NAIST-IS-DD0561029

Doctoral Dissertation

A Study on Geographical Location Based Overlay Networks for Sharing Global Ubiquitous Sensing Data

MATSUURA Satoshi

February 1, 2008

Department of Information Systems
Graduate School of Information Science
Nara Institute of Science and Technology

A Doctoral Dissertation
submitted to Graduate School of Information Science,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
Doctor of ENGINEERING

MATSUURA Satoshi

Thesis Committee:

Professor SUNAHARA Hideki	(Supervisor)
Professor YAMAGUCHI Suguru	(Co-supervisor)
Associate Professor FUJIKAWA Kazutoshi	(Co-supervisor)
Professor ESAKI Hiroshi	(Co-supervisor)

A Study on Geographical Location Based Overlay Networks for Sharing Global Ubiquitous Sensing Data*

MATSUURA Satoshi

Abstract

As the Internet covers all over the places and the price of wireless sensors diminishes rapidly, it can be expected that a large number of heterogeneous sensor networks are developing around the globe and interconnecting to share global-scale sensing data. And such data have a great effect on our daily lives, solutions of environmental problems, developments of business and lots of other application fields.

Sensor network technologies have been focusing on data collaboration in a local area, but they have been lacking for data share among lots of sensor networks. A decentralized data management mechanism is one of the essential keys to realize the goal of sharing sensing data over the globe. Many decentralized system, mainly overlay networks have been studied over the years. However, on these overlay networks some nodes unevenly have to store large data or retrieval cost become extremely high if these networks manage data on real space. This is because these works lack for considering patterns of sensor data stream and user queries besides geographical distribution of sensor networks in ubiquitous sensing environment, even though they have been tackling the problems of scalability and trying to provide distributed and self-organized systems on the Internet-scale network.

This dissertation proposes a new overlay network which can manage sensing data in terms of geographical locations. The proposed overlay network called *Mill* constructs a decentralized data management system without destroying the locality of sensing data. On this overlay network, nodes are distributed by geographical

*Doctoral Dissertation, Department of Information Systems, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-DD0561029, February 1, 2008.

location and manage data of local areas. Proposed overlay network supports both multi-scale geographical range queries and multiple-attributes queries, managing only one dimensional ID-space. This one dimensional ID-space consisting of latitude and longitude enables a routing mechanism to become simple and fast. It is also discussed that an implementation design of Mill considers patterns of sensing data stream and user queries. This implementation optimizes a routing mechanism, and almost messages from users and sensing devices are directly sent to particular nodes without searching the overlay network each time. This feature greatly reduces the retrieval cost. A Mill network is evaluated by several criteria including retrieval performance, management cost, simultaneous connections and others, through simulation experiments and evaluations of implementation. And these evaluations clarify its scalability and flexibility as well as its limitations.

Keywords:

overlay networks, geographical range queries, multiple-attributes queries, ubiquitous sensing environments

Acknowledgments

First, I especially want to thank Momoko. She willingly accepted that I decided to tackle a doctoral dissertation, and she has encouraged me for long years. Without her kind help and understanding, I think I could not achieve my research findings.

I would like to thank my supervisor Professor Hideki Sunahara. He spent a lot of time discussing topics of my research and giving me ideas and advices. He also give me opportunities to join research projects and collaborate with lots of researchers outside our university. His powerful support enable my academic life to be more exciting.

I would like to show my appreciation to Professor Suguru Yamaguchi who gave me warm and aggressive comments for my research. These comments were very precious for me.

I would like to thank Associate Professor Kazutoshi Fujikawa who spent lots of time teaching me how to write research papers and reading my papers. His comments and advices enabled me to acquire basic skills on research.

I would like to thank Professor Hiroshi Esaki for his kind and powerful guidance for my academic activity.

My sincere gratitude also goes to all members of the laboratory for their kindhearted advices and help. I enjoyed totally five years with them.

Moreover, I also thank members of Live E! projects who have been constructing and operating Live E! system together. I was very encouraged by their activities and discussions.

Finally, I would like to thank my parents and family for their understanding and support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Every places of the Earth including very tiny spaces have various kinds of states. However, inherent abilities of human beings can only sense a bit of and rough data. Development of sensing device technologies enable to collect and record lots of data automatically, correctly and constantly. As the Internet covers all over the places and the price of wireless sensors diminishes rapidly, it can be expected that a large number of heterogeneous sensor networks are developing around the globe and interconnecting to share global-scale sensing data. There is a possibility that our human beings acquire the ability to sense every states all over the globe by sensor device and network technologies, and then this impact thoroughly goads us into shifting our principles of thinking.

Today's mobile devices such as cars, weather sensors, cameras and other devices become powerful. In addition, these devices have connectivity to the Internet and equip positioning devices such as GPS sensors. If ubiquitous computing environment come out, these devices can immediately collect and provide information anywhere. If we unboundedly use large numbers of information these devices provide, these information efficient for improving daily lives, solving environmental problems, providing educational materials, business and other lots of fields.

The key technologies in distributed sensor networks are summarized as the *scalability* and *multiple-attribute search* in regard to management and provision of large amount of sensing data. Without a scalable system interconnecting lots of sensor networks we can not get cross-organizational data, even if each organization

independently manages local sensor networks across the globe. In addition, the system should support multiple-attributes queries, because users search for a wide variety of data by geographical location, time, type of sensors and lots of other attributes.

Many distributed system have been studied over the years. However, previous works do not meet the requirements on ubiquitous sensing environment, even if these works consider scalability of network systems and try to support multiple-attributes queries. Database management systems (DBMSs) are most popular system supporting multiple-attributes queries. DBMSs create indexes of data to resolve multiple-attributes queries immediately, and achieve a great performance of searching for subset of data. However, centralized systems as like DBMSs are not suitable for managing huge numbers of sensing data constantly generated over the globe.

Decentralized and self-organized systems are required in order to manage these huge data. Overlay technologies are compatible with this requirement and have been focused on the scalability. Overlay networks have tried to distribute a particular index of data (e.g., hash table, B-Trees) among lots of nodes and construct decentralized data management systems. In recent years, several studies try to support multiple-attributes queries. These overlay networks independently manage various kinds of ID-spaces for multiple-attributes queries. However, as the available number of attributes become larger, these networks have to manage additional ID-spaces. Consequently, almost ID-spaces are not utilized and the performance of retrieval becomes worse because of the high cost of ID-space management. These ID-spaces, which are obviously different from the real world, lack for considering the locality of sensing data stream and geographical locations of sensor devices or user queries.

This dissertation proposes a new overlay network which can manage sensing data in terms of geographical locations. The proposed overlay network called *Mill* constructs a decentralized data management system without destroying the locality of sensing data. On Mill network, nodes are distributed by geographical location and manage data of local areas. Mill supports multiple-attributes queries besides flexibly geographical range queries, managing only one dimensional ID-space. One dimensional ID-space enables a routing mechanism to become simple

and fast. There has to be a trade-off between advantages of routing and limitations of retrieval. However, these limitations of Mill hardly have a negative effect on multiple-attributes search in ubiquitous sensing environment, because Mill considers patterns of sensing data stream and user queries.

Chapter 2 defines criteria and reviews variety of studies related with ubiquitous sensing environment. Previous works clarify the problems and give hints for an alternative distributed data management mechanism. Chapter 3 describes the algorithm of Mill. Mill creates one dimensional ID-space from latitude and longitude by *Z-ordering* mechanism. On Mill network, users can access particular data via $O(\log N)$ messages. In addition, Mill supports flexible multi-scale geographical range queries. Simulation experiment shows the retrieval performance and management cost of overlay networks and other features. Chapter 4 presents implementation of Mill networks, focusing on deployment, operation and usability on ubiquitous sensing environment. Mill adopts HTTP as communication protocol and realizes above features without destroying interactive communication and, moreover, an optimized routing mechanism reduces retrieval cost on overlay network. Evaluations of implementation clarify the scalability of Mill and its limitations. In Chapter 5, activities of Live E! project is introduced. Live E! project have been trying to constructs information infrastructure on which users unboundedly access sensing data existing around the globe. Actually, this project have been installing weather sensors at several countries Asia, Canada and France. In this Chapter, future works of Live E! project and adaptation of Mill to the Live E! infrastructure. And Chapter 6 concludes this dissertation.

Chapter 2

Ubiquitous sensing environments

Today's sensing devices such as weather sensors, cars, and other devices become powerful. In addition, these devices have connectivity to the Internet and equip positioning devices such as GPS sensors. It is expected that the number of these devices become larger and larger. In this environment, these sensing devices can immediately collect and provide information anywhere and anytime. We call this environment *ubiquitous sensing environment*.

If we use a large number of information these devices provide, we can obtain detailed and up-to-date information. And these huge data can be applied for lots of applications. Gathering information based on geographical location can be efficient for environmental problems, educational material, businesses and our daily lives. For example, if we can gather exact torrential rainfall information of some region, this information is useful for evacuation instructions. And if we can examine distribution of temperature, we apply this information to solve the problem of heat-island effect.

2.1 Components on ubiquitous sensing environment

This section describes the main components on ubiquitous sensing environment. To overview these components is important to clarify the architecture of ubiquitous sensing environment.

- Sensor network technology
  Technologies of sensing devices have been developed and there are lots kinds of sensing devices (for examples, sensing temperature, humidity, illuminance and lots of other data). Nowadays, these sensing devices become more intelligent by embedded computer and wireless technologies enable to interconnect these sensing devices with each other. Thus, sensing devices can automatically sense and gather data by these communication mechanisms.

  Sensor network technology focuses management of dozens or hundreds of sensing devices. Considering interval between sensing data, network topology or method of packet transfer, sensor network technology contributes to save energy, improve cost of forwarding packets and equip APIs for getting particular sensing data.

- Distributed system technology
  Various hardware and software mechanisms realize distributed environment and improve scalability, application availability and other features.

  On ubiquitous sensing environment, huge data are constantly generated all over the world. Then, it is important to manage and provide sensing data in internet-scale sensor network.

  Consequently, distributed technology dominates the technical features on ubiquitous sensing environment.

The above components are related to ubiquitous sensing environment. Sensor network technology can sense and manage data in local areas, and distributed technology have a possibility for managing these sensing data in internet-scale networks.

2.2 Criteria on ubiquitous sensing environment

This section describes characteristic design criteria on ubiquitous sensing environment.

- Scalability
  In the near future, more and more data will be generated by various kinds

of sensing devices including weather sensors, automobiles, mobile phones and other sensors. 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.

- Multi-attributes search
  Sensing data have various attributes. For example, temperature data has its own value (e.g., 23.1 Celsius), place where the data is generated, creation time, type of sensor, owner of the sensor, manufacturer of this temperature sensor, degree of accuracy of the sensor and other various attributes. And users also want to search particular data by several kinds of attributes. For example, office workers want to know the current status of weather conditions around their offices and home. In this case, system need to support search mechanisms by time, geographical location and sensor types. On ubiquitous sensing environment, there are various sensors and lots of search requirements from users. Thus, a system should support multi-attributes search.

- Performance of search
  Sensing data are constantly generated. And almost these data gradually lost its values, because current status constantly is changing and generated data become older and older. Nobody want to receive yesterday's weather conditions when they get out of their offices. On ubiquitous sensing environment, a system is required not only to manage huge sensing data but to support fast search.

- Feasibility
  Considering internet-scale network, wide variety of communication devices exist and forward messages through the network. If a system depends on some specific hardwares or softwares, the system could not manage the ubiquitous sensing environment. To support a great diversity of devices, a system should consider generality of communication methodology. And a system also should not restricts the number of users and make operation difficult. In other words, a system should consider simplicity and easy to

use and operate.

In the near future, it is expected that a huge number of sensing devices can connect to the Internet and sensing environment is always changing day by day. And required functionalities of a system are also changing. To adapt this situation, a system must equip an expandability.

Feasibility of deployment which includes generality of communication methodology, expandability of a system and simplicity for using and operating.

2.3 Consideration of Practical Applications

This Section describes practical applications, and clarifies concrete scale of sensor networks, which systems should manage.

One of the most popular sensors is a weather sensor as like temperature, humidity and pressure. Data of these sensors have a great effect on our daily lives and society. Japan Meteorological Agency [39], for example, provides weather information of low or high pressure and typhoon created from data of Automated Meteorological Data Acquisition System (AMeDAS). There are 1,500 AMeDASs in Japan, and in other words, there is one AMeDAS within 20 km square. By using these sensors (AMeDASs), phenomena of meso-beta-scale can be measured, and lots of meteorological services are provided. However, there are some weather information which can not be provides by AMeDASs. Figure 2.1 shows meteorological classification by Orlanski [55]. Recently it has been said that global warming effects and other environmental problems are serious. For example, in urban areas, heat island effects are occurring, and reason of this phenomena have not been found out, because lots of skyscrapers, the huge number of cars, and other factors have complex effects on weather conditions. Considering the scale of skyscrapers and roads, it is required to measure micro-alpha-scale or micro-beta-scale. Besides urban area, in country sides high dense installations of weather sensors are useful for evacuation instructions, agriculture, construction of a road and others.

To measure weather data by micro-alpha-scale in Japan, appropriate 15,000,000 (1,500 x 100 x 100) weather sensors are needed. If every sensor generates data (1 kByte) per one minute, 2 Gbps (15M x 1k x 8 / 60) network traffic is con-

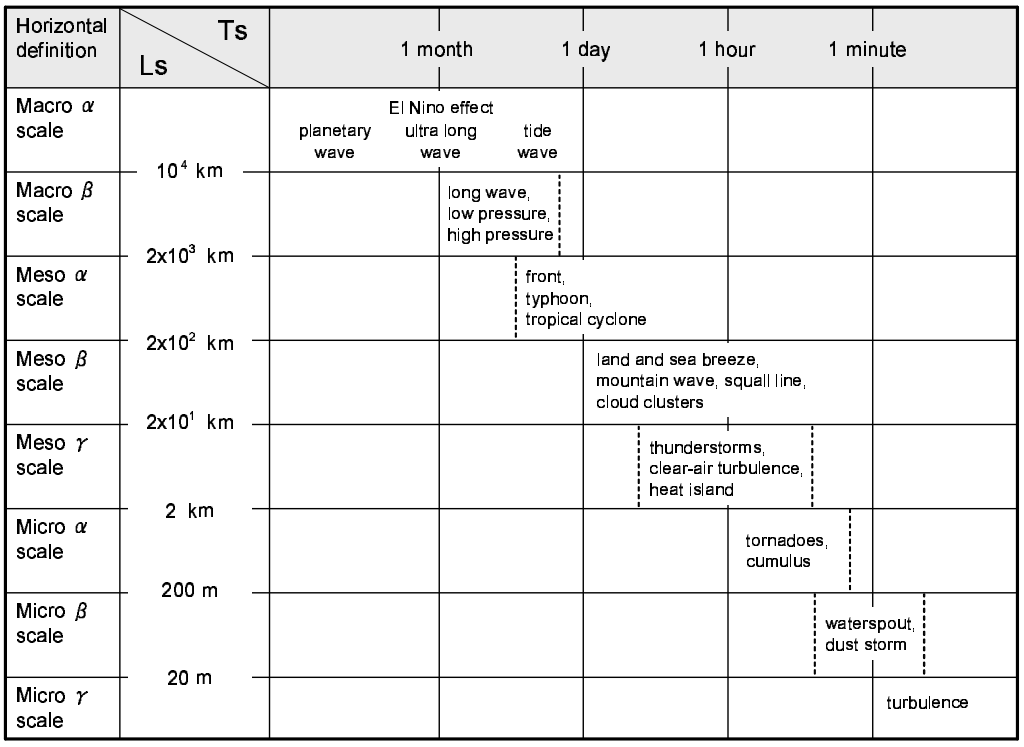


Figure 2.1: Orlanski meteorological classifications

stantly generated. It is very difficult to process these sensing data by centralized systems, because systems should process 250,000 (15,000,000 / 60) queries per second besides 2 Gbps network traffic.

Moreover, it is expected that cars connect to the Internet and collect sensing data everywhere. Sharing data cars generate, traffic congestion can be reduced and weather data can be measured in places where it is difficult to install sensors as like mountain roads. There are 75 million cars in Japan, and there are approximately 600 million cars in the world [40]. In ubiquitous sensing environment, it is required that systems have ability to manage tens or hundreds million sensors.

2.4 Related Work

Various technologies are widely studied for managing data in sensor networks or distributed environments. In this section, these technologies are categorized into three topics. Each category represents scale or design or data management.

2.4.1 sensing methodology on wireless networks

Studies of sensor networks have focused on saving energy, in other word, interval of sensing data. At Media Access Control (MAC) layer, B-MAC [57], X-MAC [8], Z-MAC [62], IEEE 802.15.4 [28], RT-Link [64] and lots of other works improve the efficacy of energy consumption. Wireless communication technologies (e.g., AODV [56], OLSR [14] and DYMO [10]) enable sensors to make networks automatically and to communicate with each other in a place where there is not the Internet.

In a next phase, it is necessary to gather several sensing data from lots of sensing devices. This subsection focuses on data collection in wireless sensor networks.

TinyDB, TAG, Synopsis Diffusion

TinyDB is one of database systems on sensor networks. TinyDB is consists of Mica Mote [27] and the networking stack as implemented in TinyOS [24]. TinyDB adopts tree topology to deliver queries to nodes and aggregate sensing data. A root node, which is usually user interface node in network, broadcasts message to other nodes on MAC-layer and build this tree topology. Figure 2.2 shows an example of tree topology and values of sensing data. Queries are described as SQL statements and queries are delivered from parent nodes to child nodes. 2.3 shows a mechanism of aggregating sensing data. Each node can sense data and aggregate values of sensors in the same time. And aggregated values are continually routed up from child nodes to parent nodes. TinyDB reduces the amount of packet transfer by this tree topology. This aggregation mechanisms is called TAG [42].

As expansion of TinyDB, ACQP (Acquisitional Query Processing) is proposed [43]. ACQP adjusts the timing of sense or aggregate data, considering frequency

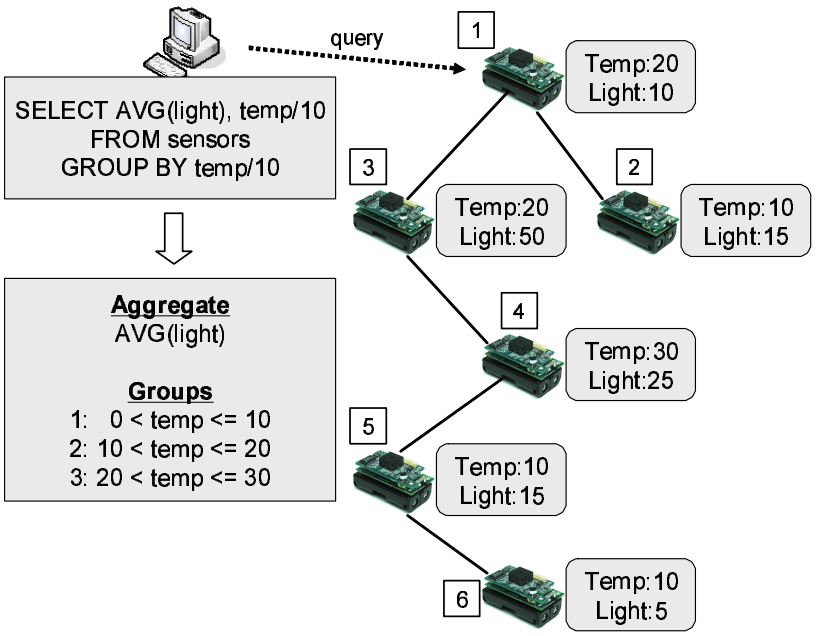


Figure 2.2: Tree topology and values of sensing data

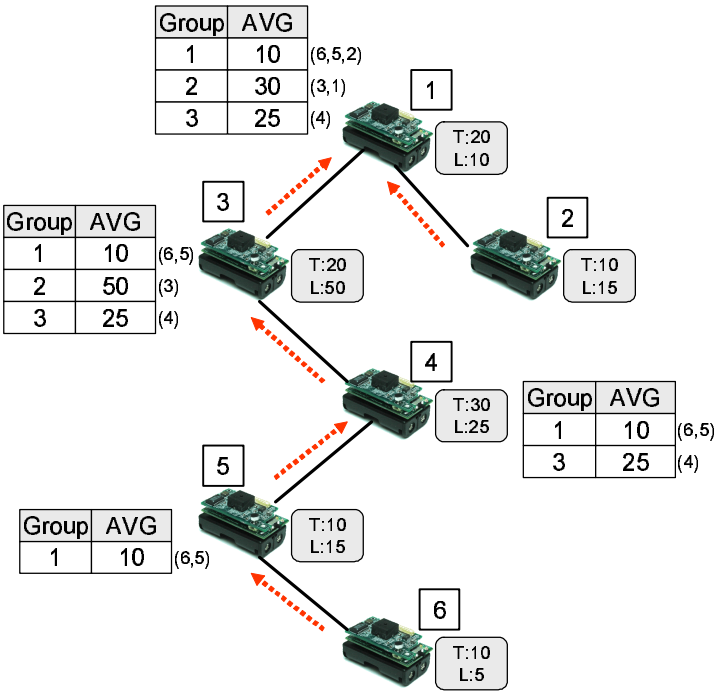


Figure 2.3: Aggregation mechanism in TinyDB

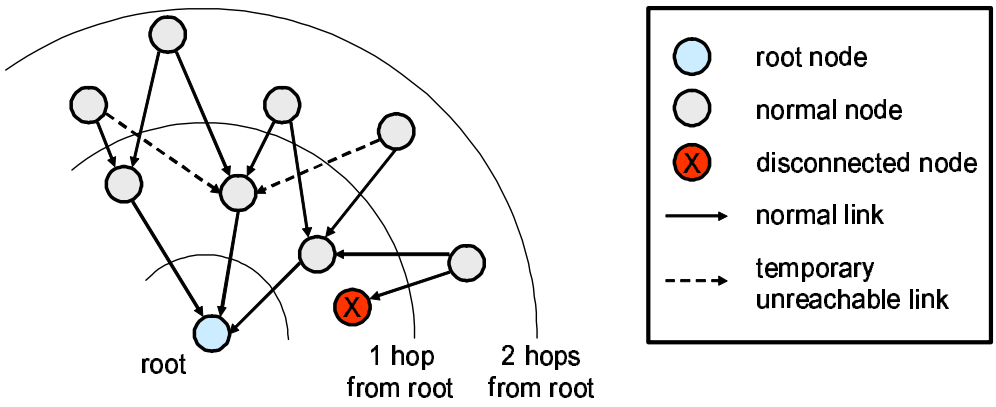


Figure 2.4: Redundant tree topology by synopsis diffusion

of requests and cost of calculations. This adjustment is efficient to save the energy of sensor devices. Lots of other methodologies [29, 22, 30, 70] are proposed for improving the performance of data collection.

