第 XXI 部

環境情報の自律的な生成・流通を可能にするインターネット環境の構築

第21部
環境情報の自律的な生成・流通を可能にするインターネット環境の構築

第1章 Live E! ワーキンググループ2006年度の活動概要

Live E! ワーキンググループでは気象センサを中心にセンサデータを誰でも自由に利用できる情報基盤の構築を行っている。

今年度は基盤システムの拡張に加え、国内の他団体や海外との連携も強化された。現在全国に気象センサが設置され、東京、倉敷、広島がとくに集中して置かれている。海外はフィリピン、タイ、カナダにセンサが設置され、今後も拡大を続ける予定である。現在のセンサ総数（vaisala）は170余りである。

以下、活動内容をまとめる。

第2章 Live E! の活動概要

Abstract

Recently, Technology of sensor networks develops rapidly. In addition, people have started utilizing many kinds of sensors. Sensors independently collect environmental information all over the world. Unfortunately those sensed data are not shared and open to the public. Therefore, we have constructed super-large-scale sensor network system to share sensed data collected from sensors all over the world. We call such a project Live E!. In a part of Live E!, To sense a public area in whole, we assume public objects (mailbox, bus stop, dumpster) uniformly allocate in a district evenly is equipped with a sensor and a wireless device. The public objects often does not have connectivity to a network. Those are Isolated Wireless Sensor Nodes (ISNs). Then, we need to consider the way of collecting from the ISNs. Accordingly, we utilize a Patrol Node (PN) that moves around ISNs, and collects sensed data from ISNs. The ISN stores the sensed data until the time PN comes back. However, because the communication time to the PN depends on the speed of the PN, ISNs can not necessarily transmit all the maintained sensed data to the PN. Therefore, we suggest that the ISN should send adaptive data according to the speed of PN. We propose a technique for transmitting adaptive data depending on the movement of the PN. In addition, we have implemented a prototype of our proposal and verified the effectiveness of our proposed system. Finally, we show that our system improves the performance of the sensor network.

Index Terms—Wireless Sensor Networks, Mobile node, Adhoc Networks, Delay Tolerant Networks

2.1 INTRODUCTION

2.1.1 Live E! Project

Recent advances in MEMS-based sensor technology have enabled the development of relatively low-cost and low-power sensors. They can be utilized not only in disaster-area but also in our daily-life such as buildings, parks and humans. Data from these sensors all over the world can be utilized for reducing the damages and finding the causes of abnormal environment such as a warm winter, heat island, and El Nino. However, these data from the whole-world sensors are not completely open to the public.

In contrast, our project aims to collect precise environmental data from sensors located in many places in the world and provide the public with the whole world environment information. We call this project Live E![132]. In Live E!, we utilize many sensors all over the world to construct precise environment information, where sensors are not only static nodes but also mobile nodes such as buses and trains.

2.1.2 Our Proposal

In a part of Live E!, to sense a public area equably, we assume public objects (mailbox, bus stop, dumpster) uniformly allocate in a district evenly is equipped with a sensor and a wireless device. The public objects often does not have connectivity to a network. Those are Isolated Wireless Sensor Nodes (ISNs). To collect from the ISNs, we introduce patrol nodes (PNs) equipped with wireless devices. A PN patrols along a regular route to carry data of Isolated Wireless Sensor Nodes (ISNs). However the movement of the PN is determined irrelevantly to the existence of the ISNs. This is because the PNs can be realized by vehicles for other specific purposes such as trucks for garbage or mail collection, buses. Therefore, all of the sensed data at an ISN cannot be transmitted to the PN at a time in some cases. Based on the above consideration, we propose the system of sending adaptive data based on the communication time with PN. In our system, we have the following three steps.

(1) We summarize all sensed data based on three parameters: time series, functions of sensors and emergency.
(2) We calculate the communication time which is related to the speed of PN.
(3) We assign priority to the summarized data which depend on the data size.

In this report, we present our system architecture and prototype implementation. The rest of the report is organized as follows: Subsection 2.2 presents related work and describes the reason for using PNs approach on isolated wireless sensor networks. Subsection 2.3 provides the background and the goals of this work. Subsection 2.4 presents the design of our system. Subsection 2.5 describes the prototype of our system and the results of experiments. Subsection 2.6 presents the conclusion and future work.

2.2 RELATED WORK

In previous works, some approaches have been proposed in isolated wireless networks. In epidemic routing[258], data at a node is copies to any passing node until the data finally reaches its destination. This method can flood the copy in a network and may induce congestion. Second, there is spray and wait[233] that improves epidemic routing. To prevent congestion, this approach limits the number of copies. However, there is a trade-offs between the number of copies and delivery rate. After all, it is necessary to increase the number of copies to improve reliability of communication. Finally, Message Ferrying System[282] is a novel approach for isolated wireless sensor networks. In this approach, the authors propose that mobile nodes collect data by moving around between isolated nodes (ISNs) at a constant cycle. We construct our system to collect data from nodes both with and without networks connectivity. Therefore, Message Ferrying System is very suitable for our system. However, the movement of the PN is determined irrelevantly to the existence of the ISNs. Therefore, all of the sensed data in ISNs cannot be transmitted to the PN at a time in some cases.

2.3 BACKGROUND AND GOALS

In a sensor network with a PN, the PN keeps moving regardless of the existence of ISN. For instance, if an ISN exists at a roadside, a bus purposely does not stop or go slowly. Therefore, the size of the data transmitted at one encounter between the PN and the ISN may be restricted by the speed of the PN. To solve this problem, we propose the following two approaches.

- An ISN beforehand summarizes the sensed data to several data sets. Depending on the degree of the change of the data, an algorithm of making a summary determines whether or not it summarizes the data. To accommodate the variation in the speed of the PN, we prepare several sets of summary with different size.

- Each set of summary is associated with its priority of transmission. The highest priority is assigned to the data set that is the most adaptive size for the speed of PN. In this regard, the data sets including emergency information is assigned higher priority than any data.

2.4 SYSTEM ARCHITECTURE

2.4.1 Assumptions and Action of Nodes

We assume that all nodes are personal computers that equipped with wireless devices. Additionally, let us assume that wireless link quality and resource of nodes are constant. We modified AODV routing protocol[186] to allow delay of a patrol. We explain both assumptions and actions of each node as follows.

1) Isolated Wireless Sensor Node (ISN)

a) Assumption of ISN

An ISN is fixed at the location and equipped with several sensor devices (i.e., temperature, humidity, atmospheric pressure, wind speed, and wind direction). Furthermore, the node does not has available connectivity to the network infrastructure.

b) Actions of ISN

ISN begins sensing and waits until it discovers a PN. When sensed data is acquired, the data is summarized depending on “time series”, “function”, and “emergency”. When the ISN discovers PN, the ISN receives a HELLO message including the speed of the PN from the PN. The ISN calculates the communication time with the PN. The ISN assigns priority to the summary data. Some time later after the PN leaves, the ISN discovers the PN again. This time the PN initiates transmitting data to the PN in such a way that a data set with higher priority is sent earlier.

2) Patrol Node (PN)

a) Assumption of PN

The PN is a mobile node and moves around at a constant cycle and predetermined route. We assume vehicles for other specific purposes such as trucks for garbage or mail collection, buses as the PN.

b) Action of PN

The PN continues broadcasting HELLO messages including own speed. The PN sends data received from ISNs to a wireless connected internet node at the terminal of the patrol.

3) Wireless Connected Network Node (WCN)

a) Assumption of WCN

The WCN is fixed nodes and has available connectivity to the network infrastructure. The WCN is the terminal of the patrol nodes. We assume that the WCN sets up in the post office, disposal center and bus terminal.

b) Action of WCN

When the WCN discovers the PN, the WCN recognizes the PN. The WCN receives a data from the PN. At the same time, the node sends the data to Live E! data base server in the internet.

2.4.2 Architectures

Our system is based on wireless communication and composed of an ISN, a PN, and a WCN as shown in Figure 2.1.

1) Initialization

When an ISN is begins its operation, it does not recognize a PN. Hence, the ISN waits until it discovers the PN for the first time. When the ISN discovers the PN, it calculates the communication time with PN based on the speed of PN contained in the received HELLO message and assigns the priority to the data sets based on the communication time.

2) Summarization of Sensed data

Sensed data is summarized depending on “time series”, “function”, and “emergency”. We define seven units of “time series”.

