2011 IEEE/IPSJ International Symposium on Applications and the Internet

Method of Locating Mirror Servers to Alleviate Load on Servers and Links

Ryota Nakamura and Hiroyoshi Miwa
Kwansei Gakuin University
Hyogo 669-1337, Japan
Email: aug60681@kwansei.ac.jp, miwa@kwansei.ac.jp

***Abstract*—Recently, large-volume contents distributed by a content delivery network (CDN) on the Internet increase the load of content delivery servers and networks, which may degrade the quality of service. To overcome this problem, some mirror servers providing the same content are located on a network, and a request is navigated to one of the mirror servers. It is important to locate the mirror servers on the appropriate place in a network, as it affects the performance of the CDN. In this paper, we address the server location problem, which determines the location of the mirror servers satisfying the following two constraints: the number of the paths to the servers in a link is small; and the number of the nodes whose nearest mirror server is the same is small. The former constraint corresponds to the alleviation of the network load, and the latter constraint corresponds to the alleviation of the server load. First, we prove that this new server location problem is NP-complete. Next, we present a heuristic algorithm and evaluate it by applying to some actual network topologies. The results show that the algorithm can determine a good server location.**

***Keywords*-Content Delivery Network; Server Location; QoS; Optimization; NP-complete; Algorithm**

I. Introduction

It is important for a contents delivery service in the Internet to ensure the comfortable accessibility of all users to the servers. For this purpose, some mirror servers that serve same contents are located in the Internet. An access is navigated to one of the mirror servers to shorten the delay time and to balance the load. We can expect the small delay time of an access and the high reliability against a failure of a network and servers by using mirror servers. In fact, a lot of contents service providers use the method for the porpose of balancing loads of servers and networks [1], [2]. As the location of mirror servers affects the performance of a content delivery service, a method for finding a better location is needed.

A lot of methods have been proposed for the server location so far. Formulation of the location problems such as p-center problem and p-median problem are well known [3]. The purpose of these problems is to shorten the delay time from all the users to servers. As these problems belong to the class of NP-hard, a lot of approximation algorithms have been proposed, e.g., a method using Voronoi diagram [4] and using linear programming [5]. Some algorihms whose approximation ratio is theoretically bounded for facility location problems are also investigated (e.g. [6], [7]).

NA(Node to Area)-connectivity [8] is known as a measure for the reliability against a failure of a network and servers by using mirror servers. If and only if a graph structure of a network is k-NA-edge-connected, the number of edge-independent paths is at least k between any vertex and a set of vertices corresponding to the nodes on which the mirror servers are located. Even if $k-1$ links simultaneously fail in a k-NA-edge-connected network, at least a path from any nodes to a mirror server remains and the service can continue. The server location problems and the network design problems in terms of NA-connectivity have been investigated; for example, a server location problem asking for a set of the nodes where the smallest number of the mirror servers are located so that the network is k-NA-edge-connected [8] and a network design problem asking for a set of links added to a network so that the resulting network is k-NA-edge-connected [9]. A server location problem considering both delay time and reliability is also investigated [10]. This problem asks for the location of the servers that satisfies the following two constraints: a network is k-NA-edge-connected and the sum of the distances from a node to all the servers is not more than a given value [10].

As described above, many measures for the server location have been considered. However, according to the increase of the traffic of a CDN, the network traffic must be also considered. In general, OSPF (Open Shortest Path First) is used as a routing protocol in the Internet and the packets are transferred on the shortest path between two nodes. As the shortest paths are determined by a network topology, many paths may pass the same link and the traffic on the link may increase. The concentration of the traffic to the link causes congestion. Therefore, it is desirable that the number of the shortest paths to the servers in a link is small for all links. In addition, the traffic concentration to a server must be avoided. In other words, the number of the nodes whose nearest server is the same is small, because, when there are some mirror servers, a request of a user is generally navigated to the nearest server. As the traffic of a CDN will account for a substantial fraction, a network service provider offering a CDN must design the network to avoid the traffic concentration to links and servers (Fig.1).

