DBSJ Journal, Vol.7, No.3
December 2008

Efficient Routing in Multi-ring Content Addressable Network

Djelloul BOUKHELEF♦ Hiroyuki KITAGAWA♠

RCAN [1] is a novel multi-ring content addressable peer-to-peer system. RCAN was proposed in the aim of improving the routing performance of CAN [4] overlays while minimizing the maintenance overhead during nodes churn in large networks. The key idea of RCAN is to equip each node with few long-links towards some distant nodes. Long-links are clockwise directed and wrap around to form small rings along each dimension. The number of rings and their sizes self-adjust as nodes join and leave the systems. RCAN is a pure P2P design, where all nodes assume the same responsibility. Unlike some existing P2P overlays, RCAN is self-organizing and does not assume any *a-priori* fixed limits for the network size or the routing state per node. Each node auto-adapts its routing state to cope with network changes.

We present in this paper an extensive study of the routing performance of RCAN under uniform and non-uniform data distributions. Experimental results show that in an overlay of n nodes, a node maintains a routing state of O(log n) long-links in average, and is able to reach any other nodes within O(log n) routing hops even in the presence of non-uniform space partitioning. Using simulation we demonstrate the full scalability and efficiency of our design and its advantages over other existing methods.

1. Introduction

Peer-to-peer (P2P) is a new emerging paradigm for organizing large-scale self-organizing distributed resource sharing systems. P2P systems are fundamentally different than traditional client-server systems, in the sense that they do not employ any central authority nor assume any global knowledge. P2P are dynamic systems by nature where nodes can join and leave the network freely. Participating nodes act simultaneously as clients and servers and exchange information and services directly with each other. Each node keeps contacts with other nodes in the system (*neighbors*).

Resource location is recognized to be at the heart of any P2P data sharing system. Scalable location services require efficient mechanisms for resources publishing and discovery that enable nodes to lookup for any resources stored in the system in an efficient and scalable way.

♦ Ph.D Student at Graduate School of Systems and Information Engineering, University of Tsukuba boukhelef@kde.cs.tsukuba.ac.jp

♠ Faculty at Graduate School of Systems and Information Engineering and Center for Computational Sciences, University of Tsukuba
kitagawa@cs.tsukuba.ac.jp

P2P systems can be classified into two categories: unstructured and structured. In unstructured systems, nodes are unaware of the resources that their neighboring nodes maintain. Lookup is typically performed by flooding the query message to all participating nodes (e.g. Gnutella). Structured P2P implement a *Distributed Hash Table* (DHT) functionality to deterministically map resources into nodes. Each node maintains information about resources its neighbors provides. DHT protocols have become an important class of P2P systems due to their scalability, routing efficiency and search completeness. Representative protocols include: CAN [4], Chord [7], Pastry [5], etc.

1.2 Content Addressable Network (CAN)

CAN [4] is a decentralized and self-organizing P2P system that provides hash-table functionality on Internet-like scale. The architectural design of CAN is a virtual multidimensional Cartesian key space. The entire key space is dynamically partitioned among the nodes present in the system such that every node possesses its individual, distinct region within the overall space.

Each node in CAN maintains contacts with its immediate neighbors (that is, managing adjacent regions). Routing information is periodically exchanged between neighbors using *heartbeat* messages. To route a message towards its target point, a node searches in its routing table for the neighbor that is closesr to the target coordinates and forwards the message to that neighbor. This task is repeated by each node along the way in a greedy manner until the message reaches its destination.

In a network of n nodes managing a d-dimensional key space, a node maintains a routing table of O(d) entries in average. This routing state enables a node to reach any other node in the system after at most O($d \cdot n^{1/d}$) hops.

One of the main advantages of CAN is the efficiency to handle nodes churn in large scale networks, due to the use of routing tables of constant size that contain local information, that require low maintenance overhead. However, in term of routing performance, CAN's greedy routing using immediate-links is not efficient, particularly in large networks when d happens to be small. The reason is that a message can only be routed through an adjacent node at each hop and there are only few neighbors in low dimensions. This may also engender long routing paths that are more vulnerable to network failures.