- Month
- Week
- Day

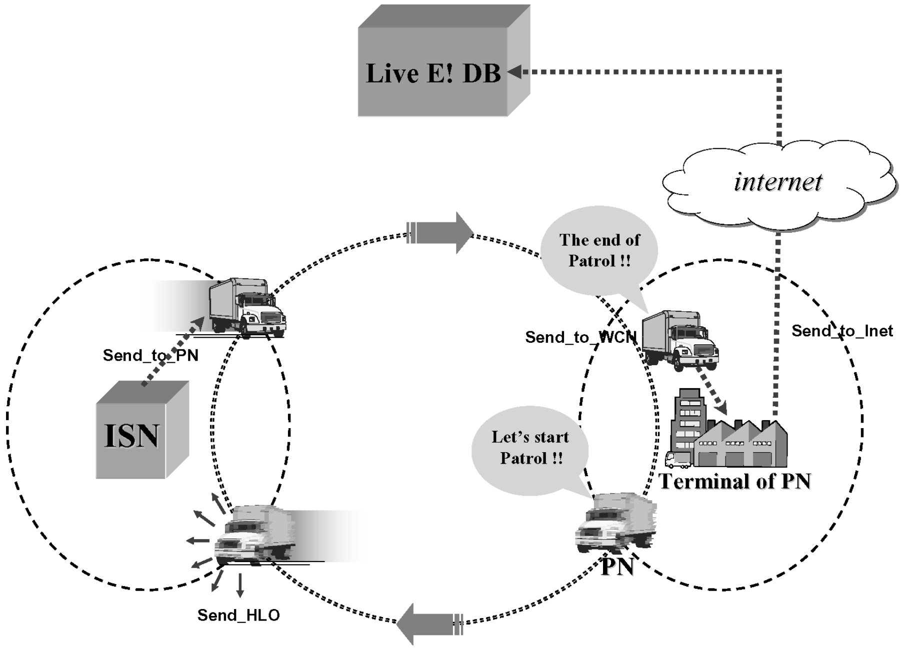

Fig. 2.1. System Architecture

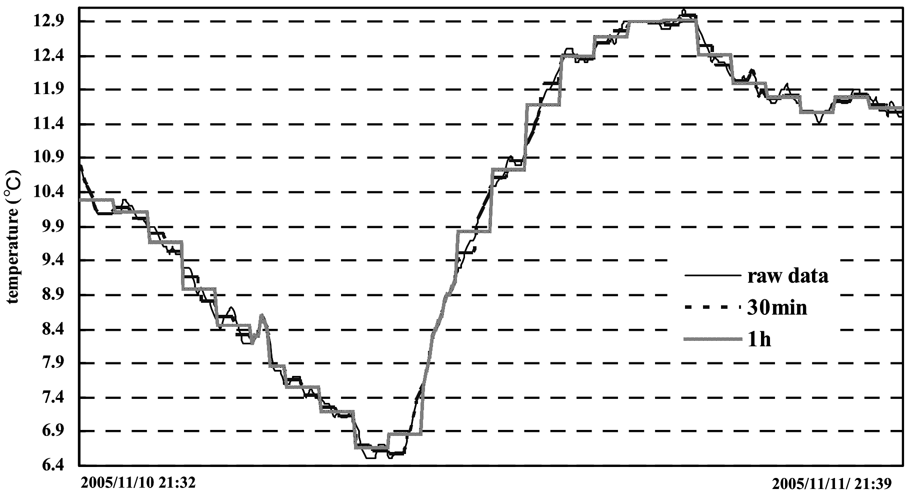

Fig. 2.2. Summarized sensed data in 1 day

- 12 hours
- 6 hours
- 30 minutes
- 5 minutes (raw data)

At the beginning, when an ISN has acquired sensed data, the node measures a change rate of data with a function (i.e. temperature) during unit of the "time series". Second, if the change rate is large which cannot be neglected, the sensed data is not summarized. In contrast, if the change is sufficiently small to be summarized, a larger unit. We verify our algorithm of summary. The verification utilizes temperature of the physical data (2005/11/10 21:32–2005/11/11/21:39) of sensor which installed at our university. Figure 2.2 shows raw data and thirty minuets, an hour summary of data during a day. The summary data during each term (thirty minutes, an hour) receives the influence of the preview state. Therefore, the value of start is different. The summary data expresses not only the average data but also detailed data.

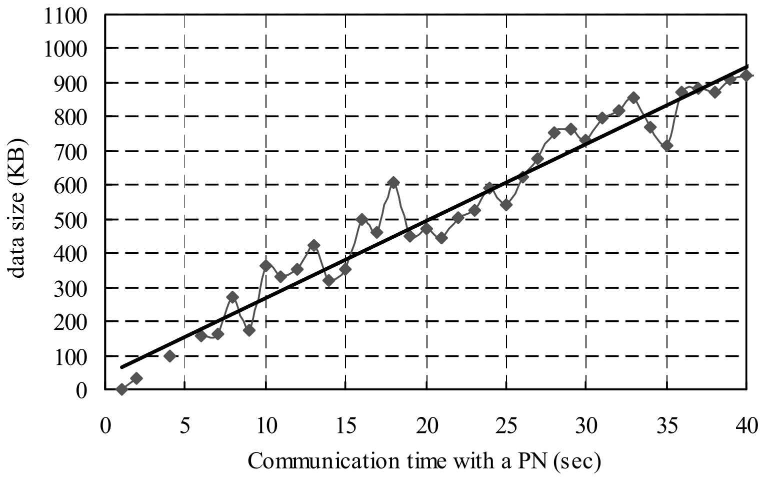

Fig. 2.3. Available transmission data size in communication time

3) Extraction of Emergency

When there is a rapid change in sensed data, that change becomes important index indicating danger in the physical world. For example, a rapid change of temperature indicates the index of fire. Therefore, a rapid change of sensed data should be transmitted to PN absolutely. We define "emergency" as the rapid change of sensed data. The sensed data is assigned the highest priority.

4) Priority Scheduler (PS)

When ISN transmits data to PN, it is necessary to forward adaptive data set sequentially according to communication time. PS assigns summarized data set to the priority for decision of forwarding order. First, PS calculates transmission data size corresponding to communication time. Second, PS searches the most adaptive data size from summarized data sets and assigns the priority.

2.5 PROTOTYPE AND EXPERIMENTATION

We implement the prototype and experiment in the physical world (surrounding of our university). In this prototype, There are three nodes (ISN, PN, and wireless connected internet nodes) equipped with wireless device (IEEE 802.11b). In order to experiment according to the difference of the speed of PN, we utilize the vehicle with different average speed as PN: "bicycle", "motorcycle".

2.5.1 Threshold of Priority Scheduler

The priority determines the threshold which is available transmission data size in communication time. When we conduct the experiment using this prototype, we should derive the threshold with two situations. Therefore, we actually measure the data size that can send in communication time. Figure 2.3 shows measurement results. Based on Figure 2.3, available transmission data size increases almost proportionally in our experimental environment. Expression 1 is obtained by the least square method.

Expression 1: $$y = 23.098x + 29.379$$ $$R^2 = 0.9554$$

Coefficient of Expression 1 is regarded as approximately one. This shows that the reliability of the regression line is high. We use Expression 1 to calculate the threshold of priority decision in our experiment.

Next, we measure relations between the speed of PN and available communication time in our experiment. Figure 2.4 shows the measurement result in two situations where PN turns around in the speed (20 km/h, 40 km/h). Each value is substituted in Expression 1, the threshold of priority is then decided and shown in Table 2.1.

2.5.2 Experimentation

We set up each node in the physical world as

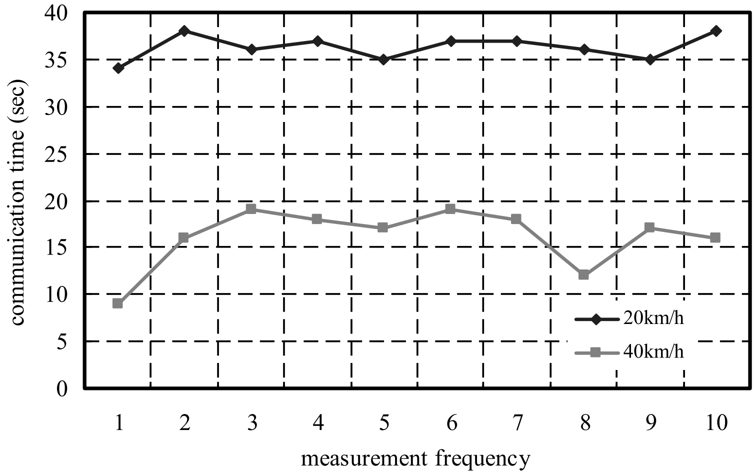

Fig. 2.4. Communication time with PN speeds

Table 2.1. Relation of the speed of node and data size

Node	Speed	Data size
motorcycle	40 km/h	260 KB
bicycle	20 km/h	722 KB

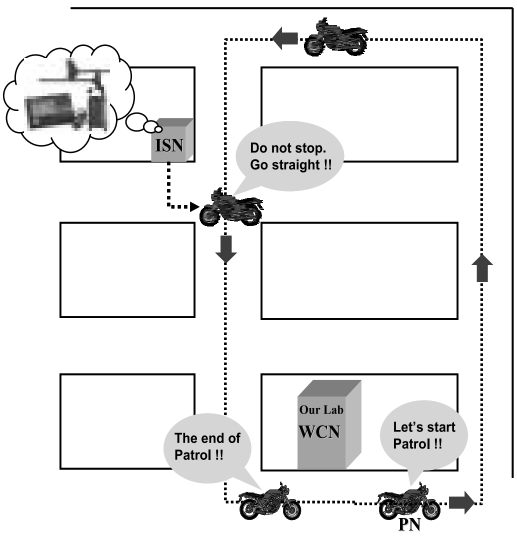

Fig. 2.5. Outline of experimentation

shown in Figure 2.5. "Bicycle", and "Motorcycle" as PN moved 20 km/h, and 40 km/h respectively. Figure 2.6 shows PN of motorcycle equipped with several devices. We verified effectiveness of our proposal.