On the other hand, other researchers propose an aggregation method called *Synopsis Diffusion* [52]. Synopsis Diffusion create a redundant tree topology, considering hop counts from a root node. Figure 2.4 shows the topology of nodes created by *Synopsis Diffusion*. Nodes have several paths to a root node. Thus, if some nodes become disconnected or some communication links become disabled, nodes can transfer packets to a root node by other paths. Compared with TAG, *Synopsis Diffusion* improves robustness, on the contrary, increases management cost by this redundant tree topology. *Tributaries and Deltas* [45] adopts a hybrid topology, comprised of tree and multi-path. To gather sensing data, the paths between nodes and a root node are important. *Tributaries and Deltas* applies multi-path topology to the paths near a root node, in order to transfer packets to the root node. On the other hand, *Tributaries and Deltas* applies tree topology to the paths far from a root node for reducing cost of packet transfer. By this hybrid topology, *Tributaries and Deltas* can satisfy both of robustness and reduction of management cost.

2.4.2 Large scale sensor networks

Sensor networks focus on managing dozens or hundreds sensor devices by Ad-Hoc network algorithms. These kinds of networks are useful to have a good grasp of buildings, farms or plants. If a system can integrate these sensor networks, users can access many kinds of sensing data in large areas and it expected that these sensing data are useful for environmental problems, saving energy in a metropolitan area, businesses and many other things.

Internet Car

Japan Automobile Research Institute (JARI) and WIDE project experiment with IPcars (taxies have some sensors and connectivity to the Internet) [66]. 300 IPcars move around Nagoya City in Japan, collecting data of vehicle speed, motion of wipers and weather information. IPcars can process queries of SNMP [9] and realize network transparency by NEMO [16]. For this reason, users can easily access data of IPcars by SNMP without considering modification of IP addresses. This experiment shows that information provided from mobile devices is very useful to know detailed weather information.

In this experiment, a client-server approach is adopted. In the near future, it will be expected that ubiquitous sensing environment come out and the number of queries for location-related information will much increase. Consequently, it is expected that servers will be much overloaded.

IrisNet

IrisNet is one of large-scale sensor networks [53, 20, 15]. IrisNet uses a two-tier architecture, comprised of two different agents. These agents are called sensing agents (SAs) and organizing agents (OAs), as shown in Figure 2.5. OAs are organized into groups and one service consists of one OA group. A group of OAs creates the DNS [49] like distributed database and provides query processing infrastructure for a specific service (e.g., parking space finder, person finder in Figure 2.5).

An OA manages a part of a tree topology database and this tree structure is suitable for a XML document tree. Figure 2.6 shows an example of XPath

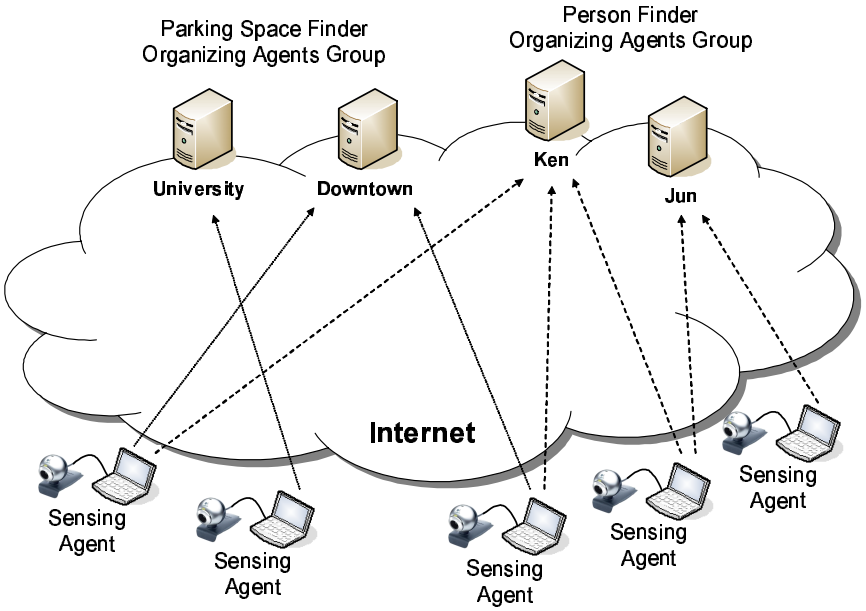


Figure 2.5: Architecture of IrisNet

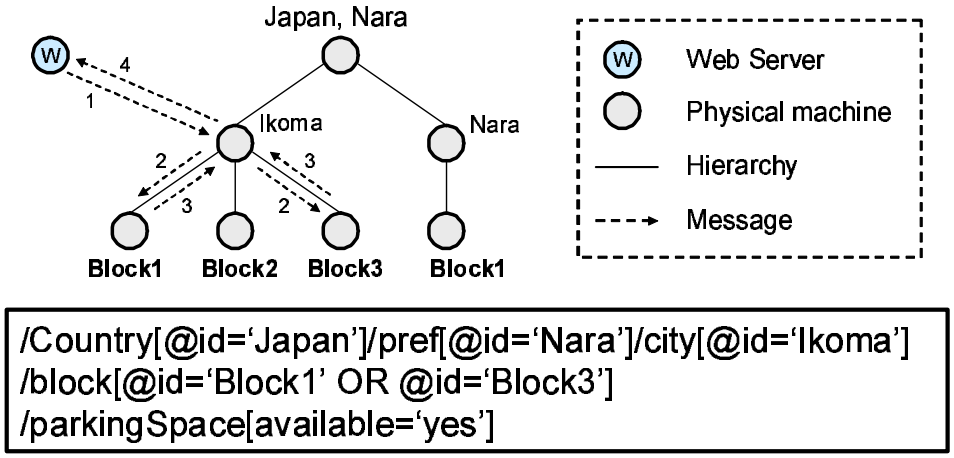


Figure 2.6: XPath query support on IrisNet

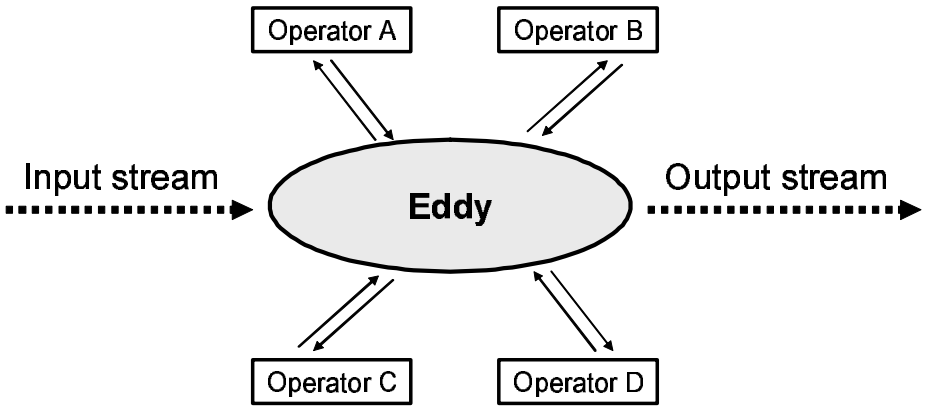


Figure 2.7: overview of Eddy

queries [69]. This XPath query means searching particular areas (Block-1 and 3 in Ikoma-city, Nara, Japan) for a parking space. A web server can process XPath queries by following the path tree of OAs.

GSN: Global Sensor Networks

GSN (Global Sensor Networks) aims at providing a flexible middleware having simplicity, adaptability, scalability and light-weight implementation [1, 2]. In order to realize efficient abstraction, GSN defines virtual sensors as interfaces by XML documents. Virtual sensors manage stream of sensing data, and one virtual sensor corresponds either to a data stream received directly from sensors or to a data stream derived from other virtual sensors. GSN manages sensor data streams based on virtual sensors. This mechanism managing sensor streams is related to TelegraphCQ's [11].

TelegraphCQ is one of DSMSs (Data Stream Management Systems) and consists of two main components. First component called CACQ (Continuously Adaptive Continuous Queries) [44] is an extension of the Eddy [5] and SteMs [63] mechanisms to support multiple continuous queries. Second component called PSoup [12] supports access to previously-arrived data, intermittent connectivity and disconnected operation. Eddy is fundamental mechanism to construct other mechanisms. Figure 2.7 shows the brief overview of Eddy. Unlike conventional

query plans (e.g., processing SQL statements on DBMSs), Eddy invokes operations by availability of queues or probability of selection on each operator. Using this technique, Eddy improves the performance of processing data streams.

Using TelegraphCQ mechanisms, GSN can flexibly manage sensor data stream based on virtual sensors. Users can access target sensing data easily through a virtual sensor or combined virtual sensors.

2.4.3 Peer-to-Peer technologies

A server-client model is most popular system for providing services through the Internet. Nowadays, every computers connects to the Internet and the cost of server rapidly are increasing day by day. In addition, centralized system like client-server model has a single point of failure. Thus, lots of decentralized and self-organized system are proposed. This subsection describes distributed mechanisms categorized for some categories, focusing peer-to-peer technologies.

DHT: Distributed Hash Table

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 lots of studies, and fundamental studies of DHTs are Chord [67], CAN [60], Pastry [65], Tapestry [71] and Kademlia [47].

Database management systems (DBMSs) create index of data in order to improve the performance of retrieval. One method of creating index is *hash table*. A DBMS manages indexes of hash table by itself. On the other hand, DHTs manage hash table by distributed nodes, as depicted Figure 2.8. Each node of DHT manages a part of the whole hash table, and this range of hash table which a node should manage is determined by ID of a nodes. During a process of joining in DHT networks, nodes calculate a random ID between zero and max value of ID space. This ID space is equals to ID space of a hash table. After joining, nodes calculate hashed-IDs of their data, and send these hashed-IDs particular nodes. Figure 2.9 shows the process of sharing hashed-IDs. In this Figure, ID-space of DHT network consists of 100 IDs (0–99). The node of ID-0 manages part of ID-space from ID-0 to ID-29, because ID of next node is 30. Each node calculates

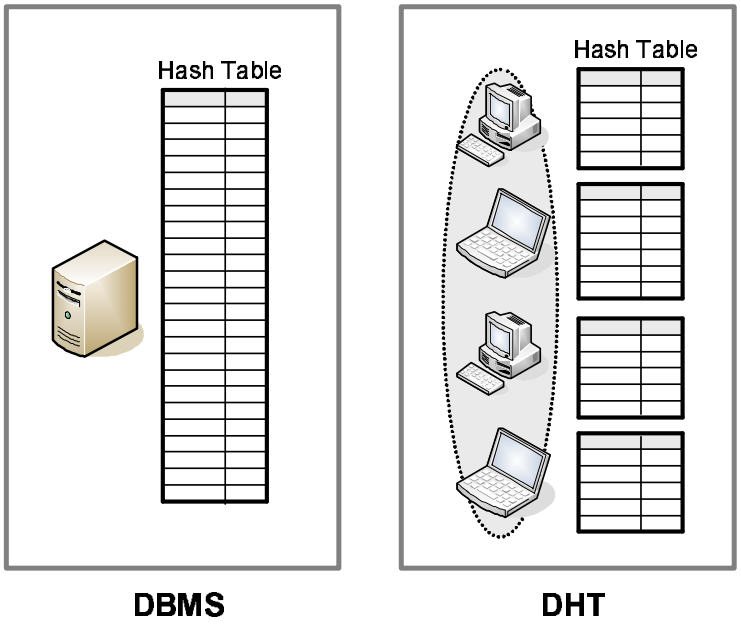


Figure 2.8: Management of Hash Table

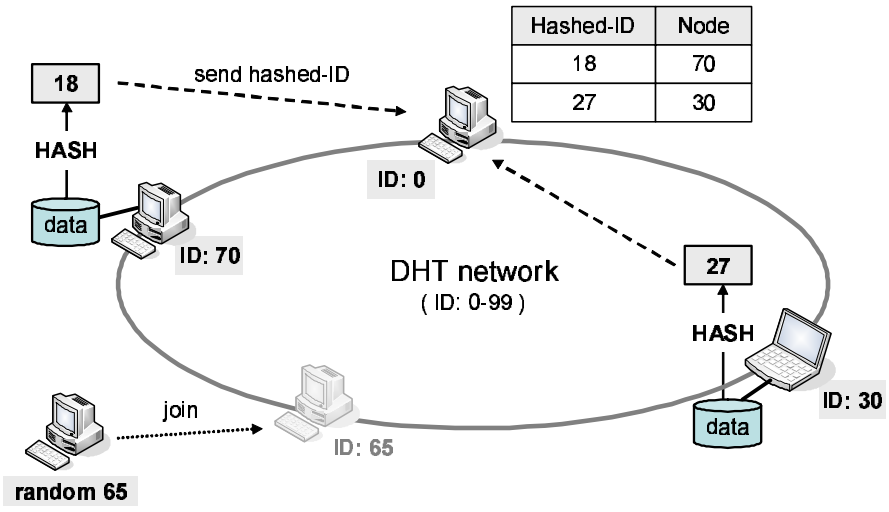


Figure 2.9: Sharing hashed-IDs on DHT

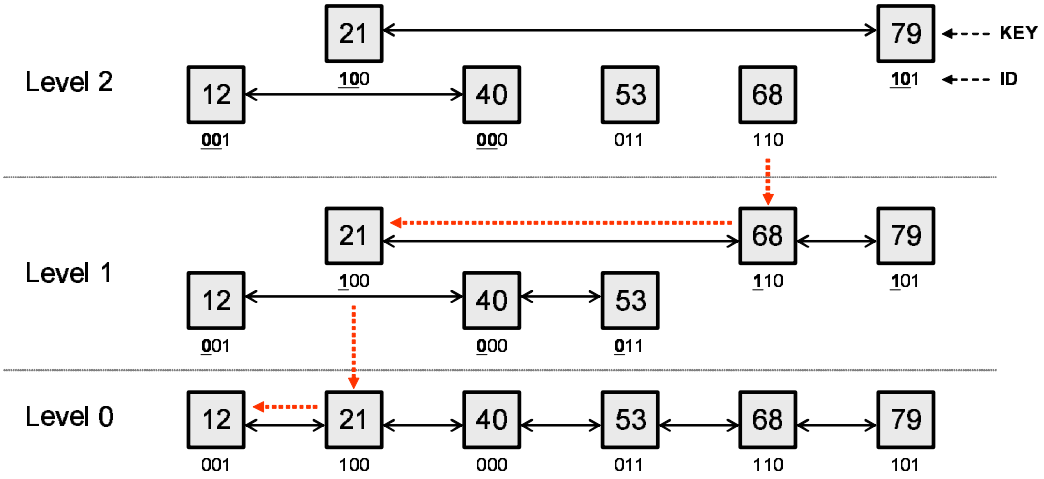


Figure 2.10: routing mechanism of Skip Graph

hashed-IDs from own data and send hash-IDs between 0 and 29 to the node of ID-0. After sharing hash-IDs, users can access particular data by calculating hash-IDs of target data and communicating nodes managing these hashed-IDs.

Each DHT has own routing mechanism based on IDs to improve the performance of search. For example, Kademlia has 160-bits ID-space by SHA-1 [19] and each node manages the distance between own node and other nodes. This distance is calculated by digit number of own ID XOR other node's ID, and managed as tree topology called k-buckets. By using k-buckets, users can search for target data via $O(\log N)$ messages.

DHTs construct structured overlay networks by hashed-ID and are scalable for the number of nodes and support fast search. Consequently, DHTs provide distributed environment for several Internet-scale applications.

P2P networks supporting Range Queries

DHTs are efficient for exact much queries but are not well suitable for range queries, because hash mechanism destroys the ordering of data. Several studies therefore focus on range queries.

Skip Graph [3] is one of structured overlay networks supporting range queries. Skip Graph adopts Skip Lists [59] as routing algorithm and nodes have two types

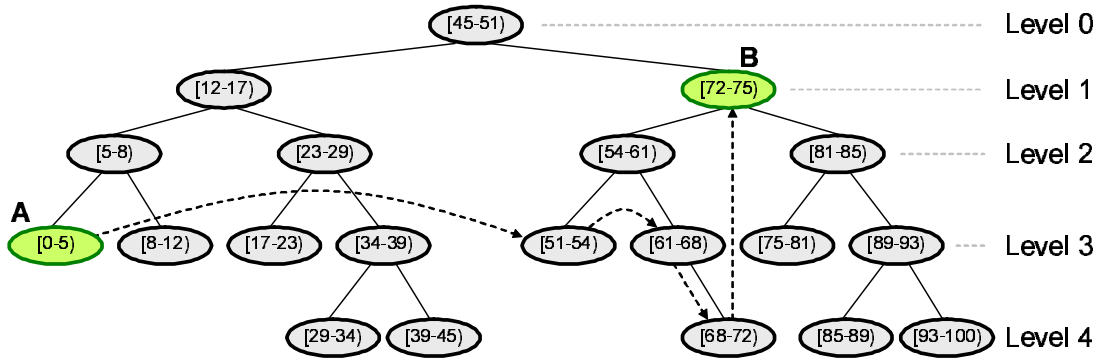


Figure 2.11: Routing mechanism on BATON

of number called 'key' and 'ID'. Figure 2.10 shows routing mechanism of Skip Graph. Nodes of Skip Graph lines up by the ordering of 'key' and create links to other nodes by 'ID'. First, nodes create links to the next nodes at level-0. Secondly, nodes create links to other nodes which manages ID matching prefix of length 1 at level-1. Similarly, nodes create links by matching prefix of length 2 at level-2. After creating links, nodes have the wide range of accessibility to other nodes, because 'IDs' are random numbers and evenly distributed. Figure 2.10 also describes the process of routing from a node (key:68) to an other node (key:12). The node (key:68) can send messages to the target node (key:12) via 2 hops. If users want to get data related with the numbers from 15 to 25, users can get these data from two nodes (key:12, key:21). Skip Graph realizes fast search and supports range queries by managing two types of number. SkipNet [21] adopts the same basic data structure as SkipGraph. SkipNet focuses primarily on the content and path locality properties of the design. On SkipNet networks, nodes store data in neighbor nodes on near subnetted networks in order to realize locality of storing and getting data.

Mercury [7] is more general overlay network than SkipNet, because Mercury supports range-based lookups on multiple-attributes. Mercury independently manages several ID-spaces and each attribute is related with one of attributes. For example there is one car, and this car has some attributes (color=#FFFFFF, price=$30,000, weight=1,500kg). In order to share information of cars, Mercury creates three ID-spaces related with color, price and weight. One ID-space is

managed as like Chord or Symphony mechanisms. Skip Graph and SkipNet can only manage one attribute. On the other hand, in Mercury network users can search cars by color, price or weight. By managing several ID-spaces, Mercury realizes multiple-attributes queries.

In the database world, B-Trees [6] is the central method of creating indexes of data. BATON [32] supporting range queries by managing balanced tree structure. In this tree structure, each node (e.g., lack mounted machine, desktop PC) corresponds to a node of the tree. BATON assign to each node a range of values. The range of values directly managed by a node is required to be to the right of the range managed by its left subtree and less than the range managed by its right subtree. This index structure is based on main-memory index called T-Tree [38]. Figure 2.11 shows the routing mechanism of BATON. Nodes manage several links to other nodes at the same level besides links to parent and child nodes. Nodes can access other nodes via $O(\log N)$ messages, by these links at the same level. If node-A want to search for data related with '74', the query can reach node-B via four messages as depicted Figure 2.11. Compared with Skip Graph and SkipNet, BATON improves load balancing by balanced tree structures and supports range queries. However, the other cost arises from maintaining tree topology. BATON* [33] is an extension of BATON in order to support multiple-attributes queries. BATON* considers the priority of attributes and reduces three attributes of low priority to one dimensional indexes by Hilbert Space Filling Curve [23]. Compared with Mercury, BATON* reduces the cost of managing ID-spaces, because of reduction in the number of attributes.