Thereby, achieving rapid lookup response requires a lookup protocol that shortens lookup paths as well as fairly distributes routing traffic to avoid nodes/links bottlenecks and to improve system's responsiveness.

1.2. Scope of this paper

RCAN [1] is novel self-scaling P2P system for multidimensional data management with a new topological and routing infrastructure. RCAN proposes an efficient way to establish long links on top of a conventional CAN overlay in order to improve its routing capabilities. Its aim is to optimize several features simultaneously: short routing path, small routing state, low maintenance cost during nodes churn, and more routing flexibility and fault-tolerance.

In this paper, we will present an experimental study of the routing performances of RCAN protocol in the presence of uniform and non-uniform data distributions. In the next section we will give the details of RCAN protocols. We will discuss some close related works in section 3. RCAN's routing infrastructure and mechanism are presented in section 4. In section 5, we will discuss experimental results. We conclude this paper with a summary and future works.

2. RCAN protocol

RCAN is a novel self-scaling multi-ring content addressable network. RCAN proposes a new P2P protocol with a new topological and routing infrastructure.

The basic substrate of RCAN is a conventional CAN overlay, where a node knows only about its immediate neighborhood. As mentioned above, the greedy routing using only neighboring nodes is not efficient and is more vulnerable to network failures. The key idea of RCAN is to consider equipping each node with few *long-range links* (*long links*, for short) towards some distant nodes (called *distant neighbors*). Long links are established in such a way that the routing path is short while the maintenance overhead for building and updating routing tables when nodes join or leave the system is very low. A node selects distant neighbors situated at distances inverse of powers of 2 on the coordinate space. The set of long links in each node is partitioned into small sub-sets, each of which is established along one dimension. Long links are clockwise-directed and wrap around the key space.

Unlike some close related P2P topologies, RCAN is a self-scaling system where no upper limits for the number of links per node or the network size are imposed. Each node maintains a routing state that automatically self-scales when the size of the network changes.

A self-organizing P2P overlay has the ability to spontaneously adapt its self to continuous changes without the need of an external or central authority. These changes include changes in node membership due churn, and imbalance in data and routing loads. In this perspective, our aim is to propose a system where each node autonomously self-adapts its local routing state to cope with changes in the network size. In a dynamic P2P system, nodes join and leave the system frequently, which may partially impair the overlay predefined structure and reduce system performance. Therefore low-overhead stabilization process is highly needed to restore the system structure and to keep the system performance at an acceptable level. Following this idea, we proposed an efficient approach to maintain the routing information during nodes churn with low communication overhead. In RCAN, at most $O(\log n)$ messages are exchanged in order to build the routing table of a newly joining node, $O(\log n)$ messages are also needed to update all the invalid entries in the other $O(\log n)$ affected routing tables.

In a self-organizing system, nodes should cooperate to ensure fair load balancing among them, and provide more routing flexibility and fault-tolerance. RCAN provides also a cost-free yet efficient mechanism to cope with moderate load imbalance. However, due to space limitation we will not go further on this aspect.

The set of long links from different nodes yields multiple independent rings along each dimension. The rings are of small size, and their maintenance is very easy. The number of rings and theirs sizes self-adjust as nodes join or leave the network. In a uniformly partitioned key space, a node is member of only one ring per dimension, i.e. the ring that intersects with its own region.

3. Related works

Content Addressable Network (CAN) [4] is based on a d-dimensional torus topology. CAN achieves $O(d\ n^{1/d})$ routing performance with only $O(d)$ routing state per node. Recently, several works have aimed to improve lookup efficiency in CAN. Their essence is to slightly increase the routing state per node in order to reduce the routing latency and provide more routing fault-tolerance.

Xu et al. proposed eCAN [9], a design for efficient routing in CAN using expressways. The objective of their work is closely related to ours. eCAN is a hierarchical scheme, that maintains neighbor pointers at different levels of the logical space. Expressways are built by taking snapshots at different points of the evolution of the system. Some peers are selected to handle the expressways. An expressway maintains links to other neighboring expressways at the same level, and also pointers to peers covered by its region. eCAN shows that it is possible to achieve $O(\log n)$ routing performance by keeping logarithmic routing information at each node in a CAN overlay. Due to hierarchical design, routing is conducted in bottom-up manner. Any query message needs to be propagated to upper-level expressways, which implies a performance bottleneck in this design at the higher level nodes that maintain expressways. Moreover, the construction of long links in eCAN depends on the joining process of nodes, which may have direct impact on the routing flexibility, scalability and fault-tolerance.

