

時間と位置を考慮したセンサオーバーレイネットワークの提案と評価

洞 井 晋 一^{†1} 松 浦 知 史^{†1}
藤 川 和 利^{†1} 砂 原 秀 樹^{†1}

センシングデバイスの高性能化と低価格化により、高精度なセンシングデータを高密度に収集することが可能になった。このようなデータの多分野への応用を考えたとき、サービスの可用性を向上させるためにはオーバーレイネットワーク技術を用いた共有が考えられる。しかし、既存のオーバーレイネットワーク技術では、センシングデータの位置と時刻を指定した検索を行うと、クエリのフラッディングが発生したり、ノード間のデータ保持量に公平性を欠いたりすることが想定される。本論文ではセンシングデータの位置と時刻を用い、オーバーレイネットワーク上でデータを集合体へ集約する手法を提案する。提案手法により、データの集合体に対して検索を行うことでクエリのフラッディングを抑制し、またデータの集合体を分散保持することでノード間のデータ保持量を公平に保つことができる。加えて、集合体を用いデータの平均値や最大値を容易に検索することを可能とした。提案手法を評価するためにエミュレータを作成し、複数の計算機による実験を行った。その結果、トラフィックを毎秒 100 バイト程度の増加に抑えつつ、センシングデータを集合体へと集約した。また、データの取得時には集約しない手法に比べてクエリの数を平均して約 13%削減した。特に、クエリの時間幅を狭くしたときは約 76%の削減を実現した。

Overlay Network Considering the Time and Location of Data Generation

SHINICHI DOI,^{†1} SATOSHI MATSUURA,^{†1} KAZUTOSHI FUJIKAWA^{†1}
and HIDEKI SUNAHARA^{†1}

Development of sensor technology enables us to use lots of and detailed information. Then, it is difficult to deal with the vast number of information generated by sensors. In order to deal with the vast data generated by sensors, we build a new information infrastructure on an overlay network. If we share sensing data on existing overlay networks, such as DHTs, overlay network nodes send a lot of queries to other nodes. In this paper, we propose an overlay network system considering time and location on which data is generated. Our proposed method supports range search, using efficient routing and reposition data. Our overlay network also supports advanced search, for example searching average or maximal value of data, because each node manages consecutive sensing data based on time and location. To estimate the performance of our overlay network, we implement an emulator. On this emulator, our proposed overlay network aggregates sensing data and suppresses traffic. Compared with overlay networks not adopting method of data aggregation, the result of emulation shows that the numbers of queries are reduced by 13%. Especially, when users search sensing data by short time span, the numbers of queries are reduced by 76%.

1. はじめに

センシングデバイスの発達により様々な情報の取得が可能となった。たとえば従来の気象観測用のセンサでは気圧や雨量といった情報を取得しているが、高機能なセンサでは二酸化炭素濃度、窒素酸化物濃度などの情報が取得可能である。こうした高機能なセンサが

低価格化していることもあり、至るところにセンサが設置されるような環境が現実的なものになってきている。センサが提供する情報（センシングデータ）は、従来の気象予報のほかに公害問題や災害対策の基本情報として重要視されている。また、公共利用だけでなくビジネスの分野においてもセンシングデータは注目されている。センシングデータをこのような多分野に応用するためには汎用的な情報基盤の構築を行う必要がある。ここで、センシングデータの情報基盤で利用される計算機を考えると、次の 2 種類があげられる。

^{†1} 奈良先端科学技術大学院大学情報科学研究科
Graduate School of Information Science, Nara Institute
of Science and Technology

- センサに併設される計算機
- データ保存用のディスクスペース

センサはインターネットへの接続性を持たないものが多いため、なんらかの計算機を利用してインターネットへ接続する。センシングデータの可用性を向上させるために、この計算機ではセンシングデータを保存し、外部に対して利用可能にしておくことが望ましい。また、センシングデータを恒久的に保存しておくためには大容量のディスクスペースが必要である。

このような環境におけるセンシングデータを用いたサービスを考えれば、ディスクスペースに保存されたセンシングデータを利用することが一番容易である。しかし、集中管理されたディスクスペースからデータを取得するサービスでは、単一点の障害に弱く可用性の低下を避けることができない。特にセンシングデータは防災への利用が期待されており、災害時に障害が発生してもサービスを提供することが求められる。

そこで、近年注目されているオーバーレイネットワーク技術（P2P 技術）と呼ばれる仮想的なネットワークを用いた分散システムを用いてセンシングデータの管理を行うことが考えられる。オーバーレイネットワーク技術を用いることでディスクスペースにデータを集中管理させるのではなく、センサに併設される計算機を用いたセンシングデータの共有が可能になる。オーバーレイネットワーク技術を用いることは可用性の向上に加え、スケーラビリティの確保、管理コストの低減などが期待できる。本論文ではセンシングデータを扱うこのようなオーバーレイネットワークを「センサオーバーレイネットワーク」と呼称する。

センサオーバーレイネットワークでは従来のオーバーレイネットワークとは異なるアプローチが必要になる。既存の主なオーバーレイネットワーク技術は、単一のファイルをいかに効率良く共有するかが主な課題である。しかし、一般的なセンシングデータは、1つ1つのデータにはそれほど価値がなく、多くのデータが集まることで意味のある情報になる。また、センシングデータは生成された位置と時間による検索が行われる。このときに位置の範囲や時間の範囲を指定することが考えられるため、連続量に対する範囲検索を可能とする必要がある。時刻を指定したデータ取得には様々なものが考えられるが、本論文では現在時刻に近いセンシングデータは単一で利用され、長い時間の経過したセンシングデータは集合体として利用されることを想定する。なお、たとえば雨が降ったときにだけ反応するようなイベントドリブンのセンサはデータの発生がないときは「0mm」のようなデータがないことを表す

データを生成することで対応可能である。まず、現在時刻に近いセンシングデータはそのときの状態を知るために取得される。一方で長い時間の経過したセンシングデータは時間経過による状態の遷移を知るために取得される。例としてセンシングデータとして気温の情報を収集することを考える。現在時刻に近いセンシングデータは、そのときの気温を知るために取得されるのに対し、1日前であったり、1カ月前のデータはそのときの気温の遷移を知るために1日分や1カ月分のデータをまとめて取得する。今までの気温の遷移を知る場合は現在時刻までのデータをまとめて取得する必要があるが、その場合、気温の遷移を更新するために現在時刻のセンシングデータを取得し続けることが考えられる。そのため、やはり現在時刻に近いデータは単一で利用されることが多いと考えられる。