In this paper, we address the server location problem considering the network traffic and the server load, which is especially necessary for a network service provider offering

978-0-7695-4423-6/11 $26.00 © 2011 IEEE
DOI 10.1109/SAINT.2011.95

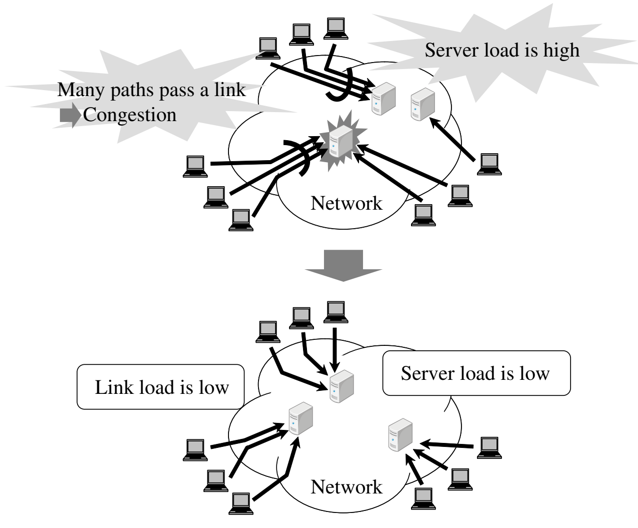

Figure 1. Server Load and Link Load.

a CDN. First, we formulate the server location problem and prove its NP-completeness. Next, we present a heuristic algorithm and evaluate the performance by numerical experiments by applying to some actual ISP backbone network topologies.

II. Server Location Problem

A. Formulation and Computational Complexity

Let $G = (V, E)$, where V and E are the vertex set and the edge set of G, respectively, be the graph representing a network structure. A server is located on a vertex in G. The vertex on which a server is located is called a server for simplicity. The location of mirror servers corresponds to vertex subset $S \subseteq V$.

We assume that the shortest path is used for communication between two nodes in a network because the shortest path routing is used in common routing protocols such as OSPF. For simplicity, we assume that the metrics of all the links in a network are the same value, one, because the metric of all links is the same value or within a short range in an actual backbone network. Furthermore, we assume that the traffic load of the requests from a node is the same for all nodes for simplicity in this paper. We define the length of a path as the number of edges included in the path. Furthermore, we define the distance between two vertices v and w in G as the length of the shortest path between v and w in G.

We assume that a request of a user is navigated to the **nearest server**, which is often used in actual navigation methods, although there are some methods to determine such a server by considering the load of the servers and the network traffic. When the distances from a vertex to some other vertices are same, the vertex has some nearest servers. The set of the vertices whose nearest server is the same is called the **neighbor set** of the server. When graph G and the set of servers S are given, a collection of the neighbor sets is referred to as $\mathcal{S}$. Fig. 2 shows that the nearest server of vertices v_1, v_2, v_3, v_4 is s_1 and that the nearest server of vertices v_4, v_5, v_6, v_7, v_8 is s_2. v_4 has two nearest vertices of s_1 and s_2. In other words, the neighbor set of s_1 has the intersection of the neighbor set of s_2, and the intersection is composed of v_4.

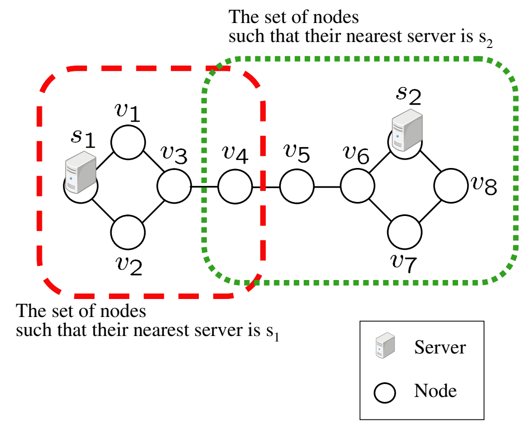

Figure 2. Neighbor set and nearest server.

If there are several shortest paths between two vertices, we assume that any path of them can be chosen. That is, even if any path of them is chosen, the servers must be located so that all required constraints are satisfied. This assumption means that the location must be determined on the safe side. When graph G and the set of servers S are given, $\mathcal{S}$, a collection of the neighbor sets of the vertices included in S, is determined. For all vertices included in a neighbor set in $\mathcal{S}$, the shortest paths to the nearest server is determined for all vertices in the neighbor set. Therefore, the multiplicity of edge e, $m(e)$, the number of the shortest paths which pass edge e, is determined. If there are several shortest paths with the same length between a vertex to its nearest server, the multiplicity is counted for all edges included in the paths. This definition is also based on the reason that the location must be determined on the safe side.

