in the network autonomously distributes accesses from clients adequately to geographically dispersed servers. The Active Anycast is based on Anycast[10)–12)] and active network technology[15),16)].

In Active Anycast, a TCP connection which is initiated by the client is autonomously set up to an adequate server by an active router. When the client has a request to the server, it sends a name resolution query to the Domain Name System (DNS) and gets a resolved Anycast address (Step 1. in **Fig. 1**). This Anycast address indicates a group of replicated servers (including an original server) which offer the same service. The initiating host sends a SYN packet whose destination address field indicates Anycast address (Step 2). The SYN packet is forwarded to an output link on the path to the closest server when it arrives at a conventional Anycast router (Step 3). When the SYN packet with the Anycast address arrives at an active router, it chooses an adequate server from all the candidate servers of the corresponding ser-

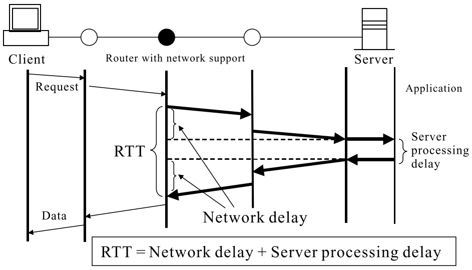


Fig. 2 RTT measurement.

vice based on the information and the policy of server selection. And this router changes the destination address of this SYN packet to the unicast address of the selected server (Step 4). Subsequently, the SYN packet is forwarded to the selected server as conventional unicast forwarding (Step 5). When the server receives this SYN packet, it replies an ACK+SYN packet (Step 7). And the client sends an ACK packet after it receives an ACK+SYN packet, which means establishment of the TCP connection (Step 8). After that, the ordinary information exchange phase is started between the server and the initiating client (Step 9). The anycast address cannot be directly used to establish a TCP connection, because anycast communication cannot guarantee that multiple packets to the same anycast address will reach the same destination. To solve this problem, IP option (record route and source route option) can be used [10),13)].

2.3 Active Anycast Server Selection

In Ref. 14), the way that an active router collects information necessary for server selection has been proposed. An active router is assumed to measure the round trip time (RTT) of a request packet and its response packet as shown in **Fig. 2** and use this RTT for server selection. This measured RTT includes both the network delay and the server processing delay, so an active router can select a good server from the standpoint of both the network delay and server load. For the server selection policy, a probabilistic server selection policy in which a router selects the server according to a probabilistic manner is applied. The probability of server selection is calculated taking account of the RTT between client and servers. When the RTT is large, selection probability should be small. This probabilistic selection prevents synchronized behavior of server selection. We apply the following simple method for calculation of the server selection probability. An active router i calculates P_{ij}, a probability of selecting server j, as follows.

$$P_{ij} = \frac{\frac{1}{RTT_j}}{\sum_{m=1}^{n} \frac{1}{RTT_m}}, \tag{1}$$

where n is the total number of servers serving the same service and RTT_m is the RTT between the router i and the server m.

3. Content Location Strategies

For content location strategies, several works have been published. Cidon, et al. [5)] and Li, et al. [6)] discuss the content location problem for a simple network model, a tree model. These results cannot be applied for the general case where many replication servers are located in the whole network and their decision affects each other, i.e., their decision in relation to which objects to be located affect total performance. Qui, et al. [7)] evaluate several content location strategies by simulation. In their evaluation, the replication server is assumed to be complete and they do not consider the behavior of each content. In Ref. 4), the content location problem is well formulated and they analyze which object is to be located in each replication server.

In Ref. 4), the content location problem is formulated as follows. Content server i in autonomous system i(i=1,2,. . . ,I), ASi, has S_i bytes of storage capacity. Object j has a size of $b_j, j \in \{1, 2, \ldots, J\}$ and a request probability p_j which is the probability that a client will request this object j. ASi has clients that request objects at aggregate rate λ_i.

$$x_{ij} = \begin{cases} 1 & \text{if content } j \text{ is stored} \\ & \quad\quad\quad \text{at content server } i \\ 0 & \text{otherwise} \end{cases}$$

The matrix of all x_{ij}'s is denoted by $\mathbf{x}$. Each object j is initially placed on an origin server. All of the objects are always available in their origin servers, regardless of the placement $\mathbf{x}$. The placement of objects to origin servers is denoted as $\mathbf{x}_o$.