既存のオーバーレイネットワーク技術を用いたとき、センシングデータの生成された位置や時間が考慮されていないために検索時に大量のクエリ（フラッディング）を発生させてしまう。オーバーレイネットワーク上でフラッディングを行うことは、ネットワークやノードに対する負荷が上昇するため避けなければならない。また、経過時間による利用頻度の違いを考慮していないためにノードが処理するクエリの数やノード間のデータの保持量に差が生じ、ノード間の公平性を欠くことが考えられる。

本論文ではセンシングデータの生成された位置と時間を利用し、センシングデータを集合体へと集約するセンサオーバーレイネットワークを提案する。センサに併設される計算機をノードの対象と想定するため、ノードがその地理位置を移動することや、頻繁に参加・離脱を繰り返すことは考慮しない。また、それなりのディスクスペースを保持していることも想定する。このようなノードによってセンサオーバーレイネットワークを構築し、センシングデータの共有を行う。生成された位置と時刻に基づいてセンシングデータを識別するため、位置や時間を指定した範囲検索が可能になる。また、生成されてからの経過時間をもとにしてセンシングデータを集約するため、範囲検索を行ったときに発生するクエリのフラッディングを抑制することができる。また、センシングデータの経過時間による利用頻度の違いを考慮し、経過時間の短いセンシングデータは少数からなる集合体となり、時間が経過するに従い多数のセンシングデータによる集合体となる。これによって、フラッディングを抑制しながらノードが処理するクエリ数や、ノード間のデータ保持量を公平に分配することができる。

また、センシングデータの集合体から演算して得られるような値、たとえば平均値や最大値といった値の検索が可能になる。センシングデータは多くのデータが集合することで意味のある情報になるため、現実の利用ではこの集約を行うことが有用である。たとえば気温の平均値や降雨量の最大値などを得る場合、従来ならば集約されていない情報を集めたうえで演算する必要があったが、提案手法では集合体を保持している計算機があらかじめ演算して提供することができる。

本論文では2章に既存のオーバーレイネットワーク技術における問題点を述べ、3章では提案手法の詳細を述べる。4章でエミュレータを用いた評価実験について説明し、5章で提案手法に対する考察を述べる。最後に6章でまとめを述べる。

2. 既存のオーバーレイネットワーク

オーバーレイネットワークの方式は個々の計算機(ノード)のトポロジにより、Structured型とUnstructured型に大別される。Structured型ではトポロジが規定されており、そのためノード間のルーティングなどが規則的に決定される。Unstructured型ではノード間のトポロジが規定されていないため、各ノードが自由に接続しオーバーレイネットワークを構成する。前述したようにセンシングデータは集合となって意味をなす場合が多い。Unstructured型のオーバーレイネットワークでは特定の規則に従ってセンシングデータを集約することが難しい。そのため、本論文ではStructured型に注目する。

2.1 DHTを用いたオーバーレイネットワーク

Structured型のオーバーレイネットワークでは分散ハッシュテーブル(DHT: Distributed Hash Table)を用いた方式が多い。DHTでは任意の(key, value)の組を用いてデータの分散保持と高速な検索を実現する。ハッシュ関数 H を用いて $H(\text{key})$ とvalueを対応させるため、keyを用いた検索に対して一意にvalueを特定することができる。また、 H を適切に選ぶことでノードの保持データ量に偏りを発生させることなく、高速にvalueを発見することができる。

しかし、DHTを用いた手法では完全一致の検索しか行うことができない。ハッシュ関数を用いるため、 $H(\text{key})$ はkeyによってまったく異なる値をとる。そのため、keyの範囲を指定した検索を行う場合、考えられるすべての $H(\text{key})$ に対して検索しなければならない。提案手法ではセンシングデータの生成された位置と時刻をkeyにする。そのため、ハッシュを用いた検索の場合、該当するすべての位置と時刻を検索しなければならない。クエリのフラッディングが発生する。

以上のことから、DHTを用いた手法ではセンサオーバーレイネットワークを構築することが困難である。

2.2 連続量を利用可能なオーバーレイネットワーク
位置や時間のような連続量を扱うことができるオーバーレイネットワークとしてはSkipnet³⁾がある。Skipnet³⁾では連続量を汎用的に扱えるため、Mercury²⁾ではこのSkipnetを用いて位置に対する範囲検索を可能にしている。また、既存のセンサオーバーレイネットワークにはMill⁴⁾やLL-net⁷⁾がある。これらの方式でも位置の範囲を指定したセンシングデータの取得を可能としている。しかし、こうした方式では時間の値をkeyとして扱うことは考慮されていない。時間は位置と同じ連続量であるが、同様に扱うことは難しい。センシングデータの生成時刻は毎回変化し、またどのセンサも同じ時刻には同じ生成時刻のセンシングデータを生成する。そのため、生成時刻を連続的な識別子として扱うとノードのデータ保持量やネットワーク負荷に偏りを発生させることになる。

以下に時間の値をkeyに用いることを考察する。ここで、オーバーレイネットワークに参加したノードは任意のkeyとして N_{key} を持つとする。あるデータの生成時刻が T である場合、 T に対応するkeyとして T_{key} を定義する。DHTやSkipnetといったオーバーレイネットワークでは T のデータは T_{key} の値に一番近い N_{key} のノードが保持する。 T から T_{key} へ変換するときに時間の連続性を損なうことがなければ、同じ時間帯のデータは N_{key} のノードが保持するため、時間をkeyにした範囲検索を行ったときにクエリをフラッディングすることがなくなる。しかし、 T から T_{key} への変換方法によっては公平性を欠く場合や、フラッディングを行ってしまう場合がある。

例として、 $T = 2007.01.01\ 00:00:00$ を T_{key} に割り当てる方法を考える。 T_{key} を14桁の数値と考え、 T を $T_{key} = 2007010100000000$ に割り当てる。しかし、この場合 $T_{key} = 2008000000000000$ といったkeyは2008年まで使われない。そのため、このkeyに近い N_{key} を保持しているノードは2008年までデータを保持することがなく、ノード間のデータ保持量が公平ではない。これに対して、 T の分と秒にだけ着目して、 T_{key} を4桁の数値にする方法が考えられる。前述の T は $T_{key} = 0000$ に割り当てられる。また、60進数を10進数に変換し、 T_{key} のとりうる範囲を0000~9999とする。ノードには0000~9999の N_{key} を割り当てることで、ある程度公平なデータ保持が可能である。しかし、この場合は1時間より長い範囲検索を行ったとき、該当する T_{key} は0000~