1) Verification of Our System Performances

We compare our system that considers the speed of PN with system only which PN is utilized. Raw data size used in the experiment is

Fig. 2.6. PN of motorcycle

Table 2.2. Data size of summary data set

time series	Data size
30 minutes	365 KB
1 hour	313 KB
6 hours	264 KB
12 hours	218 KB
Da	181 KB
Wee	163 KB
Month	148 KB

563 KB. Table 2.2 shows the size of data set that we summarized the raw data. Case of motorcycle (40 km/h), ISN can send only 260 KB. Therefore, if PN does not consider the speed of PN, PN can receive only 46.2% of raw data. However, in our system, PN can receive 100% though data is summarized. Consequently, when ISN transmits data, the efficiency of our system is better than the sensor network where PN is utilized.

2.6 CONCLUSION

We propose the system of sending adaptive data based on the communication time with PN. We summarize all sensed data based on three parameters and calculate the communication time which is related to the speed of PN. Furthermore, we assign priority to the summarized data which depend on the data size. Based on the experiment of our prototype, delivery rate has improved more than the system which does not utilize our proposal.

2.7 FUTURE WORK

For our future work, our system needs to reflect wireless link quality. This is because that there are many factors to attenuate the wireless link quality in the physical world. Now, we are implementing a system which two or more PNs move round an ISN. Additionally, we should also consider not only "a communication time with PN" but also "the orbit of PN in ISN wireless communication area of the ISN".

WIDE 報告書としては、以下の名前で報告されています。

(`wide-paper-live-e-ishizuka_2006_paper-00.pdf`)

第 3 章　Live E! ワーキンググループと ICAR ワーキンググループとの連携

3.1 はじめに

近年、急速なインターネットの拡がりとともにさまざまな情報を大規模に流通させることが可能になった。また、インターネットへの接続が容易になった気象観測ユニット（「デジタル百葉箱」など）が登場し、信頼度の高い情報を高密度に収集できるようになっている。このセンサによって集められた地球環境に関する情報（環境情報）をインターネットを用いて効率よく配信することで、多様な分野における利活用を期待できる。たとえば地球温暖化などの環境保護対策、教育における物理学関連の教育材料、災害情報や環境情報の提供といった公共サービス、ビジネスアプリケーションへの応用などが挙げられる。インターネットを通じて誰もがこれらの環境情報を自由に利用できるように Live E! プロジェクト（以下、Live E!）[132] では自由な情報基盤の構築を目指している。

Live E! では独自にセンサを設置し、環境情報を収集している。高密度にセンサを設置することを目標としているため、安価なセンサを多種類設置している。また、既に設置されているセンサに対して共通化を呼びかける活動も行っている。たとえば気象庁の取り扱っている気象情報、高速道路に設置されている風速情報などが挙げられる。こういった既存のセンサの情報は独自の情報として公開されていたり、または内部利用の目的として取り扱われている場合が多い。しかし、それぞれが独自に情報を利用することは、環境情報の密度と量を減らし、全体として不利益に繋がる可能性がある。そこで Live E! では、基本となる環境情報の所有権を放棄し、社会全体で共有することを理想として活動している。

本研究では Live E! で取り扱っている固定センサに加えて環境情報の収集場所を容易に移動可能なセンサ（移動センサ）を取り扱った。また、移動センサの対象として自動車を選択した。自動車はインターネットへの接続性を持つことが期待されている。移動センサはインターネットへの接続性が問題となるが、自動車は将来的にその問題を解決すると期待できる。また、自動車には既に多くのセンサが搭載されており、新たなセンサを設置することなく環境情報の収集が可能になる。自動車をインターネットへ接続するための活動を行っている WIDE プロジェクト [272] iCAR ワーキンググループと連携し、環境情報の収集を行った。今回の取り組みでは自動車に搭載されているセンサを用いるのではなく、iCAR ワーキンググループの開発した IPv6 センサを用いた。このセンサから収集した情報を Live E! と同形式のデータベースに蓄積し、Live E! と同様の形式で提供を行った。この取り組みにより、固定センサ・移動センサの違いを感じることなく環境情報の収集が可能になった。また、実験を行って実際の気温を測定し、気温を地図上から確認できるように実装を行った。

本報告書では、第 3.2 節において Live E! における情報管理について、第 3.3 節において、Live E! への移動センサ導入について述べる。また、第 3.4 節にて移動センサによって実際に環境情報を収集した実験について述べる。

3.2 Live E! における情報管理

本節では Live E!における環境情報の管理について説明する。第 3.2.1 節で Live E! における情報の蓄積について、第 3.2.2 節では情報の提供方法についてそれぞれ説明する。

3.2.1 情報の蓄積

Live E! ではユーザ側にセンサの違いを意識させないために、センサのもつ「機能」に着目して最小単位を定義している。複数の機能を有するセンサは、その機能ごとに分解して複数のセンサと定義している。例えば気温・湿度・気圧を測定できるセンサは、気温センサ・湿度センサ・気圧センサの複数のセンサであると定義する。2006 年 4 月現在、Live E! で取り扱う最小単位のセンサは表 3.1 のとおりである。

奈良先端科学技術大学院大学で運用している Live E! のデータベースではこの最小単位のセンサの考え方をもとにデータベースを構築している。データベースの内容は 1 つのセンサを 1 つのテーブルとして構築し、それぞれのテーブルには表 3.2 の列を設けている。各センサの値をデータベースに挿入するときは、表 3.2 の値をすべて指定する必要がある。表 3.2 の構成は、環境情報に強く関連性がある情報をまとめている。どのセンサで取得されたかという情報（ID)、いつ取得されたかという時刻 (Time)、どこで取得されたかという場所 (Latitude、Longitude、Altitude) が、取得された値 (Value) に強く関連していると考えている。このほかにも環境情報に付随する情報はある。たとえばセンサの精度や設置情報といったものである。これらの情報はセンサそのものに付随した情報と考え、ID をキーとしたプロファイル情報に格納している。以上のような環境情報の管理を行うことで、収集した環境情報を共通化して取り扱っている。

表 3.1.　センサの種類

センサ名	内容	単位
Temperature	温度データ	℃
Humidity	湿度データ	%
WindSpeed	風速データ	m/s
WindDir	風向データ	度
RainFall	雨量データ	mm/h
Pressure	気圧データ	hPa
CO2	CO_2 濃度	ppm
dew_point	露点データ	℃
Photo	画像データ	(URI を格納)

表 3.2.　各テーブルの内容

列名	内容
ID	センサを識別できる ID
Time	値取得時の Unix タイム
Value	取得した値
Latitude	値取得時の緯度
Longitude	値取得時の経度
Altitude	値取得時の高度

3.2.2 情報の提供

Live E! ではアプリケーションで環境情報を容易に利用してもらうため、SOAP/XML によるサービスから環境情報を提供している。これらのサービスは最小センサごとに用意し、新たなセンサが増えても容易にサービスの提供が可能である。サービスの内容は表 3.1 に記述されているセンサ名と同一のサービスを提供している。また、センサの取得された情報とは別に、センサそのものの情報として Profile という名前の SOAP サービスから提供している。このサービスからは前述のプロファイル情報が取得できる。

SOAP のサービスは Java によって記述し、Tomcat 上で動作する Apache Axis によって提供している。また Tomcat と Apache を連動させることにより、TCP 80 番ポートによるサービスの提供を行っている。サービスの提供するメソッドは抽象クラスである Sensor クラスに記述し、各センサは Sensor クラスを継承することで同一のメソッドを提供している。Sensor クラスに実装され、各センサが提供しているメソッドを表 3.3 に示す。

また Live E! では、Web 上からの情報提供[1] も行っている。このページはブラウザからの閲覧が可能であり、即座に情報を取得することができる。取得できる情報はグラフ形式や CSV 形式に整形されており、アプリケーションでも利用することができる。

表 3.3.　メソッドの種類

メソッド名	内容
setData	値をデータベースに格納
getCurrentData	現在の値を 1 つ取得
getDataByTimespan	指定された時間の値を取得