2.5 Summary

This section describes diverse ranges of ubiquitous sensing environment. It covers sensor network technologies, large scale sensor networks and overlay technologies. Related work is summarized as Figure 2.12 shows.

Sensor network technologies have been focusing on saving energy and data collection. Various protocol are proposed at MAC layer and improve energy efficacy by considering interval of sensing data. In addition, TinyOS, TinyDB and other studies provide SQL to get sensing data easily. These technologies are

	Multiple-attributes search	Scalability	Fast search
Sensor network	○ TinySQL	× Ad-Hoc network	△ Only local area
Large scale sensor network	○ Xpath、VS	△ Manual management of OA trees and VS	△ Lack of resource access mechanism
DHT	× Not consecutive one dimensional ID	○ Distributed management of Data and Index	○ Plaxton, XOR
Overlay supporting range query	× Consecutive one dimensional ID	○ Distribute management of Data and Index	○ Skiplist, B-Trees
Overlay supporting multiple-attributes query	○ Consecutive multi dimensional ID	△ Management of multiple attributes	△ high searching cost (product set search)

Figure 2.12: summary of related work

applied in local areas in which dozens or hundreds sensors exist.

Large scale sensor networks are studied for managing sensing data in Internet-scale networks. IrisNet provides distributed environment for sensor networks by a tree topology of OAs and supports easy access by XPath queries. However, the cost of DNS like hierarchical system is high, because parent OAs should have secondary replicas for continuation of services and ID space should be separated and managed manually. GSN can flexibly manage sensor data stream based on virtual sensors. However, GSN do not equip retrieval mechanism to search global-scale data resources for getting a particular data.

Overlay technologies provide distributed and self-organized environment. DHTs distribute indexes of data closed in databases among lots of nodes. Nodes of DHTs construct logical ID-space in order to support fast search and realize a scalable network. In addition, DHTs can process queries even if the network is continuously changing. However, there is a serious disadvantage. DHTs support only exact match lookups, because hash mechanism destroys the ordering of data. As the result, DHTs can not support range queries. Skip Graph, BATON and other

overlay networks are proposed. These overlay networks realize supporting range queries besides advantages of DHTs.

Mercury, BATON* and other works also support range queries and, moreover, support multiple-attributes queries. These overlay networks manage several ID-spaces related with attributes. A multiple-attributes query is processed in stepwise. These overlay networks can not process the query consisted of some attributes at one time, because each attribute is independent from others. If user search geographical square region in Mercury network, user have to search a large region, because square region is determined by two attribute (latitude and longitude). Similarly, the cost of searching geographical region is high in BATON* network.

In ubiquitous sensing environment, sensing devices are distributed anywhere, and the system managing these huge data should provide retrieval mechanism by which users can search a global area for particular data. Consequently, a distributed mechanism supporting geographical range queries is required. It is necessary to construct an alternative overlay network, which can provide efficient geographical range queries besides multiple-attributes.

Chapter 3

A geographical location based overlay network for sharing global-scale sensing data

To meet the requirements in Chapter 2, we propose a new P2P network system called "Mill" considering geographical location where information is generated. This Chapter 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.

It is assumed 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.

3.1 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.

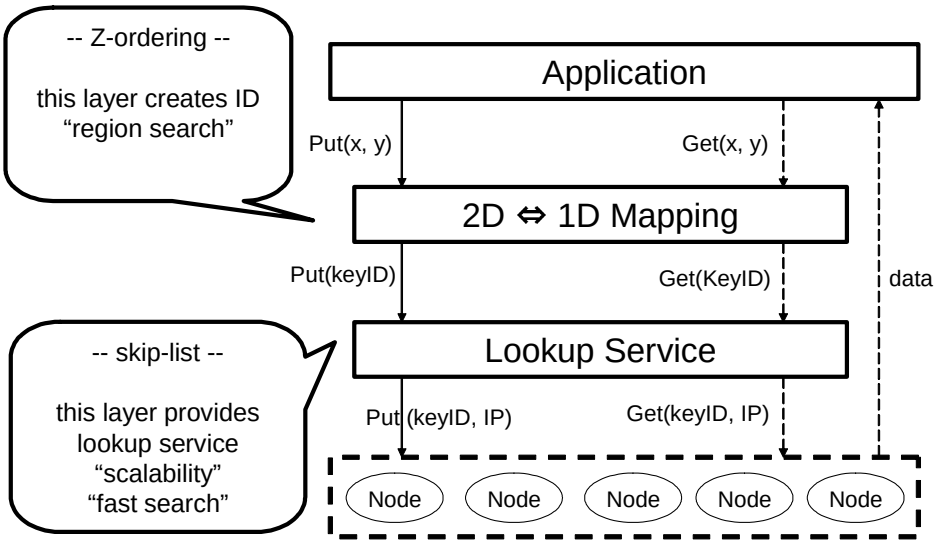


Figure 3.1: architecture of Mill

DHTs support only exact match queries because of adopting a hash function (e.g. SHA-1 [19]). 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” [50], which represents square surface of the earth. This difference enables Mill to support geographical range search.

As Figure 3.1 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.

3.2 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 earth, a size of grid cell is equals to milli-meter order. As Figure 3.2 shows, suppose that each cell is assigned a 4bit 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 Figure 3.2 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

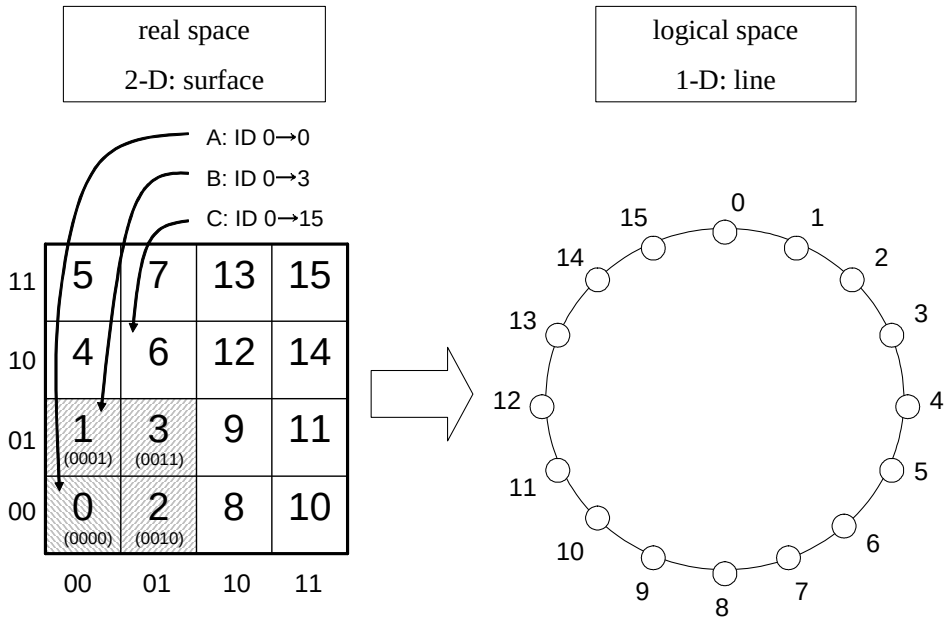


Figure 3.2: 2D-1D mapping method

3.3 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 Figure 3.3, 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 3.4 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 Figure 3.4 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:

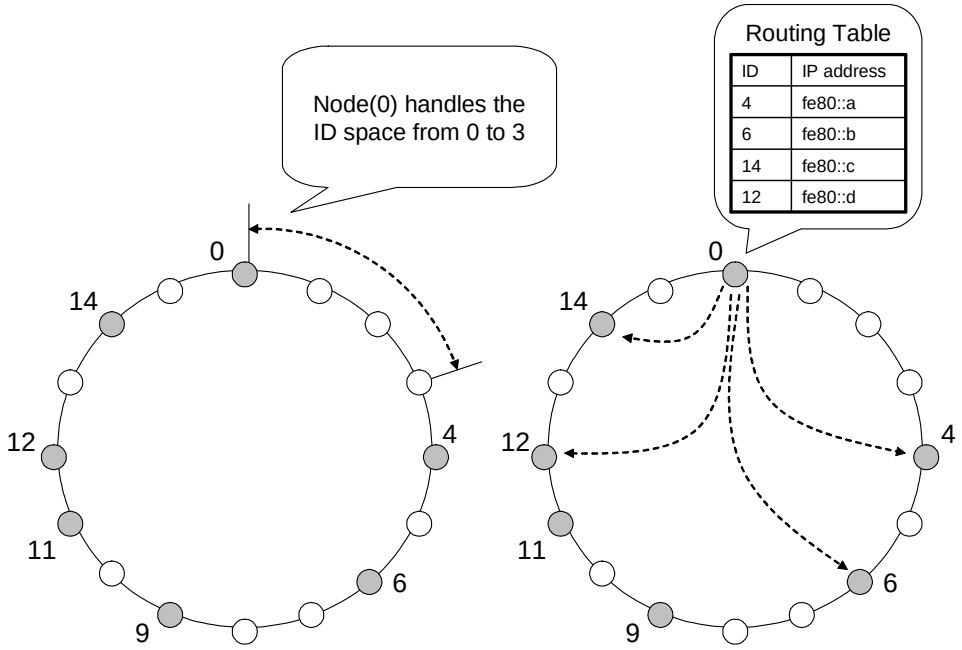


Figure 3.3: handle ID-space and routing table

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. 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

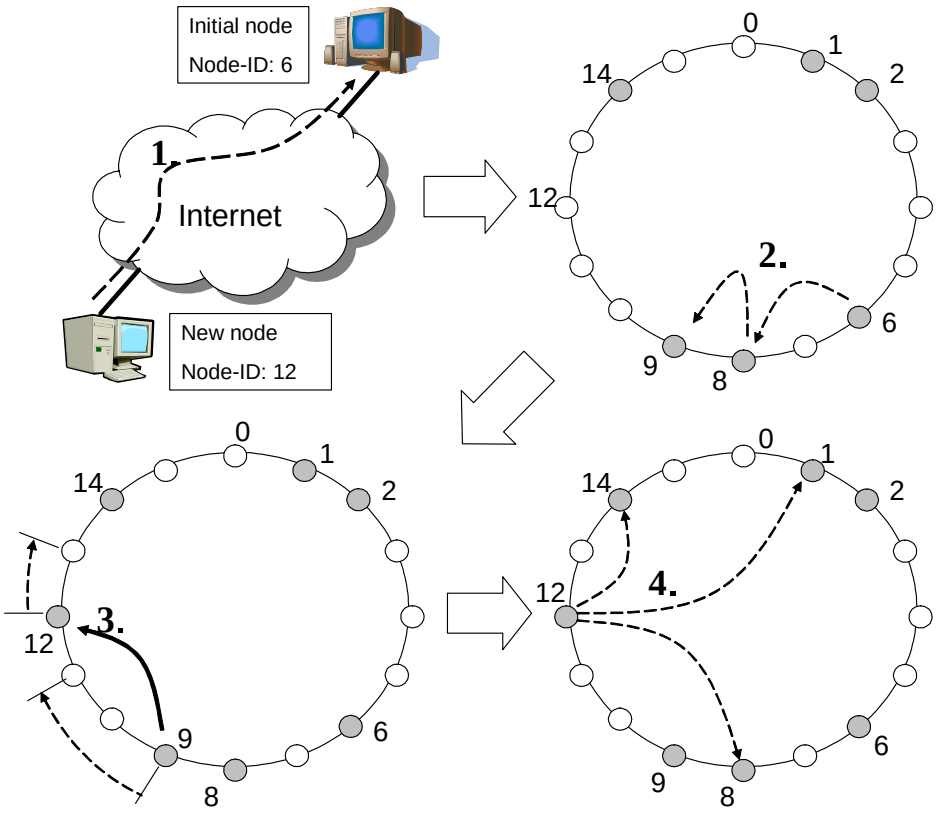


Figure 3.4: join protocol

(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.

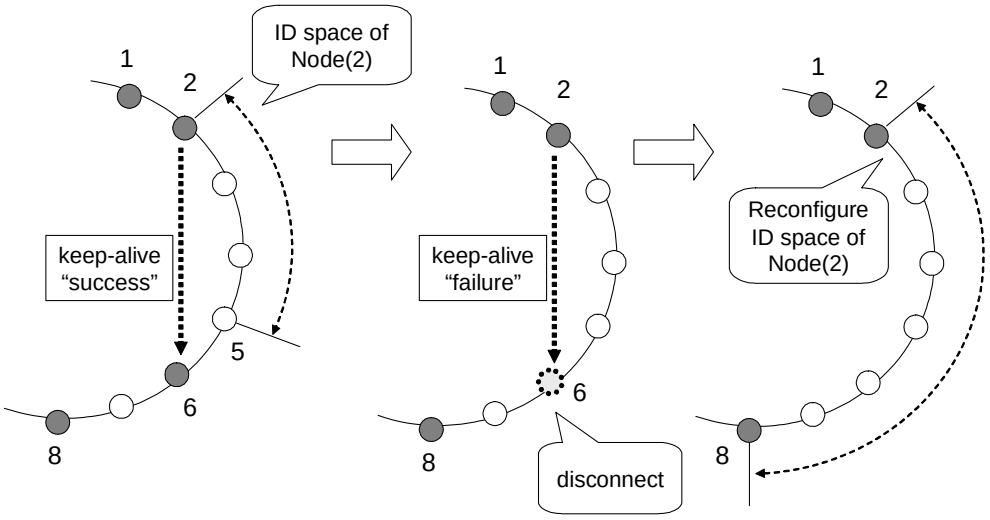


Figure 3.5: leave protocol

3.4 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 3.5 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

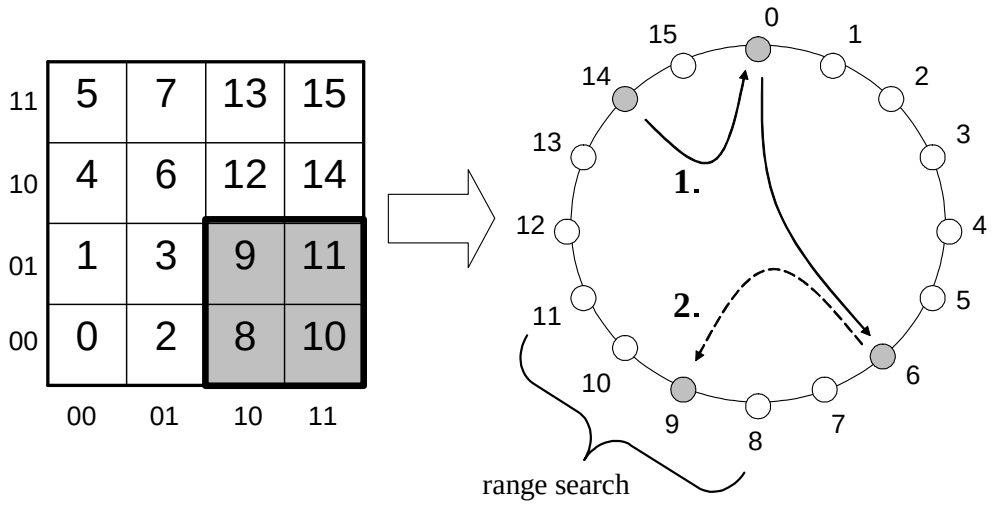


Figure 3.6: region search

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 above 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.

3.5 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 algorithm. Second, this node sends the ID and its IP-address to clockwise side node. And the 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 3.6 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.

3.6 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 3.7 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.

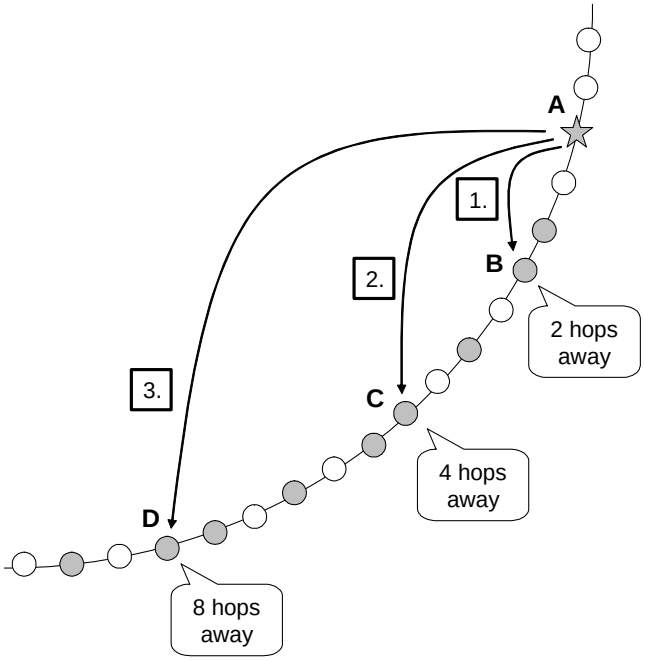


Figure 3.7: operation for creating routing table

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.

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 **??** 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

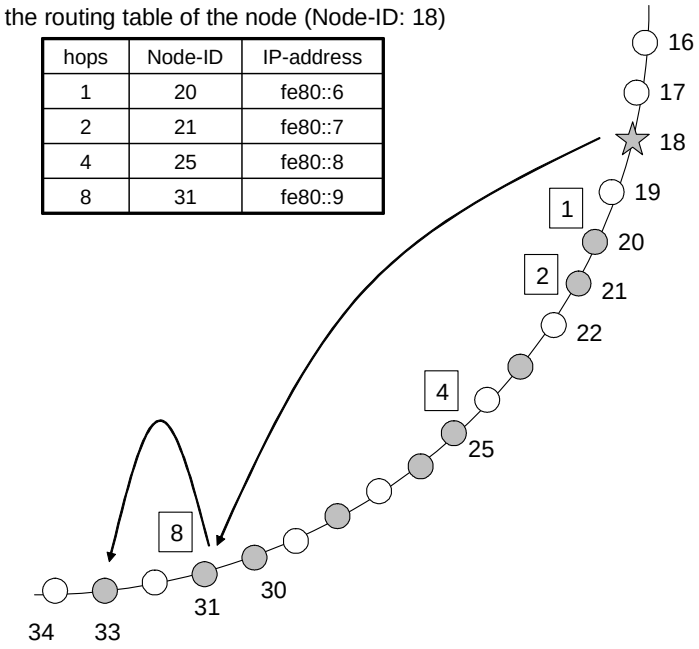


Figure 3.8: skip-list search

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 contrary 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

Table 3.1: simulation environment

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

ID-space in Mill network is effective for load balance.

3.7 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 particular node needs intensively processing many queries.

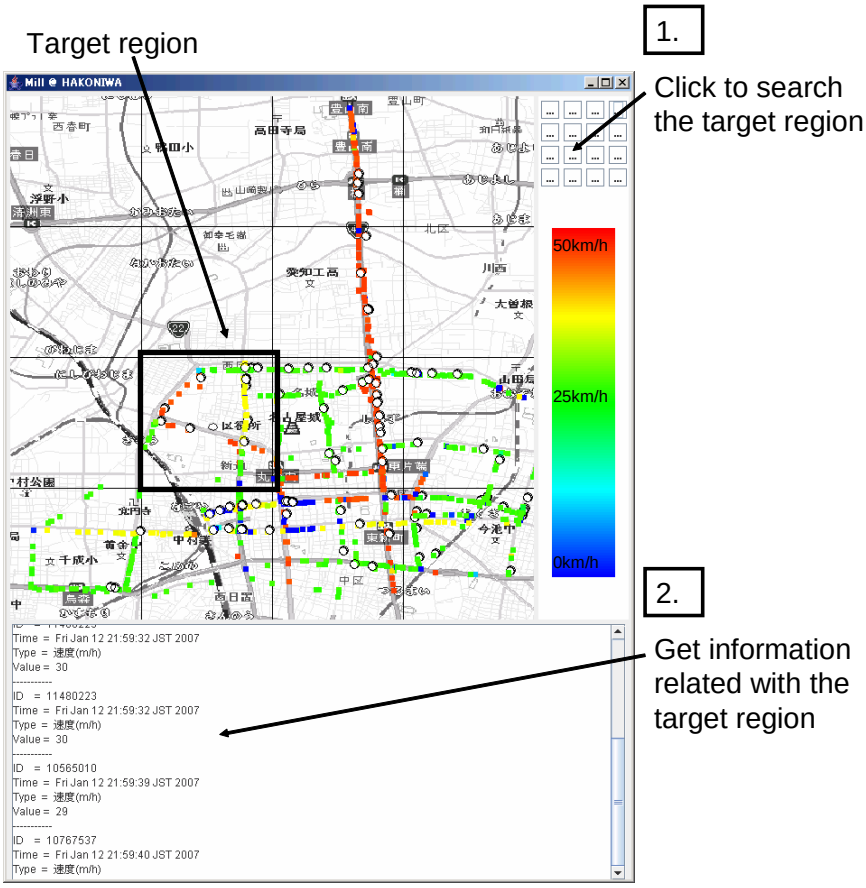


Figure 3.9: Application Example

3.8 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" [68] 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.