The storage is constrained by the space available at ASi, that is

$$\sum_{j=1}^{J} b_j x_{ij} \le S_i \qquad i = 1, \cdots, I. \tag{2}$$

The average number of hops that a request must traverse from ASi is

$$C_i(\mathbf{x}) = \sum_{j=1}^{J} p_j d_{ij}(\mathbf{x}) \qquad (3)$$

where $d_{ij}(\mathbf{x})$ is the shortest distance to a copy of object j from AS i under the placement $\mathbf{x}$.

Let $\Lambda(=\sum_{i=1}^{I} \lambda_i)$ be the total request rate of all ASs. The average number of hops from all ASs is then

$$C(\mathbf{x}) = \frac{1}{\Lambda}\sum_{i=1}^{I} \lambda_i C_i(\mathbf{x}) = \frac{1}{\Lambda}\sum_{i=1}^{I}\sum_{j=1}^{J} \lambda_i p_j d_{ij}(\mathbf{x}). \qquad (4)$$

The goal is to choose the $\mathbf{x}$ so that the cost function $C(\mathbf{x})$ is minimized. This means that the goal is to minimize the average number of inter-AS hops that a request must traverse. It is not feasible to solve this problem optimally for a large number of objects and ASs. This problem is NP-complete [4]. They proposed several heuristics to solve this problem as follows [4].

3.1 Popularity

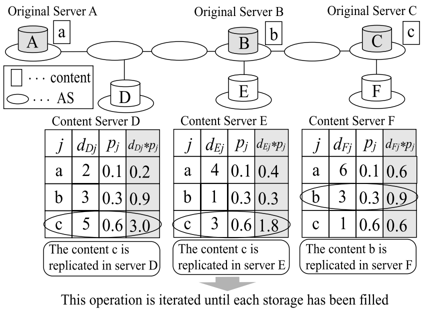
The content server in each AS stores the most popular objects. The content server sorts the objects in decreasing order of popularity and stores as many copies in this order as the storage constraint allows. The content server can estimate the popularities by observing the requests it receives from the clients. This heuristic does not require the node to get any information from outside of the AS.

3.2 Greedy-Single

Each ASi calculates

$$C_{ij} = p_j d_{ij}(\mathbf{x}_0) \quad (i \in 1, 2, \cdots, I, j \in 1, 2, \cdots, J) \qquad (5)$$

The AS then sorts the objects in decreasing order of C_{ij} and stores as many copies in this order as the storage constraint allows. The popularities are obtained as in the Popularity heuristics, but the CDN also needs information about the network topology in order to estimate the d_{ij}'s. Note that the C_{ij}'s are calculated only once under the placement $\mathbf{x}_0$ and not adjusted when copies are stored in the content server. This means that every AS stores copies independently of all the other ASs and no cooperation between ASs is required (**Fig. 3**).


Original Server A
Original Server B
Original Server C
A
a
B
b
C
c
D
E
F
□ ··· content
⬭ ··· AS
Content Server D
Content Server E
Content Server F
The content c is replicated in server D
The content c is replicated in server E
The content b is replicated in server F
This operation is iterated until each storage has been filled

j	d_{Dj}	p_j	$d_{Dj}*p_j$
a	2	0.1	0.2
b	3	0.3	0.9
c	5	0.6	3.0

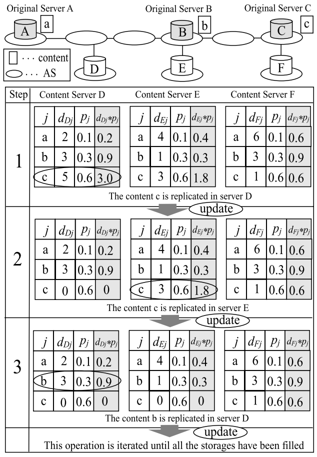
j	d_{Ej}	p_j	$d_{Ej}*p_j$
a	4	0.1	0.4
b	1	0.3	0.3
c	3	0.6	1.8

j	d_{Fj}	p_j	$d_{Fj}*p_j$
a	6	0.1	0.6
b	3	0.3	0.9
c	1	0.6	0.6

Fig. 3 Greedy-Single.