Sahin et al. [6] proposed another method to improve CAN's routing performance, by establishing long distance pointers (LDP) to random nodes in the system. The number of long-links per node is constant and independent of the networks size. As mentioned in [6], the process of selecting random nodes incurs high communication overhead, and may not guaranty a good network cover. The authors proposed another improvement based on sub-space pointers (SSP). The idea is to partition the key space into virtual non-overlapping sub-spaces. Each node then selects random nodes from each sub-space as long-range contacts.

In his seminal paper [2], Kleinberg shows that a d-dimensional grid augmented with only one additional long-range contact per node chosen at random according the harmonic distribution yield networks in which routing can be performed in $O(\log^2 n)$ expected number of hops. Symphony [3] and SCAN [8] are two overlays following the small-world model. Both of them establish k long-links per node, and achieve $O((\log^2 n)/k)$ routing performance. Symphony arranges all the participating nodes in one big ring. Each node then selects k long links using a harmonic distribution function. SCAN approximates Kleinberg's model to multidimensional spaces. However, instead of estimating the network size

like in Symphony, SCAN places an upper limit N_{max} for the network size, and chooses $k = \log N_{max}$.

Except eCAN, all the aforementioned methods are pure P2P designs. By keeping a fixed number of long links per node, they provide O(log n) lookup performance in average (k is chosen to be close to log(n) in Symphony and SCAN).

Routing state flexibility is a common drawback of LDP, SSP, and SCAN. The number of links per node is fixed and does not scale with the network size. Typically, large values for k incur high maintenance overhead. However, small values of k, and according to the small-world model [2], does not guaranty logarithmic network diameter (*maximum path length*) in large network ($n > N_{max} \Leftrightarrow k < \log n$). We also observed this property during our simulations of LDP and SSP methods.

The process of building long links during the evolution of the network was not clearly addressed in these methods. Although we can assume that long-links are maintained in a lazy manner (like in [6,9]), the maintenance of one link needs O(log n) messages in average. In general, the total cost to update the k out-going links is O($\log^2 n$). The same thing is for the departure of nodes, since k incoming links need to be re-established. Thus, the expected communication overhead for join operations is O($\log^2 n$) messages for these methods as well as in Chord.

4. Design principle

In this section we will describe the Multi-ring Content Addressable Network, RCAN. Its aim is to provide a novel approach for establishing long links on top of a CAN-like overlay in order to improve its routing performance and enhance its fault-tolerance capabilities while minimizing the maintenance overhead during nodes churn.

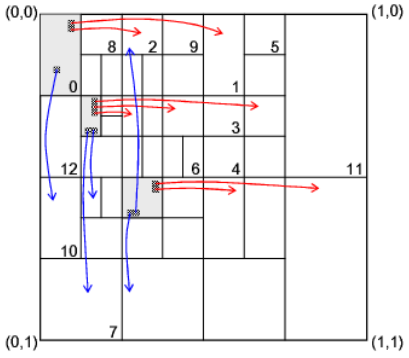


Fig. 1 Design principle of RCAN.

RCAN operates on a virtual d-dimensional Cartesian key space (Figure 1). The coordinate space warps around each dimension. A data item (*object*) is a (*key*, *value*) pair, where *key* is a d-dimensional vector and *value* is non-key information associated with the object (string, etc.).

The key space is divided into non-overlapped hyper-rectangular regions. Their sizes could be changed dynamically as nodes join (*split*) or leave the system (*merge*). Each region r is given a globally unique identifier ($r.id$) generated by applying a hash function to some data points from the region r itself.

Regions are assigned to nodes using a *one-to-one* mapping. Each node p is responsible for the data items that fall in its region r. The logical address of p is the same as its region's *id* ($p.id = r.id$). Given that nodes and regions are objectively related and their *id*s are fixed, we will simplify our notation and refer to both the node (resp. region) and its identity as p (resp. r) (without ~.*id*).