1　現在は Live E! メンバー及び WIDE プロジェクトメンバーしか閲覧することができない。

3.3 移動センサの導入

Live E! では従来の固定センサに加え、移動センサによる環境情報の収集を行った。第 3.3.1 節にて移動センサの有用性を、第 3.3.2 節にて Live E! と iCAR との連動を説明する。

3.3.1 移動センサの有用性

従来の固定センサによる情報収集は、短い時間間隔での定点観測が可能であった。しかし、1 つのセンサでは 1 地点の情報収集しか行えず複数地点の情報を得ることはできない。これに対して、移動センサでは、短い時間間隔での定点観測は難しいが、1 つのセンサで複数地点の情報収集が可能である。短い時間間隔で定点観測した情報は有益であるが、場合によっては、ある時間帯の複数地点での情報が有益となる場合がある。たとえばヒートアイランド現象の解析では、ビル風などの影響によって短い距離で気温差が発生する。この場合、気温を定点観測することに加え、複数地点での気温測定が重要である。

また、コストの面でも移動センサは有用である。短い距離の複数地点で情報収集を行う場合、固定センサを扱う方法では多量のセンサを設置することが必要になる。固定センサを高密度に設置することは現実の場合大きなコストを必要とする。しかし、移動センサを用いることは、高密度な情報収集を 1 つのセンサによって賄うことができる。車や鉄道といった既存の移動体にセンサを取り付けることで、小さなコストで広範囲の情報を収集することが可能になる。以上のように移動可能なセンサによる広範囲での情報収集は、環境情報の効果的な利用を促進することを期待することができる。

センサを搭載できる移動体としてはさまざまなものが考えられる。その中で自動車は、Navigation System などのコンピュータを搭載している点や、個人利用目的の多さから、インターネットへの接続性を期待されている。また、豊富な電力供給源、Navigation System の GPS センサ、自律的な移動が可能といった点において、移動センサとして有効な利用を期待することができる。以上のようなことから、センサを搭載する移動体として自動車を対象とした。

3.3.2 Live E! と iCAR の連動

自動車へのセンサ搭載を行うにあたり、WIDE プロジェクト iCAR ワーキンググループ（iCAR）と連携して実装を行った。iCAR とは自動車をインターネットにつなげることと、その環境におけるアプリケーションに関する活動を行っている WIDE プロジェクト内のワーキンググループである。本研究を進めるにあたり、iCAR の収集データと Live E! の収集データを連動させる作業を行った。

表 3.4. iCAR データベース

テーブル名	内容
Temperature	温度データ
Humidity	湿度データ
horizontalacceleration	水平方向加速度
verticalacceleration	垂直方向加速度
direction	方向

iCAR では収集した情報を SNMP によってサーバ側が集めている。これは Live E! で収集している方法とは違う。しかし、データの提供方法が統一していればユーザ側が同一形式のデータとして扱うことができる。そのため、Live E! と iCAR を連動させるために、データ収集方法を変更するのではなく iCAR で使用されているデータベースの内容に変更を加えた。

移動センサの情報を扱う場合、センサに対して位置情報を付随させるのではなく、収集した情報に対して位置情報を付随させる必要がある。Live E! では環境情報をより汎用的に扱うために、表 3.1 のように収集した情報に対して位置情報を付随させている。そのため、Live E! の形式では移動センサの収集した情報も、固定センサのものと同様に扱うことができる。そこで iCAR のデータベースの内容を Live E! のデータベースの形式に合わせ、Live E! の環境情報と同様に扱えるよう変更した。iCAR のデータベースは表 3.4 に示すテーブル名に変更し、各テーブルの内容は表 3.1 の内容を用意した。また、Live E! と同様に SOAP/XML による情報提供も行った。図 3.1 に iCAR と Live E! の全体概要図を示す。

3.4 実験

移動センサによって収集された環境情報を Live E! の環境情報として扱えることを実証するため、気温取得の実験を行った。実験では自動車に温度センサを設置し、インターネットを通じて温度の取得を行った。実験概要を第 3.4.1 節に、実験結果と考察を第 3.4.2 節に示す。

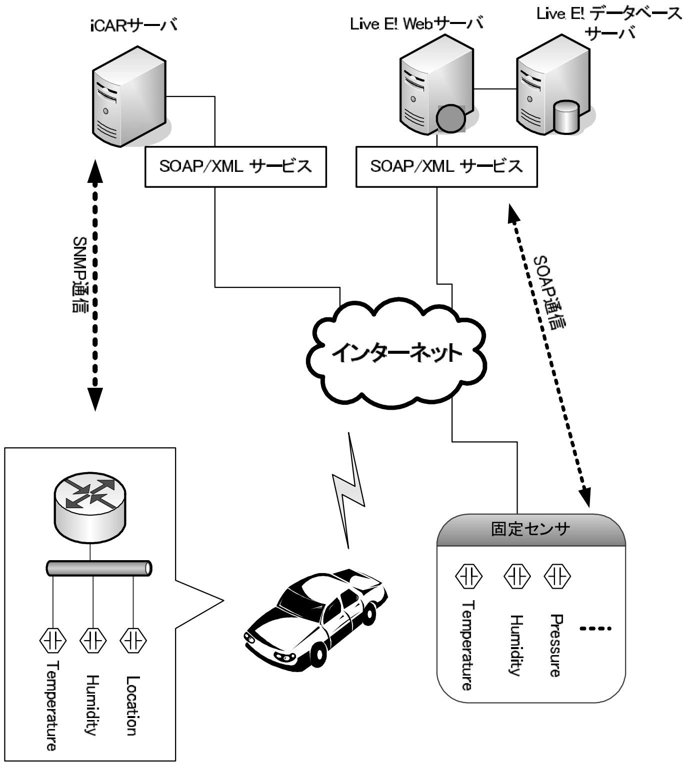

図 3.1.　全体概要図

3.4.1 実験概要

実験は奈良先端大の周辺において行った。11 時 25 分〜12 時 30 分の間に奈良先端大の周辺を走行し、温度情報を収集した。その際、なるべく外気の温度を取得するように気温センサを自動車運転席の窓際に取り付け、情報を収集した。

車載センサが収集した温度情報は Live E! サーバによる収集を行った。同様の動作が可能であることは iCAR サーバでも確認している。また収集した温度の情報は地図上で確認できるように、Web ページを作成した。地図上に温度情報を記載するために、Google Map API を用いた。作成した Web ページは、Live E! の SOAP サービスを呼び出し、返ってきた値を Google Map API に渡すことで地図上に温度情報を表示する。Web ページ上では温度情報を取得した地点にアイコンが表示され、温度によってアイコンの色が変化することで、気温の違いを視覚的に確認することができる。

以上のように今回は気温の収集を主に行ったため、方位・加速度センサは設置せずに実験を行った。今回行った実験の概略図を図 3.2 に示す。

3.4.2 結果と考察

図 3.3 に Web ページに表示された地図を示す。

実験では、取得された温度が 32.5 度〜36.0 度の間で取得された。しかし、奈良先端大に設置されている固定センサでは、同時刻帯における気温は 24.9 度〜26.1 度と記録されている。これは車内の気温が高温になっていたため、センサに車内の気温が影響し、全体的に高温に感知されたためと思われる。固定センサに一番近い位置で取得した移動センサの気温は、固定センサの気温と 8.7 度の差があった。そのため、全体の温度に対して –8.7 度の補正をかければ実際の気温に近い値が取得できると考えられる。

また、移動センサで取得した気温を時間軸に並べると、時間には関係なく気温が大きく変動していることがわかった。固定センサで取得された値は時間経過とともに気温も増加しているため、この結果は取得した地点によって気温が違うことを表しているといえる。また、値が多く取れた北西のデータを解析すると、12 時〜12 時 30 分にかけて 1.2 度の温度上昇が見られる。固定センサによって得られた同時刻帯の気温は 1.5 度上昇している。このことは、移動センサの収集した気温は外気温の変化に影響して