3.3 Greedy-Global

The CDN first calculates $C_{ij} = \lambda_i p_j d_{ij}(\mathbf{x_o})$ for all AS i and objects j. Then the CDN picks the AS-object-pair which has the highest C_{ij} and stores that copy in that content server. This results in a new placement $\mathbf{x_1}$. Then the CDN recalculates the costs C_{ij} under the new placement and pick the AS-object-pair that has the highest cost. The copy of that object is stored in the content server in that AS and a new placement $\mathbf{x_2}$ is obtained. This operation is repeated until all the storages have been filled (**Fig. 4**).


Original Server A
Original Server B
Original Server C
A
a
B
b
C
c
D
E
F
□ ··· content
⬭ ··· AS
Step
Content Server D
Content Server E
Content Server F
1
The content c is replicated in server D
update
2
The content c is replicated in server E
update
3
The content b is replicated in server D
update
This operation is iterated until all the storages have been filled

Step 1

j	d_{Dj}	p_j	$d_{Dj}*p_j$	j	d_{Ej}	p_j	$d_{Ej}*p_j$	j	d_{Fj}	p_j	$d_{Fj}*p_j$
a	2	0.1	0.2	a	4	0.1	0.4	a	6	0.1	0.6
b	3	0.3	0.9	b	1	0.3	0.3	b	3	0.3	0.9
c	5	0.6	3.0	c	3	0.6	1.8	c	1	0.6	0.6

Step 2

j	d_{Dj}	p_j	$d_{Dj}*p_j$	j	d_{Ej}	p_j	$d_{Ej}*p_j$	j	d_{Fj}	p_j	$d_{Fj}*p_j$
a	2	0.1	0.2	a	4	0.1	0.4	a	6	0.1	0.6
b	3	0.3	0.9	b	1	0.3	0.3	b	3	0.3	0.9
c	0	0.6	0	c	3	0.6	1.8	c	1	0.6	0.6

Step 3

j	d_{Dj}	p_j	$d_{Dj}*p_j$	j	d_{Ej}	p_j	$d_{Ej}*p_j$	j	d_{Fj}	p_j	$d_{Fj}*p_j$
a	2	0.1	0.2	a	4	0.1	0.4	a	6	0.1	0.6
b	3	0.3	0.9	b	1	0.3	0.3	b	3	0.3	0.9
c	0	0.6	0	c	0	0.6	0	c	1	0.6	0.6

Fig. 4 Greedy-Global.

$$C_{ij} = \lambda_i p_j d_{ij}(\mathbf{x}) \quad (i \in 1, 2, \cdots, I, j \in 1, 2, \cdots, J) \qquad (6)$$

4. Popularity-Probability

Popularity, Greedy-Single and Greedy-Global have the goal that objects are distributed to the content servers so that the total delay from each AS is minimized. This content location strategy is designed for a request routing which directs a user's request to the closest server. In this way, they can be used for Anycast routing. When a more sophisticated request routing technique, such as Active Anycast, is used, the content location strategy for Anycast, e.g. Greedy-Global, may not work well. This is because of the difference between the aims of the content location strategy and request routing. Aim of request routing of Active Anycast is to find a good server which gives optimal response time. This means Active Anycast can direct a user's access to a good server with a light load even though this server is not the closest one. Thus, for content location strategy, it is not the most important requirement that requested objects are located close to users (of course, this does not mean it is not important). It is however important that the network has an adequate amount of (the same) objects as a popular one. From these observations, we propose a new content location strategy which is applicable to Active Anycast-type request routing, i.e., a request routing taking care both of network delay and server load, *Popularity-Probability*.

In *Popularity-Probability* content location strategy, each content server decides its storage of objects according to object popularity. Since object popularity is a key factor of content location, Popularity and *Popularity-Probability* have a similar concept. However, in *Popularity-Probability*, the content server decides whether it stores a specific object or not with a probability which is predefined by its popularity. When the total number of content servers in a network is N and the request probability of object i is p_i, the expected number of content servers which store content i is Np_i. This means contents are distributed randomly in a network so that the number of replicated contents in a network is linear to its popularity. In *Popularity-Probability*, each content server can decide its storage of objects independently and there is no necessity to exchange any information among servers. So, *Popularity-Probability* is very easy to implement. On the Internet, popularities of contents follow Zipf-like distribution[17]. This distribution indicates that the number of contents with high popularity is small and many contents have low popularity. Therefore we focus on reducing the load on the server which has popular contents, because the network traffic by requests to low popular server or content may be trivial. Then, our method which distributes the popular contents among many content servers may be effective for server load balancing. Furthermore, in popularity-probability, it can be expected that contents with low popularity are also stored in content servers because of the probabilistic manner.

