IEICE TRANS. COMMUN., VOL.E90–B, NO.10 OCTOBER 2007

PAPER Special Section on New Challenge for Internet Technology and its Architecture

Mill: A Geographical Location Oriented Overlay Network Managing Data of Ubiquitous Sensors

Satoshi MATSUURA[†a)], *Nonmember*, **Kazutoshi FUJIKAWA**[†], *and* **Hideki SUNAHARA**[†], *Members*

SUMMARY With the rapid rise in the demand for location related service, communication devices such as PDAs or cellar phones must be able to search and manage information related to the geographical location. To leverage location-related information is useful to get an in-depth perspective on environmental circumstances, such as traffic conditions or weather information. To handle the large number of information and queries communication devices generate in the current ubiquitous environment, some scalable mechanism must be required. DHTs and some overlay networks supporting range search are proposed. However, these overlay networks can not process queries of geographical region search. In this paper, we propose a overlay network called "Mill" which can efficiently manage information related to the geographical location. In DHT based overlay networks, each node has responsibility to manage a part of the whole hash table. DHTs provide scalable systems and support fast search. However, DHTs are not good at solving geographical search (range search), because hash function only supports exact match. In the Mill network, each node manages a part of ID-space calculated by "Z-ordering," which represents squire surface of the earth. This structure of ID-space enables to process region queries easily and fast. And Mill supports any scale of region search. We evaluate proposed system by using traffic infomation generator called "HAKONIWA." Simulation results show that the performance of Mill is good as well as other DHT systems. In addtion, Mill provedes more efficient region search than other overlay networks supporting range search.

key words: *structured peer-to-peer system, geographical based overlay networks*

1. Introduction

Today's mobile devices such as cars, PDAs, sensors, and other devices become powerful. In addition, these devices have connectivity to the Internet and equip positioning devices such as GPS sensors. In ubiquitous computing environment, these devices can immediately collect and provide information anywhere.

If we use a large number of information these devices provide, we can obtain detailed and up-to-date information. Gathering information based on geographical location can be efficient for judging traffic condition, weather information, and other circumstances. For example, if we can gather exact rainfall information of some region, this information is useful for the people riding a bike, climbing a mountain, and doing other things. Such users want to obtain such information immediately, because they want to get not past data but current status of weather condition.

Manuscript received January 19, 2007.
Manuscript revised April 23, 2007.

[†]The authors are with the Graduate School of Information Science, Nara Institute of Science and Technology, Ikoma-shi, 630-0192 Japan.
a) E-mail: sato-mat@is.naist.jp

DOI: 10.1093/ietcom/e90–b.10.2720

Therefore, to immediately obtain some suitable information based on geographical location, a management mechanism which can handle a large number of sensing data should be required.

In this paper, we propose a new approach which can handle information in terms of location. To simplify the management of the location-related information, we convert two dimensional coordinates into one dimensional circumference. Using this technique, our P2P network named Mill, which can flexibly search arbitrary region for location-related information. Mill has a good performance as well as DHTs. Mill can search information by $O(\log N)$ and process queries in ubiquitous environment and does not require a special node (e.g. central server). In addition, Mill can flexibly search location-related information from small region to large region. Mill does not adopt a hash function. The strategy of assigning ID is quite different from DHTs. Mill can search consecutive IDs at one time. Therefore, Mill reduces the number of queries for a region search. The results of simulation shows that Mill can search a paticular geographical region fast and scalable to increasing the number of nodes.

The rest of this paper is structured as follows. Section 2 describes requirements for information management and retrieval on ubiquitous computing environment. Section 3 shows the related work. Section 4 presents the mechanism of Mill and explains several of its properties. Section 5 shows Mill's performance through simulations with a traffic information generator. Finally, we summarize our contribution in Sect. 6.

2. Requirements for Ubiquitous Environment

In ubiquitous computing environment, there are many devices including mobile phones, desktop PCs, web cameras and sensor devices. If these devices provide information around where they exist, we may obtain valuable information. However, there are several requirements that we have to cope with.

- Scalability
  In the near future, the number of information provided by mobile devices and other devices will much increase. A centralized system which handle such information as like client-server model will much overloaded. It is required to handle information and search queries issued by a large number of devices all over the

place.
- Region search
  If sensor devices provide location-related information, we try to obtain useful information. For example, someone wants to know weather information around his/her office and traffic condition between his/her home and office. Therefore, a flexible search mechanism is required, which can be applied to an arbitrary size of geographical region.
- Fast search
  If we try to obtain traffic condition or weather information, the up-to-date information must be provided. Therefore, a search mechanism should be fast.

3. Related Work

Japan Automobile Research Institute (JARI) [1] experimented with IPcars (taxies have some sensors and connectivity to the Internet). This experiment showed that information provided from mobile devices is useful to know detailed weather information. In this experiment, a client-server approach was adopted. In the near future, it will be expected that ubiquitous computing environment come out and the number of queries for location-related information will much increase. Consequently servers will be much overloaded.

To decentralize information and queries, peer to peer (P2P) networks are widely studied. Especially, P2P networks with distributed hash table (DHT) have been proposed in many studies [2]–[5]. DHTs are scalable for the number of mobile nodes and are effectively adapted for entry and separation of nodes. DHTs can process queries even if the network is continuously changing. However, there is a serious disadvantage. DHTs support only exact match lookups because of adopting a hash function. If users search a range consisted of consecutive IDs, lots of queries whom the number is equal to the number of consecutive IDs should be processed. Therefore, the exact match mechanism is not suitable for searching a particular region.