9999 となるためすべての N_{key} のノードに対してクエリを送信しフラッディングを発生させてしまう。

以上のように、既存のオーバーレイネットワークではセンシングデータの特徴を考慮することが難しく、特に時間に対して key を割り当てるのが難しい。実現するセンサーバレイネットワークではデータ保持量の公平性を保ち、クエリのフラッディングを抑制しながら位置と時間に対して key を割り当てる必要がある。

3. データの生成時刻と位置を考慮したセンサーバレイネットワークの提案

以下に本論文で提案するセンサーバレイネットワークの方針を示す。

- (1) 経過時間を用いたデータの集約
- (2) 地理的な位置に基づいたデータの分散化

時間に対して key を割り当て (1) を実現することで、近い時間に発生したセンシングデータは集合体へと集約されていく。そのため、時間の範囲を指定した検索を行ったときに集合体を保持するノードにだけクエリを送信することで検索時のフラッディングを抑制することができる。また、集合体を保持するノードを適切に選ぶことでノード間のデータ保持量を公平にすることができる。

(1) と同時に (2) を実現することによって、地理的な位置に基づいたセンシングデータの可用性を向上する。また、センサーバレイネットワークではセンサに併設される計算機が用いられるため、センサの多い地域には多くのノードを見込むことができる。センシングデータが多い地域では多くのノード群によって共有され、各ノードのデータ保持量を公平にすることができる。

提案手法では (2) と (1) を同時に実現するために、ノードに 3 次元の ID (x, y, τ) を割り当てる。この (x, y, τ) がオーバーレイネットワークの ID となり、 X 軸、 Y 軸、 T 軸によって 3 次元の ID 空間を構築する。 x, y はノードの地理的な位置を表した ID であり、 τ は保持するデータの時刻を定めるための ID となる。この 3 次元の ID 空間はノードやデータの増加があっても十分に足りる必要があるため、非常に大きな数値を扱えるようにする必要がある。また、空間は 2 分割されていくため 2 の累乗の必要がある。そのため、想定している各軸は $0 \sim 2^{32}$ の大きさとする。また、この ID をもとにしてノードは領域を管理する。領域は $(x_s, y_s, \tau_s), (x_e, y_e, \tau_e)$ で表され、この 2 点を始点と終点とし、各軸に平行な辺を持った直方体を領域とする。

センシングデータにも 3 次元の ID が割り当てられる。センシングデータの場合は (x_D, y_D, t) となり、

x_D, y_D はデータの生成位置、 t はデータの生成時刻である。ノードはデータの ID が

$$x_s \leq x_D < x_e, y_s \leq y_D < y_e, \tau_s \leq \tau_t < \tau_e$$

の条件を満たす (x_D, y_D, τ_t) のデータを保持する。 τ_t は t が変換された値である。つまり、時間に対して ID を割り当てることは t を τ_t に変換することに相当する。ノードに 3 次元の ID と領域を割り当てる方法を 3.1 節で説明し、時刻 t を τ_t に変換する方法を 3.2 節で説明する。

3.1 3 次元 ID 空間の構築方法

ノードに対して 3 次元の ID (x, y, τ) と領域 $(x_s, y_s, \tau_s), (x_e, y_e, \tau_e)$ を割り当てる手法を以下に説明する。提案手法では 3 次元の ID 空間を構築し、その空間をノードが分割保持することで ID と領域が決定される。 n 次元の ID 空間を用いたオーバーレイネットワークとしては CAN⁵⁾ がよく知られている。CAN では n 次元トーラスの ID 空間を用いてオーバーレイネットワークを構築する。CAN に参加するノードは、すでに参加しているノードから ID 空間の半分が割り当てられる。また、隣接のノードどうしによるリンクを形成することで、ノード数 N に対して $O(\sqrt[3]{N})$ の平均探索時間で検索が可能である。この CAN の手法を用いて 3 次元 ID 空間を構築する。位置の情報である (x, y) はノードの実際の地理位置をもとにしてあらかじめ定める。たとえば緯度と経度の値を 0 から 2^{32} の大きさの (x, y) に変換する。ノードはセンサが併設されているか、センサの多い地域に設置されることを想定しているため、ノードの密度とセンシングデータの密度が一致し、ノード間での公平性を保つことができる。しかし、 (x, y) の位置によっては領域の半分を分け与えることができない場合がある。どのような座標にノードを配置しても領域を分割するために、担当領域の条件を定め、その条件の下で領域分割のアルゴリズムを用いた ID 空間の分割を行う。以下に領域分割のアルゴリズムと担当領域の条件を示す。

領域分割のアルゴリズム

- 1 新規ノードは (x, y) と、ランダムな (τ) を定める。
- 2 新規ノードは (x, y, τ) の座標を管理しているノードに対して参加の要求を行う。
- 3 参加の要求を受けたノードは X 軸または Y 軸による分割を試みる。
- 4 分割後の領域が条件 a, b (下記に示す) を満たさない場合は T 軸による分割を行う。
- 5 条件 b を満たし、条件 a を満たさない場合は T 軸における座標の移動を行う。
- 6 T 軸による分割が条件 b を満たさない場合はダミー

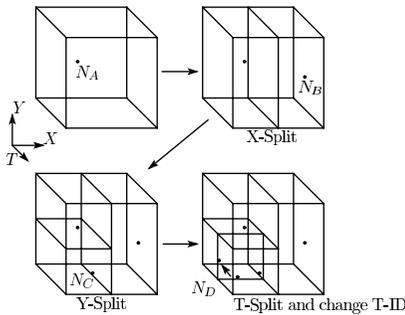


図1 領域分割
Fig.1 Process of dividing ID space.