The number of the nodes included in the neighbor set of a server corresponds to the load of the server, and the number of the shortest paths that pass an edge corresponds to the load of the link. For a network service provider offering a CDN, the location of the servers must satisfy the constraints that the number of the nodes included in the neighbor set of a server is small for all servers and that the number of the shortest paths that pass an edge is small for all edges. We formulate this server location problem as follows:

Server Location Problem with Restricted Loads on Servers and Links (SLRL)
INSTANCE: *Undirected graph* $G = (V, E)$*, positive inte-*

gers k, r, and c.

QUESTION: *Is there a vertex subset* $S = \{s_1, s_2, \dots, s_k\} \subseteq V$ *such that* $|V_i| \leq r$ $(i = 1, 2, \dots, k)$ *where* V_i *is the neighbor set of* s_i $(i = 1, 2, \dots, k)$ *and that* $m(e) \leq c$ $(e \in E)$*?* ∎

The size of a neighbor set corresponds to the load of the server to which the nodes in the neighbor set access. The smaller the size is, the lower the server load is. The multiplicity of an edge corresponds to the load of the link. Thus, a vertex subset S, the solution of this problem, corresponds to the set of the nodes to which the servers are located so that the server load and the network load are low.

We show the computational complexity of SLRL.

Theorem 1 *SLRL is NP-complete.*

Proof: The SLRL belongs to NP, because, when graph G and its vertex subset S are given, we can check in polynomial time whether the size of each neighbor set is equal or less than r and the number of shortest paths that pass each edge is no more than c.

Next, we show that 3SAT, which is an NP-complete problem [11], can be reduced to SLRL in polynomial time.

Let $U = \{u_1, u_2, ..., u_n\}$ be a set of variables and $C = \{C_1, C_2, ..., C_m\}$ be a set of clauses in an instance of 3SAT. In this proof, we assume that no instance includes a trivial clause that is satisfied by any truth assignments. We construct an instance of SLRL $(G = (V, E), k, r, c)$ from an instance (U, C) of 3SAT as follows (Fig. 3):

Let $V \leftarrow \emptyset$ and $E \leftarrow \emptyset$. For each $C_i \in C$ $(i = 1, 2, \dots, m)$, let c_i be a vertex, and for each $u_j \in U$ $(j = 1, \dots, n)$, let t_j and f_j be a vertex, respectively. Let $V \leftarrow \{c_1, c_2, \dots, c_m\} \cup \{t_1, t_2, \dots, t_n, f_1, f_2, \dots, f_n\}$. Let $E \leftarrow \{(t_i, f_i)\}$ $(i = 1, 2, \dots, n)$. For each $C_i \in C$ $(i = 1, 2, \dots, m)$, if $u_j \in C_i$, let $E \leftarrow E \cup \{(c_i, t_j)\}$, and if $\overline{u_j} \in C_i$, let $E \leftarrow E \cup \{(c_i, f_j)\}$. For each $i = 1, 2, ..., m$, let $V \leftarrow V \cup \{v_{ij}\}CE \leftarrow E \cup \{(t_i, v_{ij}), (f_i, v_{ij})\}$ $(j = 1, 2, \dots, m)$. Let $k = n$, $r = 2m + 2$, and $c = 1$. This transformation can obviously be executed in polynomial time.

We show that $(G = (V, E), k, r, c)$ has a solution, if and only if (U, C) has a solution.

Suppose that (U, C) has a solution, that is, there is a truth assignment to satisfy all the clauses. Then, we locate a server on t_i if u_i is true and on f_i if u_i is false $(i = 1, 2, \dots, n)$. The size of the vertex subset on which the servers are located, k, obviously satisfies that $k = n$. The nearest server of vertex c_j corresponding to clause C_j is the server t_i or f_i corresponding to the literal that satisfies C_j. If t_i (resp. f_i) is the server, t_i (resp. f_i) is the nearest server of f_i (resp. t_i) and v_{ij} $(j = 1, 2, \dots, m)$. Therefore, the number of the vertices included in the neighbor set of each server is $m + 2 + m = 2m + 2 \leq r (= 2m + 2)$. In addition, the number of the shortest paths that pass each edge is one $(\leq c (= 1))$. Consequently, this server location satisfies the constraints. That is, (G, k, r, c) has a solution.