SkiptNet [6] is one of structured peer-to-peer networks. SkipNet dose not use hashed-IDs but consecutive IDs. Using comsecutive IDs, SkipNet supports range search on one dimension. Mercury [7] is a SkipNet based overlay network. Mercury manages several overlay networks to supports range search by several attributes. For example there is one car, and this car has some attributes (color=#FFFFFF, price=$30,000, weight=1,500 kg). SkipNet can only manage one attribute. On the other hand, in Mercury network users can search cars by color, price or weight. However, Mercury can not proess the query consisted of some attributes at one time, because each attribute is independent from others. If user search geographical squire region in Mercury network, user have to search a large region, because squire region is determined by two attribute (latitude and longitude).

There are several P2P networks considering location. However, these P2P networks have some defects in dealing with location-related information. LL-net [8] is location based P2P network. This P2P network defines an area as a square region divided by latitude and longitude. LL-net is optimized for context-aware service, and this P2P network is efficient to find where node is and what services node has. LL-net has two kinds of special nodes (super peer and rendezvous peer). The super peer manages information about all rendezvous peers. All other peers should know the super peer in advance. A rendezvous peer exists per an area. This peer manages normal peers in its area.

Table 1 Summary of related work.

	DHTs	Mercury	LL-net
fast search	○	○	△
region search	×	△	○
scalability	○	○	×

Table 1 shows the comparison of related work by using the results of evaluation (Sect. 5). In DHTs networks, a large number of queries are generated for region search because of a hash function. In Mercury networks, users can search a paticular range on one dimension. However, users need to search a large region to get information of a paticular geographical region, because each attribute is independent from others in Mercury networks. LL-net improves cost of search by hierarchical ID-space. On the contrary, herarchical ID-space makes lots of special nodes and increases management cost for maintaining overlay networks.

4. Mill: A New Geographical-Based Peer-to-Peer Network

To meet the requirements in Sect. 2, we propose a new P2P network system called "Mill" considering geographical location where information is generated. This section describes the mechanism of Mill. Mill has several protocols, which are join and leave, maintenance overlay network, store and search, optimization of queries on searching several regions, maintenance of routing tables, and other protocols. Due to the space limitation, we explain join, store, and search protocols.

We assume that nodes in P2P networks are consisted of home agents (HA) of mobile devices, rack-mounted PCs managing sensor devices and other stable computers. In other word, nodes in P2P networks need having ability to store data, process queries and connect to the Internet stably.

4.1 Overview

If we deal with information based on geographical location, a P2P network system must support region search. In mobile environment, it is difficult to comprehend exact location of mobile devices in advance. Therefore, when searching a particular point, we do not know whether we acquire some information or not.

DHTs support only exact match queries because of

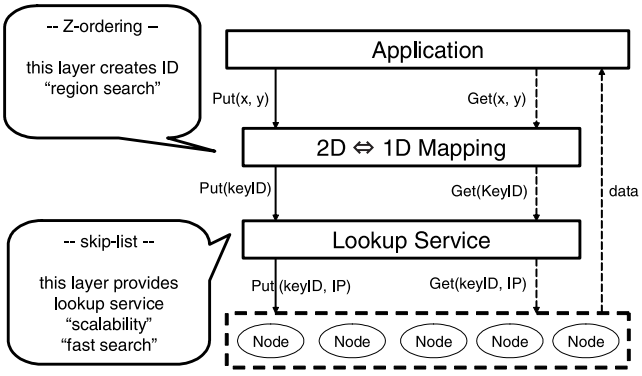


Fig. 1 Architecture of Mill.

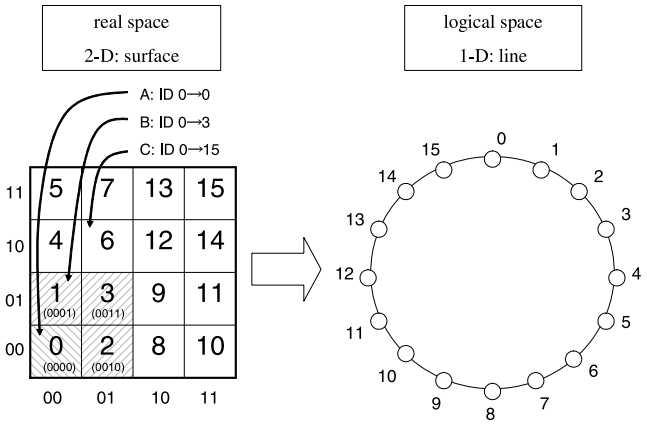


Fig. 2 2D-1D mapping method.

adopting a hash function (e.g. SHA-1 [10]). If we search a particular region, we should search all points in the region. For example, if a DHT system expresses a region as 10 bits, we should search 1024 points. Exact match causes the large number of queries on region search.

In DHT based networks, each node has responsibility to manage a part of the whole hash table. On the other hand, in Mill network, each node manages a part of ID-space which is calculated by using "Z-ordering," which represents squire surface of the earth. This difference enables Mill to support geographical range search.

