

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

Hirokazu Miura<sup>1</sup> and Miki Yamamoto<sup>2</sup>

<sup>1</sup> Faculty of Systems Engineering, Wakayama University  
930 Sakaedani, Wakayama, 640-8510 Japan  
miurah@sys.wakayama-u.ac.jp

<sup>2</sup> Graduate School of Engineering, Osaka University  
2-1 Yamadaoka, Suita-shi, Osaka, 565-0871 Japan  
yamamoto@comm.eng.osaka-u.ac.jp

**Abstract.** Recently the content distribution networks (CDNs) are 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 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* provides robust CDNs.

**Keywords:** Content distribution networks, content location strategy, request routing.

## 1 Introduction

In 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]. The aim of this approach is to 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. 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 effect of integration of these techniques.

When a client would like to select a good (replication) server to obtain a 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 server's location and load. An Anycast server selection is more sophisticated way of guiding a client's request to one of many hosts[10][11][12]. A packet destined for an Anycast address will be delivered to one of the hosts with Anycast address, ideally the closest one from 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 of 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 base of the assumption that Anycast is used for request routing.

In this paper, we claim that there is significant difference between aims of content location strategy and request routing. And we claim that these aims should be 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 load of servers to be balanced, content location strategy should work well together with this strategy. From these observation, 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 a quite simple operation, i.e. a specific object is located in a content server with the probability which are given from its relative popularity. By this simple operation, objects are randomly located inside a network and an object with high popularity has larger number of copies inside a network than lower-popularity objects. With *Active Anycast* strategy, 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 will open a new possible network design, the *robust CDN*.

Remainder of the paper is structured as follows. Section 2 describes about request routing technology and introduces Anycast and *Active Anycast*. Section 3 explains about previously published content location strategies in detail. Section 4 claims necessity of robust CDN and proposes a new integration of request routing and content location strategy, i.e. *Active Anycast* and *Popularity-Probability*. Section 5 shows simulation results which investigate 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 viewpoint of improving latency in obtaining objects. URL approach is the simplest one and some modifications of them have been proposed[3]. This approach assumes request routing decision is made at client side, so it can be categorized into end-to-end approach. In the paper, we focus on network support approach which makes use of active network technology. As a network support approach, we explain about Anycast and Active Anycast, in detail.

### 2.1 Anycast

In Anycast technology, an Anycast address can indicate a group of servers offering the same service[10][11][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 selection of the closest server without an end-user's 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][11][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 service 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).