ノードの作成を行い、手順2からやり直す。
担当領域の条件

条件 a 1つの領域には1つのノードしか存在しない。

条件 b 領域の1辺を他の1辺と比較すると等しいか
または2:1の関係にある。

条件 a と条件 b により、新規ノードには領域が半分
ずつ割り当てられていく。そのため、ノードの粗密に
応じて公平に領域が割り当てられる。

領域の分割例を図1に示す。図1ではまずノード
NA が ID 空間に存在している。そこに NB が参加の
要求を行い、X 軸による分割が行われる。次に NC
が NA に対して参加の要求を行う。X 軸による分割
を行うと条件 b を満たさないため Y 軸による分割
を行う。最後に ND が NC に参加の要求を行う。X 軸
による分割も Y 軸による分割も条件 b を満たさない
ため、T 軸による分割が行われる。ND の座標がその
ままでは条件 a を満たさないため、ND の (τ) を変更
する。このように各辺を2等分割にすることで領域
の大きさが公平に決定される。

3.1.1 ダミーノード

上記の領域分割のアルゴリズムにおける6のよう
に、T 軸上での移動を行っても条件 a, b を満たさな
い場合がある。このような場合は、参加の要求を受け
たノードがダミーノードを作成する。ダミーノードは
X 軸または Y 軸に分割可能な座標に割り当てられ、
通常のノードと同じように領域の半分が割り当てら
れる。ダミーノードに領域を分配した状態で再度新規
ノードに領域の割当てを試み、この場合でも領域の分
割を行えない場合はさらにダミーノードを作成し、領
域の割当てを繰り返す。ダミーノードが2つ作成され
ると条件 b を必ず満たすため、ダミーノードが大量に
発生することはない。

ダミーノードは通常のノードと同様に割り当てられ
た ID 空間を管理するが、以下の部分が通常のノード
とは異なっている。

- ダミーノードは作成したノードの計算機資源を利用する。
- ダミーノードに参加要求が行われた場合、ダミーノードの全領域を新規ノードに割り当て、ダミーノードは消滅する。

以上のような T 軸方向への ID の変更やダミーノードの作成によって、どのような座標にノードを配置しても領域を分割することが可能である。

3.2 経過時間によるセンシングデータの集約

次にデータの生成時刻である t を τt に変換する方法について説明する。ノードはセンシングデータが生成されてからの経過時間をもとに τt を計算する。そのため、時間が経過するに従って τt は変化し、データを保持するノードは変化していく。提案する手法では時間の経過したセンシングデータほど、長い時間での集約が行われる。たとえば最近1日の間に生成されたセンシングデータは多数のノードが分散して保持しているが、1カ月前の1日分のセンシングデータは少数のノードが保持している。このような経過時間を用いた集約を行うことで、公平なデータ保持とフラッディングを抑制した時間検索を実現する。以下に τt へ変換する手法の詳細と、この手法によって発生するトラフィックについて説明する。

3.2.1 時刻 t の τt への変換

ノードは集約を行うために以下に示す共有指数 n を用いる。

$$n = \frac{\tau_{max}}{|\tau_s - \tau_e|} \tag{1}$$

ここで τmax は T 軸の最大値であり、2⁶⁴ のように非常に大きな値である。共有指数 n はノードの持つ (x, y) の領域を n ノードによって共有していると推定した値である。この共有指数 n と集約を行うときの時刻 tnow を用い、次のようにして時刻 t を τt へ変換する。τ を集約を行うノードの T 軸上の ID とする。

- (1) t を2進数で表し、下位 log₂(tnow - t) - 1 ビットを tscope' とする。
- (2) tscope' の上位 log₂ n ビットを tscope とする。
- (3) τ の先頭 log₂ n ビットを tscope と入れ替えたものを関数の戻り値とする。

図2に生成時刻を用いたデータの集約手法の例を示す。図2ではノード B が t = 1101 1101 0101 のデータを、経過時間が 110 1011 のときにどのノードへ送信するかを計算する。なお、本来ならば経過時間の桁が上がったときに集約が行われるため、100 0000 のときが集約の対象となる。この例では、集約のタイミングをノードが独自に判断することを想定し、経過時

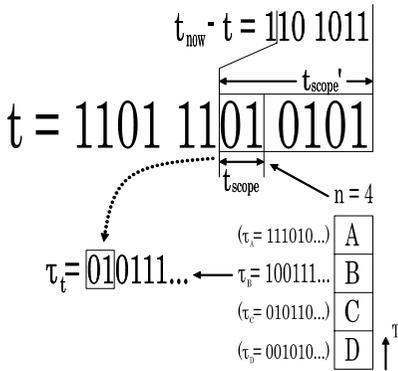


図 2 生成時刻を用いたデータの集約

Fig. 2 A method of data aggregation by generation time.

間を 110 1011 に設定している．集約のアルゴリズムにおける (1) では経過時間のビット数から 1 を引いたビット数を計算し，時刻の下位ビットを取り出す．そのため， $t_{scope}' = 01 0101$ となる．次に (2) では共有指数が $n = 4$ となるため， t_{scope}' の先頭 2 ビットとノードの T 軸上の ID である τ_B の先頭 2 ビットを入れ替え， $\tau_t = 010111\dots$ となる． τ_t はノード C の領域であるため，ノード C へとデータが送信される．この操作をデータごとに繰り返すことにより，ノード C には $t = 1001 1101 0000$ から $t = 1001 1101 1111$ までのデータが集まる．この範囲のデータ検索が行われるとノード C へのみクエリが送信されるため，フラッシングを抑制することができる．また， T 軸上に均等にデータが配置されるため，データ保持量を公平に保つことができる．このような変換を行いデータを集合体へと集約することで，公平性を保ちつつデータの生成時刻に key の割当てが行える．

3.2.2 集約によって発生するトラフィック

センシングデータを集約するためにデータはノード間を移動し，その結果定常的なトラフィックが発生する．時間が経過するにともない発生したセンシングデータが増加するため，トラフィックが増加していくことが考えられる．しかし，このトラフィックの増加は対数的な値にとどまる．これを以下に示す．

ここで時間 t_i が経過したときに，集約の対象となるセンシングデータの数を求める．集約は 2 進数で表した経過時間の桁が上がったときに行われるため， m を任意の正の整数とすると $t_i = 2^m = 1, 2, 4, 8, 16 \dots$ となる t_i のデータが対象となる． t_i は正の値をとる必要があるため， $0 < m < \lfloor \log_2 t_i \rfloor$ の関係となる．例として時刻 0 からセンシングデータが生成され始め，時刻 100 のときの集約を考える．集約の対象となるセンシングデータは経過時間 t_i が 1, 2, 4, 8, 16, 32, 64

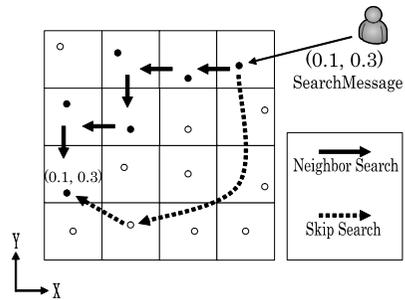


図 3 任意座標の検索

Fig. 3 Process of search for arbitrary coordinate.