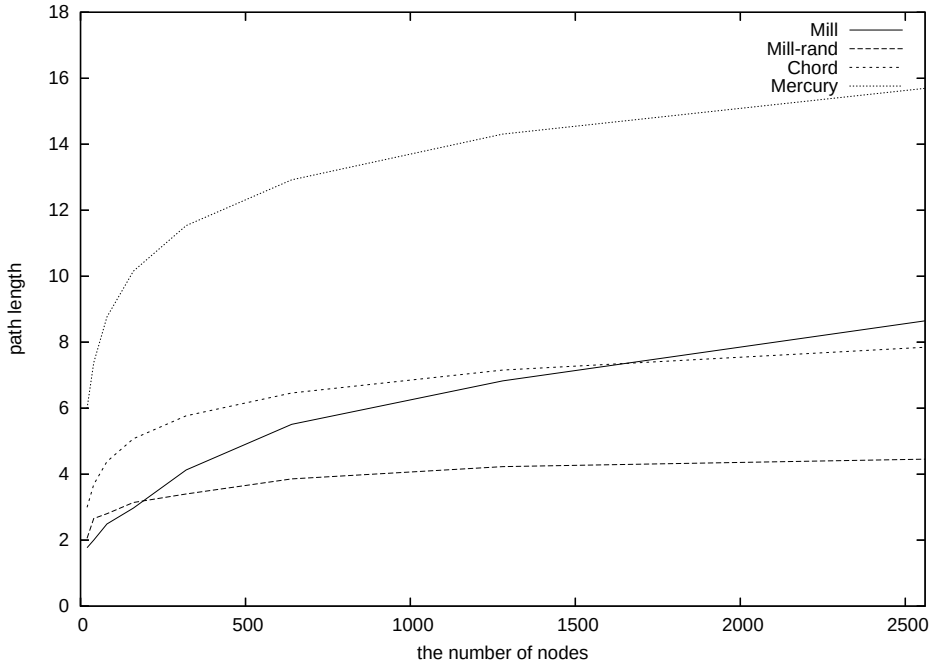


Figure 3.10: node vs path length

3.8.1 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 3.9 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 50km/h on the north-south highway.

3.8.2 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 Figure 3.10 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 limited 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" [68]. 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 the difference in density of nodes within the ID-space. If it is assumed that the Round Trip Time (RTT) of mobile phones is about 500 milli-seconds, queries reach destination nodes within 3 or 4 seconds on Mill network.

Secondly, we compare the performance attributed to data structure on each overlay network. As Figure 3.11 shows, we express the whole ID-space as 64 (32 + 32)bits. It is assumed that the density of nodes is uniform. And it is expressed that 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 particular 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 $2 \log N + 2m \times 2^{32}/2^{i/2}$.

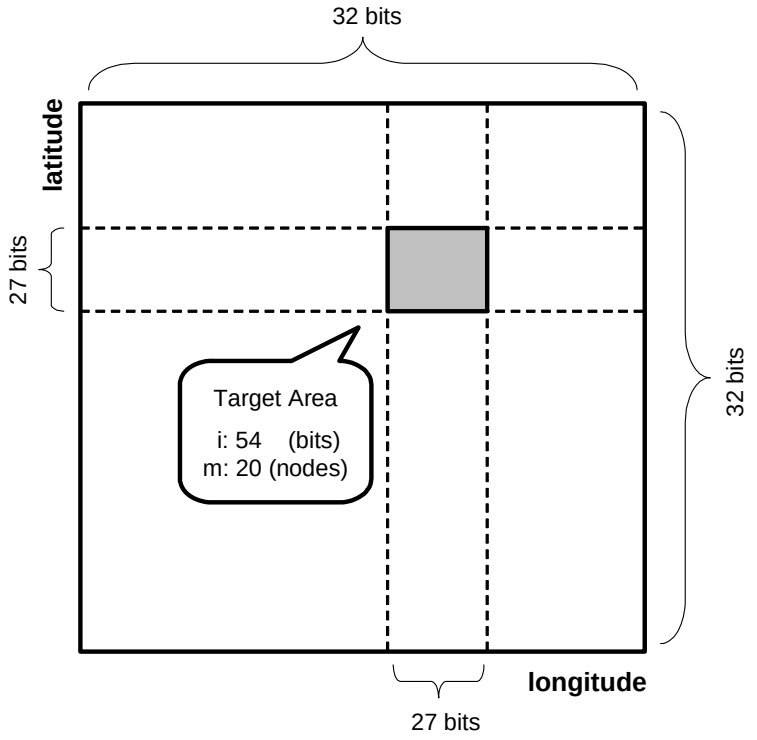


Figure 3.11: example of searching a region

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.

LL-net has hierarchical ID-space to improve search mechanism. LL-net solves range queries 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 ID-space.

Discussion of Z-ordering algorithm

As above section, the advantages of proposed method is discussed. However, Z-ordering is not always useful for searching various types of square regions. Here, detailed features of Z-ordering is discussed.

If users search the regions (parts of ID-space) which is divided by power of 4 bits, users can attain sensing data by just one query. This is the best case of search by Z-ordering. If regions of search are changed longitudinally or latitudinally, the cost of retrieval increases. Figure 3.12 shows changes of search regions, and Figure 3.13 shows the numbers of queries and hops related with the difference from the best case. It is assumed that the density of nodes and size of search region are the same as Figure 3.11 shows.

In Figure 3.13 "long" means longitudinal direction, "lati" also means latitudinal direction, "hop" means the number of hops and "query" means the number of queries. Misalignment from the best case decreases retrieval performance. However, as Figure 3.13 shows, this disadvantage is not serious, because the higher limit of retrieval cost is fixed (3 queries, 54 hops) by searching a little large region (1.5 times the best case) including a target region. A diagonal misalignment causes the worst case, and 2.5 times region compared with the best case is searched. Then, Mill network should process 4 queries and 78 hops in higher

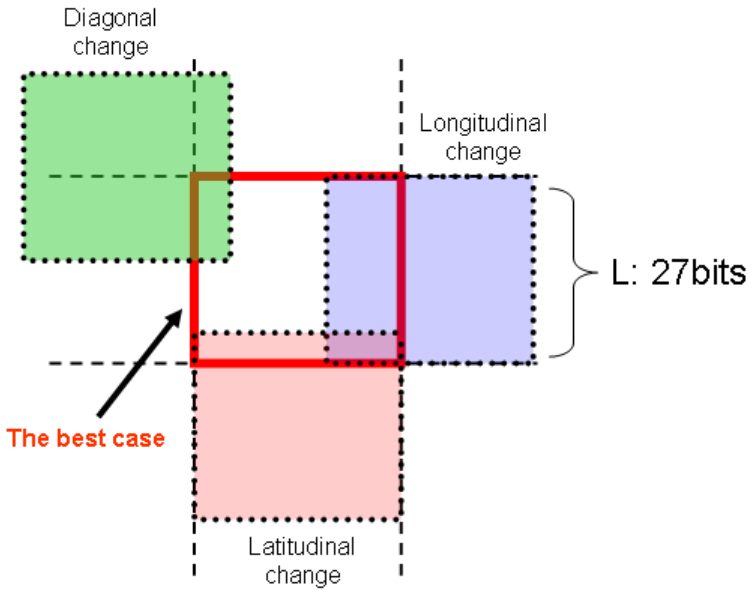


Figure 3.12: changes of search region

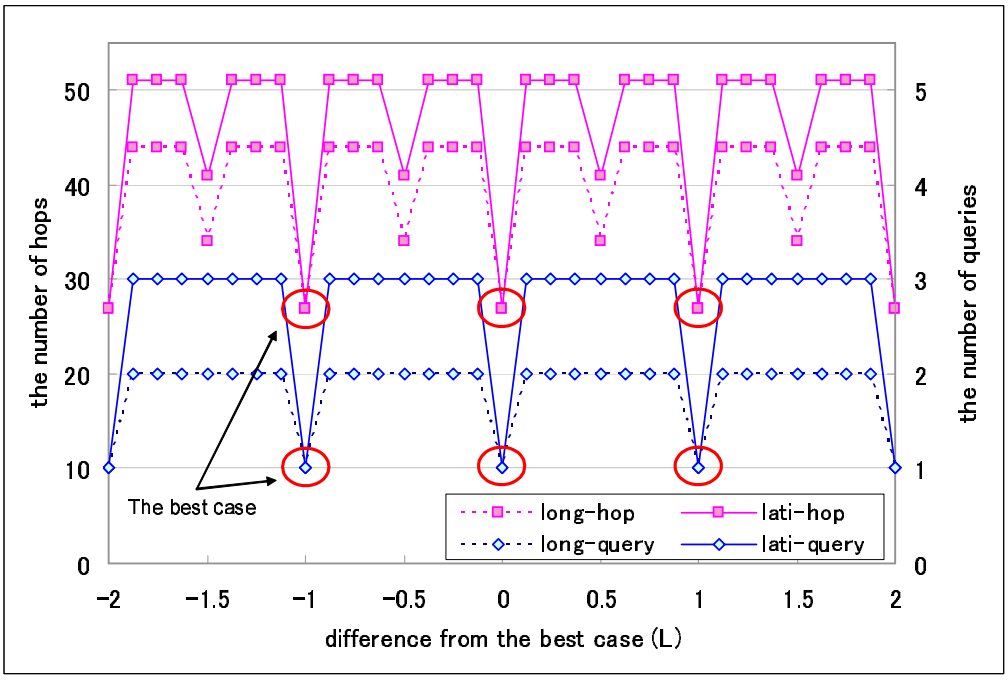


Figure 3.13: features of z-ordering (changes of regions)

limit.

On Mill network, misalignment of search region causes degradation of search performance, however compared with DHTs, Mercury and other overlay networks, this disadvantage is not serious. If multi-scale grid interface whose one grid cell represents the best case is provided, users can change the scale and pick up several target grid sells. This user interface causes little restriction and Mill network always processes queries of searching regions consisting of the best cases.

On the other hand, latitudinal or longitudinal increase of search region causes serious effect on the retrieval performance. If search regions are increasing as Figure 3.14 shows, the number of hops and queries are also increasing as shown in Figure 3.15. The larger a target region becomes, there are more nodes in it. For this reason, increase of search region basically causes degradation of retrieval performance on any overlay networks. To make things worse, on Mill network, retrieval performance becomes more diminished than other overlay networks, because elongated rectangle region is divided into squares (the best case) and some queries which is the same number of squares are processed. As the Figure 3.15 shows, the longer one side of rectangle becomes, the worse retrieval performance becomes, and the retrieval advantage of Mill is gradually diminished. However, there is a method preventing from degrading retrieval performance. Each query which is related with square region is independent from other queries, and can be processed concurrently. Consequently, some queries generated to search elongated rectangle region can be independently and concurrently processed by mechanism of threads programing. In addition, considering that users usually search square or circular region and rarely research extreme elongated rectangle region, Z-ordering is suitable for almost cases.

3.8.3 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 Figure 3.16, “Hakoniwa-stable” means that IDs of nodes are determined by HAKONIWA

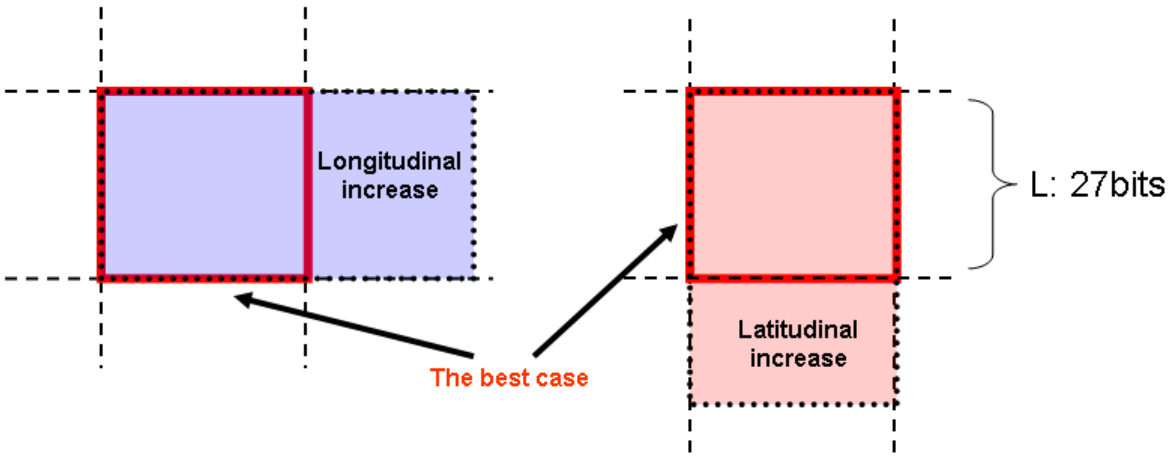


Figure 3.14: increase of search region

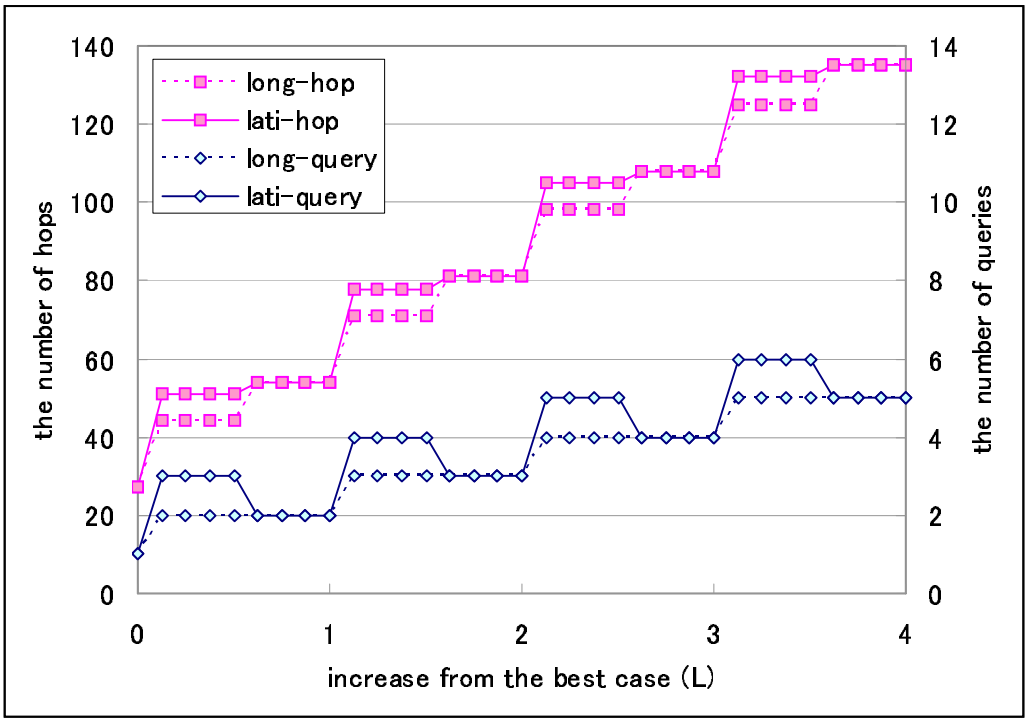


Figure 3.15: features of z-ordering (changes of regions)

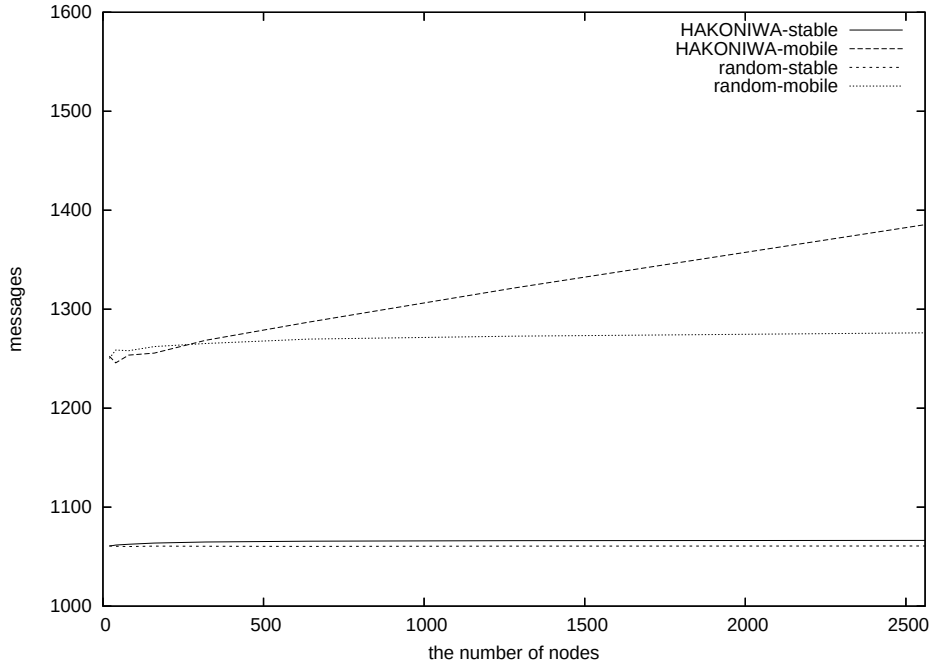


Figure 3.16: nodes vs messages

(traffic information generator) and nodes don't often join and leave from overlay network because the status of link is stable. It is assumed 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, it is assumed 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

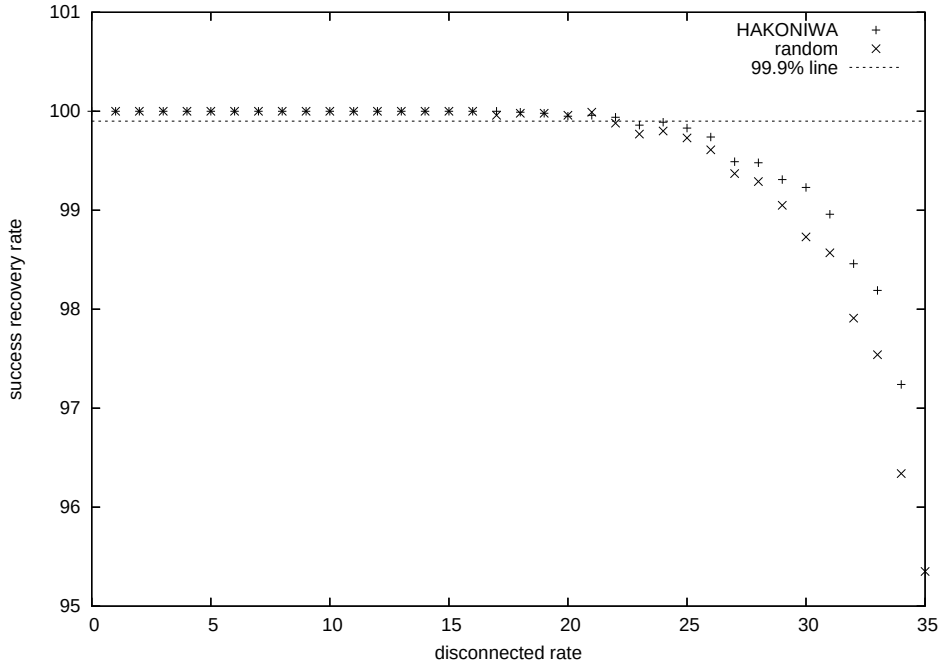


Figure 3.17: disconnected vs recovery

data generated by mobile devices.

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

3.8.4 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 Figure 3.17 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.

3.9 Related Work

There are several P2P networks considering location. However, these P2P networks have some defects in managing location-related information. GHT[61] is a location based P2P network. This P2P network defines an area as a square region divided by latitude and longitude. GHT adopts a routing algorithm based on GPSR[34]. GHT supports region search. However the performance of search is not good and this network is not scalable.

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 particular range on one dimension. However, users have to search a large region to get information of a particular geographical region, because each attribute is independent from others in Mercury networks. It is difficult to handle sensing data based on geographical location in these overlay networks.

In GHT network, users can search two dimensional surface for target regions. However, the performance of search is low ($O(\sqrt{N})$) and fixed size of grid disturbs multiscale geographical search. And GHT dose not consider event driven behavior and feasibility of deployment.

3.10 Summary of Mill algorithm

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 addition, 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 square region. By this method, users can search any size of square

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.

Chapter 4

An implementation methodology of geographical overlay networks suitable for ubiquitous sensing environment

4.1 Introduction

Today's sensing devices such as weather sensors, cars, and other devices become powerful. In addition, these devices have connectivity to the Internet and equip positioning devices such as GPS sensors. It is expected that the number of these devices become larger and larger. In this environment, these sensing devices can immediately collect and provide information anywhere and anytime. We call this environment *ubiquitous sensing environment*.

If we use a large number of information these devices provide, we can obtain detailed and up-to-date information. And these huge data can be applied for lots of applications. Gathering information based on geographical location can be efficient for environmental problems, educational material, businesses and our daily lives. For example, if we can gather exact torrential rainfall information in some region, this information is useful for evacuation instructions. And if we can examine distribution of temperature, we apply this information to solve the

problem of heat-island effect.