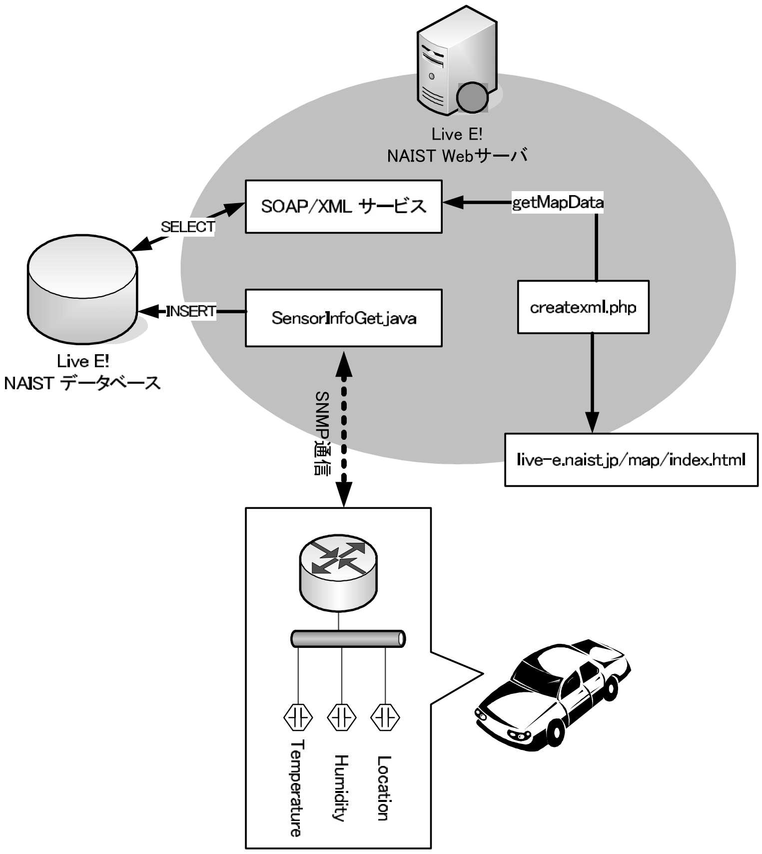

図 3.2.　実験環境

図 3.3.　実験結果

いると推測することができる。

ただし、データの量が少なく設置条件も悪いため、正確な検証を行ったとは言い難い。正確な検証を行うには高密度な固定センサと比較する必要がある。また、自動車の速度と気温の関係などにも留意する必要があるため、実際の気温として扱うためにはより高度な補正が必要である。

3.5 おわりに

今回行った実装によって、移動センサの情報をLive E!の環境情報として扱えることを確認した。また、移動センサを扱うことによって、従来の固定センサでは得られなかった広範囲・高密度の環境情報の収集が可能であることを確認した。

今後、こういった移動センサによる情報収集として、さまざまな可能性が残されている。たとえば、船舶・飛行機といった移動体にセンサを取り付けることで固定センサの設置が難しいような場所の環境情報も容易に得ることができる。また、移動センサが固定センサの環境情報を拾い上げる方法も考えられる。Live E!では固定センサの情報を扱うためにインターネットへの接続性が必要である。しかし、インターネットへの接続が無い地点での定点観測を行う方法として、移動体がサポートできる可能性がある。たとえば、固定センサが収集した情報を近隣を通過する自動車に預け、その自動車が代理で情報を提供することで、インターネットへの接続が無い地点で

あっても定点観測の情報提供が可能になる。定期的に自動車が通るような場所、たとえばバスやトラックの通り道といった場所であれば、自動車が環境情報を拾い上げる枠組みは、インターネットへの接続を提供することよりコストの面で有利であると考えられる。今後は環境情報を取得できるセンサを増やすと共に、新たな情報収集モデルの構築、そして大量の情報を管理することに関して研究を進めていきたい。

謝辞

今回の実験を行うにあたり協力していただいたWIDE プロジェクト iCAR ワーキンググループの皆様、Live E! プロジェクトの皆様に心より感謝いたします。

WIDE 報告書としては、以下の名前で報告されています。

(`wide-paper-live-e-shinido_2006_paper-00.txt`)

第 4 章　P2P を利用したセンサデータの分散管理

Abstract

With the rapid rise in the demand for location related service, communication devices such as PDAs or cellar phones must be able to search and manage information related to the geographical location. To leverage location-related information is useful to get an in-depth perspective on environmental circumstances, traffic situations and/or other problems. To handle the large number of information and queries communication devices create in the current ubiquitous environment, some scalable mechanism must be required. In this report, we propose a P2P network system called "Mill" which can efficiently handle information related to the geographical location. To simplify the management of the location related information, we convert two dimensional coordinates into one dimensional circumference. Using this technique, Mill can search information by $O(\log N)$. Mill does not adopt any flooding method, and it reduces the amount of search queries compared with other P2P networks using flooding. DHT networks also do not leverage flooding and have good features. Simulation results show that the performance of Mill is good as well as other DHT networks. DHTs support only exact match lookups. The exact match is not suitable for searching information of a particular region. Mill provides an effective region search, by which users can search flexibly location-related information from small regions to large regions.

4.1 Introduction

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

If we use a large number of information these devices provide, we can obtain detailed and real time information. Gathering information based on geographical location can be effective for judging traffic situation, weather condition, and other circumstances. For example, if we gather rainfall information based on geographical location throughout a city, we know where rain clouds are exactly. This information is useful to the people riding a bike, climbing a mountain, and doing other things. However, if we can not immediately obtain this information, the value of information will be lost. Therefore, to immediately obtain some suitable information based on geographical location, a management mechanism which can handle a large number of mobile devices should be required.

Japan Automobile Research Institute (JARI)[110] experimented with IPcars (taxi; has some sensors and connectivity to the Internet). This experiment showed that gathered information from mobile devices is useful to create detailed weather information. In this experiment, a client-server approach was adopted. In the near future, it will be expected that ubiquitous computing environment come out and the number of queries for location-related information will much increase. Then servers will be much overloaded.

To decentralize information and queries, peer-to-peer (P2P) networks are widely studied. Especially, P2P network with distributed hash table (DHT) are proposed in many studies[197, 202, 238, 283]. DHTs are scalable to the number of mobile devices and are effectively adapted for entry and separation of nodes. DHTs can answer queries even if the network is continuously changing. However, there is serious defect. DHTs support only exact match lookups because of adopting a hash function. If we deal with location-related information, exact match will be disturbance. Despite geographical distance, all information is assigned absolutely different ID by a hash function. If some information is geographically close, assigned IDs are not relevant. Therefore, the exact match mechanism is not suitable for searching a particular region.

There are several P2P networks considering location. However, these P2P networks have some defects in dealing with location-related information. LL-net[117] is location based P2P network. This P2P network defines an area as a square region divided by latitude and longitude. LL-net is optimized for context-aware service, and this P2P network is efficient to find where node is and what services node has. LL-net has two kinds of special nodes (super peer and rendezvous peer). The super peer manages information about all rendezvous peers. All other peers should know the super peer in advance. A rendezvous peer exists per an area. This peer manages normal peers in its area. Besides, LL-net can not deal with attribute of time and can not gather location-related information such as temperature, speed and other values of sensors, because LL-net manages not location of information but location of nodes.

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

The rest of this paper is structured as follows. Subsection 4.2 describes requirements for information management and retrieval on ubiquitous computing environment. Subsection 4.3 presents the mechanism of Mill and explains several of its properties. Subsection 4.4 shows Mill's performance through simulations. Finally, we summarize our contribution in Subsection 4.5.

4.2 Requirements for ubiquitous environment

In ubiquitous computing environment, there are many devices including mobile phones, desktop PCs, web cameras and sensor devices. If we gather information from these devices, we can obtain valuable information. However, there are several requirements that we have to cope with.

- Scalability
  In the near future, the number of information created by mobile devices and other devices will much increase. And centralized system as like client-server model will much overloaded. It is required to handle information and search queries created by a large number of devices all over the place.
- Region search
  If sensor devices provide location-related

information, we try to obtain useful information. For example, someone wants to know weather condition around his/her office and traffic situation between his/her home and office. Therefore, a flexible search mechanism is required, which can be applied to arbitrary size of region.

- Fast search

  If we try to obtain traffic situation or weather condition, the up-to-date information must be provided. Therefore, a search mechanism should be fast.

4.3 Mill: A new Geographical-based peer-to-peer network

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

Overview

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

DHTs support only exact match queries because of adopting a hash function (e.g. SHA-1[63]). 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.

To support region search, Mill adopts not a hash function but a mapping mechanism optimized for location-related information. Using this mapping mechanism, Mill reduces search queries compared with other DHTs.

The architecture of Mill is similar to the DHTs. As Figure 4.1 shows, the architecture is hierarchy structure. 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 resolved via $O(\log N)$ messages.