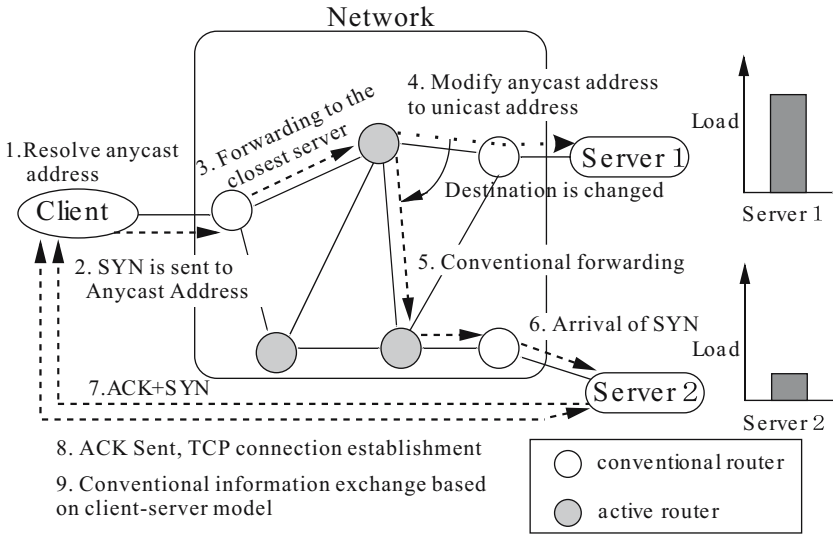


Fig. 1. Active Anycast

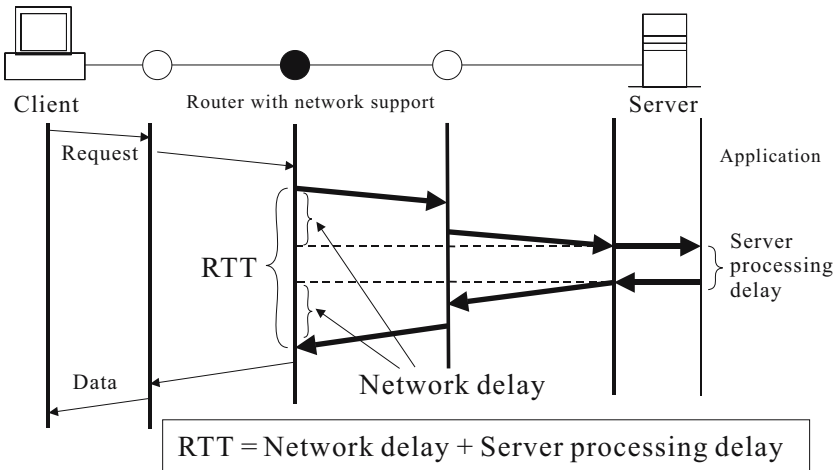


Fig. 2. RTT measurement

### 2.3 Active Anycast Server Selection

In [14], the way that an active router collects information necessary for server selection has been proposed. An active router is assumed to measure 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 of the network delay and the server processing delay, so an active router can select a good server from the viewpoint of both of 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 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 a 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.

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

where  $n$  is 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 content location problem for simple network model, a tree model. These results cannot be applied for general case where many replication servers are located in the whole network and their decision affects each other, i.e. their decision of which objects to be located affects total performance. Qui et al.[7] evaluates several content location strategies by simulation. In their evaluation, replication server is assumed to be complete and they do not consider behavior of each content. In [4], content location problem is well formulated and they analyze which object to be located in each replication server.

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

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

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 AS $i$ , that is

$$\sum_{j=1}^J b_j x_{ij} \leq S_i \quad i = 1, \dots, I. \quad (2)$$

The average number of hops that a request must traverse from AS $i$  is

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

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

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

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

Content server in the 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 AS $i$  calculates