In ubiquitous sensing environment, sensing data is generated anywhere and anytime. To manage and provide a large number of sensing data, we have been developing geographical location oriented overlay network called "Mill"[46].

In this Chapter, we discuss the implementation of "Mill", which is suitable for ubiquitous sensing environment. In this environment, queries from users and sensing devices have some characteristic patterns. Recent years, we have installed weather sensors and share these information on Live E! project[41]. In Live E! networks, almost users constantly get weather information of the same region, because users want to know weather conditions around their home, office or other particular places. And sensors constantly generate weather data of the same region, because almost sensors are equipped on a particular location.

Mill provides distributed environment based on geographical location and supports multiscale region search by $O(\log N)$ hops. In addition, the number of queries of users and sensors are minimized by iterative routing in Mill network. And Mill supports an event driven mechanism on geographical overlay network.

The rest of this Chapter is structured as follows. Section 4.2 describes requirements in ubiquitous sensing environment. Section 4.3 shows the related work. Section 4.4 and 4.5 present the mechanism of Mill and explain its implementation. Section 4.6 describes evaluations of Mill's implementation and its shows how many users and sensing devices Mill can handle. Finally, we summarize our contribution in Section 4.7.

4.2 Requirements for ubiquitous sensing environment

In ubiquitous sensing environment, there are lots of devices including weather sensors, web cameras and other sensor devices. To manage these huge number of data, we have proposed an overlay network called "Mill", which supports "scalability", "multiscale region search" and "fast search". However, there are several requirements that we have to cope with.

- handling repetition of similar queries

Sensing devices such as weather sensors constantly generate data and users need to constantly get these data to know the current status. For example, users check current status of temperature, wind speed or other weather information. On existing overlay networks, users need to search overlay network each time they want to get data. In this case, relaying nodes should pass many same or similar queries from one node to another. This operation wastes time and network resources.

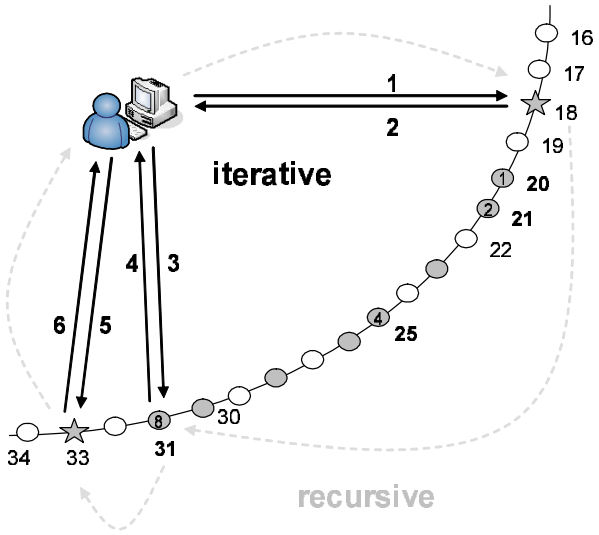
If overlay networks support that users in a particular region can directly access to data generated by sensing devices in the same region, it is possible to reduce the almost all relaying queries in overlay networks.

Systems managing ubiquitous sensors should consider the characteristics of query patterns from users and sensing devices.

- supporting event driven behavior
  There is a requirement that users want to receive an alert message. For example, users want to receive a message when it is starting to rain. To realize these alert services, a polling method is useful, but it might waste network resources. To save network resources, an event driven method is required.

  On existing overlay networks, users should install some special softwares to get data from overlay networks. If some nodes pushing alert messages can know some users information in advance, special softwares enable users to receive alert messages when a particular event occurs. However, the setup cost of installing special softwares restricts the number of users. Sensing data, such as temperature and rainfall, are highly public data. In order to enable lots of users to use such public sensing data, overlay networks should support an event driven method without special softwares.

- considering flexibility of system operation
  In overlay networks, using some special software restricts the number of users and makes operation difficult. And in ubiquitous sensing environment, the number of sensing devices becomes larger and larger. To adapt this situation, overlay networks must consider flexibility of system operation



the routing table of the node (Node-ID: 18)

hops	Node-ID	IP-address
1	20	fe80::6
2	21	fe80::7
4	25	fe80::8
8	31	fe80::9

Figure 4.1: iterative routing

which includes easy installation, easy operation and flexibility for user's environments.

4.3 An implementation toward characteristics of ubiquitous sensing data

To meet the requirements in section 4.2, we implement Mill based on HTTP, because HTTP meets these requirements and additionally has some good features for ubiquitous sensing environment. This section describes implementation of Mill.

4.3.1 Handling repetition of similar queries

We have operated Live E! network, in which weather information mainly is managed [51]. In this sensor network, users tend to search the same place to get constantly current status of that place, since people are interested in situation of their home or office. And sensing devices constantly measure data of the same place, because almost sensors are equipped on a particular location. Consequently

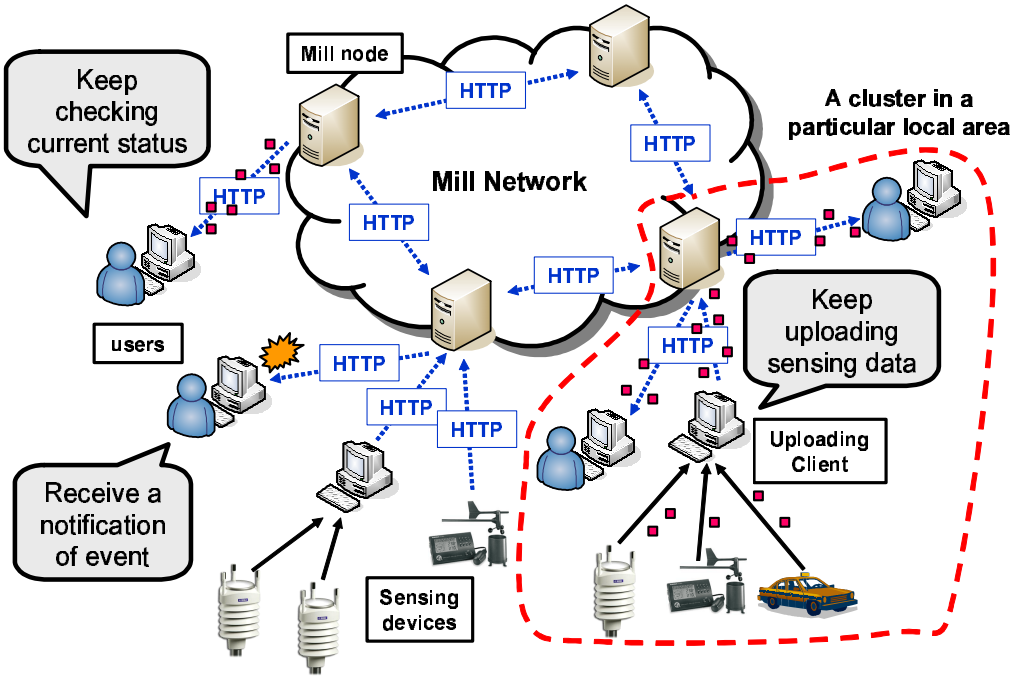


Figure 4.2: Mill network based on HTTP

similar or same queries for uploading and downloading sensing data are frequently generated. In this case, relaying nodes should pass many similar or same queries from one node to another on an overlay network. This operation wastes time and network resources.

To handle these queries, we adopt skiplist with iterative routing on our overlay network. Figure 4.1 shows a process of iterative routing. Unlike recursive routing, by using iterative routing, users communicate with several relaying nodes until a query reaches up to a target node. After reaching a target node, users can cache routing information (e.g., IP-address, ID-space) of the target node and its neighbor nodes, because of features of skiplist. The routing table made by skiplist enables a querying node to especially communicate with neighbor nodes of a target node.

Figure 4.2 shows that users and sensing devices communicate with nodes of Mill. In Mill network, sensing data are managed based on geographical location by nodes. Consequently, users constantly communicate with the same node, if

they want to know the current status of a particular region. Caching information created by skiplist with iterative routing enables queries and sensing data to dynamically being clustered in a particular local area, and nodes and sensing devices can directly connect to a target node. In addition, this cache is effective to search adjacent area of the target node. And sensing devices also directly upload data to the target node in the same way. Caching mechanism by iterative routing optimizes the path of queries from users and sensing devices. For this reason, the number of relayed queries can be significantly reduced. And data related with a particular region are automatically gathering at a particular node. Like DBMSs, methods of creating index are efficient for multiple-attributes search if data are managed at one place. In fact, on Mill network, first of all users must search by geographical location. This restriction enables to support multiple-attributes queries. After searching by geographical location, users reach to a target node, and can search by other any attributes and get data from the target node.

On ordinary overlay networks sharing files, queries are relayed from a node to another node, and after retrieval users download target files. In these cases, relayed queries do not cause serious problems, because basically size of target files are several MBytes, sometimes several GBytes and download cost is much larger than retrieval cost. In other words, one data of file have some values and usefulness. On the contrary, one sensing data hardly has some values, because each sensing data is very small and lacks of reliability. Gathering data by time, geographical location or other attributes, sensing data become valuable data. To get valuable sensing data, lots of queries are generated and these queries cause some traffic on overlay networks. Like DHTs and other overlay networks, if every queries are relayed among nodes, additional large traffic causes constantly. As Section 3.8.2 describes, management of ID space has great effect on the performance of retrieval, and in addition, it is important to consider query patterns and features of data.

Users or sensing devices communicate with nodes on Mill network through HTTP. HTTP is suitable for an iterative routing, because HTTP basically consists of one request and one response. The request and response corresponds to query and reply of iterative routing. Users easily get sensing data on Mill network by some web browsers or other HTTP clients, without installing special

software. In addition, iterative routing is efficient for avoiding performance reduction caused by time-out. Recursive routing is fast, but querying clients can not know relaying nodes. For this reason, if time-out happens, a querying node can not do anything. On the contrary, by using iterative routing, clients can communicate with all relaying nodes. A querying node therefore can find out nodes causing time-out and change routing path, picking up an other node by own routing table.

4.3.2 Supporting event driven behavior

There is a requirement that users want to receive an alert message. For example, users want to receive a message when it is starting to rain or temperature gets over 35 degrees Celsius. To realize these alert services a polling method is useful, but this method might waste network resources. To save network resources, an event driven method is required.

To realize an event driven method, we apply a data pushing mechanism called "Comet" on an HTTP server. A server-side application does not respond to the requests from client-side applications, and the connections between a server and clients remain until the event occurs. When the event occurs, the server pushes the data to the clients through the already-established connection.

On existing overlay networks, users should install some special softwares to get data from overlay networks. This setup cost restricts the number of users. On Mill network, users easily can receive event driven messages as well as getting sensor data without installing particular softwares.

4.3.3 Considering flexibility of system operation

In ubiquitous sensing environment, the number of sensing devices becomes larger and larger. To adapt this situation, overlay networks must consider flexibility of system operation which includes easy installation, easy operation and flexibility for user's environments.

To realize these requirements, we adopt HTTP as an underlying communication protocol of "Mill". Using HTTP, we can define services for responding users and sensing devices with URLs. Mill nodes have own software version, and URLs

Table 4.1: environments of evaluations

CPU	AMD Opteron252 x 2
Memory	DDR400 4GB
Hard Disk	Ultra320 SCSI 10000rpm 150GB
NIC	Gigabit Ethernet (PCI-X)
OS	Linux 2.6.17-10 (ubuntu 6.10)
Language	Perl 5.8.8
Data Base	SQLite 3.3.5

are managed by software of Mill in each version. As software of Mill is upgraded, URLs basically increase with new services. Accessing to a node of Mill, Users get lists of services (URLs) and its explanations with some web browsers. If users want to receive some services, they just specify URLs. For this reason, users can receive services without special software. HTTP is one of the most popular protocol and its interconnectivity is high between different sites. Adopting HTTP enables operation of overlay networks easy. In addition, we use lots of software assets for HTTP. For examples, SSL can be used for encrypted communication and redundancy technology for web sites can be used for improving application availability of each node.

Moreover, adopting HTTP also provides flexibility for developer's environments. For examples, if we want to improve a performance of database or add new functions, we can easily exchange a RDBMS or write additional code. The reason is that an interface is represented by a URL and URLs can be handled independently from the internal implementation. In other word, URL hides the internal implementation of software. For this reason, adopting HTTP provides flexibility of developer's environment. However, it is a little difficult to practically exchange databases, because the way of accessing to data is different on each system. We therefore implement an O/R mapper of Mill in order to enable almost RDBMSs to be operable on Mill network. Mill considering easy installation, expandability and flexibility of development, and as the result Mill provides flexibility of system operation.

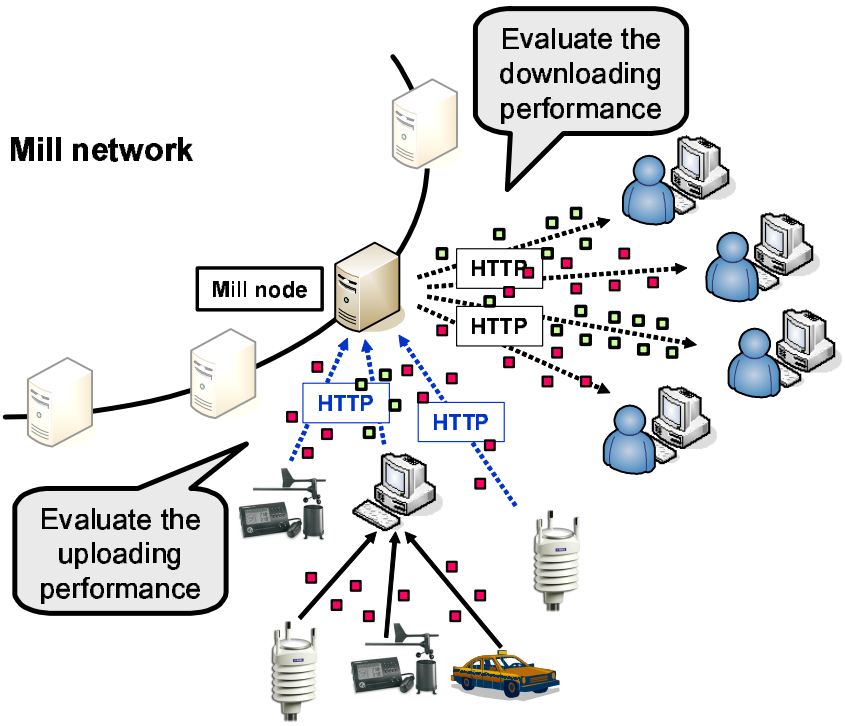


Figure 4.3: environment of evaluations

4.4 Evaluation

In this section, we evaluate the performance of Mill. We have implemented nodes of Mill by Perl. Table 4.1 shows the specification of implementation and experimental environment. We measure the performance of downloading data, uploading data and notification of events. Figure 4.3 shows the environment of evaluations. One node on Mill network consisting of total 10 nodes processes lots of queries from sensing devices or users. Through this experiment, we clarify the performance limitations.

As shown in above section, Mill realizes localities of processing sensing data besides the accessibility to the whole geographical region. On Mill network, some clusters related with particular local areas are dynamically created. One cluster consists of one Mill node and a large number of sensing devices and users, and almost all communications are processed within each cluster. For this reason, if

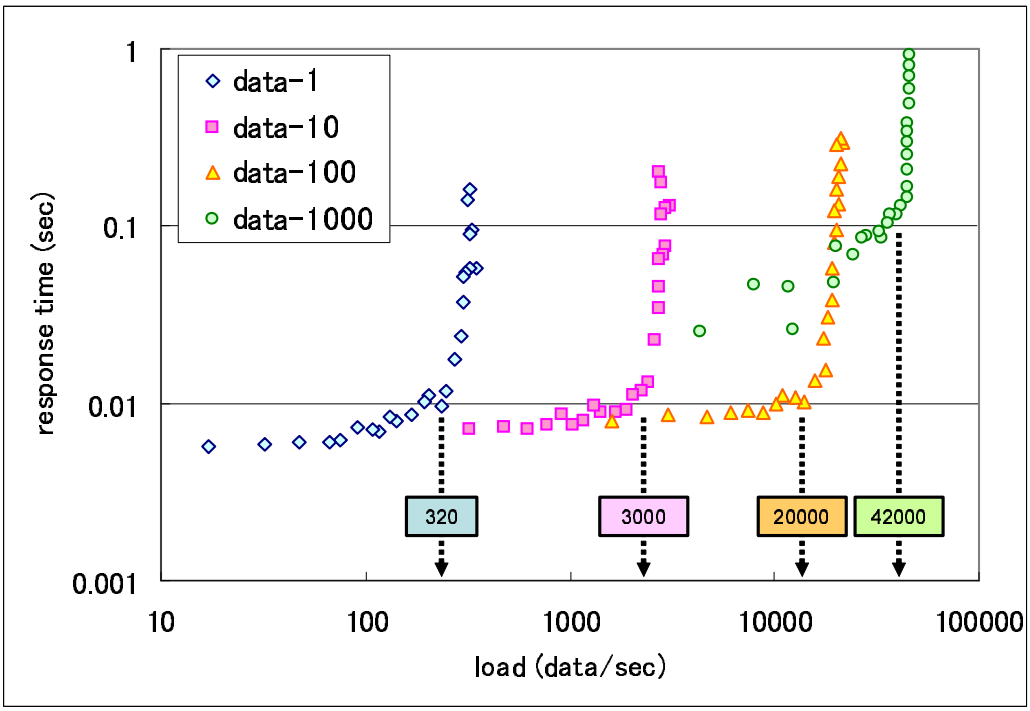


Figure 4.4: Downloading sensing data

we evaluate the performance of one Mill node, the performance of the whole of Mill network is almost clarified.

We have evaluated performance [46] of routing algorithm, messaging cost and robustness in Mill network on HAKONIWA [68]. HAKONIWA is a traffic simulator on which lots of cars are moving around and generating sensing data (e.g., velocity, direction) in a particular city (e.g., Nagoya, Tokyo). On this environment being close to real world, Mill network consisting of thousands of nodes is able to achieve its goals of logarithmic-hop routing, scalability and robustness as an overlay network. In other words, we have evaluated the performance of the overlay network (Mill) consisting of lots of nodes on the simulator. This time we therefore implement nodes of Mill, and evaluate the performance of one Mill node in order to clarify the performance of the whole of Mill network in real environment.

4.4.1 Downloading sensing data

We evaluate a performance of downloading data. As we describe in the above section, users tend to get constantly sensing data of the same place. In most cases, users directly communicate with target nodes. Therefore, we evaluate performance limits of downloading data per node.

Figure 4.4 shows the result of downloading performance. The "response time" means the time from when a user sends a query to a target node to when the user receives the answer of the query. One "data" is consisted of *ID* of sensor, *sensor type*, *time* when data is generated, *Mill-ID* (made from latitude and longitude) where data is generated and *value* of a sensor. The example format (JSON) of one data is as follows. Users can get this data, if users access to URL (http://192.168.0.1/get/data?last=1&sensor_type=Pressure) of a target node assigned "192.168.0.1" as IP-address.

```
{
  "id":1,
  "mill_id":"0xed01944ec1361d9e",
  "sensor_type":"Pressure",
  "time":"2007-09-24T11:06:44+09:00",
  "value":"985.2"
}
```

In Figure 4.4, "data-1" means that users get one sensing data per query. One Mill node can response to approximately 320 queries per one second. And "data-10" means that users get 10 sensing data per query. Then, one Mill node can response to approximately 300 queries per one second. This result shows that one Mill node can process almost 9.3 times the amount of data traffic by merging sensing data, compared with "data-1".

As users want to get larger data, the performance of downloading data become better. On the contrary, response time becomes worse and the efficacy also becomes worse. As Figure 4.4 shows, "data-100" provides better performance, considering both downloading data and response time.

In the Live E![41] network, weather sensors upload current data once a minute. Therefore, if every users want to keep getting current 10 kinds of data, one Mill

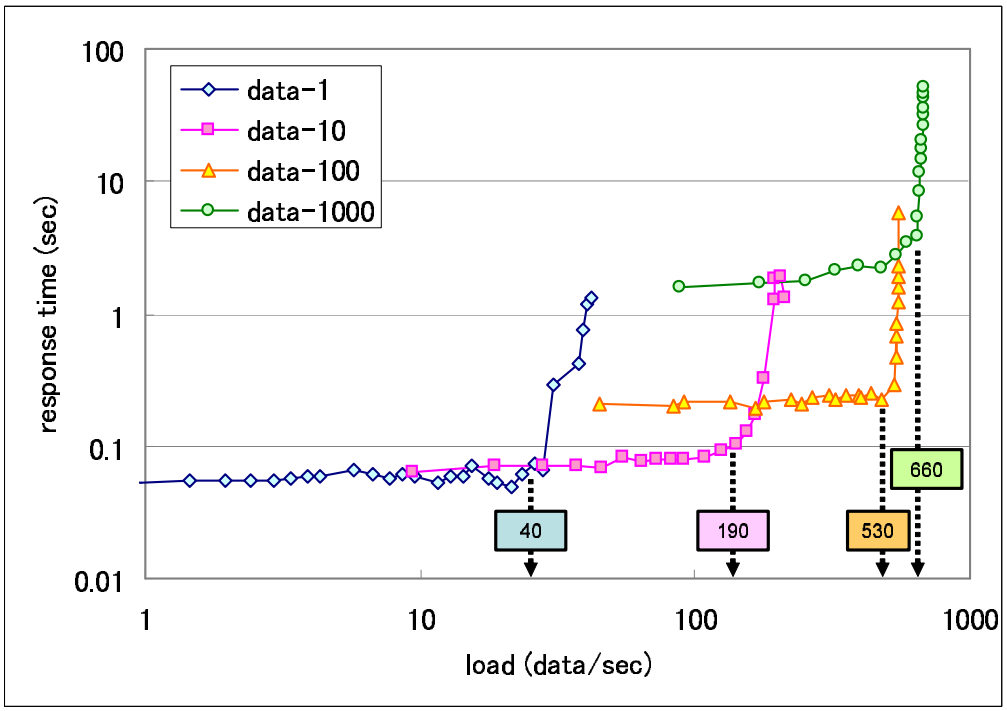


Figure 4.5: Uploading sensing data

node can handle 18,000 ($= 3000(data/sec) \div 10(data) \times 60(sec)$) users on performance limits.