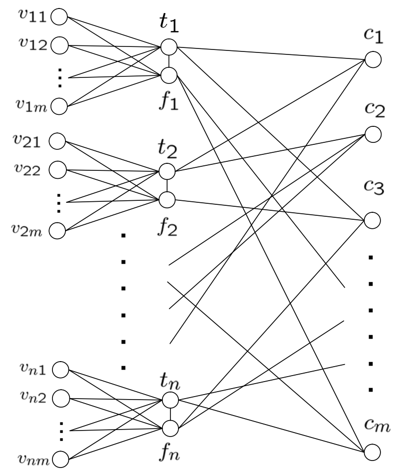

Figure 3. Transformation from 3SAT to SLRL in Polynomial Time.

Conversely, suppose that (G, k, r, c) has a solution. Let G_i $(i = 1, 2, \dots, n)$ be the induced subgraph by t_i, f_i, and v_{ij} $(j = 1, 2, \dots, m)$. Assume that no server is located in G_i. The $m+2$ vertices in G_i have the nearest server outside G_i; however, the cut size of G_i is at most m. This constradicts that $c = 1$, since there must be an edge where 2 or more shortest paths pass. Therefore, one or more server must be located in G_i. As $k = n$, just one server must be located in G_i, and no servers are located on $\{c_1, c_2, \dots, c_m\}$. Assume that a server is located in v_{ij}. v_{ij} is the nearest server of all the vertices in G_i. As the degree of vertex v_{ij} is two and the number of the vertices in G_i is $m+2$, the constraint that $c = 1$ cannot be satisfied. Consequently, all the servers are located on either t_i or f_i exclusively $(i = 1, 2, \dots, n)$. Let u_i be true (resp. false) if a server is located on t_i (resp. f_i). If no server is located on all the vertices adjacent to vertex c_j, the shortest path from c_j to its nearest server passes edge (t_i, f_i) where c_j is adjacent to t_i or f_i. This contradicts that $c = 1$. Therefore, this truth assignment satisfies all the clauses.

Thus, 3SAT is reduced to SLRL in polynomial time.

Consequently, SLRL is NP-complete. ■

B. Algorithm for SLRL

In the rest of the paper, we address the optimization version of the problem minimizing the maximum size r of the neighbor sets for all servers. The number of the servers is usually restricted because of the cost constraint, and the number of the shortest paths in a link is also restricted because of the link capacity. Therefore, the maximum size of the neighbor sets should minimized to alleviate the load of the servers.

As the optimization problem is NP-hard from Theorem 1, we cannot expect a polynomial time algorithm. Therefore, we propose a heuristic algorithm to this optimization problem. We describe the basic idea of the algorithm as follows.

Let S be the set of the already located servers. The algorithm finds server v so that the maximum size of the neighbor sets of $S \cup v$ is minimized for all vertices $v \in V$, and the algorithm greedily adds the server v successively to S.

Algorithm GreedyLocation

INPUT: $G = (V, E)$, k, c
OUTPUT: S

```
S ← ∅
while |S| < k
    Choose vertex v such that the maximum size of
    the neighbor sets of S ∪ {v} is the minimum
    for all vertices v ∈ V
    under the constraint that m(e) ≤ c for all edges e.
    If there is no vertex satisfying this constraint,
    choose v under the constraint that
    max_{e∈E} m(e) is the minumum.
    S ← S ∪ {v}
end while
if max_{e∈E} m(e) > c then Output infeasible.
Output S
```

It takes $O(n + m)$ to determine the distance from a vertex to the other vertices by using the breadth-first-search, where $n = |V|$ and $m = |E|$, since we assumed that the metrics of all links are the same. Therefore, it takes $O(n(n+m))$ to determine the neighbor sets of the servers. When the algorithm adds a server, as at most n vertices are checked whether the constraints are satisfied, $O(n^2(n+m))$ is necessary. As the number of the added servers is k, $O(kn^2(n+m))$ is necessary in total.

III. Performance Evaluation

In this section, we evaluate the performance of the proposed algorithm **GreedyLocation**. We use the structures of the commercial ISP backbone networks that are publicly available on the web site of the Cooperative Association for Internet Data Analysis (CAIDA) [12].