Join: When a new node q joins the system, it contacts an existing node p to share the load with. Node p splits its region into two equal parts. One of which is handed to q. Logically, on the virtual splitting tree p is a parent of q. But on the flat key space their region are also sibling. A region is split along one dimension each time, in a cyclic way (dimension 0,1,…d-1; 0,1, etc).

Leave: When a node q is about to leave the system, it contacts one of its neighbors, let's say p, hands its regions and data items to p and then leaves the system. If the node p is the latest sibling of q, it will extend its responsibility by merging its region with the region of q. Otherwise; p handles temporarily the tow regions, and will look for the latest sibling of q's region in order to finalize the merge process.

RCAN is a decentralized self-organizing content addressable network. Multiple splits and merges can be conducted independently at the same time. As a result, regions with different extents may coexist in the key space. Each node maintains its local region and keeps contact with its neighboring nodes. Only little information is exchanged between neighbors to perform split or merge operations. No information is spread out to other distant nodes or committed to a central server.

5. Routing

For routing purpose, each participating node maintains certain routing information about its neighboring nodes.

In RCAN, the routing state of each node consists of two types of links: *short links* towards adjacent neighbors and *long links* towards distant nodes. When we say that a node p has a link towards a node q, this means that a direct communication channel is established between the two nodes, and through which p can send messages to q.

5.1 Short links

Short links are contact information (network address, region's extent, etc.) about adjacent nodes. A node has an average of O(d) immediate neighbors, independent of the network size [4]. A node stores its short links in a local routing table, called **CTable**. We assume that **CTable** is maintained by the underlying CAN overlay, for example by exchanging heartbeat messages between neighbors.

5.2 Long links

Long links are at the core of our work. In this section, we show how to establish and maintain long links and how to build RCAN's multi-ring infrastructure. Routing using long-links is also presented in this section.

The routing state of a node is augmented with small number of unidirectional links towards some distant nodes in the system. Assume that a node n keeps k_i long

links along the i-th dimension (Figure 2). Denote them $l_i = \{l_i^0,\ldots,l_i^{k_i-1}\}$, where l_i^j is the distance between n and the node pointed by the j-th link. l_i is an ordered set, that is, $l_0 < l_2 \ldots < l_{k_i-1}$. The value of $l_i^j = 2^j \times w_i$, where w_i is the size (width) of the region of n along the i-th dimension. k_i indicates also how many time the region of n had been split along the i-th dimension.

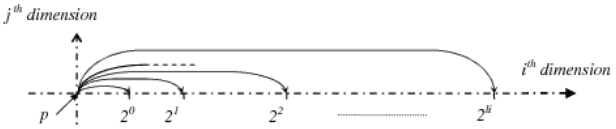


Fig. 2 Long links along the i-th dimension.

Each set l_i of long links built along the i-th dimension is stored in a separate routing table, called **LTable**$_i$.

From the above definition we can see that the number of long links originating from each node is proportional to the size of its region. A node, hence, can adapt dynamically the number of long links by establishing additional links when its region is split (node join), or dropping extra links when nodes leave the system. Experiments show that the total number of long links per node $k = \Sigma k_i$ is tightly bounded by O(log n).

5.3 Multi-ring topology

Long links are parallel to data axes and wrap around the key space. Long links originating from nodes whose origins are situated on the same line form a ring along each dimension. Multiple small rings are hence formed. In a regular grid, a node is member of one ring along each dimension, that is, the ring intersecting with its region.

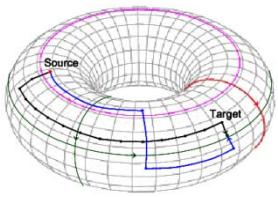


Fig. 3 Routing using long-links (*blue line*) and short-links (*black line*) in a 2d toroidal RCAN.

The number of rings and the number of nodes per ring self-scales when the network size changes. In a key space where its regions may have different sizes, a node may temporarily become a member of more than one ring along each dimension because its region is large and it intersects with several rings.