Compared with other overlay networks, Mill reduces almost all relaying queries, because users can directly access to nodes of Mill. If users get certain data on 1000-node network, the cost of search is approximately 10 hops on existing overlay networks. The relaying cost is almost equal to the cost of getting data, because size of sensing data is very tiny. Mill improves the cost of search by approximately 90% on 1000-node network. This improvement is also efficient if sensing devices upload data.

4.4.2 Uploading sensing data

First, we evaluate a performance of uploading data. Sensor devices directly communicate with target node, because almost sensors are equipped on a particular location. For this reason, we evaluate performance limits of uploading data per

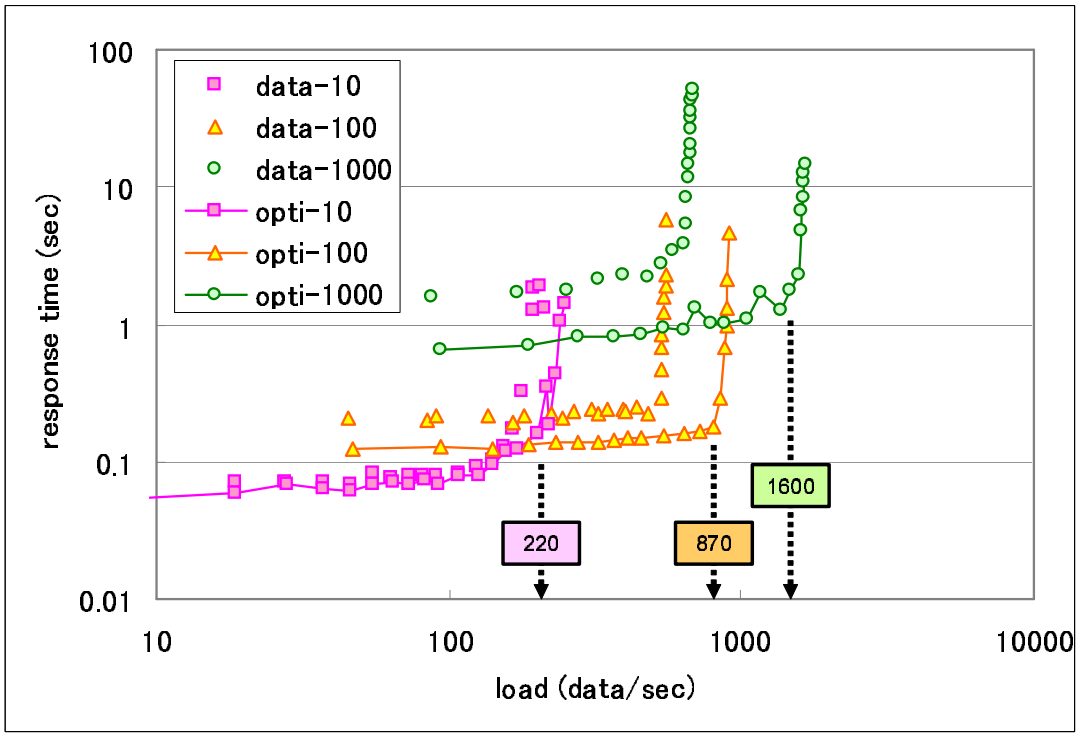


Figure 4.6: Efficacy of optimized format

node.

Figure 4.5 shows the result of uploading performance. One "data" is consisted of *ID*, *Mill-ID*, *sensor type*, *time*, and *value*. Uploading clients use *latitude* and *longitude* instead of *Mill-ID*. As sensing devices (uploading clients) put larger data, the performance of uploading become better. In addition, poor sensing devices can save their energy, uploading several data at once.

However, "data-1000" is too large to improve the performance. In order to upload sensing data, uploading clients have to register before uploading data. When a node of Mill receive sensing data from uploading clients, the node checks the sensing data whether data are sent by a registered sensor. This process of checking data makes the performance worse.

Secondly, we optimize the format for uploading sensing data. And we compare the performances of normal uploading and optimized uploading. Ordinary

weather sensors have some functions. For example, these sensors can measure temperature, humidity, pressure and other several items of weather. Aggregating sensing data sent from the same sensor device, nodes of Mill reduces the number of checking sensing data.

Figure 4.6 shows the result of comparison. The graph legend of "opti" means that nodes of Mill can checks data blocks at once by optimized format. As the data becomes larger, the efficacy of optimization becomes better.

If a sensing device measures 10 kinds of data and send data to a node per one minutes, one Mill node can handle 13,000 ($\approx 220(data/sec) \times 60(sec) = 13200$) sensors on performance limits. If an uploading client aggregates 10 sensors, that means sending 100 "data" to a node per one minute, one Mill node can handle 52,000 ($\approx 870(data/sec) \times 60(sec) = 52200$) sensors per node.

As Section 2.4 describes, 15,000,000 weather sensors are needed to measure weather conditions by micro-alpha-scale. If one set of this weather sensor has ten kinds of sensors, Mill network consisting of 2,900 ($\approx 15,000,000 \times 10 \div 52,000$) nodes can manage this scale sensor networks. The bottleneck of uploading performance is writing to the database system. Mill abstracts internal implementations by HTTP, and also implements O/R mapper to exchange database systems easily. It is expected to improve uploading performance by exchange of database and existing techniques of performance tuning, and the less number of nodes can manage larger scale sensor networks.

Considering data replications, the larger number of nodes is needed. Replication process is as following. After receiving sensing data, a node sends next node replication of these sensing data. If a node suddenly becomes disconnected, next node recover the lack of ID space and manages ID space including the disconnected node managed, and provides sensing data including the disconnected node had. As the Figure 4.6 shows, uploading bulk data is efficient. However, to send replication data at long interval is in danger of losting large data if nodes leave from overlay networks. And then, in addtion long delay can be occured when user getting current sensing data. There is a trade-off between interval of uploading replication data and lack of sensing data. These parameters are depends on kinds of applications. In most cases, lack of several sensing data can not cause serious effects on applications, because time-series data at around these trouble and

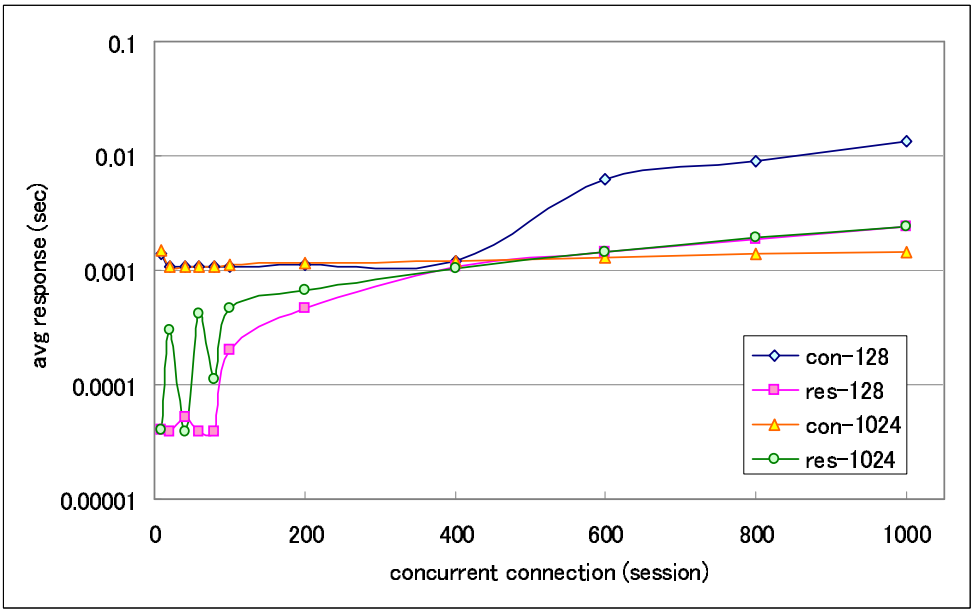


Figure 4.7: average response time

neighboring data complement lacks of data. An important factor is increasing management cost of a recovering node. If a recovering node manage additional ID space which is equals to a disconnected node managed, the recovering node constantly has to process approximate the twice number of queries. Then, if the uploading cost the recovering node processes exceeds the performance limit, many sensing data constantly lost. There is another approach to avoid lacking sensing data. Implementations of user-interface are hidden by communication protocols, and it is possible to improve robustness of a particular node by constructing a cluster system as like redundant web servers or database systems. The cost of recovering overlay networks is high, and therefore, it is necessary to consider provision of preventing from exeeding the performance limits, if providing practical applications.

4.4.3 Event driven messages

Mill supports event driven method by mechanism of “Comet”. Here we evaluate the performance of response time when a particular event occurs.

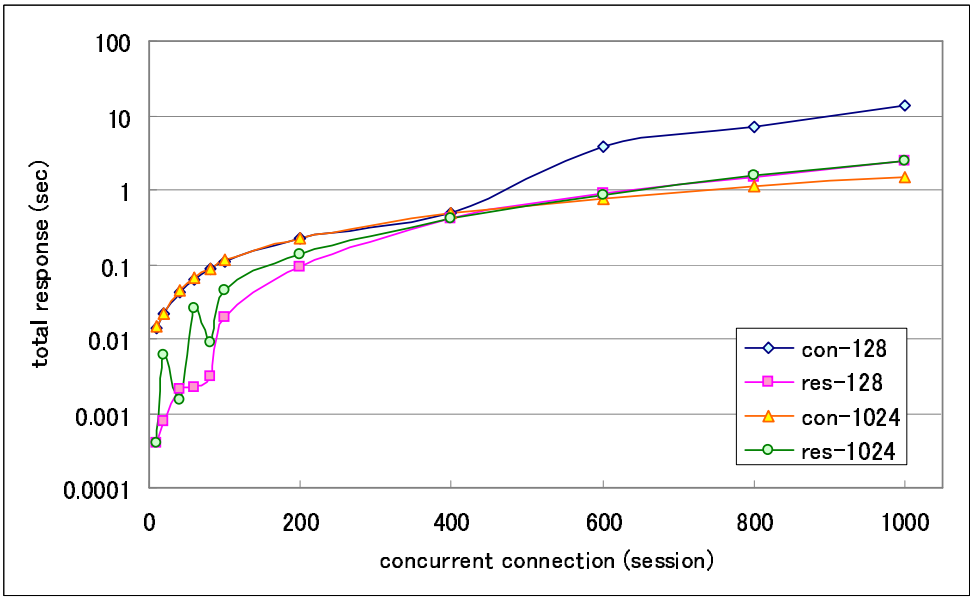


Figure 4.8: total response time

Figure 4.7, 4.8 shows the result of performance if a Mill node handles concurrent connections. In these Figures, graph legend of “con” means that users connect to a Mill node and ”res” means that a Mill node respond to users. And “128” means that the maximal value of “SOMAXCONN”. The legend of “1024” is also the maximal value of “SOMAXCONN”. Unix operating systems defines the number of listen queue by “SOMAXCONN”.

As Figure 4.7 shows, the time of connection and response are almost constant and it is milliseconds. As the number of concurrent connections becomes larger, the total response time linearly becomes longer. Because Mill adopts an asynchronous IO system at handling HTTP requests. Figure 4.8 shows that if the number of concurrent connection is 1024, the response time is approximately 2 seconds. This result means that users can receive an alert message at least in 2 seconds. This response time is useful to lots of applications, which are alerting start of rain, transgressing 35 Celsius and others.

Push mechanism reduces the larger number of queries than polling mechanism does. On the other hand, by using push mechanism nodes of Mill manages concurrent connections from users, and then nodes consumes larger memories than

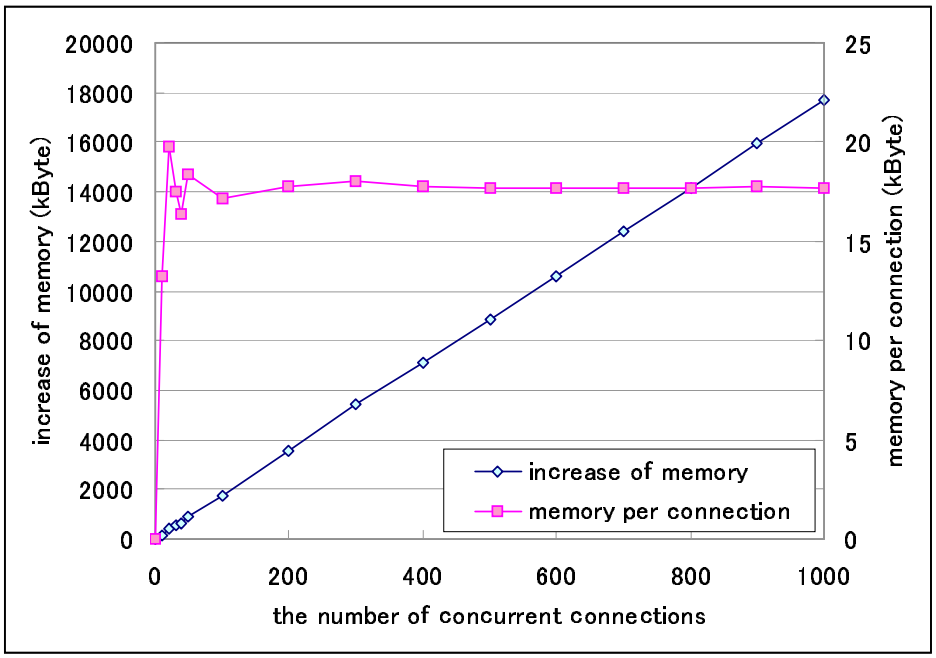


Figure 4.9:

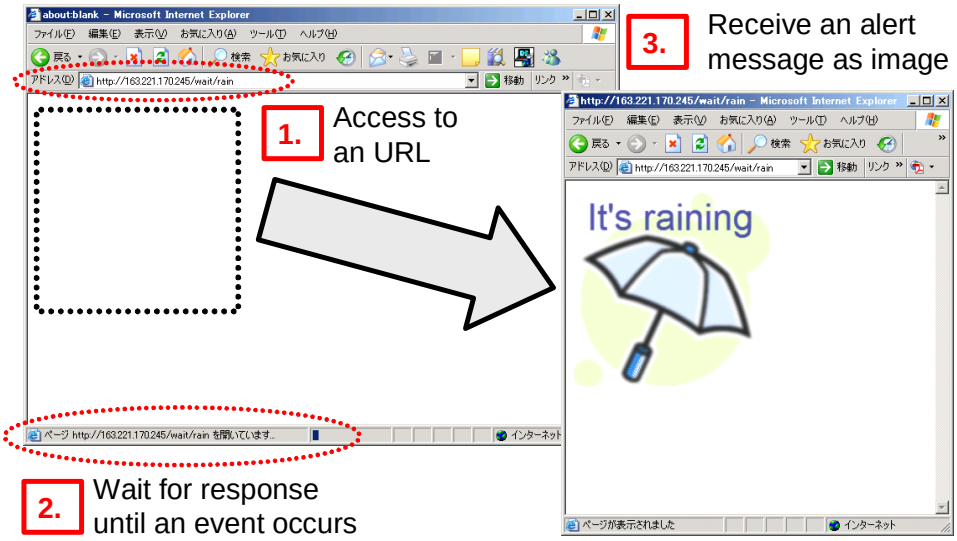


Figure 4.10: example of event driven behavior

by using polling. In other words, there is a trade-off between communication lines and memory space. Figure 4.9 shows the increase of memory consumption related with the number of concurrent connections. As the Figure shows, if 1,000 users concurrently connect to a nodes, the node approximately consumes additional 18 MBytes memories. Machines used for this experiment have 4 GB memory, and can approximately manage 222,000 connections. For this reason, memory consumption caused from concurrent connections does not become serious problems.

Figure 4.10 shows that a user receives an alert message of rainfall by an ordinary web browser. Without installing any special softwares, users can easily receive some services.

4.5 Summary of Implementation of Mill

In the ubiquitous sending environment, sensing devices can measure lots of kinds of data anywhere and anytime. To share these huge sensing data through the Internet, we implement a geographical location based P2P network called "Mill". Mill provides distributed environment based on geographical location and supports multiscale region search by $O(\log N)$ messages.

Our implementation of "Mill" handles *repetition of similar queries*, supports *event driven behavior* and considers *feasibility of deployment* in ubiquitous sensing environment. We adopt HTTP as a communication protocol and iterative routing as a routing algorithm to realize the above three requirements.

The results of evaluation show that one Mill node can handle ten thousands of users and sensing devices, and thousands of Mill nodes can manage weather data of micro-alpha-scale in Japan. If we can install Mill nodes different places on the Internet, we will be able to manage the huge data in ubiquitous sensing environment.

Chapter 5

Construction information infrastructure for sensing the Earth

The Live E! project is an open research consortium among industry and academia to explore the platform to share the digital information related with the earth and our living environment. We have getting a lot of low cost sensor nodes with Internet connectivity. The deployment of broadband and ubiquitous networks will enable autonomous and global digital information sharing over the globe. In this paper, we describe the technical and operational overview of Live E! project, while discussing the objective, such as education, disaster protection/reduction/recovery or business cases, and goal of this project activity.

5.1 Introduction

Recent natural disaster, e.g., hurricane or global warming, or heat island effect in the metropolitan let increase the attention and interesting on grasping the detailed status of space. This is because, due to these disasters, our social life and business activity could be seriously degraded or sometimes be damaged. When we could realize the relationship of cause and effect, it would be expected to reduce and protect the damage by these disasters. The structure change of metropolitans in the developed countries and rapid inflation of cities in Asian

countries make complex and difficult to realize the real status and tendency of global weather system. As symbolized by Kyoto Protocol proposed by United Nation in February of 2005, so called COP3 [37], it is realized as the urgent and serious global agenda to reduce or to stop the global warming effect.

We have getting a lot of low cost sensor nodes with Internet connectivity. The deployment of broadband and ubiquitous networks will enable autonomous and global digital information sharing over the globe, using these sensor nodes. When these wide variety of sensor nodes are autonomously connected and the sensor information let available to all the node on the Internet space, different types of sensor information, e.g., video or still image captured by Web camera, temperature, location, IR-image or chemical, can be integrated for data analysis. Then, we will be able to create so wide variety of applications and possibilities. When these sensor nodes are connected with broadband Internet, these information can be available even in real-time fashion. We have realized and proposed the activity, called Live E! Project [41], that is a sensor networks sharing all the digital information related with the at large status of the Earth for any purpose and for anyone.

We expect that these digital information will be used for various purposes, e.g., educational material, public service, business cases, by the deployment of effective and safe physical space for all the human being. Internet has been originally invented and developed to use and to share the expensive high performance computers, remotely. In these days, the jobs executed at the computer were numerical calculation for particular work to provide more effective working environment for researchers. Digital information has the following five primitives; generation, collection, circulation, processing, sharing. Through these five primitives, the computer system can improve and innovate the life-style of human being or the professional/commercial activities. Also, we have realized that the ICT can contribute to improve and innovate the human life and industrial activities more effective and safer.

In following sections, we describes the technical and operational overview of Live E! project, while discussing it's objective, such as education, disaster protection/reduction/recovery or business cases, and goal of this project activity.