となるデータ，すなわち生成時刻が 99, 98, 96, 92, 84, 68, 36 のセンシングデータであり，その個数は 7 個である．この関係から集約の対象となるデータの数は対数的な増加になるため，トラフィックも対数的な増加にとどめることができる．

3.3 オーレイネットワークの構築

上記に示したオーバーレイネットワークをノードは自律的に構築する．提案手法は ID 空間の構築手法として CAN と近似の手法を用いているため，以下に簡単にオーバーレイネットワークの構築手法を説明する．

3.3.1 ノードの参加

3.1 節で示した領域分割のアルゴリズムに従い，オーバーレイネットワークに参加するノードはすでに参加しているノードから領域を割り当てられる．また，領域が隣り合っているノードの情報を受け取り保持しておく．

3.3.2 ルーティングテーブルの作成

提案手法では CAN で用いられているような検索を実現する隣接ノードリストを用いることで，参加しているすべてのノードに対してクエリを到達可能としている．また，高速な検索を実現するスキップノードリストを導入することで，高速にクエリの解決を可能としている．この 2 つのルーティングテーブルを用いることで，隣接ノードとのリンクのみの維持を行いつつ，高速な検索を可能としている．

隣接ノードリスト

提案するオーバーレイネットワークでは CAN と同様に隣接したノードに対するリンクを形成することで，任意の座標の検索を可能にする．各ノードは隣接したノードの情報を格納したリストを隣接ノードリストとして保持する．検索が行われる様子を図 3 の実線に示す．なお，図 3 では簡単のため 2 次元の ID 空間としている．3 次元空間で構築する場合は隣接する方向を 6 方向にすることで，同様のリストを構築可能である．図 3 では座標 (0.1, 0.3) を担当しているノードを検索する．検索を行うノードは，最初に (0.1, 0.3) が

自身の領域に含まれているかを検査し、含まれていない場合は隣接ノードリストに記載されているノードのうち、(0.1, 0.3) の点に最も近いノードへ検索クエリを送信する。クエリを受け取ったノードは同様の検索を繰り返すため、最後には (0.1, 0.3) を担当するノードにクエリが到達する。また、隣接ノードリストに記載されている情報が正確であるかを定期的に確認し、検索に使用するリンクの維持を行う。

スキップノードリスト

隣接ノードリストを用いた検索はノードの数が増えると、検索に必要な時間が長くなる。そこで提案手法では Chord⁶⁾ などに用いられているスキップノードリストを利用する。スキップノードリストには ID 空間上で 2 の累乗離れた領域に対し、その領域に属する任意のノードを記載する。この 2 の累乗離れた領域を仮領域とし、スキップノードリストには各仮領域とその仮領域に属する任意のノードの IP アドレスの組を記録する。仮領域は近くの領域ほど狭く、遠くの領域ほど広くなるため、近いノードの経路は密に、遠いノードの経路は疎に保持する。図 3 の破線はスキップノードリストによる検索を示している。隣接ノードリストを用いた手法では 5 個のクエリで検索しているのに対し、スキップノードリストでは 2 個のクエリで目標とする座標まで到達している。

しかし、このスキップノードリストではリンクの維持を行わない。これは、隣接ノードリストによって検索クエリは確実に到達できることが保証されているため、スキップノードリストによる確実な到達性が必要ないためである。提案手法では、スキップノードリストの構築やリンクの維持のためにトラフィックを流さず、検索を行ったときにその結果をスキップノードリストに保存する。そのため、初期のスキップノードリストは空の状態、検索が行われるたびに埋められ、高速な検索が実現されていく。提案手法ではデータの再配置のために検索がある程度行われるため、時間の経過にともない高速な検索が可能になる。

3.3.3 ノード離脱時のネットワーク再構築

最後にオーバーレイネットワークからノードが離脱するときについて説明する。オーバーレイネットワークに参加するノードはそれぞれが独立に管理された計算機であるため、非同期に参加と離脱を行う可能性がある。オーバーレイネットワークの再構築に関して、ダミーノードを用いた再構築手法と、データの複製保持による欠損回避について以下に説明する。

ダミーノードを用いた再構築手法

提案手法では ID の構築に CAN と近似の方式を用

いている。また、既存の様々なオーバーレイネットワークと比較したとき、ノードに ID を割り当てているという点が共通しているため、参加と離脱に対し既存のオーバーレイネットワークの手法を用いることができる。また、提案手法ではダミーノードを生成することができるため、離脱したノードの領域にダミーノードを生成することで、離脱したノードの代替となることができる。このダミーノードはノードから参加の要求を受けると、保持している領域すべてを参加ノードに譲渡する。そのため、離脱したノードが再び参加してきたときはダミーノードの領域を譲渡するだけでよい。

データの複製保持による欠損回避

ノードの離脱によってオーバーレイネットワークから欠損するデータは、そのノードが保持していたデータである。提案手法のオーバーレイネットワークでは、ノードは近い位置で発生したデータを集合体として保持しているため、ノードの離脱によって近い位置のデータを欠損することになる。センサに併設された計算機の利用を想定した場合、離脱したノードによって他のノードから発生したデータが欠損することは好ましくない。この問題に対して、オーバーレイネットワーク上で共有するデータを元データの複製（複製データ）とすることで対処する。元データを最初に保持しているノードは、複製データを生成し、集約先のノードに送信する。複製データを受け取ったノードは、複製データを次に集約する時刻になるまで保持し、集約する時刻になった後は破棄可能とする。元データを保持しているノードは集約する時刻になるたびに複製データを集約先のノードに送信する。このように複製データを共有することによって、ノードが離脱しても、離脱したノードの保持していた元データが欠損するだけで、他のノードの元データは欠損されずに済む。

4. 評価実験

提案手法を用いてセンサオーバーレイネットワークを構築した場合、検索時に削減可能なクエリ数を調べることで、フラディングをどの程度抑制できるかを評価する。また、センシングデータの集約によって発生するトラフィックやデータ保持量の偏りを調べることで実環境における利用が妥当であるかを評価する。この評価を行うために提案手法をエミュレータとして実装し、TCP/IP を用いた複数計算機間の実験を行った。以下に詳細を説明する。

4.1 エミュレータを用いた評価実験

提案手法を評価するためにエミュレータを作成し、実験を行った。作成したエミュレータは以下のような

表 1 実験環境

Table 1 Experimentation environment.

ノード数	5,001
データ発生間隔	150 ~ 90 [sec]
データ発生時間	18,000 [sec]

特徴がある .

- ノードの動作をエミュレートする .
- 通信には TCP/IP を用いる .
- 複数の TCP ポートを用い , 1 台の計算機で複数ノードをエミュレートする .