In the following experiments, the number of the servers, k, is 10% of the number of all the vertices in a graph, and the upper bound of the multiplicity of all the edges, c, is $\lceil (|V|/(k \cdot \Delta)) \cdot d \rceil$ where Δ is the maximum degree and d is a positive integer. $\lceil (|V|/(k \cdot \Delta)) \rceil$ is the lower bound of the maximum of $m(e)$ for all edges $e \in E$. In fact, as $|V|/k$ is the lower bound of the maximum size of the neighbor sets, $|V|/(k \cdot \Delta)$ is the lower bound of the number of the shortest paths that pass an edge. The value of c corresponds to the capacity of the links in a network. The value of d corresponds to a margin of the capacity of a link. When $d = 1$, the capacity is equal to the lower bound of the number of the shortest paths that pass a link.

We evaluated the performance for the various ratio of the number of the servers to the number of all the vertices and for the various value of d; however, we had the similar results. Therefore, we show only the results in the case that the ratio is 10% and d is four.

First, we show the ratio of r by the proposed algorithm to $r^*(= |V|/k)$. r is the maximum size of the neighbor sets and r^* is the lower bound of r. Note that the approximation ratio of the algorithm to the optimum is less than or equal to this value, r/r^*. We evaluate the performance of the algorithm by the upper bound of the approximation ratio. We show the results in Table I. The blank in the column of r means that the network does not satisfy the constraint of c.

Table I
Networks and the approximation ratio.

Networks	n	m	# of Servers	r*	r	r/r*
above_net	22	25	3	8	–	–
AGIS	82	92	9	10	26	2.60
Allegiance Telecom	53	88	6	9	20	2.22
At Home Network	46	55	5	10	13	1.30
ATT	93	154	10	10	–	–
BBN Planet	41	49	5	9	14	1.56
Cable and Wireless	19	33	2	10	13	1.30
Cable Internet	8	7	1	8	8	1.00
CAIS Internet	37	44	4	10	11	1.10
CompuServe Network Services	16	23	2	8	11	1.38
CRL Network Services	35	50	4	9	12	1.33
DataXchange Network Inc	8	24	1	8	8	1.00
EPOCH Networks Inc	29	30	3	10	14	1.40
Eunet	28	30	3	10	24	2.40
Exodus	14	19	2	7	10	1.43
Genuity	48	53	5	10	14	1.40
GeoNet Communications Inc	13	15	2	7	11	1.57
GetNet International	5	6	1	5	5	1.00
GlobalCenter	9	36	1	9	9	1.00
GoodNet	27	58	3	9	19	2.11
IDT Corp	15	18	2	8	8	1.00
ipf net	5	5	1	5	5	1.00
iSTAR Internet Inc	20	22	2	10	14	1.40
MindSpring	41	45	5	9	23	2.56
Nap Net LLC	6	7	1	6	6	1.00
Netrail Incorporated	17	21	2	9	12	1.33
PSINet	78	110	8	10	16	1.60
Qwest	14	26	2	7	9	1.29
RISQ Network	13	12	2	7	8	1.14
RNP	27	35	3	9	13	1.44
Savvis Communications	28	56	3	10	17	1.70
ServInt Internet Services	23	34	3	8	11	1.38
Sprint	22	39	3	8	12	1.50
Telstra Internet	21	24	3	7	13	1.86
UUNET	128	321	13	10	–	–
Verio	35	72	4	9	17	1.89
VisiNet	11	13	2	6	9	1.50
XO Communications	33	38	4	9	12	1.33

The proposed algorithm can determine the server location for 30 networks of the actual networks in Table I so that the server location satisfies the constraints and the approximation ratio is no more than two.

For the networks with the ratio r/r^* is more than two, as the maximum degree is large, c is too small to locate the servers so that the ratio is small. Similarly, for the networks that the algorithm cannot determine the feasible solution, the maximum degree is large.

Fig. 4 and Fig. 5 show the networks and the server locations for AtHomeNetwork and Genuity, respectively.

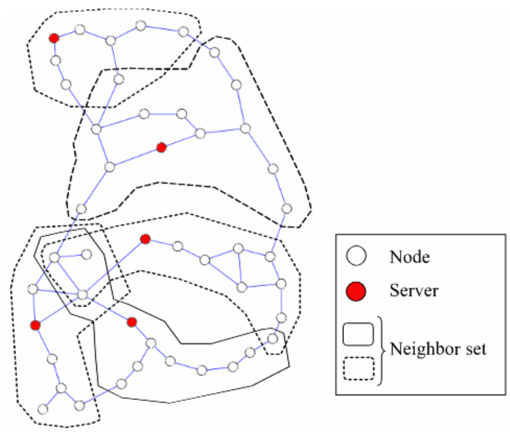

Figure 4. The network topology and the server location of AtHomeNetwork.