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

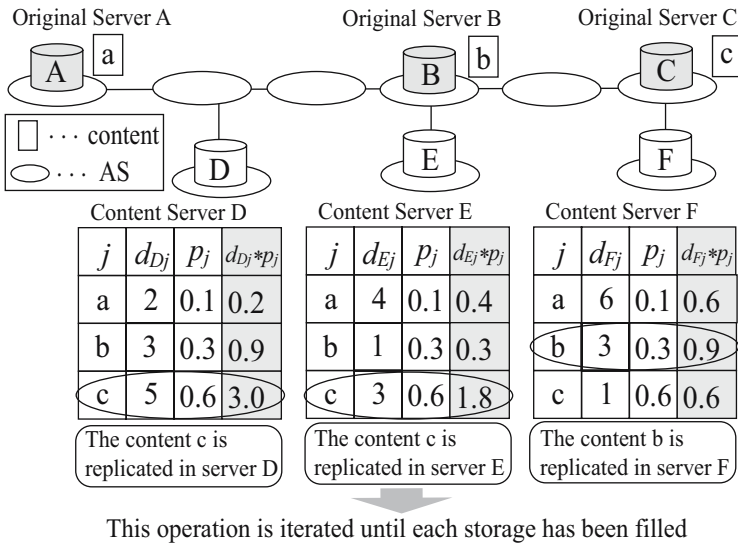


Fig. 3. Greedy-Single

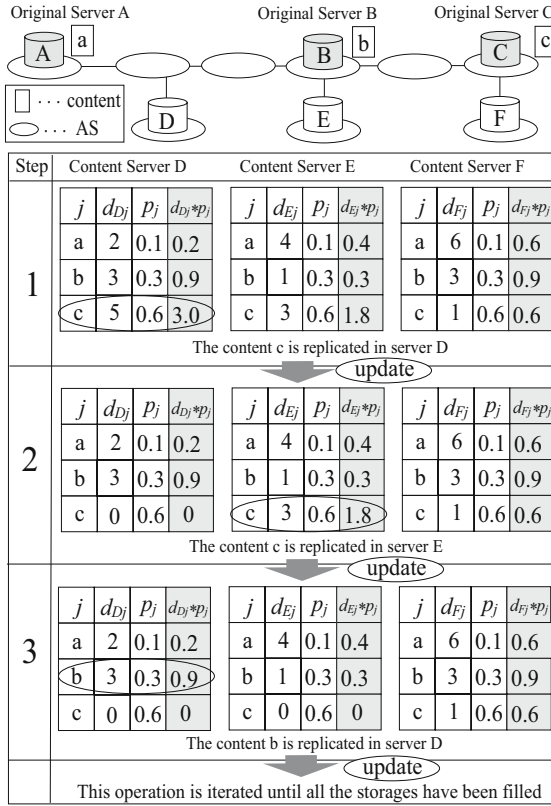


Fig. 4. Greedy-Global

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).

### 3.3 Greedy-Global

The CDN first calculates  $C_{ij} = \lambda_i p_j d_{ij}(\mathbf{x}_0)$  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 iterated until all the storages have been filled (Fig.4).

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

## 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. These content location strategy is designed for a request routing which directs a users request to the closest server. So, they are designed for Anycast routing. When more sophisticated request routing technique, such as *Active Anycast*, is used, content location strategy for Anycast, e.g. Greedy-Global, may not work well. This is because of difference between aims of 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 of 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 rather important that network has adequate amount of (the same) objects for a popular one. From these observation, we propose a new content location strategy which is applicable to *Active Anycast*-type request routing, i.e. request routing taking care both of network delay and server load, *Popularity-Probability*.

In *Popularity-Probability* content location strategy, each content server decides its storing objects according to object popularity. From the meaning that object popularity is a key factor of content location, Popularity and *Popularity-Probability* has the similar concept. However, in *Popularity-Probability*, content server decides whether it stores a specific object or not with probability which is predefined by its popularity. When the total number of content servers in a network is  $N$  and request probability of object  $i$  is  $p_i$ , expected number of content servers which store content  $i$  is  $Np_i$ . This means content are distributed randomly in a network so that the number of replicated content in a network is linear to its popularity. In *Popularity-Probability*, each content server can decide its storing objects independently and there is no necessity to exchange any information among servers. So, *Popularity-Probability* is very easy to be implemented.

## 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 technology. For request routing technology, Anycast and *Active Anycast* is applied.



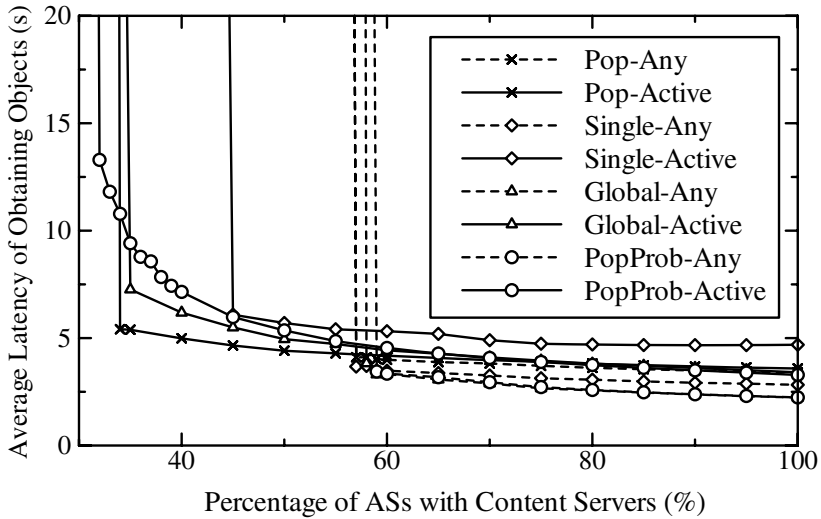


Fig. 5. Average latency characteristics

### 5.1 Performance of Request Routing and Content Location

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

- The network model of the random graph with 100 ASs is used. 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 capacity of 1.0 requests/sec.
- Accesses to servers are generated by Poisson process. The access arrival rate indicates aggregate access arrival rate to each router 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 objects, i.e. capacity of each server is 10 copies.

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 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 that delay characteristics diverges. This is because utilization of servers inside a network becomes larger than 1, i.e. servers are in overload status when sufficient number of 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 gives 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, combination of *Popularity-Probability* and *Active Anycast* can distribute adequate number of contents inside a network and guide users request with satisfying server load balance.