以下に実験の詳細を述べる .

4.1.1 実験概要

実験は 5 台の計算機を用いて行い , 表 1 の環境において実験を行った . 5 台の計算機はそれぞれ 1,000 ノードを生成し , オーバレイネットワークを構築する . これに評価用のノードを 1 ノード加え合計で 5,001 ノードをエミュレートし , 評価実験を行った . 全ノードがオーバレイネットワークに参加できた段階から , 5 時間にわたってセンシングデータを発生させ , オーバレイネットワーク上で共有を行った . センシングデータの集約はデータが発生したときに行われるため , 集約のタイミングはノードのデータ発生間隔と同じである

この実験では以下の内容を調べた .

- センシングデータの保持状況
- 集約に必要なトラフィック
- センシングデータの保持量
- データ取得時にクエリの送信されるノード数

これらの内容を調べることで , 提案手法のオーバレイネットワークが公平性を実現しながらフラッシングを行わずに検索が行えていることを確認する .

4.1.2 センシングデータの保持状況

5 時間の実験が終わった段階でそれぞれのノードが保持しているセンシングデータの生成時刻を調べた . 結果を図 4 に示す . 図 4 は横軸がセンシングデータの生成時刻 , 縦軸がノードの T 軸における位置を表している . センシングデータは時刻 0 から発生し , 不規則な間隔でデータを 5 時間発生させている . 図 4 に示すように生成時刻が 18,000 付近のデータは , T 軸に均等に分散しているのに対し , 時間の経過したデータは特定のノードに集約されていることが分かる . 図 4 では黒い矩形になっている部分とバーコード状の模様になっている部分が形成されている . これは , ノードの粗密によって T 軸の分割数が異なるためである . 図 4 では T 軸の大きさが 0.125 と 0.0625 の 2 種類のノードが混在しているため , それぞれの大きさに保持しているデータが描画されている .

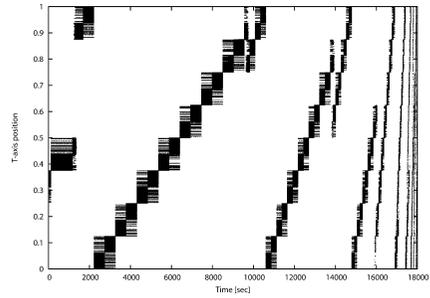


図 4 ノードの保持しているセンシングデータ
Fig. 4 Map of sensing data on T -axis.

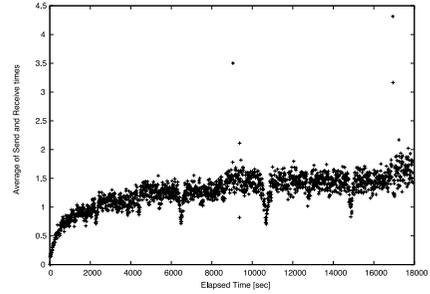


図 5 平均送受信データ個数の推移
Fig. 5 The average number of transmitted and received data.

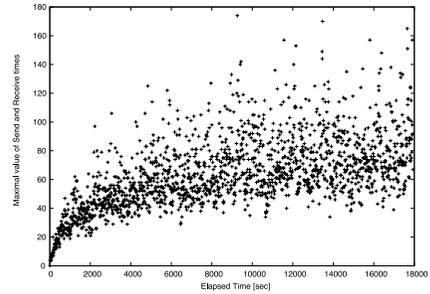


図 6 最大送受信データ個数の推移
Fig. 6 The maximal number of transmitted and received data.

4.1.3 集約に必要なトラフィックの検証

提案手法では , 3.2.2 項で述べたように集約に必要なトラフィックは対数的な増加にとどまると計算上求められる . 実験では各ノードごとにセンシングデータを送受信した回数を記録し , 提案手法のトラフィックに対する検証を行った . ノードは 10 秒間に送受信したデータの個数を記録し , 全ノードがその 10 秒間に送受信したデータ個数の平均値と最大値を求める .

図 5 に送受信したデータ個数の平均値を , 図 6 に最大値を示す . 図では横軸を時間 , 縦軸を送受信回数の平均値 , 最大値としている . この図から集約時に送受

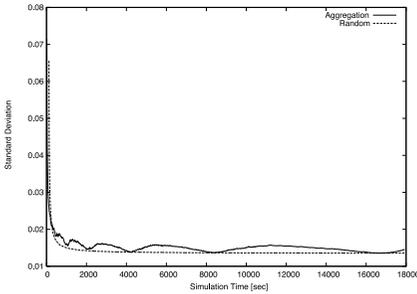


図 7 データ保持量の標準偏差の変化

Fig. 7 Standard deviation of amount of data stored.

信されたデータの個数は対数的な増加にとどまっていることが分かる．実験結果から値の分布を調べたところ，平均値は 1.0~2.0 個に 83.40%の値が収まり，最大値は 60~110 個に 80.68%の値が収まっている．これは 10 秒間の値であるため，毎秒に換算すると平均値は 0.1~0.2 個，最大値は 6~11 個と推定できる．

4.1.4 センシングデータ保持量の偏り

任意のノード N_i のセンシングデータの保持量を d_{N_i} ，全体のセンシングデータの数を d_{Au} としたときノード N_i の保持しているデータが全体のセンシングデータの何割であるかを示す値， $r_{N_i} = d_{N_i}/d_{Au}$ をもとにし，各時間における r_{N_i} の標準偏差を調べる．なお，全ノードにおけるデータ保持量の平均値を \bar{d} とする． r_{N_i} の標準偏差の変化を知ることによって，センシングデータ保持量の偏りが時間経過とともにどのように変化するかを知ることができる．以下の式 (2) を用いて標準偏差 σ を記録した．なお， m はノードの数を表す．

$$\sigma = \sqrt{\frac{1}{m} \sum_{i=1}^m \left(\frac{d_{N_i} - \bar{d}}{d_{Au}} \right)^2} \quad (2)$$

提案手法を用いた場合と，データを保持するノードをランダムに決定した場合とでこの σ を比較した．これは，データをほぼランダムに保持するハッシュを用いた手法と比較するためである．提案手法を用いた場合の偏りが，ハッシュによってデータを保持した場合の偏りとどの程度の差があるのかを調べた．

図 7 に標準偏差 σ の時間経過による変化を示す．提案手法を実線に，ランダムに保持した場合を破線に示す．図 7 から，提案手法はランダムに保持した場合の標準偏差とほぼ同じ値を得られていることが分かる．

4.1.5 データ取得時に削減したクエリ数

提案手法ではセンシングデータを集約しているため，データの取得時に従来よりも少ないクエリでデータの取得が可能である．そこで，様々な位置や時刻を指定

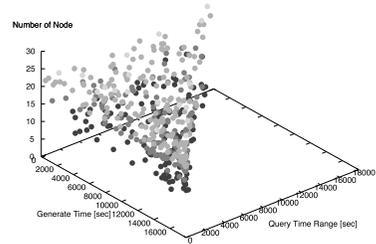


図 8 集約していない場合のクエリ送信先ノード数

Fig. 8 The number of destination nodes of a query (No aggregation).