2D-1D mapping

Mill divides two dimensional space into a grid cell by latitude and longitude. A grid cell is a small square region. If Mill expresses the surface

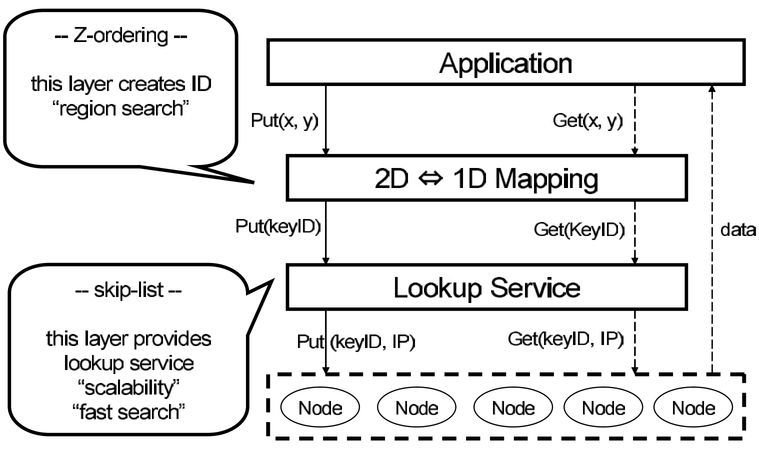

Fig. 4.1. Architecture

of the earth as 64 bits, a size of grid cell is equal to milli-meter order. As Figure 4.2 shows, suppose that each cell is assigned a 4 bit identifier. Mill manages these IDs as one dimensional circular IDs, and each Mill node is responsible for a part of circular IDs. The number of ID is created 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 (only 4-bit) to explain simply, however in real use Mill's ID space is large 64-bits. A particular region can be expressed as a range between "Start-ID" and "End-ID". For example, in Figure 4.2 ID range $(0, 0)$ correspond a square cell (region A), ID range $(0, 3)$ correspond quarter of the whole square (region B), and ID range $(0, 15)$ correspond the whole square (region C). In fact, Mill expresses a particular square region as a consecutive of cell IDs and can search range of IDs at one time for information. 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, I summarize the features of "Z-ordering" (including the features without explanation)

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

In some cases, altitude is needed. In urban areas, buildings are usually multi-story ones. Then, sensor devices are installed on different floors. Therefore, it is necessary to distinguish a sensor on first floor as being different from another sensor on second floor. It is easy to expand Mill network into 3 dimensions. The IDs of Mill network are created by alternating x-bit, y-bit and z-bit in 3 dimensions.

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. As Figure 4.3 shows an example, the overlay network is consist of 7 nodes (0, 4, 6, 9, 11, 12, 14). The node whose ID is 0 handles a part of the circular ID space from ID 0 to 3. This node has 4 connections with other nodes ID 4, 6, 14, and 12.

A new node joins Mill network by the following protocol. Figure 4.4 shows an example.

1. The new node creates 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 define this node as initial node. And the new node

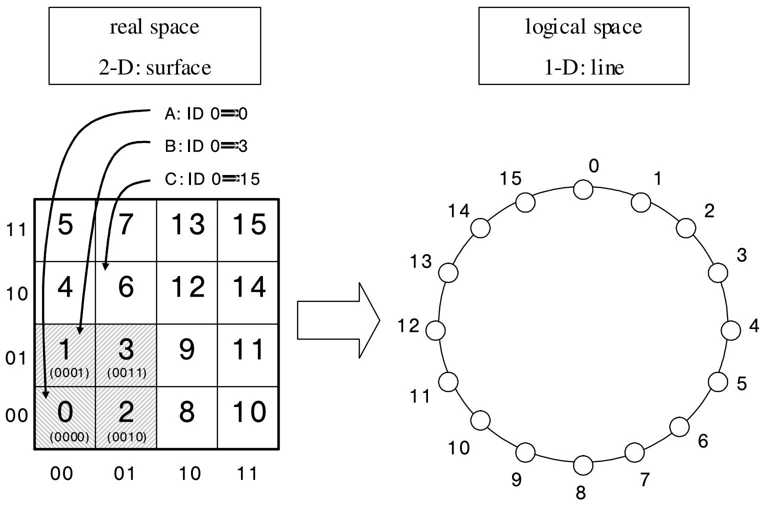

Fig. 4.2. 2D-1D mapping method

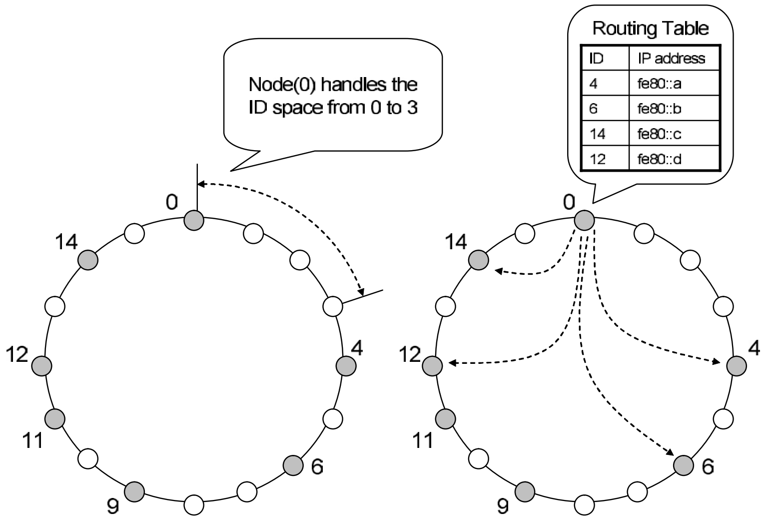

Fig. 4.3. Handle ID-space and routing table

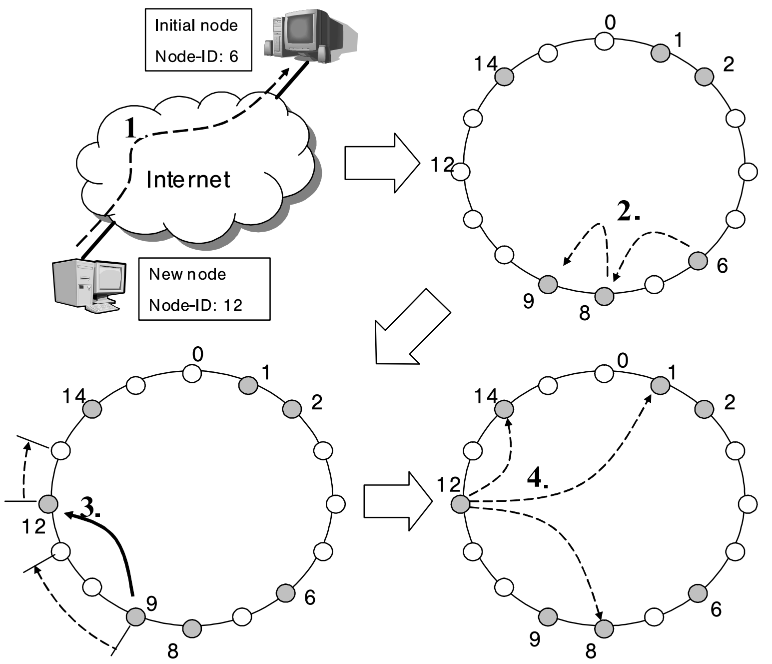

Fig. 4.4. Join protocol

sends Node-ID and IP-address to the initial node. As Figure 4.4 shows, the new node creates 12 as Node-ID according to its actual location. Then, the new node sends Node-ID (12) and IP-address to the initial node (Node-ID: 6).

2. The initial node sends this message to clockwise side node, and the clockwise side node sends this message in the same way. As each node send 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

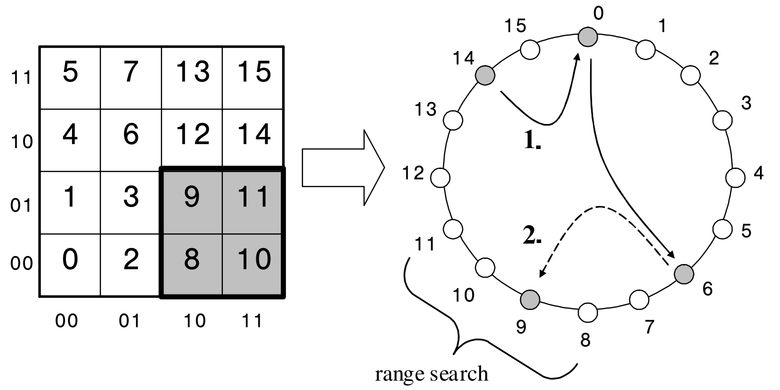

Fig. 4.5. Region search

this node also informs the new node about Node-ID and IP-address of neighbor nodes. The node (Node-ID: 9) assigns the ID space $(12, 13)$ to the new node and informs the new node about Node-ID and IP-address of neighbor nodes (Node-ID: 8, 9, 14, 1). And the node (Node-ID: 9) reassigns the ID space $(9, 10, 11)$ by itself.

4. The new node informs neighbor nodes about own Node-ID and IP-address. Then neighbor nodes update their routing table. The new node (Node-ID: 12) informs neighbor nodes (Node-ID: 8, 14, 1) about own Node-ID and IP-address.

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

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 got. This ID is created by the 2D-1D mapping mechanism. Second, this node sends the ID and own 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 4.5 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 meaning that information is not found.

In practice use, information consists of ID, time, type, and value. Therefore, Mill can supports not only region search and but also time based search and type based search.