5.4 Routing algorithm

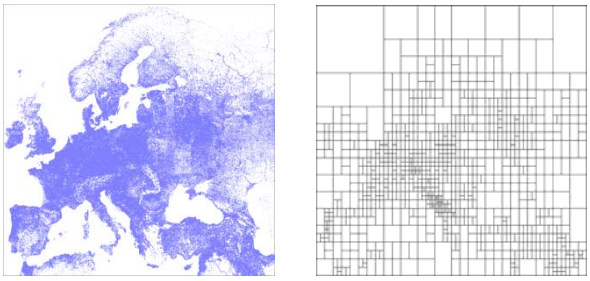
In RCAN we consider a *hop-by-hop* greedy routing approach. A node uses only local routing information to decide the next step. To forward a message, the current node consults its routing tables and determines the node that is strictly closer to the target among its *immediate* and *distant* neighbors, according to a well defined metric[1], and forwards the message through that node. If there are many neighbors at the same *expected* distance to the target, the node may choose one at random.

The routing process in RCAN consists of solving the routing path along one ring at each step. During the routing process, rings can be used in an arbitrary order (Figure 3). A good feature of RCAN multi-ring topology is that there are many paths with *almost* the same expected distance between any pair of nodes. This property enables more flexibility in the selection of the path to route a message to its target, and gives RCAN routing mechanism more robustness against node and links failures, by offering each node more choices to select the best neighbor through which it route a message to its target.

6. Performance evaluation

In order to demonstrate the effectiveness of our proposal, we have implemented a protocol of RCAN in C++ and performed series of experiments using different data patterns on networks with up to 2^{16} nodes.

We have analyzed the behavior of R-CAN under uniform and non-uniform data patterns. Uniform data set is generated using uniform random number generator. To emulate non-uniform data distribution, we used a real-world data that consists of the coordinates of more than 680000 main towns in Europe (Figure 4).



(*a*) *Europe dataset* (*b*) *1000 nodes overlay*

Fig. 4 Europe dataset (a) contains the 2d coordinates[2] of about 686511 main towns in Europe. (b) The key space is partitioned into 1000 regions, each of which is assigned to a node in RCAN.

Our measurements are mainly concentrated on the routing performance: average and maximum path length, routing state per node, number of messages processed by a node (*routing load*), number messages traversing each link (*link congestion*), etc. Due to space constrains we will show only lookup performance measurements.

To offer some comparative measurements we also run our scheme against SSP (an improvement of LDP [6]). SSP was obtained from RCAN by establishing long links randomly according to [6]. Unlike RCAN, SSP uses a fixed number (k) of long links per node independent of the

[1] RCAN uses clockwise Manhattan distance as metric function. Other metrics may also be used.

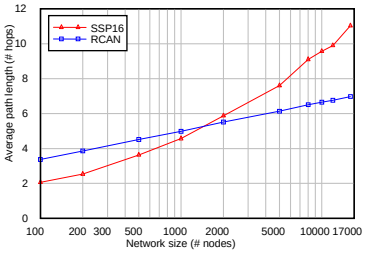
[2] In a preprocessing step, the geographical coordinates are normalized from (*longitude* , *latitude*) system to [0,1[^{2}.

network size. For fair comparison we choose (k = 16) and we employ the same greedy routing algorithm as RCAN. In what follows we will call it SSP16. In spite of its simplicity, we believe that SSP16 emulates SSP method with enough accuracy to allow a fair comparison.

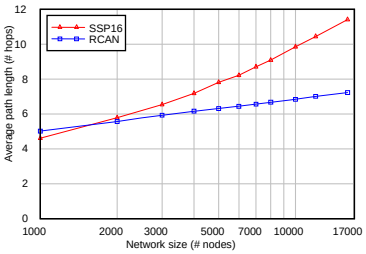
6.1 Average path length

In the following experiments we evaluate RCAN in term of routing performance and compare it to SSP16. These measures focus essentially on the average and maximum routing path as function of the network size in both uniform and non-uniform space partitioning.

Figures 5 (*a*) and (*b*) plot the average of the measured path lengths of the lookup requests with respect to the network size. The path length measures the number of hops per lookup request.