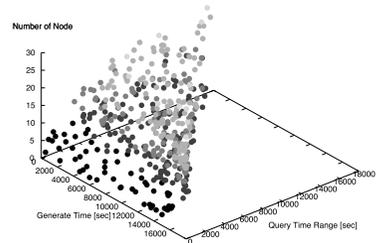


図 9 集約した場合のクエリ送信先ノード数

Fig. 9 The number of destination nodes of a query (Aggregation).

した検索を行い，クエリをどの程度削減できるかを調べた．5 時間の実験が終了した状態で検索を行い，その際に何ノードにクエリを送信したかを調べた．この検索は 0~18,000 の任意の時刻を始点とし，その値より大きな任意の時刻を終点としたランダムな時間範囲を指定したクエリを用いた．500 回の検索に対して集約していない場合にクエリが送信されるノード数と，集約した場合にクエリが送信されるノード数の比較を行う．実験ではランダムな位置をもとにした一定範囲に対するデータ取得を行う．集約していない場合は，該当する位置のノードすべてにクエリの送信を行う．まず， T 軸の値が最も小さいノードに対してクエリを送信し， T 軸の値の順に逐次的にクエリを送信する．一方で，集約した場合は該当する位置の一部のノードにのみクエリを送信する．クエリの時刻から，データを保持しているノードの T 軸の値を逆算し，そのノードに対して逐次的にクエリを送信する．

図 8 に集約していない状態においてクエリが送信されるノード数，図 9 に集約した状態においてクエリが送信されるノード数を示す．図において Generate Time と記された軸が検索の始点となる時刻，Time of Range Query と記された軸が検索の時間幅を表している．また，表 2 にクエリが送信されるノード数の平均値を示す．集約していない場合は平均して 18.04

表 2 クエリ送信ノード数の平均値

Table 2 The average number of destination nodes of a query.

集約の有無	平均クエリ送信ノード数
集約なし	18.04
集約あり	15.68
集約なし(時間幅が 1,000 秒以下)	18.89
集約あり(時間幅が 1,000 秒以下)	5.00

ノードに対してクエリが送信されるのに対し、集約した場合は平均して 15.68 ノードへのクエリ送信に抑えられている。これはクエリの送信されたノードを約 13%削減したことになる。また、集約した場合は検索の時間幅が短いほどクエリ数が少ない。これは時間幅が短いほど対象となるデータが少なくなったためである。データが少なくなることで集合体を保持しているノード数が少なくなり、送信されるクエリ数も削減されている。一方で集約していない場合は、時間幅が短い場合も対象となる地域のすべてのノードに対してクエリを送信しなければならない。特に時間幅を 1,000 秒以下に限定した場合は平均して 5.00 ノードに対するクエリ送信に抑えられており、集約しない場合に比べるとクエリが送信されるノードを約 76%削減している。このことから時間を用いた集約により、フラッシングを抑制しながらセンシングデータを取得できていることが分かる。なお、図の右半分の領域は検索の始点となる時刻と時間幅の組合せが不可能であるため、実験の結果が存在していない。

5. 考 察

エミュレータを用いた実験から提案手法についての考察を行う。以下に集約を行ったことによって発生した得失と、提案手法が有効となるような環境、今後の課題について考察する。

5.1 集約による損失

集約する過程において発生したトラフィックは図 5 と図 6 から読み取ることができる。3.2.2 項で示したように集約によって送受信されたデータの個数は対数的な増加にとどまり、平均値は毎秒 0.10~0.20 個、最大値は毎秒 6~11 個である。センシングデータを XML で表し 1 つのデータを 500 バイトと仮定すると、平均値で毎秒 50 バイト~100 バイト、最大値でも毎秒 3,000 バイト~5,500 バイトの増加である。この値は現在のネットワークインフラを考慮した場合、ほとんど負荷を与えないといえる。

次にデータ保持量の公平性について考察する。ノードはセンシングデータの集合体を保持するため、他の

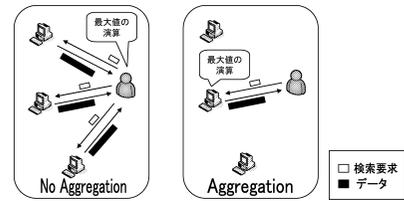


図 10 集約による演算の効果

Fig. 10 An effect of method for network traffic.

ノードよりも大きな集合体を保持するノードが存在する場合、データ保持量が公平ではなくなる。しかし、提案手法では図 4 のようにセンシングデータは T 軸に対してほとんど均等に割り振られている。また、図 7 で示したようにデータ保持量の偏りはハッシュを用いた手法と同等の値に抑えることができています。そのため、ノードのデータ保持量の偏りは従来のオーバレイネットワークと比べても問題がないと考えられる。

5.2 センシングデータの集約による利点

センシングデータを集約したことにより、4.1.5 項に示したように検索クエリを平均して約 13%削減することができた。これは 2.36 個のクエリを削減したことになる。そのため、検索クエリを XML で表し 1 つのクエリを 300 バイトと仮定すると、実験では検索によるトラフィックを 708 バイト削減したことになる。集約によるトラフィックが平均して毎秒 100 バイトであると仮定した場合、7.08 秒に 1 回以上検索が行われる環境では、トラフィックの削減を期待できる。

また、利用者はセンシングデータを利用するとき、集約された値を利用することが多い。たとえば 1 日の平均気温や 1 週間の積算降水量のような値である。こうした値は多くのセンシングデータを用いて演算され、利用者に提供される。センシングデータの集約が行われていない場合、それぞれのノードからセンシングデータを入手したうえで演算を行う必要がある。しかし、提案手法では生成時刻による集約を行っているため、ノードがあらかじめ演算して利用者に情報を提供することができる。図 10 を例にして説明する。図 10 では左に集約しない場合の最大値の取得を、右に集約した場合の最大値の取得を示している。どちらの場合も利用者は 3 つのノードが担当する地域に対して気温の最大値について取得を行う。集約しない場合、ノードは 3 つのノードに対してクエリを送信する。要求を受けたノードは最大値を演算しデータを送信する。3 つのノードからデータを受け取った後に収集したデータの中から最大値を演算し、この地域における気温の最大値を得ることができる。これに対して集約した場合、3 つのノードから生成されたセンシングデータは