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 power of two hops away nodes as like 1, 2, 4, 8, 16 hops away. First, to know the information of a 4 hops away node, each node communicates with a 2 hops away node. Second, to know the information of an 8 hops away node, each node communicates with a 4 hops away node. Repeating this communications, a size of routing table is larger. The maximum size of rouging table is no more than $O(\log N)$. This routing table is likely to have more entries for closer nodes and fewer entries for further nodes. This list structure is called *skip-list*.

Figure 4.6 shows a search example and a clockwise routing table of the node (Node-ID: 18). The node gets the information related with ID (34) by the following search protocol.

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

This routing table reduces the searching cost to

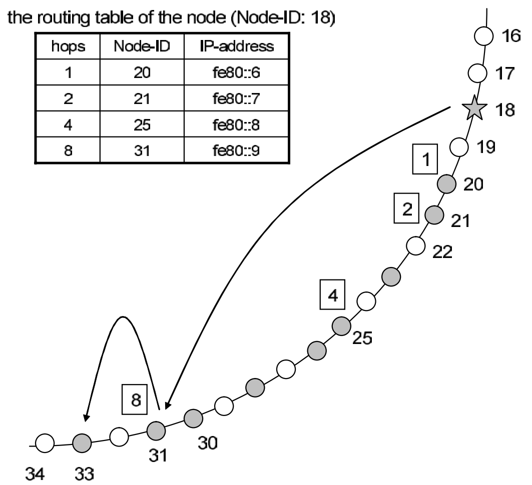

Fig. 4.6. Skip-list search

$O(\log N)$. 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.

Load balance

DHTs use hash-function for creating 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 creating IDs. On the face of things, information is not distributed in Mill network. However, in fact, information is distributed. I 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 has 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 these areas, lots of information is generated, however the size of IDs each node has is small. The size of IDs is inverse of density.

The information amount 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.

4.4 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 4.1 shows the simulation environment.

Application example

We make a sample application on the simulator.

Table 4.1. Simulation environment

CPU	Pentium4 2.4 GHz
Memory	1 GB
OS	WindowsXP SP2
programing language	Java 2 SDK ver1.4.2
the number of node	10 → 2,560
ID-space	2^{24} (4,096 × 4,096)
transfer method	random walk

This application creates the weather information. In this application 100 mobile devices are moving around, sensing temperature. After running the application, every node communicates with some other nodes and creates Mill network. Figure 4.7 shows a snapshot of this application. Small circles represent mobile nodes and dots do information of temperature. First, users determine a target region and click the button related with the target region. Second, users get information on the target region. The temperature information is plotted on the target region as dots. After we search several region, we can see the atmospheric temperature profile.

Path length

We define the path length as the number of nodes relaying a search query. As Figure 4.8 shows, if a searching method is skip-list search, the path length is $O(\log N)$ and if a searching method is clockwise liner search, the path length is $O(N)$. In fact, the path length is almost half of $\log N$, because each node has clockwise and counterclockwise information of nodes on a routing table.

Figure 4.9 shows how many search queries reach the destination. In the 160 nodes system, 80% of search queries reach the destination through 4 hops. In most cases, almost all search queries reach the destination through 8 hops. As the Round Trip Time (RTT) of mobile phones is around 500 (msec), search queries reach the destination by 3 or 4 (sec).

Now, we compare our Mill network system with other P2P network system. We express target region as "i" bits, the number of nodes in the target region as "m" nodes, and the number of nodes in the 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 the Mill network, a node searches 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 16, 20, and 10000 respectively, each search cost is as follows.