As Fig. 1 shows, the architecture is hierarchy structure. The architecture of Mill is similar to the DHTs. In Mill network, if an application stores or searches location-related information, the application just specifies the latitude (y) and longitude (x). The 2D-1D mapping layer converts x and y into key-ID. This layer corresponds to a hash function of DHTs. The lookup layer searches a particular node based on this key-ID. In case that there are N nodes, a query can be processed via $O(\log N)$ messages.

4.2 Z-Ordering: Represent 2D Surface as Consecutive IDs

Mill divides two dimensional space into a grid cell by latitude and longitude. A grid cell is a small square region. If each grid cell is represented by 64 bit-ID for the surface of the earch, a size of grid cell is equals to milli-meter order. As Fig. 2 shows, suppose that each cell is assigned a 4 bit identifier. Mill manages these IDs as one dimensional circular IDs, and each Mill node is responsible for a part of circular IDs. ID of each cell is generated by alternating x-bit and y-bit. For example, if an x-bit is '00' and a y-bit is '11,' a cell ID is '0101.' This method is called "Z-ordering." Here, ID-space is very small, that is only 4-bits, to explain simply, however in real use Mill's ID space is represented by 64-bits. A particular region can be expressed as range between "Start-ID" and "End-ID." For example, in Fig. 2 ID range (0, 0) corresponds to a square cell (region A), ID range (0, 3) corresponds to quarter of the whole square (region B), and ID range (0, 15) corresponds to the whole square (region C). In fact, Mill expresses a particular square region as a consecutive of cell IDs and can search some information by range of IDs at one time. Mill searches location-related information by a few queries against arbitrary size of region. Because of this feature, Mill can reduce the number of search queries.

Here, we summarize the features of "Z-ordering."

- locality of ID
  - region search, load-balance
- consecutive ID
  - reduce search queries
- create one-dimension ID
  - simple management, fast & simple search

4.3 Join Protocol

Each node has a responsibility to handle a part of the circular ID-space. And each node communicates with two clockwise side nodes and two counterclockwise side nodes. In Fig. 3, our overlay network consists of 7 nodes (0, 4, 6, 9, 11, 12, 14). The node whose ID is 0 handles the part of the circular ID-space from ID 0 to 3. This node has 4 connections with other nodes whose IDs are 4, 6, 14, and 12.

A new node joins Mill network by the following protocol. Figure 4 shows an example of joining Mill network.

1. A new node is assigned an ID from the actual location (x, y). We define this ID as Node-ID. The new node knows an IP-address of at least one node in advance. We call this node "initial node." And the new node sends Node-ID and IP-address to the initial node. As Fig. 4 shows, the new node creates 12 as Node-ID according to its an actual location. Then, the new node sends Node-ID (12) and IP-address to the initial node (Node-ID: 6).
2. The initial node sends this message to clockwise side node, and the clockwise side node sends this message in the same way. As each node sends the message repeatedly, finally this message reaches a particular node which handles the ID-space including the Node-ID.

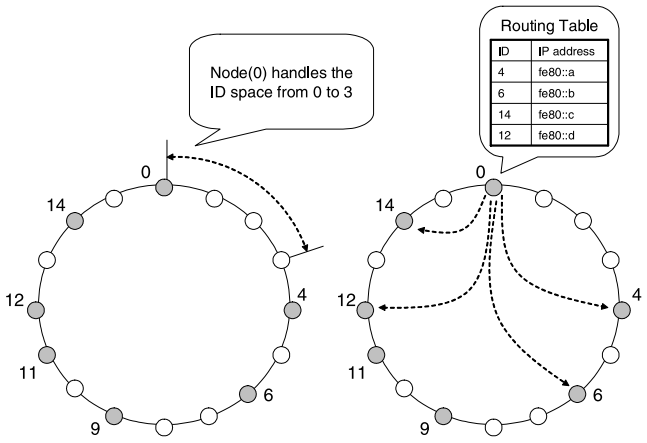


Fig. 3 Handle ID-space and routing table.

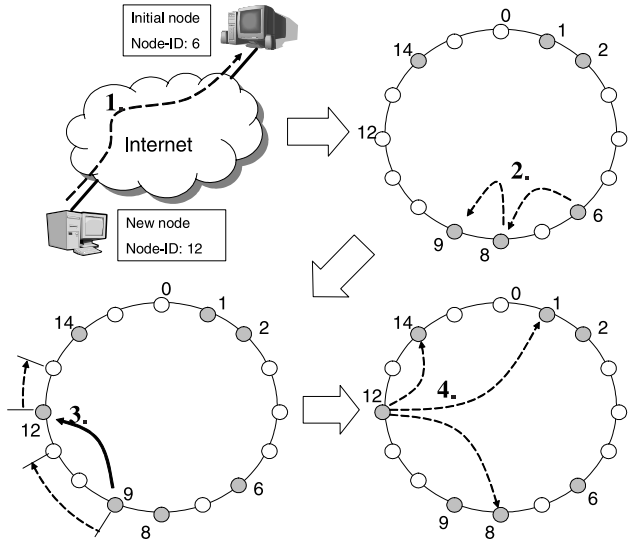


Fig. 4 Join protocol.

The initial node (Node-ID: 6) sends the message to the node (Node-ID: 8). And the node (Node-ID: 8) sends the message to the node (Node-ID: 9) which handles the ID-space including the new node's Node-ID (12).