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

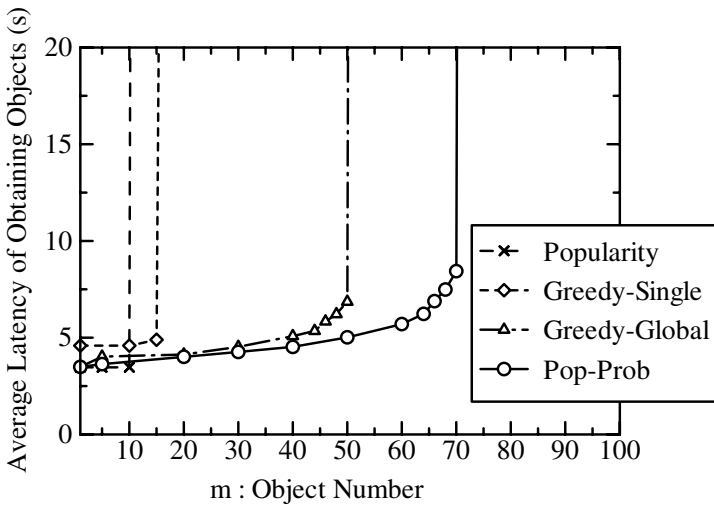


Fig. 6. Performance in the case where the popularity changes in 10 ASs

## 5.2 Simulation Results: Robustness

Another important performance for CDN is robustness. For content location strategy, e.g. *Popularity* and *Popularity-Probability*, measured or predefined information about popularity of 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 viewpoint that 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  $p_j$  change from the original design. It is modeled as the situation where the request probability of 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

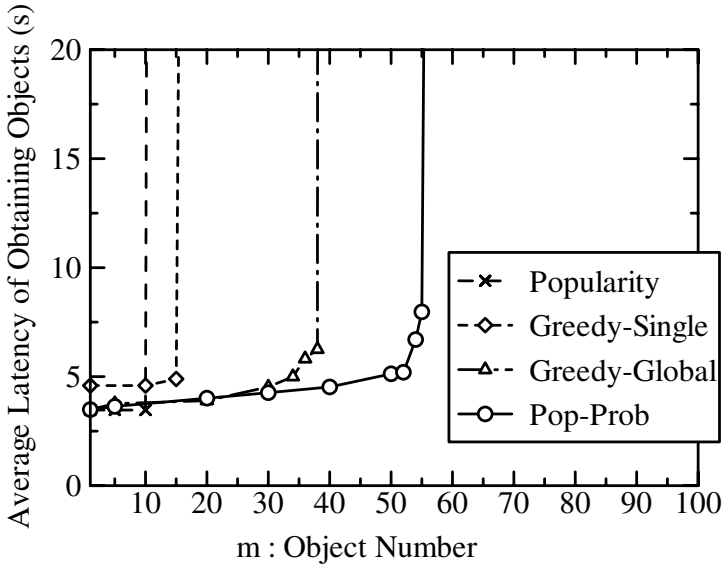


Fig. 7. Performance in the case where the popularity changes in 50 ASs

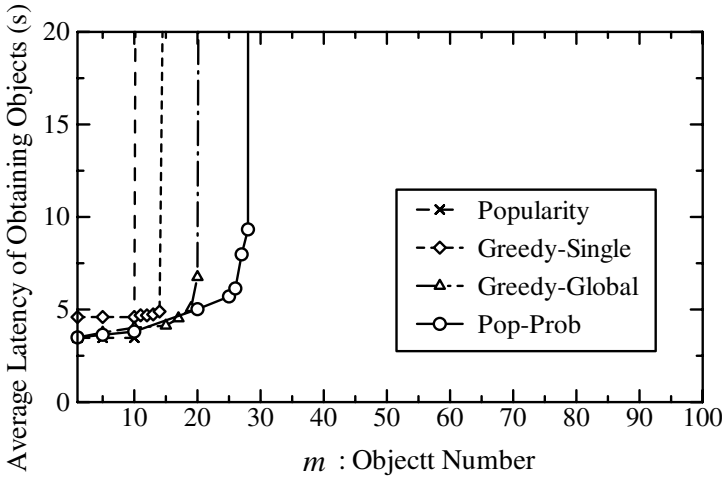


Fig. 8. Performance in the case where the popularity changes in 100 ASs

is assumed that the request probability of most popular object is  $\hat{p}_1$  and  $m$ 'th popular object is  $\hat{p}_m$  originally. Each request probability  $p_1, p_m$  become as follows after 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 Fig.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. *Active Anycast* is used as request routing in above all figures.