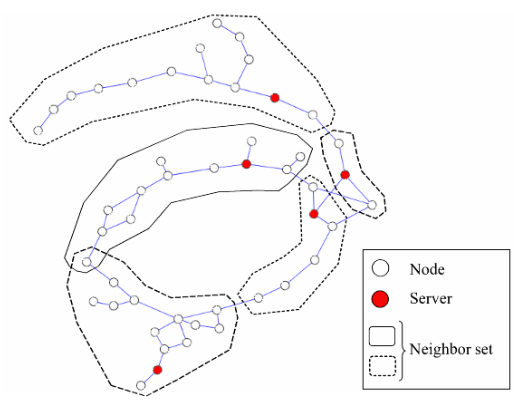

Figure 5. The network topology and the server location of Genuity.

Fig. 6 shows the relationship between r and the number of the servers for the network topology of AGIS. Fig. 7 shows the relationship between the maximum of the multiplicity and c. According to the increase of the number of the servers, r^* and c decrease from these definitions. Similarly, the maximum size of the neighbor set and the maximum multiplicity by the proposed algorithm also decrease. We can observe that the approximation ratio is small independently of the number of the servers in Fig. 6. When k is 8 or 9, the maximum size of the neighbor sets is larger than that when k

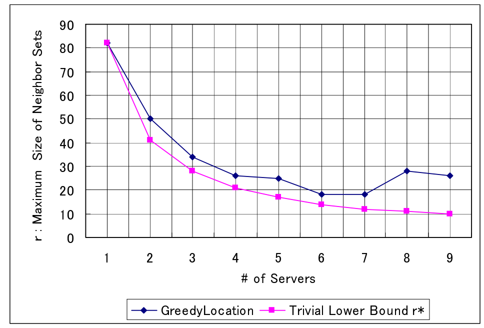

Figure 6. The maximum size of the neighbor sets in AGIS.

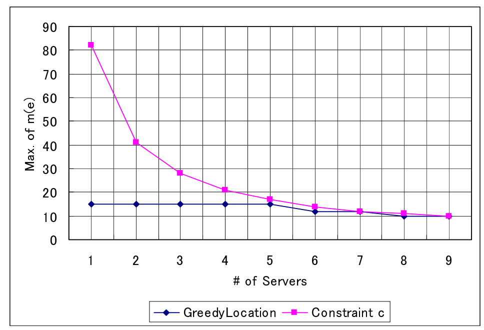

Figure 7. The maximum of the multiplicity in AGIS.

is 7. As c is too small when k is 8 or 9, the algorithm cannot find the server location that satisfies the constraint of c and that r is small. The maximum multiplicity is almost constant independently of the number of the servers in Fig. 7.

Similar properties can be observed not only in the network of AGIS, but also in the networks of AllegianceTelecom and AtHomeNetwork (Fig. 8, Fig. 9, Fig. 10, Fig. 11).

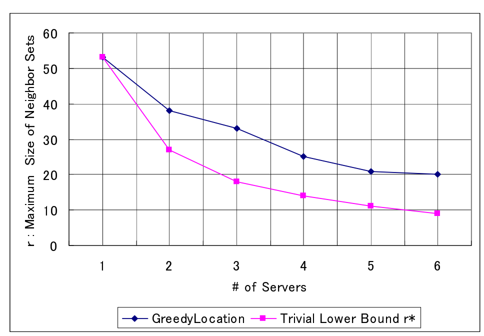

Figure 8. The maximum size of the neighbor sets in Allegiance Telecom.

When the number of the servers is large, it is expected that the load on a server and a link is small; in other words, r is small and the maximum of the multiplicity is also small. These results show that the approximation ratio of the proposed algorithm is small independently of the number of the servers, unless the link capacity is too small.

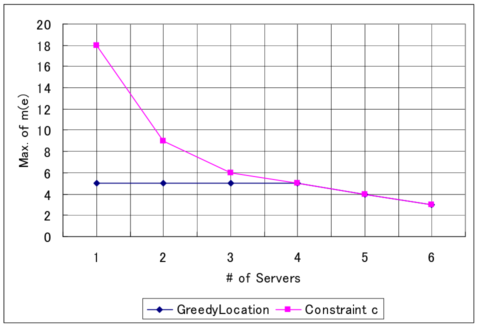

Figure 9. The maximum of the multiplicity in Allegiance Telecom.