3. The node which handles ID-space including Node-ID assigns a part of ID-space to the new node and reassigns own ID-space. And this node also informs the new node about Node-ID and IP-address of neighbor nodes. The node (Node-ID: 9) assigns the ID-space (12, 13) to the new node and informs the new node about Node-ID and IP-address of neighbor nodes (Node-ID: 8, 9, 14, 1). And the node (Node-ID: 9) reassigns the ID-space (9, 10, 11) by itself.
4. The new node informs neighbor nodes about own Node-ID and IP-address. Then neighbor nodes update their routing table. The new node (Node-ID: 12) informs neighbor nodes (Node-ID: 8, 14, 1) about own Node-ID and IP-address.

Through the join protocol, the new node can be assigned particular ID-space and knows neighbor nodes.

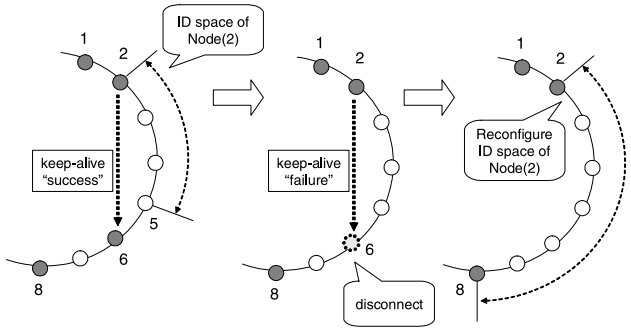


Fig. 5 Leave protocol.

4.4 Leave Protocol

Each node constantly sends keep-alive messages to neighbor nodes on its routing table. If a link of node is suddenly disconnected, a next node finds out the disconnection by keep-alive messages, and the next node tries to recover the overlay network.

A node leaves Mill network by the following protocol. Figure 5 shows an example of leaving Mill network.

1. Each node has a responsibility for a part of ID-space, and constantly sends keep-alive messages to neighbor nodes. A node (Node-ID: 2) has a responsibility for ID-space from ID-2 to ID-5, and this node sends a keep-alive message to a node (Node-ID: 6).
2. If some nodes disconnect from Mill network, a neighbor node finds out the disconnection by lack of keep-alive message. The node (Node-ID: 2) finds out the disconnection of the node (Node-ID: 6) by failure of keep-alive.
3. After detection of disconnection, a next node tries to reconfigure its own ID-space including the ID-space of disconnected node. After reconfiguration of ID-space of the node (Node-ID: 2), this node has a responsibility for ID-space from ID-2 to ID-7. This ID-space includes the ID-space which the node (Node-ID: 6) had.

If a link of node is disconnected, Mill network recovers ID-space of overlay network by itself through avove processes. If a node wants to leave, leaving process is simpler than the case that a sudden disconnection occurs. Instead of detection by lack of keep-alive messages, leaving node sends a leave-message to next node. After accepting the leave-message, the next node reconfigures ID-space as like the second and third operation of above processes.

4.5 Store and Search Protocol

A message flow of store protocol is similar to join protocol. First, if a node gets information, the node records the ID of the location where the information is generated. This ID is calculated by Z-ordering algolithm. Second, this node sends the ID and its IP-address to clockwise side node. And the

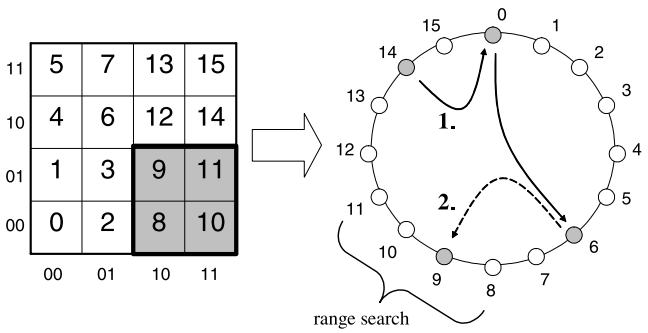


Fig. 6 Region search.

clockwise side node sends this message to the next clockwise side node. Sending the message clockwise, a particular node which handles the ID-space including the ID is received this message. This node manages the ID with the IP-address.

The search protocol is similar to the store protocol. If a node wants to get some location-related information, a search query including "StartKey-ID" and "EndKey-ID" is issued. Figure 6 shows an example. The node (Node-ID: 14) wants to search the region from ID-8 to ID-11. In this case, StartKey-ID is 8 and EndKey-ID is 11. First, the search query is sent clockwise until the node handles the ID-space including 8 is found. Second, the node (Node-ID: 6) replies to the node (Node-ID: 14) with IDs and IP-addresses related with ID-8. And this node sends the search query to the clockwise side node (Node-ID: 9). The node (Node-ID: 9) replies to the node (Node-ID: 14) with IDs and IP-addresses related with ID-9, 10, and 11. Then the node (Node-ID: 14) knows IP-addresses of nodes which have information related with ID-8, 9, 10, and 11. Connecting to these IP-addresses, the node gets information. If the nodes (Node-ID: 6, 9) do not find any information, they reply to the node (Node-ID: 14) with the message, which represents that information is not found.

4.6 Improvement of Routing Algorithm

The clockwise liner search is not scalable, because a search query is sent through a sequence of O(N) other nodes toward the destination. To reduce a searching cost, each node manages information of nodes which are power of two hops away, as like 1, 2, 4, 8, 16 hops away. Figure 7 shows the process of creating routing table.