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

## 6 Conclusion

In this paper, we have claimed that there is significant difference between aims of content location strategy and request routing. 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 thought that request routing and content location strategy works well together when content location strategy is aim to manage the number of objects according to their popularity. From these observation, 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 storing 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. 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 robust CDN in the meaning that CDN is tolerable to change of user request tendency. Our proposed integration of request routing and content location strategy in CDN will open a new possible network design, the *robust CDN*.

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 insights obtained in the paper can be applied to general CDNs. In the case that link capacity of each content server is different, content location strategy should take into account not only popularity of contents but also server's network situation. For example, popular contents should be located at a server with good network situation. We would like to leave this issue for our further research.

## References

1. M. Day, B. Cain, G. Tomlinson, and P. Rzewski, "A model for content Internet-working(CDI)," draft-day-cdn-model-09.txt(work in progress), Nov. 2001.
2. A. Barbir, B. Cain, F. Douglis, M. Green, M. Hofmann, R. Nair, D. Potter, and O. Spatscheck, "Known CN request-routing mechanisms," draft-cain-cdn-known-request-routing-04.txt(work in progress), Nov. 2001.
3. L. Wang, V. Pai and L. Peterson, "The Effectiveness of Request Redirection on CDN Robustness," in Proc. of the Fifth Symposium on Operating Systems Design and Implementation (OSDI'02), Boston, MA, December 2002.
4. Jussi Kangasharju, James Roberts, and Keith W. Ross, "Object Replication Strategies in Content Distribution Networks," Sixth International Workshop on Web Caching and Content Distribution, Boston, Massachusetts, USA, June 2001.
5. I. Cidon, S. Kutten and R. Soffer, "Optimal Allocation of Electronic Content," in Proc. of IEEE INFOCOM, Anchorage, USA, April 2001.
6. B. Li, M. Golin, G. Giuseppe, F. Italiano, X. Deng and K. Sohraby, "On the Optical Placement of Web Proxies in the Internet," in Proc. of IEEE INFOCOM, New York, USA, March 1999.
7. L. Qui, V. Padmanabhan and G. Voelker, "On the Placement of Web Server Replicas," in Proc. of IEEE INFOCOM, Anchorage, USA, April 2001.
8. T.P. Brisco, "DNS support for load balancing," Internet Request for Comments (RFC 1794), April 1995.
9. E.D. Katz, M. Butler, and R. McGrath, "A scalable HTTP server: The NCSA prototype," Computer Networks and ISDN syst., vol.27, pp.155-164, 1994.
10. R. Engel, V. Peris and D. Saha, "Using IP anycast for load distribution and server location," Proc. of IEEE Globecom Global Internet Mini Conference, pp.27-35, Nov. 1998.
11. C. Partidge, T. Mendez, and W. Milliken, "Host anycasting service," Internet Request for Comments (RFC 1546), 1993.
12. S. Weber and L. Cheng, "A Survey of Anycast in IPv6 Networks," IEEE Communications Magazine, vol.42, no.1, pp.127-132, January 2004.
13. H. Miura, M. Yamamoto, K. Nishimura, H. Ikeda, "Server load balancing with network support: Active anycast," The Second International Working Conference on Active Networks(IWAN2000), pp.371-384, 2000.
14. H. Miura and M. Yamamoto, "Content Routing with Network Support Using Passive Measurement in Content Distribution Networks", IEICE Transactions on Communications, Special Issue on Content Delivery Networks, Vol.E86-B, No.6, pp.1805-1811, June 2003.
15. D. Tennenhouse, J. Smith, W. Sincoskie, D. Wetherall, and G. Minden, "A survey of active network research," IEEE Communications Magazine, vol.35, no.1, pp.80-86, January 1997.
16. D. Tennenhouse and D. Wetherall, "Towards an active network architecture," ACM Computer Communication Review, vol.26, no.2, pp.5-18, April 1996.