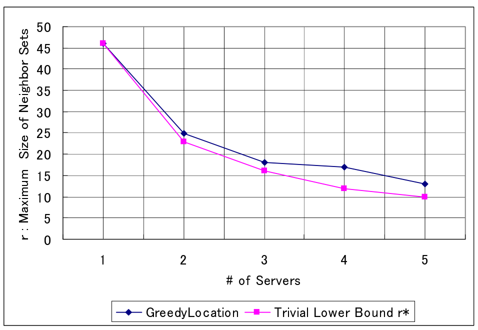

Figure 10. The maximum size of the neighbor sets in AtHomeNetwork.

IV. Conclusion

We addressed the server location problem considering the network traffic and the server load, which is especially necessary for a network service provider offering a CDN. Such a network service provider must take into account the traffic to avoid the traffic concentration to a link and a server. The number of the nodes included in the neighbor set of a server corresponds to the load of the server, and the number of the shortest paths that pass an edge corresponds to the load of the link. For a network service provider, the location of the servers must satisfy the constraints that the number of the nodes included in the neighbor set of a server is small for all servers and that the number of the shortest paths that pass a link is small for all links.

First, we formulated this new server location problem and proved its NP-completeness. Next, we proposed a heuristic algorithm and evaluated the performance by numerical experiments by applying some actual ISP backbone network topologies. The results showed that the algorithms perform well, since the approximation ratio of the algorithm is small, unless the link capacity is too small.

For the future work, it remains to improve the algorithm. The algorithm adds the server greedily, but the order of the server addition affects the results. We seek a meta-heuristic algorithm such as the local search to determine better order of the server addition.

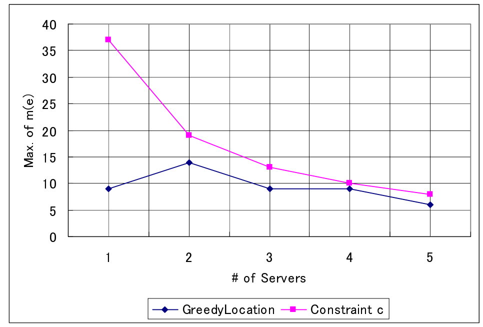

Figure 11. The maximum of the multiplicity in AtHomeNetwork.

Acknowledgment

This work was partially supported by the Japan Society for the Promotion of Science through a Grant-in-Aid for Scientific Research (S) (18100001) and (C) (20500080).

References

[1] Akamai Technologies, http://www.akamai.com

[2] Edge Cast Networks, http://www.edgecast.com/

[3] S. L. Hakimi, "Optimum distribution of switching centers in a communication network and some related graph theoretic problems", *Operations Research*, Vol 13, pp. 468-475, 1965.

[4] M. Erwig "The graph Voronoi diagram with application", Networks, Vol. 36, pp. 156-163, 2000.

[5] M. S. Daskin, "A new approach to solving the vertex p-center problem to optimality: Algorithm and computational results", *Communications of the Operations Research Society of Japan*, 45:428-436, 2000.

[6] M. Charikar, S. Guha, "Improved Combinatorial Algorithms for Facility Location Problems", SIAM J. Comput., Vol. 34, pp. 803–824, 2004.

[7] J. Byrka, "An optimal bifactor approximation algorithm for the metric uncapacitated facility location problem", Lecture Notes in Computer Science, Vol. 4627, pp. 29-43, 2007.

[8] H. Ito, M. Yokoyama, "Edge Connectivity between Nodes and Node-Subsets", Networks, Vol. 31, No.3, pp. 157-164, 1998.

[9] H. Miwa, H. Ito, "Sparse Spanning Subgraphs Preserving Connectivity and Distance between Vertices and Vertex Subsets", IEICE Trans. Fundamentals, Vol. E81-A, No 5, pp. 832-841, 1998.

[10] R. Nakamura, A. Hashimoto, H. Miwa, "Methods of Locating Mirror Servers with High Connectivity and Small Distances", Proc. International Conference on Intelligent Networking and Collaborative Systems, pp. 353-356, Barcelona, Spain, Nov. 4-6, 2009.

[11] M. R. Garey and D. S. Johnson, "COMPUTERS AND INTRACTABILITY", W. H. FREEMAN, 1979.

[12] http://www.caida.org/data/