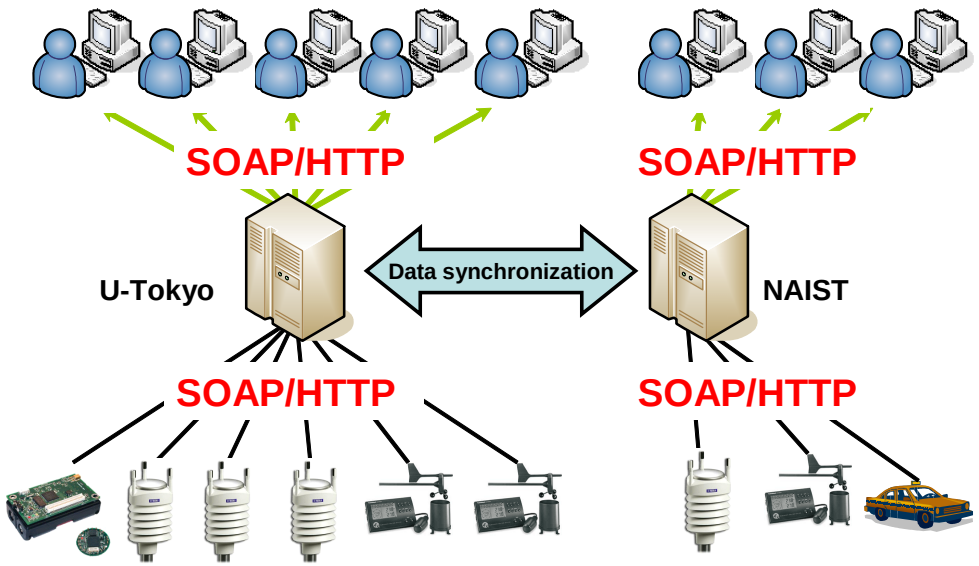


Figure 5.1: integrate interface by SOAP/HTTP

5.2 overview of Live E! project

Live E! Project is a R&D consortium founded by WIDE project and IPv6 Promotion Council, Japan. This project is aiming to establish the platform to share all digital information, generated by sensor devices all over the world. Now, these devices are installed and operated individually by each organization.

The information should be related with the live environmental information of the Earth. By sharing this digital information, we will be able to create new applications, which can contribute to safety and effective space (environment). Some applications, such as basic information for the protection of environment (e.g., a heat-island phenomena in metropolitan), educational material, public service, public safety or business applications, could use the common digital information for different purposes. Figure 5.1 shows the overview architecture of Live E! project. The working agendas of Live E! are installing a large number of sensors, combining every organization and individual to preserve global environment, and contributing environmental problems, education and businesses.

5.3 Live E! deployment

We have realized that the weather station with Internet connectivity has the following three application areas with single device.

(1) For Educational Material

Weather information data is useful data for education and for research on geophysics. There are wide variety of educational program on geophysics, related with the weather system, from the elementary school to the college. Actually, in Live E! project, we have worked on the educational program in the elementary school in Minato-ku in Metropolitan Tokyo, and on the engineering program in some high schools collaborating with university in Hiroshima.

(2) For Public Service, e.g., disaster reduction/recovery

Weather information is very important and critical information for the case of disaster. In these days, we have a lot of natural disasters, such as flooding or earth-quake. Grasping the detailed information for the disaster case is useful for proactive and reactive program. These are disaster protection, reduction, and recovery. For example, the detailed weather information on the road or at the evacuation sites, the people could take appropriate evacuation path. Also, none knows the exact and detailed data on the heat-island effect at metropolitan. We must grasp the real status of town with large number of weather sensors.

In the fiscal year of 2006, Live E! project will deploy the Internet weather sensors for public service at Minato-ku in metropolitan Tokyo [48], at Kurashiki-city in Okayama [35], at Hiroshima-city and other local governments. Minato-ku is focusing on the disaster protection and reduction against some natural disasters (e.g., flooding or earth-quake), and focusing on the understanding the detail of heat-island effect of metropolitan Tokyo. Kurashiki-city is focusing on the disaster protection and reduction, against the flooding due to heavy rain. Figure 5.2 shows the current status of weather around Kurashiki City.

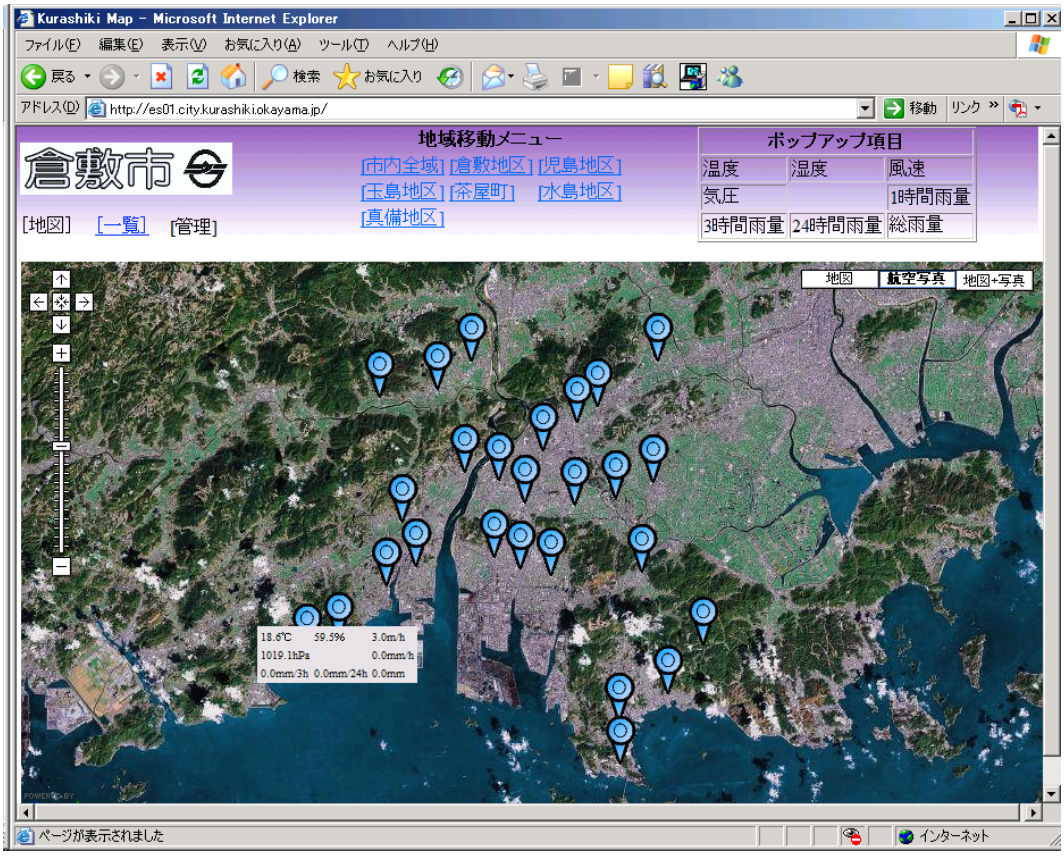


Figure 5.2: sensor data of Kurashiki City

(3) For Business case

There are a lot of potential business applications, by the use of weather information. One of example would be effective taxi dispatching using the rainfall information. Dispatching the taxi cabs around the area, getting the rain, leads to higher income by the increase of customer. The other example would be electricity power company. Once they can operate total energy control and management system, they could reduce the amount of investments on the power generator or power supplying system, which is very expensive facility for them. Seriously, they may start to think about total portfolio for business investments.

Table 5.1: normalized name & unit

name	unit
Temperature	centigrade
Humidity	%
Pressure	hPa
RainFall	mm/h
WindSpeed	m/s
WindDir	degree

Table 5.2: examples of profile data

name	explanation
address	address of sensor
location	installation location
ipAddr	IPv4 address
ip6Addr	IPv6 address
latitude	latitude of sensor
longitude	longitude of sensor
altitude	altitude of sensor

5.3.1 Data normalization and access method

In order to share sensing data with every organization and individual, first we need to normalize data format and access method. In this sub-section we introduce our data format with some examples.

The weather information is expressed by XML and transferred among the servers and clients using the SOAP. The sensor is not defined by node, but is defined by each sensor function. A typical internet weather station has multiple sensors in a single station in it, which sensors are temperature, humidity, pressure, rainfall, wind-speed and wind-direction. This typical weather station has 5 objectives defined by XML. Each objective has their own profile information, such as location, sensor type, IP address or name.

As shown in Table 5.1, we basically adopt MKS-system on our platform. If a

Table 5.3: examples of SOAP web-services

getCurrentDataAll() - get current data from all sensors
getCurrentDataByAreaRect(x1, y1, x2, y2) - get data generated in particular region
getDataByTimespan(sensorID, startTime, endTime) - get data by time-span
getCurrentDataByType(sensorType) - get current data of specific sensor type
getProfileAll() - get all profile data
getProfileByType(sensorType) - get profile data of specific sensor type
putData(xmlDocument) - put sensing data
setProfile(xmlDocument) - set profile data

new kind of sensor is installed, project member will define its name and the unit (recently, we installed CO2 sensors, illumination sensors, acceleration sensors and other sensors). By SOAP web-services, we create a user interface (Figure 5.1). As databases are synchronized between two locations, users can select web services which they want to use.

As the table 5.2 shows, we also provide attribute information of sensors called "profile data". We normalize and provide sensing data and profile data as XML document. Application creators freely combine sensing data with profile data to meet each requirement.

Example web-services we prepared are shown in Table 5.3. Here, we introduce an example way to retrieve sensor data. The most popular web-service is "getCurrentDataAll()". By this function, user gets current data generated by all sensors. Following Perl script is an example code to do it.Of course, any language and operating system can access to sensor data by using SOAP.

```
#!/usr/local/bin/perl -w
use encoding 'UTF-8', STDOUT => 'shiftjis';
```

```
<sensorGroup address="東京都文京区弥生2-11-16 " class="combinedSensor" id="live-
e.org/WXT510/03000005c3a2/" latitude="35.712194" location="東大情報基盤センター"
longitude="139.76775" sensorModel="WXT510" sensorVendor="vaisala">
<sensor id="live-e.org/WXT510/03000005c3a2/Temperature" sensorType="Temperature">
<value time="2006-07-25T19:37:33.0000000+09:00">25.8</value>
</sensor>
<sensor id="live-e.org/WXT510/03000005c3a2/Humidity" sensorType="Humidity">
<value time="2006-07-25T19:37:33.0000000+09:00">73.3</value>
</sensor>
<sensor id="live-e.org/WXT510/03000005c3a2/Pressure" sensorType="Pressure">
<value time="2006-07-25T19:37:33.0000000+09:00">1009.4</value>
</sensor>
<sensor id="live-e.org/WXT510/03000005c3a2/RainFall" sensorType="RainFall">
<value time="2006-07-25T19:37:33.0000000+09:00">0</value>
</sensor>
<sensor id="live-e.org/WXT510/03000005c3a2/DayRainFall" sensorType="DayRainFall">
<value time="2006-07-25T19:37:33.0000000+09:00">0.64</value>
</sensor>
<sensor id="live-e.org/WXT510/03000005c3a2/WindDir" sensorType="WindDir">
<value time="2006-07-25T19:37:33.0000000+09:00">174</value>
</sensor>
<sensor id="live-e.org/WXT510/03000005c3a2/WindSpeed" sensorType="WindSpeed">
<value time="2006-07-25T19:37:33.0000000+09:00">0.9</value>
</sensor>
</sensorGroup>
```

Figure 5.3: example of reply as XML document

```
use SOAP::Lite;

$service = SOAP::Lite -> service(
   'http://example.com/axis/DataProvider?wsdl');
$currentData = $service->GetCurrentDataAll();
print $currentData;
```

Figure 5.3 shows the result XML documents when above example Perl script executed. We standardized XML format. Using other SOAP web-services, user always receives data as same XML format. Defining the name of web-services and XML format contributes to create application easily. In addition, as the above XML documents shows, we define a sensor-group. Sensor-group organizes individual sensors. In above XML document, class of sensor-group is "combined-

Sensor". This sensor-group organizes a multiple sensor as one of sensors. We define a new class of sensor-group for several purposes. For example, class of "zipCode" is to manage a particular area, and class of "highAccuracy" is to pick up high performance sensors. Of course, we create new web-services by combining several classes. Sensor-group improves a creativeness of raw sensing data.

Here, we introduce some of our applications created by SOAP web-services. Figure 5.4 shows the example of data displaying. On the web site, user can search each place for detailed data, which are graph of sensors, XML (or CSV) data and pictures of web cameras. And, with Google Earth ,we show the status of clouds and path information of a typhoon with sensor data. Live E! Project doesn't observe cloud and typhoon. Collaborating with other organizations, our project provides these data. We enhance the collaboration with others.

We also provide educational materials. We made a information viewer displaying real time sensor data. Students of elementary school constantly check and record the data of this viewer. Students will plot a graph with these data and discuss the difference of daily weather and local climate, and compare climate of home town with other place. Students will take interest in science with such actual experiences.

We have installed more than 100 stations and let them on-line. Some of stations have installed in Philippine by the collaboration with ASTI, or have installed in Thailand by the collaboration with PSU. For dense installation of weather sensors, we collaborate with Minato-ku in metropolitan Tokyo. Several sensors are installed at some elementary schools at Minato-ku and Kurashiki-city in Okayama. We will install about 30 stations in this fiscal year of 2006. By this installation, Minato-ku can have about 2 km mesh weather station network. And Kurashiki-city is focusing on the disaster protection and reduction against the flooding due to heavy rain.

5.4 Future enhancement of Live E!

- XML based Publish/Subscribe system
  The more sensor devices and users increase, the more data and queries should be processed. The most popular web-service is "getCurrentDataAll()".

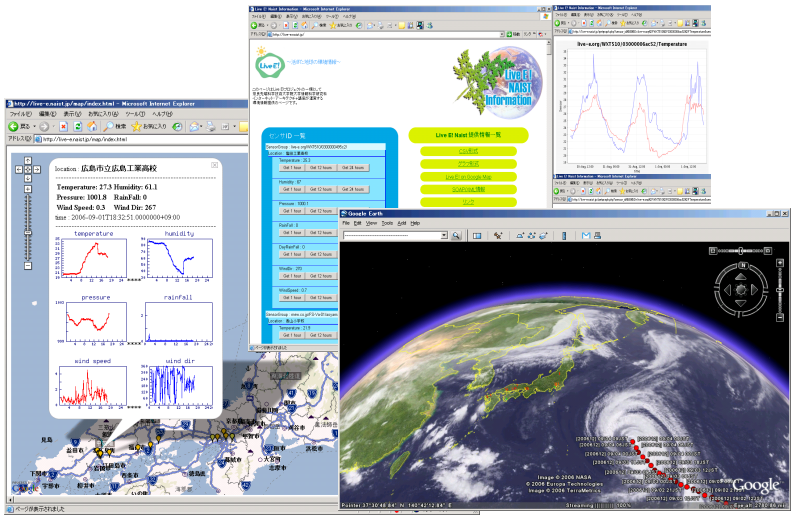


Figure 5.4: data displaying

Users are very interested in real time data. In order to provide this web-service stable, we create XML based Publish-Subscribe system. This system don't store archived data but only provide current data. If users register for subscribing data, they constantly receive current data of all sensors from one of XML router. From now, we will try to set up of an application layer multicast based on user requirements to provide all web-services stable.

- Integration with InternetCAR system
  WIDE project has long time worked on the R&D activity, where automobiles connect to the Internet. Automobiles can be realized as the mobile sensor node, running on the surface of the earth. We are now integrating the "InternetCAR" into our platform [17]. In addition, we are trying to use ad-hoc network for sensors not having the connectivity to the internet. Patrol nodes (ex. Post office cars, garbage trucks or buses) gather the sensing data via ad-hoc network [31].

- Support small nodes
  There are lots of microcomputers not having the ability to translate XML document. In order to integrate lots of data created by microcomputers

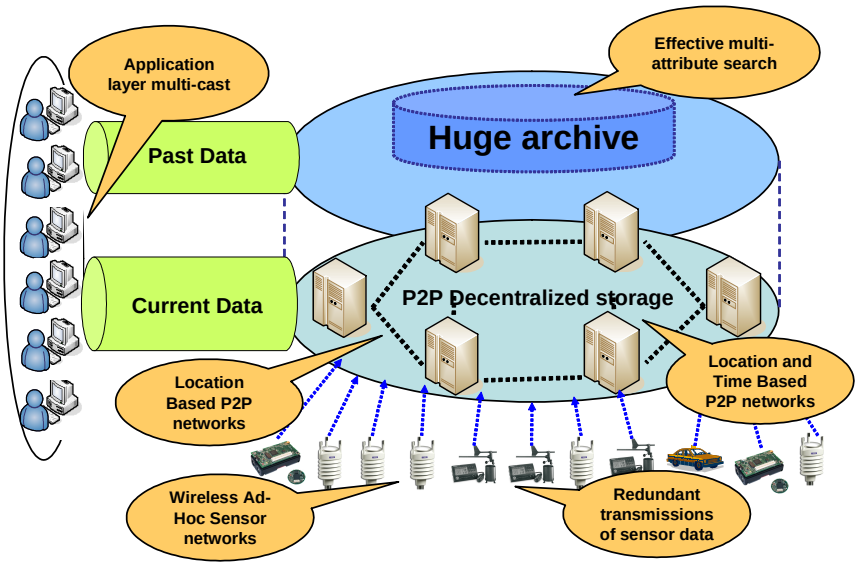


Figure 5.5: next Live E! architecture

generate, we study about XML translational machinery [54].

- Scalable and autonomous data collection system
  The current system is a kind of client-server system. This is just fine for small scale system. However, the system must come up with the increase of sensors and the increase of type of sensors. Distributed and autonomous management and operation is mandatory. We are studying about geographical location based overlay network [46].

As shown in Figure 5.5, next phase we focus on scalable and autonomous data collection and provision. We are evaluating the requirements of future sensor network, and integrate several kinds of researches. And we have been introducing proposed geographical overlay network into Live E! network.

5.5 Summary of Live E! project

The Live E! project is an open research consortium among industry and academia to explore the platform to share the digital information related with the earth

and our living environment. Using the low cost weather sensor nodes with Internet connectivity, we deployed the nation-wide sensor network. The network has accommodate more than 100 station, and has two of dense installation. The application of this weather station network is for disaster protection/reduction/recovery and for educational material for various level of students.

In the near future, the number of weather sensors becomes much larger. Live E! system therefore will have to adopt scalable data management system. Through actual operations, we have investigated patterns of sensing data stream and user queries. These instances give great hints to implement proposed overlay network.

Chapter 6

Conclusion

Development of sensor technologies and the global diffusion of the Internet enables us to expect that lots of heterogeneous sensor networks are developing all over the places and interconnecting to share ubiquitous sensing data. In addition, recent sensing devices become intelligent and also have an ability to easily connect to the Internet. Moreover, the price of these device rapidly diminishes.

A decentralized system combined with scalability and multiple-attributes search is a key mechanism to realize a goal of sharing global-scale sensing data. Many decentralized system, mainly overlay networks have been studied over the years. However, on these networks some nodes have to manage locally-concentrated data or particular queries cause large retrieval costs, in order to manage data on real space. This is because these works lack for considering patterns of sensor data stream and user queries besides geographical distribution of sensor networks, although they have been focusing on scalability and trying to construct distributed and self-organized systems on the Internet-scale network. It is necessary to reconstruct the ID-spaces in overlay networks without detracting advantages of previous works.

6.1 Contribution

This dissertation proposes a new overlay network to manage global ubiquitous sensing data. The proposed the overlay network called *Mill* constructs a decentralized data management system destroying the locality of sensing data. On

Mill network, nodes are distributed by geographical location and manage data of local areas. Mill supports multiple-attributes queries besides multi-scale range queries, managing only one dimensional ID-space. This one dimensional ID-space consisting of latitude and longitude is calculated by *Z-ordering* method. One dimensional ID-space enables a routing mechanism to become simple and fast.

The simulation experiments show that queries are resolved via $O(\log N)$ messages and proposed overlay network has scalability and robustness. These features are equal to advantages DHTs and other works have. In addition, Mill supports multiple-attributes queries and multi-scale geographical range queries. There has to be a trade-off between advantages of routing and limitation of retrieval. However, these limitations of Mill hardly have a negative effect on multiple-attributes search in ubiquitous sensing environment, because Mill considers patterns of sensing data stream and user queries. In Live E! project, we have actually installed hundreds weather sensors and provided several services for users. These experiences clarify the patterns of sensing data stream and user queries and give great hints to design the implementation of Mill network. Lots of sensing devices and users tend to access the same nodes, because almost sensing devices immobile or can not move fast and almost users interested in particular places. Considering above situations, an optimized routing mechanism extremely reduce retrieval cost on the overlay network. And Publish/Subscribe mechanism implemented by *Comet* also reduces lots of communication between users and target nodes.

Evaluations of implementation clarify the processing performance of queries and its scalability and limitations. These evaluations also show that one Mill node can manage several tens of thousands of sensing devices or users. The implementation of Mill constructs a loosely-coupled system by distribution of nodes based on geographical location and its communication protocols, therefore it is possible that Mill network realizes one of scalable global-scale data management systems by developing nodes all over the places and its shows one of feasibility of managing global ubiquitous sensing data.

6.2 Future Directions

This dissertation describes a new overlay network and its implementation for managing ubiquitous sensing data. On a proposed overlay network, users can search multi-scale geographical range for particular data. This achievement of constructing data management system raises necessity of managing calculations. Useful information are created from particular data through several calculations and used for some applications. These processes of calculations cause some costs. In contrast, replications of calculated data cause very little costs. Users can create more complex and useful information based on calculated data if data of replications can be stored and reused on overlay networks.

It is remarkable that in some cases calculation methods are more important than calculated data themselves, because sensing data have a feature as stream. It should be founded to share calculation methods besides calculated data, in order to introduce these feedback mechanism into overlay networks. It is expected that construction of calculation management layer isolated from data management layer and sharing processes of calculations are essential keys to better utilize global-scale sensing data.