1. After joining Mill network, each node has routing information of neighbor nodes. Node-A (star shape) has routing information of a node which is two hops away (Node-B). First, Node-A communicates Node-B to get a routing information of a node which is 4 hops away (Node-C).
2. Secondly, Node-A communicates Node-C to get a routing information of a node which is 8 hops away (Node-D). Node-C can probably reply with information of Node-D, because Node-C also adds routing information as like Node-A does.
3. Thirdly, Node-A communicates Node-D to get a routing information of a node which is 16 hops away. If Node-D does not have routing information of target node, Node-A tries to communicate Node-D again, after a while.

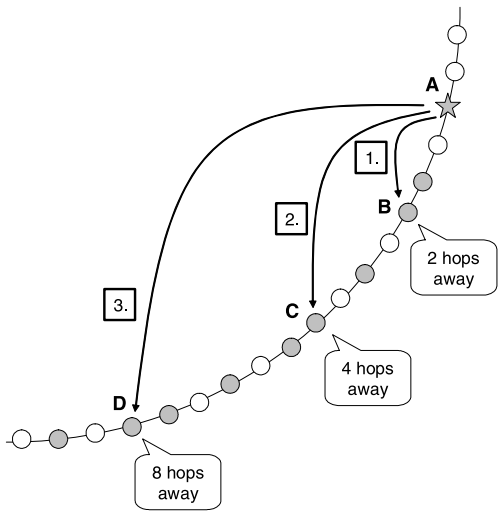


Fig. 7 Operation for creating routing table.

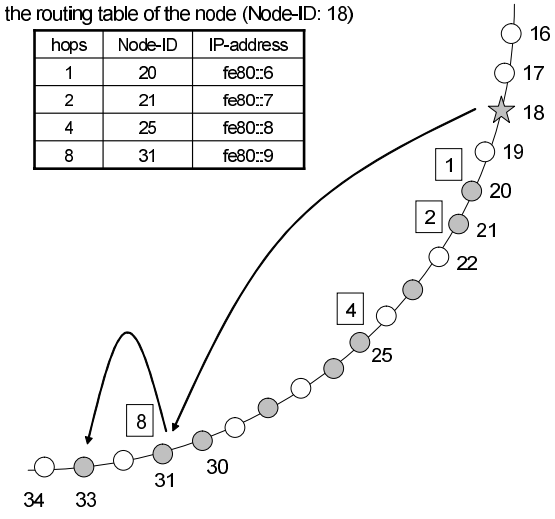


Fig. 8 Skip-list search.

Repeating this operation, the size of routing table becomes larger. Through above processes, the maximum size of routing table is as much as $O(\log N)$. This routing table is likely to have more entries for the closer nodes and fewer entries for the distant nodes. This list structure is called "skip-list."

Figure 8 shows an example of search and a clockwise routing table of the node (Node-ID: 18). The node gets the information related with a certain region whose ID is 34 by following search protocol. Here, we express node (Node-ID: 18) as Node-18 and id (ID: 34) as ID-34.

1. The Node-18 compares ID-34 with Node-IDs on the routing table.

2. On the routing table, the closest Node-ID is 31 (8-hop away node)
3. The Node-18 sends the search query to the Node-31
4. The Node-31 passes the search query to the Node-33
5. The Node-33 handles the ID-space including ID-34 and reply to the Node-18 with IP-addresses related with the ID-34.

This routing table reduces the searching cost to $O(\log N)$. This routing table also enhances stability of Mill network. Mill network can recover itself to find alive nodes by using this routing table even if several nodes are disconnected at the same time.

A search method of Mill is similar to the one of Chord. Chord also adopts a skip-list search. Mill creates a routing table based on existence of nodes. On the other hand, Chord creates a routing table based on ID-space. In DHTs networks, a system dose not need to consider density of nodes, because hash function assigns random IDs to nodes. On the contraty in real space, density of nodes is changing by location. Mill responds the difference of node density by creating routing table based on existence of nodes. This routing table and separation method of ID-space in Mill network is effective for load balance.

4.7 Load Balance

DHTs use hash-function for assignment of IDs. Based on hashed IDs, information generated by nodes is distributed. On the other hand, Mill does not use hash-function but Z-ordering algorithm for calculating IDs. We explain how to distribute information in Mill network as follows.

In Mill network, each node has responsibility for a part of ID-space. The size of IDs each node manages is determined by the distance between one node and next node. In the area where density of nodes is high, the distance between two nodes is short. In such area, lots of information is generated, however the size of IDs each node has is small. The size of IDs is inverse of density.

The amount of information each node has is not effected by density of nodes because of locality of Z-ordering. Every node manages almost same size of information, and load balance is realized in Mill network.

On the other hand, in SkipNet and Mercury network, load of nodes is affected by the density of nodes, because each nodes has a random ID and need having responsibility to manage a part of ID-space defined by this random ID. In SkipNet and Mercury network, if the density of nodes is high, a patitular node needs intensively processing many queries.

5. Evaluation

In this section, we evaluate the performance of Mill system. We have made a simulator to evaluate Mill system by Java 2 SDK. Table 2 shows the simulation environment. We use a traffic information generator called "HAKONIWA" [9] for transfer method of sensor devices. HAKONIWA creates information of cars which are latitude, longitude, speed, and direction. These cars are moving around in Nagoya city in Japan.

Table 2 Simulation environment.