(*a*) *Uniform data set*



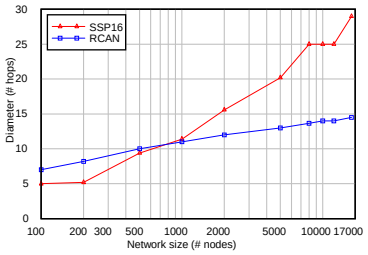
(*b*) *Europe data set*

Fig. 5 Average path length

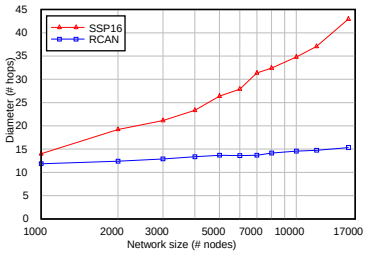
According to [6], SSP achieves almost logarithmic average path length for (k > log n). However, for large values of n, the path length is SSP becomes larger. This is because the number of links per node is fixed and do not scale with the network size. RCAN, however, delivers perfectly logarithmic path length and keeps almost the same logarithmic performance even in the presence of non-uniform space partitioning. This is because RCAN has a better self-scaling routing state that is able to cope with large scale data distributions and network changes.

6.2 Maximum path length (*network diameter*)

Figures 6 (*a*) and (*b*) depict the network diameter as function of the network size. The network diameter measures the maximum path lengths between any pair of nodes in the network. We can see that the maximum path length in SSP increases faster with the network size, and it is almost two times bigger than in RCAN for large network settings. This fact becomes more visible in the case of non-uniform space partitioning.



(*a*) *Uniform data set*



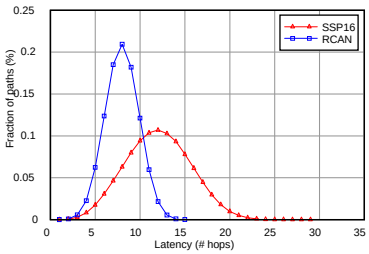
(*b*) *Europe data set*

Fig. 6 Maximum path length

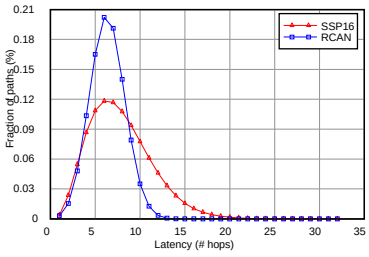
Figures 6 (*a*) and (*b*) show that RCAN's network diameter is perfectly aligned to the logarithm of the network size, even in the case of non-uniform space partitioning.

6.3 Latency distribution

In order to understand more deeply the routing process, we measured the distribution of path lengths. Figures 7 (*a*) and (*b*) depict the routing latency distribution for RCAN and SSP for uniform and non-uniform space partitioning. The latency measures the frequency of path lengths (number of hops) in a network of 5000 nodes. In this experiment, each node initiates a lookup query towards every other node in the network.



(*a*) *Uniform data set*



(*b*) *Europe data set*

Fig. 7 Path length distribution

We can see on figures 7 (*a*) and (*b*) that the variances of path length distributions in SSP16 are higher than in RCAN although RCAN maintains an average of 12.36 long links per node for a network of 5000 nodes, compared to 16 long links per node in SSP16.

In case of non-uniform distribution, RCAN maintain almost the same behavior. However, the variance of latency in SSP becomes much more important.

In order to understand this phenomenon, we have measured the node diameter distribution for non uniform data (Figure 8). Node diameter here measures, the maximum path length between each node and all the other nodes in the network.

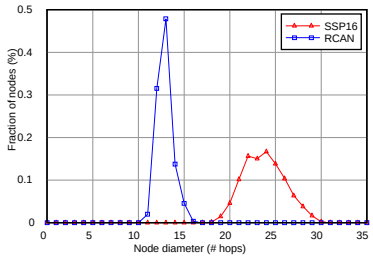


Fig. 8 Node diameter distribution (Europe data)

Figure 8 shows that node diameter distribution in RCAN is much better than in SSP, and its variance is much smaller. Witch demonstrates why the network diameter is increasing rapidly in case of SSP for large networks. This is because random node selection in SSP does not cope very well with network size especially in the presence of non-uniform space partitioning.