5. Performance Evaluation

In this section, we evaluate the performance of the combination of the content location strategy and request routing technologies by computer simulation, and investigate the environment where each technology works effectively. In that evaluation, Popularity, Greedy-Single, Greedy-Global and *Popularity-Probability* are applied as a content location strategy. For request routing technique, Anycast and Active Anycast is applied.

5.1 Experimental Setting

To investigate the environment where each content location and request routing technology works effectively, we investigate the average delay of obtaining objects with various percentages of ASs where the content servers are located. We make the following assumptions.

- Waxman random graph is used for our AS network model with 100 ASs. The average AS degree is 4 as the parameter of this Waxman model. Each AS has a router and at most one content server.
- The link capacity between any nodes is 25.0 requests/sec.
- The server is modeled as M/M/1 queueing model with a capacity of 1.0 requests/sec.
- Requests from clients to all 100 contents are generated by the Poisson process. The request arrival rate indicates the aggregate request arrival rate to each AS from users connected directly to it and is 30.0 requests/sec.
- The number of contents stored in the content server is 10% of all 100 contents, i.e. the capacity of each server is 10 copies. The popularities of contents are assigned from Zipf-like distribution[17] whose parameter is 1.0.

5.2 Performance of Request Routing and Content Location

Figure 5 shows the average latency of obtaining objects vs. the percentage of the number of ASs which have a content server in the network. A solid line and a dotted line show the latency of obtaining objects with Anycast and Active Anycast, respectively. As shown in this figure, with any combination of request routing technology and content location strategy there is some area where delay characteristics diverge. This is because the utilization of servers inside a network becomes larger than 1, i.e., servers are in overload status when a sufficient number of

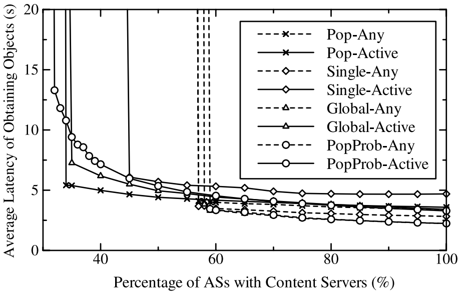


Fig. 5 Average latency characteristics.

content servers are not prepared in a network. However, the percentile of AS's, i.e. the number of content servers, which gives delay divergence is varied for each combination. *Popularity-Probability* and Active Anycast combination give the smallest value of this divergence point. This means this combination needs the smallest number of content servers in order to stabilize delay characteristics. Thus, the combination of *Popularity-Probability* and *Active Anycast* can distribute an adequate number of contents inside a network and guide users' requests with a satisfying server load balance.

We also evaluated the performance in the case where the server capacity is 30%. Simulation results for this situation show that there is no significant difference between the results for a 10% case. As the server capacity becomes larger, the total performance of CDN is improved, of course. However, there is a tendency for the combination of *Popularity-Probability* and Active Anycast to need the smallest number of content servers in order to stabilize delay characteristics.

5.3 Simulation Results: Robustness

Another important performance factor for CDN is robustness. For content location strategy, e.g., Popularity and *Popularity-Probability*, measured or predefined information about the popularity of an object is necessary. When there is some error on its estimation or temporal change of popularity, there may be some performance degradation in CDN. We evaluate robustness from the standpoint of how average latency characteristics are degraded with these errors. In this paper, we investigate the effect on the integration of request routing and content location in the case where the request probability of the objects p_j changes from the original design. It is modeled as the situation where the request probability of the most popular object is replaced with the request probability of m'th ($m \geq 2$) popular object. When the contents are sorted in decreasing order of popularity, it is assumed that the request probability of the most popular object is $\hat{p}_1$ and m'th popular object is $\hat{p}_m$ originally. Each request probability p_1, p_m becomes as follows after the change.