CPU	Pentium4 2.4 GHz
Memory	1 GB
OS	WindowsXP SP2
programing language	Java 2 SDK ver1.5
the number of node	$10 \rightarrow 2560$
ID-space	2^{24} ($4{,}096 \times 4{,}096$)
transfer method	traffic generator

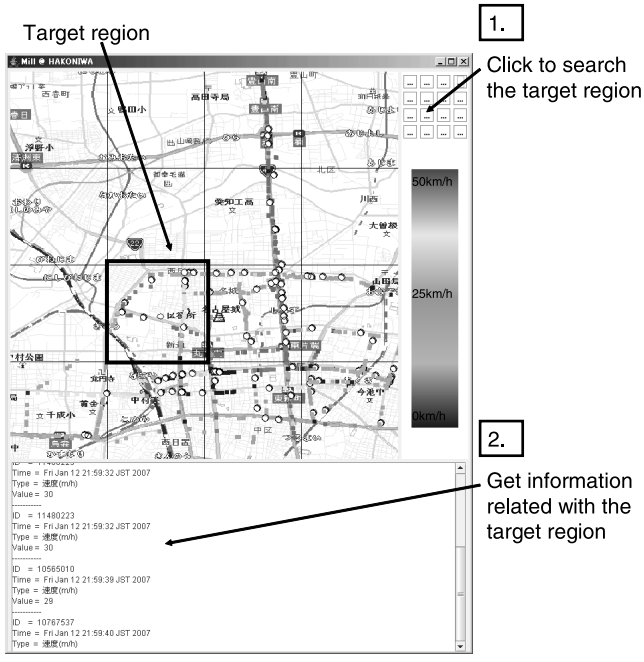


Fig. 9 Application example.

5.1 Application Example

We make a sample application on the simulator. This application creates the traffic information. In this application 100 mobile devices (cars) are moving around, sensing speed. After running the application, every node communicates with some other nodes and make Mill network. Figure 9 shows a snapshot of this application. Small circles represent mobile nodes and dots do information of speed. Users determine a target region and click the bottom related with the target region. After that, users get information on the target region. The speed information is plotted on the target region as dots. After we search several region, we can see the state of traffic in Nagoya city. For example, cars run more than 50 km/h on the north-south highway.

5.2 Search Path Length

First, we compare the performance of routing algorithms

adopted by some overlay networks. We define the path length as the number of nodes relaying a search query. On each overlay network, a node tries to search one ID by using its own routing algorithm. As Fig. 10 shows, the path length is $O(\log N)$ in Mill, DHTs and Mercury networks. In Mill network, the path length is almost half of $\log N$, because each node has clockwise and counterclockwise information of nodes on a routing table. There is a little difference. However, each algorithm has almost same performance in limitted environment where a node tries to search one ID.

We evaluate a path length of Mill in two circumstances. In one circumstance, sensing devices move around randomly. In another circumstance, sensors move on roads and its motion is simulated by "HAKONIWA" [9]. In the latter circumstance which is similar to real world, the performance of search is slightly worse than the one by random walk, because the size of routing table of nodes in circumstances of "HAKONIWA" is larger than th one of nodes in circumstance of random walk, due to difference in density of nodes within the ID-space. Assume that the Round Trip Time (RTT) of mobile phones is about 500 milli-seconds, in Mill network search queries reach the destination within 3 or 4 secconds.

Secondly, we compare the performance attributed to data structure on each overlay network. As Fig. 11 shows, we express the whole ID-space as 64 (32 + 32) bits. We assume that the density of nodes is uniform. We also express target region as "i" bit-length, the number of nodes in the target region as "m" nodes, and the number of nodes in the whole network as "N" nodes.

DHTs only support exact match queries. In a DHT network, a node searches all points in a target region. Then, the search cost is $2^i \times \log N$.

In Mercury network, a node searches a paticular ID with messages more than twice as much as in Mill and DHTs. Besides, it is the most significant defect that Mercury needs to search a large area. For example, if a range query is related with Nagoya City, Mercury needs to search north and south of Nagoya even including Russia and Australia. Mercury also needs to search east and west of Nagoya on the circumference of the earth, because Mercury network has to manage ID-space of latitude and ID-space of longitude independently. In Mercury network, a node respectively searches a stripe region of latitude and longitude related with a target region. Then, the search cost is $2 \log N + 2m \times 2^{32}/2^{i/2}$.

In a Mill network, a node can search sequential IDs at a time. In the target region, a search query is sequentially sent from a node to a node. Then, the search cost is $\log N + m$. Let "i," "m," and "N" be 54, 20, and 20480 respectively, each search cost is as follows.

- $DHT : 2^{54} \times \log 20480 = 2.58 \times 10^{17}$
- $Mercury : 2 \times \log 20480 + 2 \times 20 \times 2^{32}/2^{27} = 1309$
- $Mill : 1/2 \log 20480 + 20 = 27$

In DHTs networks, a node should search all points in a target region. In Mercury network, a node should search a external stripe region. On the contrary, in Mill network, a node just communicates with nodes in target region. So, the performance of Mill is better than other overlay networks. However, if "m" increases, the performance of Mill becomes worse. The factor of performance degradation is sequential search in the target region. It seems that adopting routing table is effective in the target region to solve this degradation. We need to consider the relation among the quality of information, the density of nodes, and the routing table. This optimization is a future work.