7. Conclusion and Future Works

RCAN is a novel self-organizing topological structure proposed to overcome the weakness of greedy routing in CAN. Its key idea is to equip each node with additional long-range links. Each node maintains a routing state that self-scale logarithmically with the network size. RCAN's multi-ring infrastructure gracefully adapts itself to cope with moderate changes in the network size.

Our solution is simple and efficient and shows that even a small extension can lead to significant improvements on different aspects. We must emphasize that RCAN is by no mean the first that uses long links for this purpose. It's not also the first to achieve logarithmic routing performance using logarithmic routing state. However, to the best of our knowledge, RCAN is the first CAN-like overlay that provides a completely decentralized mechanism that provides a self-scaling routing state.

We presented in this paper a detailed experimental study of its routing performance. Using simulation, we demonstrated the full scalability and efficiency of RCAN compared to CAN as well as other existing methods.

Acknowledgment

This research is partly supported by Grant-in-Aid from Core Research for Evolutional Science and Technology (CREST), Japan.

References

[1] Boukhelef, D. and Kitagawa, H.: "Multi-ring Infrastructure for Content Addressable Networks", Proc. of the 16th International Conference on Cooperative Information Systems (CoopIS'08), pp. 193–211(2008).

[2] Kleinberg, J.: "The small-world phenomenon: An algorithmic perspective", Proc. of the 32nd annual ACM symposium on Theory of computing (STOC '00), pp 163–170 (2000).

[3] Manku, G. S., Bawa, M. and Raghavan, P.: "Symphony: distributed hashing in a small world". Proc. of the 4th Conference on USENIX Symposium on Internet Technologies and Systems (USITS'03), pp. 10–10 (2003).

[4] Ratnasamy, S., Francis, P., Handley, M., Karp, R. and Shenker, S.: "A Scalable Content Addressable Network". Proc. of the ACM-SIGCOMM, pp. 161–172 (2001).

[5] Rowstron, A. I. T. and Druschel, P.: "Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems", Proc. of the IFIP/ACM International Conference on Distributed Systems Platforms Heidelberg, pp. 329–350, (2001).

[6] Sahin, O. D., Agrawal, D. and Abbadi, A. E.: "Techniques for efficient routing and load balancing in content-addressable networks". Proc. of the 5th IEEE International Conference on Peer-to-Peer Computing (P2P'05), pp. 67–74 (2005).

[7] Stoica, I., Morris, R., Karger, D., Kaashoek, M. F. and Balakrishnan, H.: "Chord: A scalable peer-to-peer lookup protocol for internet applications", IEEE/ACM Transactions on Networking 11(1):17–32, (2003).

[8] Sun, X.: "SCAN: a small-world structured P2P overlay for multi-dimensional Queries", Proc. of the 16th International Conference on WWW (WWW '07), pp. 1191–1192 (2007).

[9] Xu, Z. and Zhang, Z.: "Building low-maintenance expressways for P2P systems", Technical Report HPL-2002-41 41, HP Laboratories, Palo Alto, (2002).

Djelloul BOUKHELEF

He received B.Sc. and M.Sc. degrees in Computer Science from the National Institute of Computer Science, Algeria in 1998 and 2002 respectively. He is currently a Ph.D student at Graduate School of Systems and Information Engineering, University of Tsukuba.

Hiroyuki KITAGAWA

He received B.Sc. degree in Physics and M.Sc. and Dr.Sc. degrees in Computer Science, from the University of Tokyo, in 1978, 1980, and 1987, respectively. He joined Institute of Information Sciences and Electronics, University of Tsukuba in 1988. He is currently a full professor at Graduate School of Systems and Information Engineering and at Center for Computational Sciences, University of Tsukuba. His research interests include integration of heterogeneous information sources, WWW and databases, structured documents, XML and semi-structured data, multimedia databases, and data mining. He is a member of ACM, IEEE Computer Society, the Database Society of Japan, IEICE, IPSJ, and JSSST. Also, he is an IEICE Fellow and an IPSJ Fellow.