$$p_1 = \hat{p}_m, \quad p_m = \hat{p}_1$$

The x-axis shows the object number m to be replaced and y-axis shows the latency of obtaining objects in **Figs. 6**, **7** and **8**. Figures 6, 7 and 8 show the performance in the case where the popularity of objects changes in 10%, 50% and 100% of all ASs in the network, respectively. As we mentioned in the previous section,

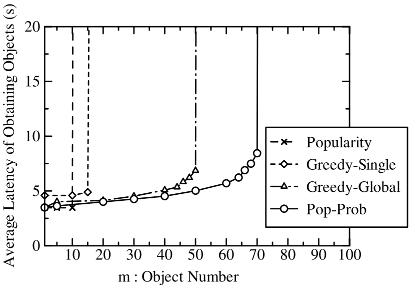


Fig. 6 Performance in the case where the popularity changes in 10 ASs.

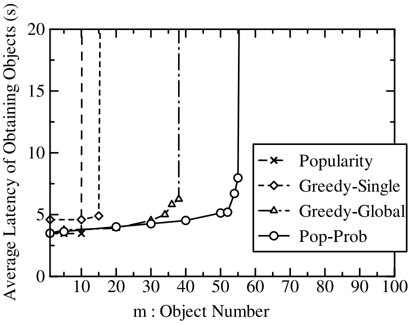


Fig. 7 Performance in the case where the popularity changes in 50 ASs.

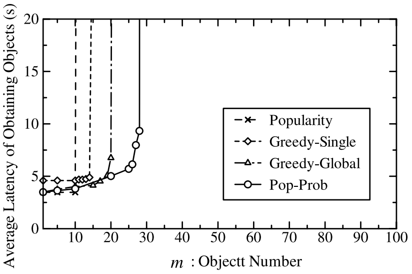


Fig. 8 Performance in the case where the popularity changes in 100 ASs.

each content location strategy with Active Anycast can show a better performance than the combinations of Anycast routing show. This is because Anycast cannot select the adequate server from the candidates based on server load or network congestion. So, Active Anycast is used as request routing in all the above figures.

As shown in these figures, the combination of *Popularity-Probability* and Active Anycast gives the best performance from the standpoint of robustness because it can hold stable delay characteristics even when a large error happens in popularity pre-estimation.

6. Conclusion

In this paper, we have claimed that there is a significant difference between the aims of content location strategy and request routing. The aim of request routing of *Active Anycast* is to find a good server which gives optimal response time. Thus, for content location strategy, it is not the most important requirement that requested objects are located close to users. We believe that request routing and content location strategy work well together when content location strategy is designed to manage the number of objects according to their popularity. From these observations, we have proposed a new content location strategy, *Popularity-Probability*. In *Popularity-Probability*, objects are randomly located in content servers inside a network according to its popularity. In *Popularity-Probability*, each content server can decide its storage of copies independently and there is no necessity to exchange any information among servers.

We have evaluated our proposed integration of request routing and content location strategy, i.e., the combination of *Active Anycast* and *Popularity-Probability*. We compare the average latency of obtaining objects in our proposed integration with the various combinations of previously proposed request routing techniques: Anycast and *Active Anycast*, and content location strategies, Popularity, Greedy-Single and Greedy-Global. Our simulation results show that our proposed integration gives stable CDN in that CDN is tolerable to a change of user request tendency. Our proposed integration of request routing and content location strategy in CDN may open a new possible network design.

In our evaluation in this paper, we assume link capacities of all links are homogeneous. For content servers, we think their network situation should be good, i.e., their available bandwidth should be large. This is because these content servers will be prepared by network carriers or service providers. Thus, we believe that the insights obtained in the paper can be applied to general CDNs. Where the link capacity of each content server is different, content location strategy should take into account not only the popularity of contents but also the server's network situation. For example, popular contents should be located at a server with a good network situation. We would like to leave this issue for our further research.

References

1) Day, M., Cain, B., Tomlinson, G. and Rzewski, P.: A model for content Internetworking (CDI), draft-day-cdnp-model-09.txt (work in progress) (Nov. 2001).