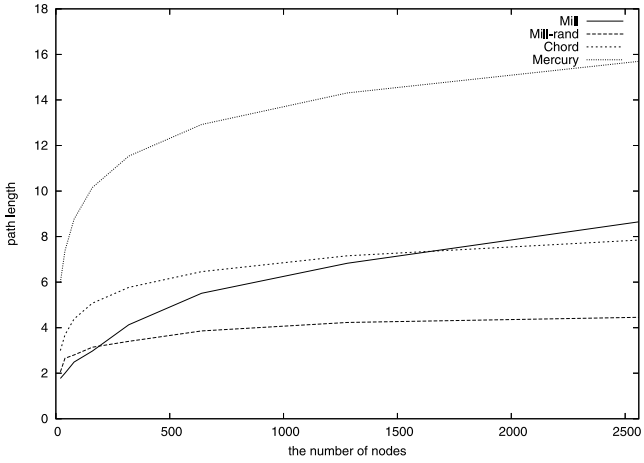


Fig. 10 Node vs. path length.

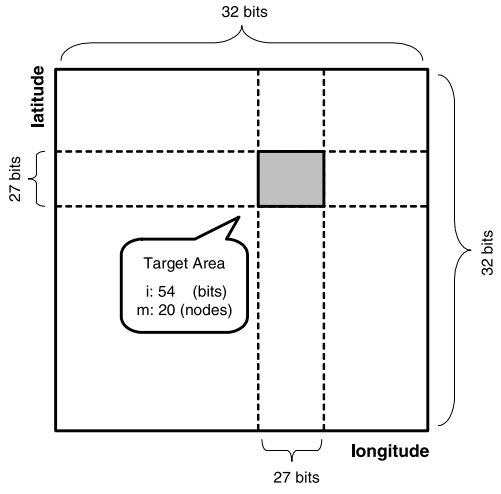


Fig. 11 Exsample of searching a region.

LL-net has hierarchical ID-space to improve search mechanism. LL-net solves range queris via almost $O(\log N)$ by creating a number of hierarchical layers. However, as the number of hierarchical layers becomes larger, the management cost increases. If one of the layers consists of very small areas, user can search very a small region. On the contrary, as the number of rendezvous peers increases, the super peer should manage a large number of rendezvous peers.

The performance of DHTs and LL-net are directly affected by the size of ID-space. One of the Mill's advantages is that the performance of Mill is not affected by the size of

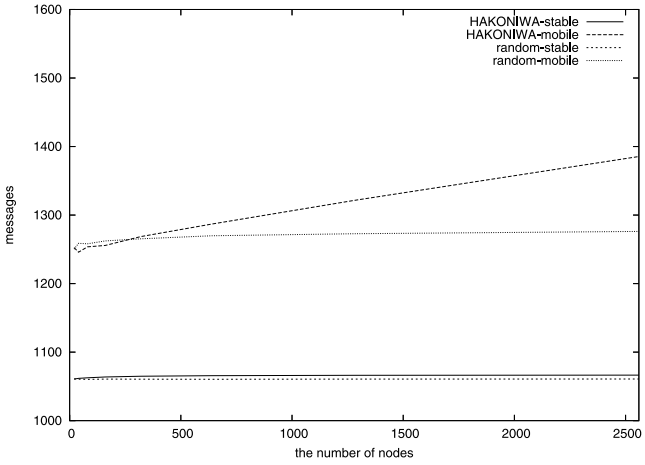


Fig. 12 Nodes vs. messages.

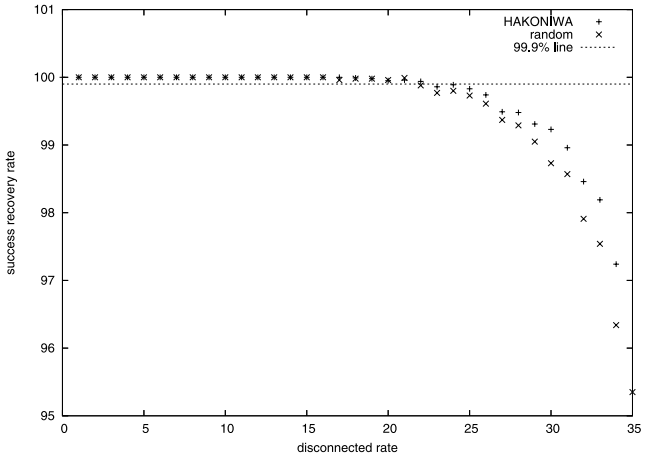


Fig. 13 Disconnected vs. recovery.

ID-space.

5.3 Management Cost

There are 2 types of messages in Mill network. One type is join-leave message. This message is sent if a node joins or leaves from Mill network. Another type is keep-alive message. A node sends this message to recognize a link status of other nodes.

We evaluate the number of processed messages per node. In Fig. 12, "Hakoniwa-stable" means that IDs of nodes are determined by HAKONIWA (traffic infomation generator) and nodes don't often join and leave from overlay network because the status of link is stable. We assume that nodes are stable machines having wired-link as like home agents of mobile devices, rack-mounted PCs managing sensor devices and other stable computers. And "random-mobile" means that IDs of nodes are determined by function of random or hash and status of network is unstable. In this case, we assume that nodes are mobile nodes having wireless link (e.g. PHS, wireless-LAN) as like cars, PDAs and other mobile devices.

In the case of HAKONIWA, if nodes repeatedly join and leave from overlay network, the management cost are increasing as the number of nodes increases. The reason is that lots of recovery message are generated in the place where the density of nodes is high. However, if status of link is stable, management cost is constant.