- *DHT*: $2^{16} \times \log 10000 = 603609$
- *Mill*: $\log 10000 + 20 = 29$

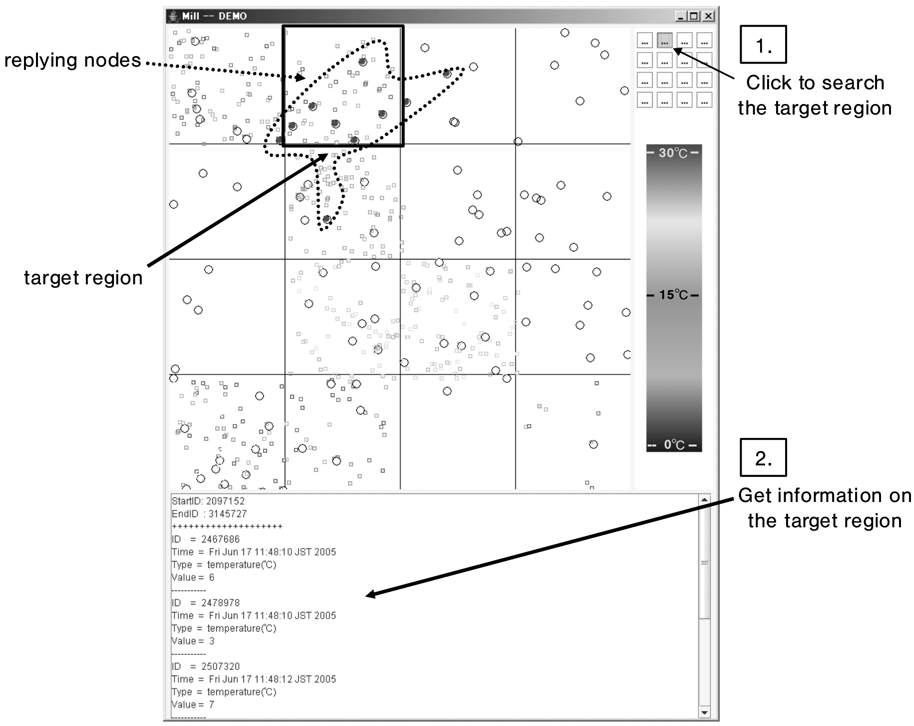

Fig. 4.7. Application Example

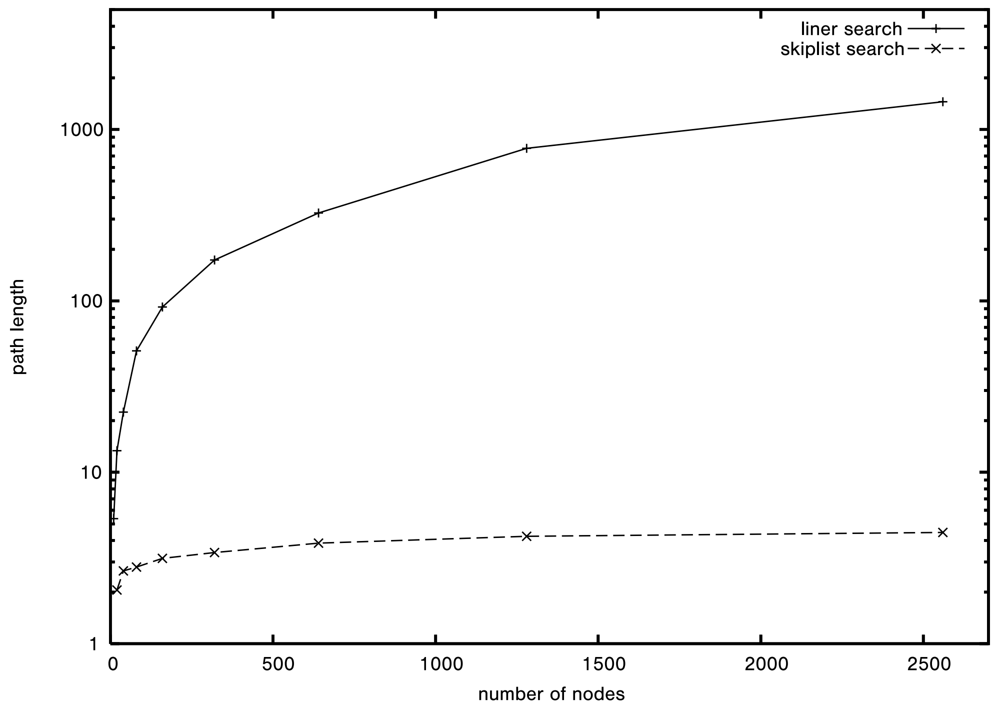

Fig. 4.8. Node vs pathlength

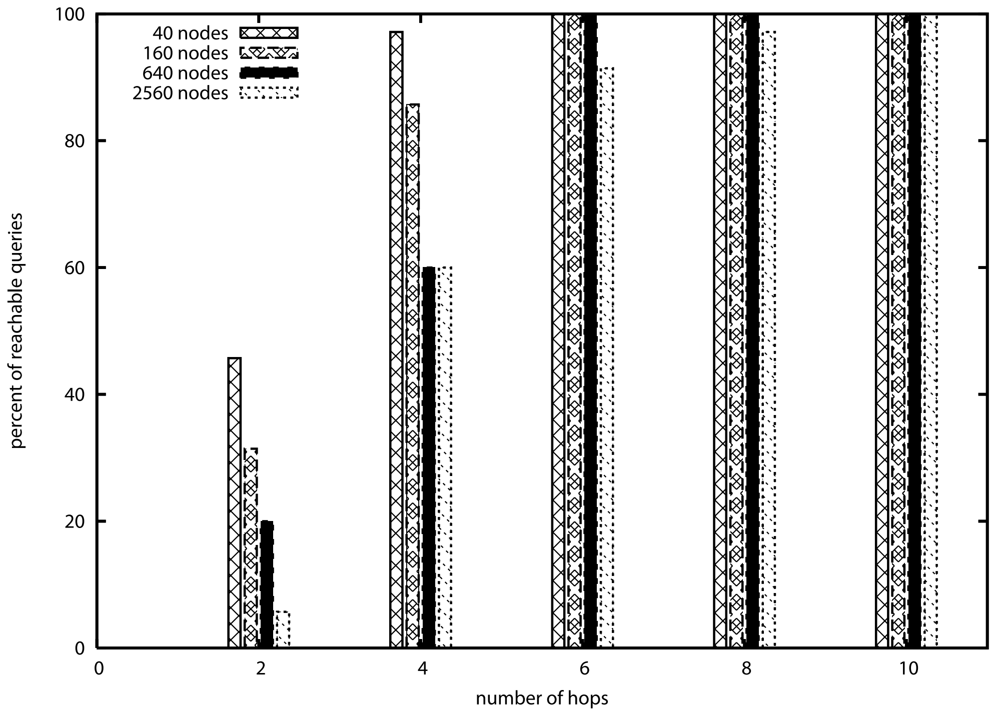

Fig. 4.9. Hops vs reachableQueries

On region search, DHTs create much larger queries than Mill does. However, if "m" increases, the performance of Mill becomes worse. The performance-degrading factor 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 between the quality of information, the density of

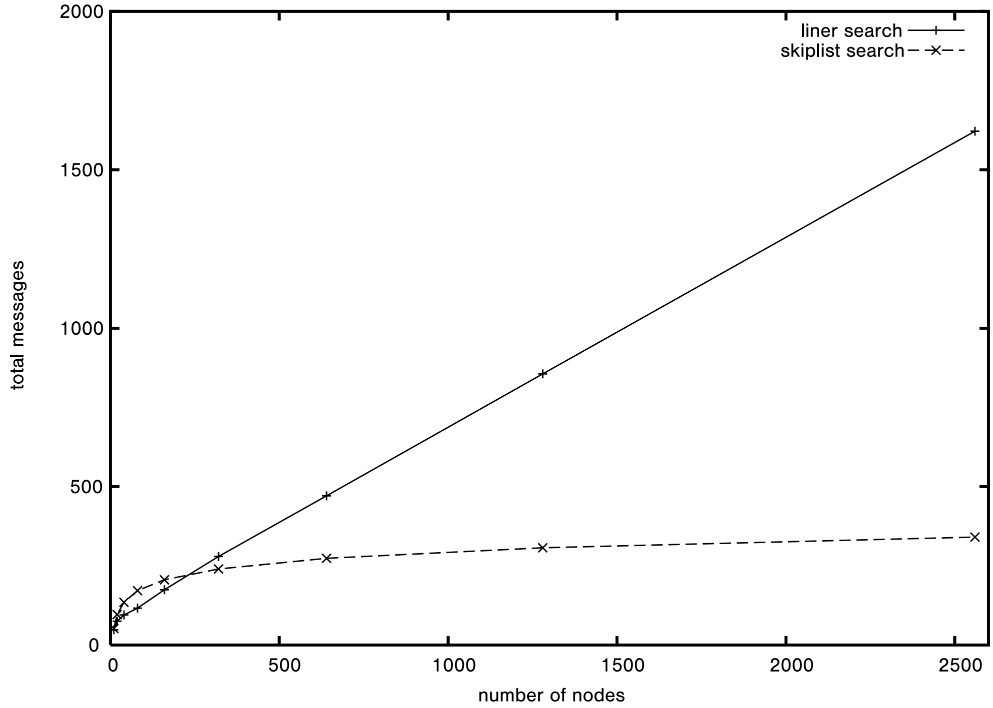

Fig. 4.10. Nodes vs messages

nodes, and the routing table. This optimization is one of the future works.

LL-net has hierarchical ID spaces to improve search mechanism. 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 related with the size of ID space.

Management cost

In a message of Mill, there are 2 types. One type is join-leave message. This message is sent if a node joins or leaves 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. As Figure 4.10 shows, if a searching method is clockwise liner search, the number of processed messages becomes larger as the number of nodes increases. On the contrary, if using skip-list search, the number of processed messages is almost $O(\log N)$.

In the Mill network, the order of search cost and maintenance cost is $\log N$. In DHT networks, the order of these costs is also $\log N$. $O(\log N)$ is effective for the increasing number of nodes, and Mill and DHTs are scalable to the number of nodes. In the LL-net network, information of every area is centrally managed by the super peer. Therefore, it seems not to be a scaleable to the number of nodes.

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 handles ID-space and circular ID-space is not divided anywhere. If Mill network works well, each node can put and get information. In this simulation experiment, a particular percent of nodes is forced to be disconnected

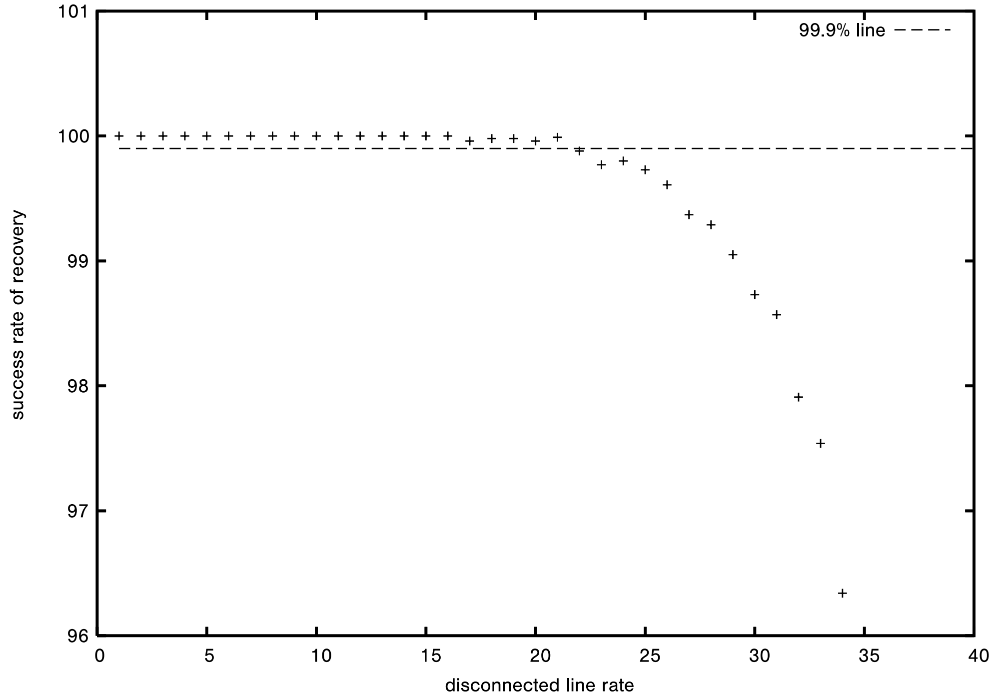

Fig. 4.11. Disconnected vs recovery

at once. Alive nodes try to recover Mill network to find other nodes by adopting a routing table. As Figure 4.11 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.

4.5 Concluding remarks

In the ubiquitous computing environment, communication devices can provide information anywhere and anytime. Therefore, information should be shared among communication devices based on geographical location. Mill enables communication devices to share information based on geographical location. In an N-node network, Mill can search information by $O(\log N)$ and each node maintains routing information for about $O(\log N)$ other nodes. Mill can recover the overlay network, even if 15% nodes become disconnected at the same time. In addition, Mill does not adopt any flooding methods, therefore Mill can reduce the number of search queries compared with other P2P adopting flooding mechanism. DHTs also do not adopt flooding methods and have good features. However, DHTs only support exact match queries. Exact match queries is not suitable to searching a particular region, because users should search all points in the region. Mill can search consecutive IDs at one time by 2D-1D mapping mechanism. Mill can also support effective region search and users flexibly search information from a small region to a large region.

WIDE 報告書としては、以下の名前で報告されています。

（wide-paper-live-e-matsuura_2006_paper-00.txt を参照）

第 5 章　まとめと今後の展開

プロジェクトが拡大し、国際的な交流も頻繁に行われるようになってきた。今まで準備してきた資産を有効活用するためにも英語ドキュメントの整備や

各国の担当者とのコミュニケーションを今まで以上に活発に行っていく必要がある。

また分散環境、無線を中心テーマに研究および基盤システムの拡張、改良を引き続き行っていく。