2) Barbir, A., Cain, B., Douglis, F., Green, M., Hofmann, M., Nair, R., Potter, D. and Spatscheck, O.: Known CN request-routing mechanisms, draft-cain-cdnp-known-request-routing-04.txt (work in progress) (Nov. 2001).

3) Wang, L., Pai, V. and Peterson, L.: The Effectiveness of Request Redirectionon CDN Robustness, *Proc. 5th Symposium on Operating Systems Design and Implementation* (*OSDI'02*), Boston, MA (Dec. 2002).

4) Kangasharju, J., Roberts, J. and Ross, K.W.: Object Replication Strategies in Content Distribution Networks, *6th International Workshop on Web Caching and Content Distribution*, Boston, Massachusetts, USA (June 2001).

5) Cidon, I., Kutten, S. and Soffer, R.: Optimal Allocation of Electronic Content, *Proc. IEEE INFOCOM*, Anchorage, USA (Apr. 2001).

6) Li, B., Golin, M., Giuseppe, G., Italiano, F., Deng, X. and Sohraby, K.: On the Optical Placement of Web Proxies in the Internet, *Proc. IEEE INFOCOM*, New York, USA (Mar. 1999).

7) Qui, L., Padmanabhan, V. and Voelker, G.: On the Placement of Web Server Replicas, *Proc. IEEE INFOCOM*, Anchorage, USA (Apr. 2001).

8) Brisco, T.P.: DNS support for load balancing, Internet Request for Comments (RFC 1794) (Apr. 1995).

9) Katz, E.D., Butler, M. and McGrath, R.: A scalable HTTP server: The NCSA prototype, *Computer Networks and ISDN Syst.*, Vol.27, pp.155–164 (1994).

10) Engel, R., Peris, V. and Saha, D.: Using IP anycast for load distribution and server lo-

cation, *Proc. IEEE Globecom Global Internet Mini Conference*, pp.27–35 (Nov. 1998).

11) Partidge, C., Mendez, T. and Milliken, W.: Host anycasting service, *Internet Request for Comments* (RFC 1546) (1993).

12) Weber, S. and Cheng, L.: A Survey of Anycast in IPv6 Networks, *IEEE Communications Magazine*, Vol.42, No.1, pp.127–132 (Jan. 2004).

13) Miura, H., Yamamoto, M., Nishimura, K. and Ikeda, H.: Server load balancing with network support: Active anycast, *The 2nd International Working Conference on Active Networks* (*IWAN2000*), pp.371–384 (2000).

14) Miura, H. and Yamamoto, M.: Content Routing with Network Support Using Passive Measurement in Content Distribution Networks, *IEICE Trans. Comm., Special Issue on Content Delivery Networks*, Vol.E86-B, No.6, pp.1805–1811 (June 2003).

15) Tennenhouse, D., Smith, J., Sincoskie, W., Wetherall, D. and Minden, G.: A survey of active network research, *IEEE Communications Magazine*, Vol.35, No.1, pp.80–86 (Jan. 1997).

16) Tennenhouse, D. and Wetherall, D.: Towards an active network architecture, *ACM Computer Communication Review*, Vol.26, No.2, pp.5–18 (Apr. 1996).

17) Breslau, L., Cao, P., Fan, L., Phillips, G. and Shenker, S.: Web caching and Zipf-like distributions: Evidence and implication, *Proc. IEEE Infocom*, New York, March 21–25 (1999).

(Received March 8, 2007)
(Accepted October 2, 2007)
(Released January 9, 2008)

Hirokazu Miura received his B.E. and M.E. in communications engineering from Osaka University in 1999 and 2001 respectively. He joined the Department of Computer and Communication Science at Wakayama University in 2003, where he is currently a research associate. His research interests include content distribution networks and wireless networks. He is a member of IEEE and IEICE.

Miki Yamamoto received his B.E., M.E., and Ph.D. in communications engineering from Osaka University in 1983, 1985, and 1988. He joined the Department of Communications Engineering at Osaka University in 1988. He moved to the Department of Electrical Engineering and Computer Science of Kansai University in 2005, where he is a professor. He visited the University of Massachusetts at Amherst in 1995 and 1996 as a visiting professor. His research interests include multicast communications, high-speed networks, wireless networks, and the evaluation of performance of these systems. He is a member of IEEE, ACM, and IEICE.