For sharing sensing data generated by mobile devices, nodes of overlay networks are stable nodes and mobile devices should have a role of not an overlay node but a data generator. And stable nodes should gather and manage sensing data generated by mobile devices.

In this stable stuation, management cost of nodes in Mill network is constant and Mill is scalable to increasing the number of nodes.

5.4 Robustness

We evaluate the robustness of Mill network. It is important to work Mill network even if link status of nodes is continuously changing. We define the normal condition of Mill network as that every node appropriately manages ID-space and connect to neighbor nodes. In other word, in this condition, each node can correctly put and get information. In this simulation experiment, a particular percent of nodes is forced to be disconnected at once. Alive nodes try to recover Mill network to find other nodes by using a routing table. As Fig. 13 shows, Mill network can recover itself perfectly (100%) until disconnected rate is about 15%. Each routing table has information of neighbor and distant nodes, and Mill network can recover itself by this routing table even if several nodes become disconnected at once.

6. Concluding Remarks

In the ubiquitous computing environment, communication devices can provide information anywhere and anytime. Therefore, information should be shared among communication devices based on geographical location. Mill enables communication devices to share information based on geographical location. In an N-node network, Mill can search information by $O(\log N)$ and each node maintains routing information of $O(\log N)$ other nodes. Management cost of each node is constant if the number of nodes is increases. In addtion, Mill can recover the overlay network, even if 15% nodes become disconnected at the same time.

The search mechanism by using "Z-ordering" is efficient to get data in a geographical squire region. By this method, users can search any size of squire region. Mill provides fast and flexible geographical search mechanism. If nodes of Mill network are consisted of stable machines and nodes manage data generated by sensing devices in each place, Mill contribute to manage and share data in ubiquitous sensing environment.

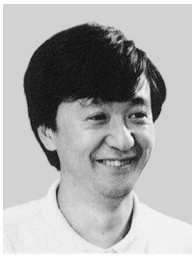
References

[1] JARI: Japan Automobile Research Institute, http://www.jari.or.jp/
[2] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Schenker, "A

scalable content-addressable network," ACM SIGCOMM, pp.161–172, 2001.
[3] I. Stoica, R. Morris, D. Karger, M.F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lockup service for Internet applications," ACM 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, pp.149–160, 2001.
[4] B. Zhou, D.A. Joseph, and J. Kubiatowicz, "Tapestry: A fault tolerant wide area network infrastructure," Sigcomm, Tech. Report UCB/CSD.01.1141, 2001.
[5] A. Rowstron and P. Druschel, "Pastry: Scalable, decentralized object location and routing for largescale peer-to-peer systems," IFIP/ACM Int'l Conf. Distributed Systems Platforms (Middleware), pp.329–350, 2001.
[6] N.J.A. Harvey, M.B. Jones, S. Saroiu, M. Theimer, and A. Wolman, "SkipNet: A scalable overlay network with practical locality properties," Proc. Fourth USENIX Symposium on Internet Technologies and Systems (USITS'03), March 2003.
[7] A.R. Bharambe, M. Agrawal, and S. Seshan, "Mercury: Supporting scalable multi-attribute range queries," ACM SIGCOMM Computer Communication Review, pp.353–366, 2004.
[8] Y. Kaneko, K. Harumoto, S. Fukumura, S. Shimojo, and S. Nishio, "A location-based peer-to-peer network for context-aware services in a ubiquitaous environment," Proc. 2005 Symposium on Applications and the Internet Workshops (SAINTW'05), pp.208–211, March 2005.
[9] T. Hino, M. Sato, K. Uehara, K. Imamura, H. Akabane, H. Haruta, R. Horiguchi, and H. Sunahara, "Hakoniwa — Application developement environments for Internet car systems," 11th World Congress on ITS, Nagoya, Oct. 2004.
[10] D. Eastlake and P. Jones, "SHA-1: Us secure hash algorithm 1," RFC3174, Sept. 2001.

Satoshi Matsuura received the B.S. degree in Physics from Ritsumeikan University in 2003, and the M.E. degree in Information Systems from Nara Institute Science and Technology in 2005. From 2005, he has been a Ph.D. candidate at Nara Institute Science and Technology. His research interest includes overlay networks and sensor networks.

Kazutoshi Fujikawa is an Associate Professor in Graduate School of Information Science, Nara Institude of Science and Technology since 2002. He was a visiting researcher of the Multimedia Communications Laboratory at Boston University from March 1996 through January 1997. He has been engaged in the project of bio/IT related R&D called "BioGrid Project," which consists of several institudes in Kansai area of Japan. His research interests widely cover distributed systems and multimedia systems. Now he is very interested in Grid systems for scientific simulations. He received the M.E. and Ph.D. degrees in information computer sciences from Osaka University in 1990 and 1993, respectively. He is a member of IEEE and ACM.



Hideki Sunahara received the B.S. and M.S. degrees in electrical engineering from Keio University in 1983 and 1985, respectively. He received the Ph.D. in computer science from Keio University in 1989. Currently, he is a professor in the Information Technology Center, Nara Institute of Science and Technology, Ikoma, Japan. His research focuses on multimedia communication systems, digital libraries, computer architecture, parallel processing, distributed systems, operating systems, and computer networks. He is a member of the ACM, IEEE, Internet Society, JSSST, and IPSJ. He is also a board member of the WIDE project of Japan.