References

[1] K. Aberer, M. Hauswirth and A. Salehi. "Infrastructure for data processing in large-scale interconnected sensor networks." Proceedings of The 8th International Conference on Mobile Data Management (MDM'07). May. 2007.

[2] K. Aberer, M. Hauswirth and A. Salehi. "A Middleware For Fast And Flexible Sensor Network Deployment." Proceedings of the 32nd international conference on Very large data bases (VLDB'06). pages 1199–1202. Sep. 2006.

[3] J. Aspnes and G. Shah. "Skip Graphs." Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms (SODA2003). pages 384–393. Jan. 2003.

[4] ASTI (Advanced Science and Technology Institute), `http://www.asti.dost.gov.ph/`

[5] R. Avnur and J. M. Hllerstein. "Eddies: Continuously Adaptive Query Processing." Proceedings of ACM Special Interest Group on Management of Data (SIGMOD'00). pages 261–272. May. 2000.

[6] R. Bayer and E. McCreight. "Organization and maintenance of large ordered indexes." Acta Informatica Vol.1, No.3, pages 173–189. Springer Berlin / Heidelberg. Sep. 1972.

[7] Ashwin R. Bharambe, Mukesh Agrawal, Srinivasan Seshan. Mercury: supporting scalable multi-attribute range queries ACM SIGCOMM Computer Communication Review, pages 353–366, 2004

[8] M. Buettner, G. V. Yee, E. Anderson and R. Han. "X-MAC: A short preamble MAC protocol for duty-cycled wireless sensor networks." Proceedings of the 4th international conference on Embedded networked sensor systems (SenSys'06). pages 307–320. Nov. 2006.

[9] J. D. Case, M. Fedor, M. L. Schoffstall and J. R. Davin. Simple Network Management Protocol (SNMP). RFC 1157.

[10] I. Chakeres and C. Perkins. Dynamic MANET On-demand (DYMO) Routing (draft-ietf-manet-dymo-10). Internet-Draft (Standards Track).

[11] S. Chandrasekaran, O. Cooper, A. Deshpande, M. J. Franklin, J. M. Hellerstein, W. Hong, S. Krishnamurthy, S. R. Madden, F. Reiss and M. A. Shah. "TelegraphCQ: Continuous Dataflow Processing for an Uncertain World." Proceedings of First Biennial Conference on Innovative Data System Research (CIDR 2003). Jan. 2003.

[12] S. Chandrasekaran and M. J. Franklin. "PSoup: a system for streaming queries over streaming data." The International Journal on Very Large Data Bases (VLDB Journal). Vol.12, pages 140–156. Aug. 2003.

[13] I. Clarke, O. Sandberg, B. Wiley and T. W. Hong. "Freenet: a distributed anonymous information storage and retrieval system." Proceedings of International workshop on Designing privacy enhancing technologies: design issues in anonymity and unobservability. pages 46-66. Jul. 2000.

[14] T. Clausen and P. Jacquet. Optimized Link State Routing Protocol (OLSR). RFC 3626.

[15] A. Deshpande, S. Nath, P. B. Gibbons and S. Seshan. "IRIS: Internet-scale Resource-Intensive Sensor Service." Proceedings of the 2003 ACM SIGMOD international conference on Management of data. pages 667–667. Jun. 2003.

[16] V. Devarapalli, R. Wakikawa, A. Petrescu and P. Thubert. "Network Mobility (NEMO) Basic Support Protocol." RFC 3963.

[17] Shinichi Doi, Satoshi Matsuura, Hideki Sunahara: An Infrastructure for Environmental Information Gathered by Fixed-point and Mobile Sensors, Multimedia, Distributed, Cooperative and Mobile Symposium(2006), DICOMO2006, IPSJ Symposium Series Vol.2006, No.6, 801–804.

[18] Dumbo project, `http://www.interlab.ait.ac.th/dumbo/`

[19] D. Eastlake and P. Jones. US Secure Hash Algorithm 1 (SHA1). RFC 3174.

[20] P. B. Gibbons, B. Karp, Y. Ke, S. Nath and S. Seshan. "IrisNet: An Architecture for a WorldWide Sensor Web." IEEE Pervasive Computing, Volume 2, Number 4 (October-December 2003).

[21] N. J. A. Harvey, M. B. Jones, S. Saroiu, M. Theimer, and A. Wolman. "SkipNet: A Scalable Overlay Network with Practical Locality Properties." In Proceedings of Fourth USENIX Symposium on Internet Technologies and Systems (USITS 03), Mar. 2003.

[22] J. Heidemann, F. Silva and C. Intanagonwiwat. "Building Efficient Wireless Sensor Networks with Low-Level Naming." Proceedings of the eighteenth ACM symposium on Operating systems principles (SOSP 2001). pages 146–159. Oct. 2001.

[23] D. Hilbert. "Über die Stetige Abbildung Einer Linie auf Ein Flächenstück." Mathematische Annalen, vol. 38, pages 459–460, 1891.

[24] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister. "System Architecture Directions for Network Sensors." In Proceedings of the 9th International Conference on Architectural Support for Programming Languages and Operating Systems, pages 93–104, ACM, Nov. 2000.

[25] Hiroshima-City, `http://www.city.hiroshima.jp/`

[26] CO2 data of Hiroshima-City, `http://lab.ipc.hiroshima-cu.ac.jp/co2/`

[27] M. Horton, D. Culler, K.pister, J. Hill, R. Szewczyk, and A. Woo. "MICA, The Commercialization of Microsensor Motes. In sensors Magazine, pages 40-48, Apr. 2002.

[28] IEEE, "Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)." IEEE-802.15.4-2003, 2003.

[29] C. Intanagonwiwat, R. Govindan and D. Estrin. "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks." Proceedings of the 6th annual international conference on Mobile computing and networking (MobiCOM 2000). pages 56–67. Aug. 2000.

[30] C. Intanagonwiwat, D. Estrin, R. Govindan and J. Heidemann. "Impact of Network Density on Data Aggregation in Wireless Sensor Networks." Distributed Computing Systems, 2002. Proceedings. 22nd International Conference (ICDCS'02). pages 457–458. Nov. 2001.

[31] Hiroki Ishizuka, Kenji Sasaki, Satoshi Matsuura, Makoto Kamiya, Hideki Sunahara, Hiroshi Esaki: Collecting Adaptive Data for Isolated Wireless Sensors with Patrol Nodes in Live E!, International Workshop on Future Mobile and Ubiquitous Information Technologies, Proceeding of FMUIT2006, pages 249–253. 2006.

[32] H. V. Jagadish, B. C. Ooi and Q. H. Vu. "BATON: A Balanced Tree Structure for Peer-to-Peer Networks." Proceedings of the 31st international conference on Very large data bases (VLDB'05). pages 661–672. Sep. 2005.

[33] H. V. Jagadish, B. C. Ooi, K. Tan, Q. H. Vu. and R. Zhang. "Speeding up Search in Peer-to-Peer Networks with A Multi-way Tree Structure." Proceedings of the 2006 ACM SIGMOD international conference on Management of data. pages 1–12. Jun. 2006.

[34] B. Karp, and H. T. Kung. "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks" In Proceedings of ACM/IEEE International Conference on Mobile Computing and Networking(Mobicom2000), pages 243–254, Aug. 2000.

[35] Kurashiki-City, `http://www.city.kurashiki.okayama.jp/`

[36] sensor data of Kurashiki-City, `http://es01.city.kurashiki.okayama.jp/`

[37] Kyoto Protocol by United nation, `http://www.cop3.org/`

[38] T. J. Lehman and M. J. Carey. "A Study of Index Structures for Main Memory Database Management Systems." Proceedings of the 12th international conference on Very large data bases (VLDB'86). pages 294–303. Aug. 1986.

[39] Japan Meteorological Agency `www.jma.go.jp/`

[40] Japan Ministry of Internal Affairs and Communications Statistics Bureau. http://www.stat.go.jp/

[41] Live E! project, http://www.live-e.org

[42] S. Madden, M. Franklin, J. Hellerstein and W. Hong. "TAG: A tiny AGgregation service for ad-hoc sensor networks." Proceedings of Fifth Symposium on Operating Systems Design and Implementation (OSDI2002), pages 131–146. 2002.

[43] S. Madden, M. Franklin, J. Hellerstein and W. Hong. "TinyDB: An acuisitional query processing system for sensor networks." ACM TODS, pages 122–173. 2005.

[44] S. Madden, M. Shah, J. M. Hellerstein and V. Raman. "Continuously Adaptive Continuous Queries over Streams." Proceedings of the 2002 ACM SIGMOD international conference on Management of data. pages 49–60. Jun. 2002.

[45] A. Manjhi, S. Nath and P. B. Gibbons. "Tributaries and Deltas: Efficient and Robust Aggregation in Sensor Network Streams." Proceedings of the 2005 ACM SIGMOD international conference on Management of data. pages 287–298. Jun. 2005.

[46] S. Matsuura, K. Fujikawa, H. Sunahara. "Mill: A geographical location oriented overlay network managing data of ubiquitous sensors." IEICE Transactions on Communications, Vol.E90-B, pages 2720–2728, Oct. 2007.

[47] P. Maymounkov and D. Mazieres. "kademlia: a peer-to-peer information system based on the XOR metric." Proceedings of the 1st International Workshop on Peer-to-Peer Systems (IPTPS '02). Mar. 2002.

[48] Minato-ku, Tokyo, http://www.city.minato.tokyo.jp/

[49] P. V. Mockapetris and K. J. Dunlap. "Development of the Domain Name System." In Proceedings of SIGCOMM '88. pages 123–133. Apr. 1988.

[50] G. M. Morton. “A computer Oriented Geodetic Data Base; and a New Technique in File Sequencing.” Technical Report, IBM Ltd. Ottawa, Canada, 1966.

[51] M. Nakayama, S. Matsuura, H. Esaki and H. Sunahara. “Live E! Project: Sensing the Earth.” Second Asian Internet Engineering Conference AINTEC 2006,Lecture Notes in Computer Science, Volume 4311/2006, pages 61–74. Springer Berlin / Heidelberg. Nov. 2006.

[52] S. Nath, P. B. Gibbons, S. Seshan and Z. R. Andrson. “Synopsis Diffusion for Robust Aggregation in Sensor Networks.” Proceedings of the 2nd international conference on Embedded networked sensor systems (SenSys’04). pages 250–262. Nov. 2004.

[53] S. Nath, Y. Ke, P. B. Gibbons, B. Karp and S. Seshan. “IrisNet: An Architecture for Enabling Sensor-Enriched Internet Service.” Intel Research Pittsburgh Technical Report IRP-TR-03-04, June 2003.

[54] H. Ochiai, H. Esaki. “The Application of XML Translator for a Large-Scale Sensor Node Network.” IEICE 2006, A-21-5. 2006.

[55] I. Orlanski. “A rational subdivision of scales for atmospheric processes.” Bulletin of the American Meteorological Society, Vol. 56, No.5, pages 527-530. May. 1975.

[56] C. Perkins, E. Belding-Royer and S. Das. Ad hoc On-Demand Distance Vector (AODV) Routing. RFC 3561.

[57] J. Polastre, J. Hill and D. Culler. “Versatile low power media access for wireless sensor networks.” Proceedings of the 2nd international conference on Embedded networked sensor systems (SenSys’04). pages 95-107. Nov. 2004.

[58] PSU (Prince of Songkla University), `http://www.psu.ac.th/`

[59] W. Pugh. “Skip Lists: A Probabilistic Alternative to Balanced Trees.” Communications of the ACM Vol.33, pages 668–676, Jun. 1990.

[60] S. Ratnasamy, P. Francis, M. Handley, R. Karp and S. Schenker "A Scalable Content-Addressable Network." ACM SIGCOMM, pages 161–172, 2001

[61] S. Ratnasamy, B. Karp, L. Yin, F. yu, D. Estrin, R. Govindan, and S. Shenker. "GHT: A Geographic Hash Table for Data-Centric Storage" In Proceedings of ACM International Workshop on Wireless Sensor Networks and Applications(WSNA2002), pages 78–87, 2002.

[62] I. Rhee, A. Warrier, M. Aia and J. Min. "Z-MAC: A hybrid MAC for wireless sensor networks." Proceedings of the 3rd international conference on Embedded networked sensor systems (SenSys'05). pages 90-101. Nov. 2005.

[63] V. Roman, A. Deshpande and J. M. Hellerstein. "Using State Modules for Adaptive Query Processing." Proceedings of 19th International Conference on Data Engineering (ICDE2003). pages 353–364. Mar. 2003.

[64] A. Rowe, R. Mangharam and R. Rajkumar. "RT-Link: A Time-Synchronized Link Protocol for Energy-Constrained Multi-hop Wireless Networks." Proceedings of the IEEE 3rd Sensor and Ad Hoc Communications and Networks (SECON'06). pages 402–411. Sep. 2006.

[65] A. Rowstron and P. Druschel. "Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems." In IFIP/ACM Int'l Conf. Distributed Systems Platforms (Middleware), pages 329–350, 2001.

[66] H. Sunahara, M. Sato, K. Uehara, K. Aoki and J. Murai. "IPCar: Building the Probe Car System with the Internet." IEICE Transactions on Information and Communications engineers, Vol.J85-B, pages 431–437, Apr. 2002.

[67] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek and H. Balakrishnan. "Chord: A scalable peer-to-peer lookup service for internet applications." Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM 2001). pages 149–160, Sep. 2001.

[68] H. Tetsuji, S. Masaaki, U. Keisuke, H. Hitoshi, I. Kei, H. Ryota, A. Hirokazu and H. Sunahara. "Hakoniwa - application development environments for

internet car systems." 11th World Congress on ITS Nagoya, Aichi2004. Oct. 2004.

[69] World Wide Web Consortium (W3C). "XML Path Language (XPath) Version 1.0." `http://www.w3.org/TR/xpath` Nov. 1999.

[70] J. Zhao and R. Govindan. "Understanding Packet Delivery Performance In Dense Wireless Sensor Networks." Proceedings of the 1st international conference on Embedded networked sensor systems (SenSys'03), pages 1–13. Nov. 2003.

[71] B. Zhou, J. Kubiatowicz and D. A. Joseph. "Tapestry: An Infrastructure for Fault-tolerant Wide-area Location and Routing." Technical Report UCB/CSD-01-1141, Computer Science Division, U. C. Berkeley. Apr. 2001.

Publications

Journal

- Satoshi Matsuura, Kazutoshi Fujikawa and Hideki Sunahara. Mill: A Geographical Location Oriented Overlay Network Managing Data of Ubiquitous Sensors. *IEICE TRANSACTIONS on Communications*, E90-B, No.10, pages 2720–2728. Oct. 2007.

International Conference

- Satoshi Matsuura, Hiroki Ishizuka, Hideya Ochiai, Shinichi Doi, Shinichi Ishida, Masaya Nakayama, Hiroshi Esaki and Hideki Sunahara. Live E! Project: establishment of infrastructure sharing environmental information. In *Proceedings of International Symposium on Applications and the Internet Workshops (SAINT-W 07)*, No.67. Jan. 2007.

- Satoshi Matsuura, Kazutoshi Fujikawa and Hideki Sunahara. Mill: Scalable Area Management for P2P Network based on Geographical Location. In *Proceedings of Twelfth Annual Scientific Conference on Web technology, New Media, Communications and Telematics theory, Methods, Tools and Applications (Euromedia 2006)*, pages 46–52. Apr. 2006.

- Satoshi Matsuura, Kazutoshi Fujikawa and Hideki Sunahara. Mill: An Information Management and Retrieval Method Considering Geographical Location on Ubiquitous Environment. In *Proceedings of International Symposium on Applications and the Internet Workshops (SAINT-W 06)*, pages 14–17. Jan. 2006.

Technical Report

- Satoshi Matsuura, Hiroki Ishizuka, Hideya Ochiai, Shinichi Doi, Shinichi Ishida, Masaya Nakayama, Hiroshi Esaki and Hideki Sunahara. Live E!: control system for sensor stream and geographical location based overlay network managing Ubiquitous sensors. In *IPSJ Multimedia, Distributed,*

Cooperative, and Mobile Symposium (DICOMO 2007), pages 1161–1167. Jul. 2007. In Japanese.

- Satoshi Matsuura, Kazutoshi Fujikawa and Hideki Sunahara. A method for managing and retrieval information related with geographical location on peer-to-peer network. In *IPSJ 46th Programing Symposium*, pages 83–94. Jan. 2005. In Japanese.

Joint Paper: Journal

- Shinichi Doi, Satoshi Matsuura, Kazutoshi Fujikawa and Hideki Sunahara. Overlay Network Considering the Time and Location of Data Generation. *IPSJ Journal*, Vol.49. Feb. 2008. (to be appeared) In Japanese.

Joint Paper: International Conference

- Shinichi Doi, Satoshi Matsuura, Kazutoshi Fujikawa and Hideki Sunahara. Overlay Network Considering the Time and Location of Data Generation. In *Proceedings of International Symposium on Applications and the Internet Workshops (SAINT-W 07)*. No.61. Jan. 2007.

- Masaya Nakayama, Satoshi Matsuura, Hiroshi Esaki and Hideki Sunahara. Live E! Project: Sensing the Earth. Proceedings of the Second Asian Internet Engineering Conference (AINTEC 2006), Lecture Notes in Computer Science (LNCS), Volume 4311/2006, pages 61–74. Springer Berlin / Heidelberg. Nov. 2006.

- Hiroki Ishizuka, Kenji Sasaki, Satoshi Matsuura, Makoto Kamiya, Hideki Sunahara and Hiroshi Esaki. Collecting Adaptive Data for Isolated Wireless Sensors with Patrol Nodes in Live E!. Proceedings of the International Workshop on Future Mobile and Ubiquitous Information Technologies (FMUIT'06). pages 249–253. May. 2006.

Joint Paper: Technical Report

- Shinichi Doi, Satoshi Matsuura, Kazutoshi Fujikawa and Hideki Sunahara. Overlay Network Considering the Time and Location of Data Generation. *Technical Report of the IPSJ Special Interest Group on System Evaluation*, 2007-EVA-22-(2), Vol.2007, No.82(20070802), pages 7-14. Aug. 2007. In Japanese.

- Shinichi Doi, Satoshi Matsuura, Kazutoshi Fujikawa and Hideki Sunahara. VS-RS: Creation of a Virtual Sensor based on a Real Sensor. *Technical Report of the IPSJ Multimedia, Distributed, Cooperative, and Mobile Symposium (DICOMO 2007)*, pages 1502–1509. Jul. 2007. In Japanese.

- Masato Yamanouchi, Shinichi Doi, Satoshi Matsuura, Kazutoshi Fujikawa and Hideki Sunahara. Auto configuration method of sensor location. *Technical Report of the IPSJ Multimedia, Distributed, Cooperative, and Mobile Symposium (DICOMO 2007)*, pages 60–65. Jul. 2007. In Japanese.

- Hideya Ochiai, Satoshi Matsuura, Masato Yamanouchi, Hideki Sunahara and Hiroshi Esaki. Evaluation of an Embedded OLSR Node for IP Sensor System. *Technical Report of the IEICE*, Vol.107, No.151, IA2007-26, pages 97–102, Jul. 2007. In Japanese.

- Hideya Ochiai, Satoshi Matsuura, Hideki Sunahara and Hiroshi Esaki. Architecture for Centralized Management of Sensor Data Upload Stream. *Technical Report of the Eighth Workshop on Internet Technology (WIT2007)*, pages 90–96. Jun. 2007. In Japanese.

- Shinichi Doi, Satoshi Matsuura, Kazutoshi Fujikawa and Hideki Sunahara. Overlay Network Considering the Time and Location of Data Generation. *Technical Report of the IPSJ 48th Programming Symposium.* pages 65–76. Jan. 2007. In Japanese.

- Shinichi Doi, Satoshi Matsuura and Hideki Sunahara. An Infrastructure for Environmental Information Gathered by Fixed-point and Mobile Sensors. *Technical Report of the IPSJ Multimedia, Distributed, Cooperative,*

and Mobile Symposium (DICOMO 2006), pages 801–804. Jul. 2006. In Japanese.

- Kenichi Ishibashi, Takahiro Yoneda, Satoshi Matsuura, Yojiro Uo and Hideki Sunahara. Tag Match: A Communication Support System for Participants of An Event. *Technical Report of the IPSJ Multimedia, Distributed, Cooperative, and Mobile Symposium (DICOMO 2006)*, pages 581–584. Jul. 2006. In Japanese.

- Hiroki Ishizuka, Satoshi Matsuura, Hideki Sunahara and Hiroshi Esaki. Live E! Project – A information infrastructure leveraging weather sensors. *Technical Report of the IEICE 4th Sensor Networks Conference* 2006-05-SN, No.20. May. 2006. In Japanese.

- Shinichi Doi, Satoshi Matsuura, Hirokazu Kobayashi, Hiroki Ishizuka, Makoto Kamiya, Masahiko Kimoto, Hiroshi Esaki and Hideki Sunahara. Live E!: A study of weather data distribution and its utilization. *Technical Report of the Internet Conference Work in Progress (IC2005)* pages 145. Oct. 2005. In Japanese.