1つのノードに集約されている．そのため1つのノードにクエリを送信することで，この地域における気温の最大値を得ることができる．このように集約した場合は集約していない場合に比べて冗長となるデータを送受信する必要がない．提案手法では集約されたセンシングデータが複数のノードに分散している可能性がある．その場合は，それぞれのノードから演算された値を取得し，取得した値から演算しなおす必要がある．最悪の場合はすべてのノードに対してクエリを送信する必要があるが，多くの場合は検索によるトラフィックを削減することができる．

次にデータ配信の方式における利点について説明する．ID空間において同じXY平面を共有するノードはセンシングデータを時間的に分散保持する．そのため，ノードは近くのセンサの情報を自動的に保持することになる．これは登録しておいた場所に応じてセンシングデータが自動的に送信されてくることになり，publish/subscribe形式のデータ配信に近い．そのためノードは近くのセンシングデータを保持し，データ取得のクエリを送信しなくても利用者へデータを提供することができる．このようなセンシングデータの提供方法が行えることも集約による利点である．

5.3 有効となる環境

提案手法は既存の手法にはない，センシングデータの集約を行うという特徴がある．センシングデータはノード間で再配置されながら，集合体へと集約される．この集合体を保持しているノードに対して検索クエリを送信することで，時間と位置を指定した検索に対しクエリのフラディングを抑制しながらセンシングデータの取得を可能としている．検索クエリによるトラフィックが減るため，5.2節に示したように平均して7.08秒に1度以上の検索が行われれば，オーバーレイネットワークを流れる全体のトラフィックを減らすことにもなる．また，既存の手法では検索が集中して行われるような環境においてクエリの処理やトラフィックの増加により，アプリケーションの可用性が低下することが想定される．提案手法では位置の範囲に対するクエリのフラディングを抑制することにより，この可用性低下を防ぐことができる．一方で提案手法ではセンシングデータをノードが協調して集約することなどから，ノードが頻繁に離脱するような環境ではセンシングデータの集約による効果をあまり期待できない．こうした理由から，提案手法はセンサオーバーレイネットワークのバックボーンの役割となることが望ましい．安定動作するノードによってオーバーレイネットワークを構築することで，データを生成時刻に応じて

集約する．ユーザはデータを取得するときにノードに対してクエリを送信し，クエリを受け取ったノードは，オーバーレイネットワークに対してデータの検索を行う．このようにノードは複数のユーザからのクエリを処理することで，1つのノードは多くの検索を行うことになる．

以上のような提案手法が既存の手法に比べて有効となる環境の例として，Live E!プロジェクト¹⁾における小中学校へのセンサ設置の活動がある．この活動では，小中学校に設置されている百葉箱をインターネットへの接続性を持ったセンサに置き換え，センシングデータをインターネット上で広く共有するとともに，各小中学校での教育利用に役立てることが目的である．各小中学校ではセンサから取得したデータをその学校に設置されている計算機に蓄積し，インターネット上にデータを公開する．現在はサーバにデータをアップロードしているが，全小中学校の数は約3万校にもおぼるため，サーバに集中するトラフィックやデータ量の増加に対応するため早期に改良する必要がある．そこで，各学校に設置された計算機によってオーバーレイネットワークを構築し，センサオーバーレイネットワークを構築する．小中学校に設置された計算機は安定した動作を期待できるため，提案手法を用いることでセンシングデータを集合体へと集約することが期待できる．各学校では1台の計算機がオーバーレイネットワークのノードとなり，ユーザの端末からのクエリを処理する．学校の児童たちがユーザとなるため，特に授業などの時間帯においてオーバーレイネットワークに流れる検索クエリ数は非常に多くなることが想定される．提案手法ではこの検索クエリを削減することでアプリケーションレベルでの可用性を維持することが期待できる．

5.4 今後の展開

提案手法を実環境で利用するためには，現実のネットワークとの乖離を考慮する必要がある．本論文でのオーバーレイネットワークではノードの地理的な位置に基づき，近い位置のノードどうしてネットワークを構築している．しかし，現実のネットワークポロジは必ずしも地理的な位置に基づいて構築されていない．理想的なネットワークポロジではネットワークトラフィックの負荷が分散されていたとしても，現実を利用される経路では分散されていない可能性がある．図11はそうした場合の1つの例である．AからEのノードへ順に経路が存在しているとして，現実のネットワークポロジを考えていない場合は，図の右側のように非常に効率の悪い経路になる可能性がある．

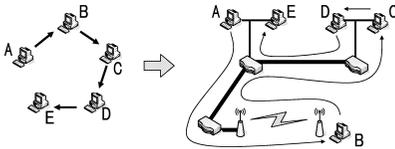


図 11 オーレイネットワークと現実のネットワークの乖離
Fig. 11 Difference between overlay networks and real networks.

こうした問題への対処としては、ネットワークの状況に応じてオーレイネットワークの経路を動的に変化させることが考えられる。1つの例としては通信にかかった時間を計測し、経路制御表の内容を更新する方法が考えられる。たとえば通信にかかった時間が長いノードほど現実のネットワークでは遠いノードであると仮定し、その経路はなるべく使わないようにする。このような動的に経路を変化させる方法は Unstructured 型のオーレイネットワークに多く見られる。Structured 型のオーレイネットワークに Unstructured 型のオーレイネットワークを組み合わせることで、こうした問題に対処できると考えられる。

6. 結 論

本論文では時間と位置を考慮したセンサーオーバーレイネットワークを提案した。センシングデータの情報基盤では、位置と時間による検索が不可欠である。そのため、位置の範囲を指定した検索、時間の範囲と周期を指定した検索を実現する必要があった。ノードの位置に応じてオーレイネットワークを形成し、局所的なセンシングデータの配置を実現することで、位置の範囲を指定した検索を実現した。また、センシングデータの生成位置と時刻を用いて集合体へと集約した。この提案手法によって位置と時間を指定したセンシングデータの検索を行ったとき、クエリの送信されるノードを従来よりも平均 18%削減することができた。また、クエリの時間幅を狭めた場合はクエリの送信されるノードを平均して約 74%削減することができた。この集約によって発生した定常的なトラフィックは平均して毎秒 75~100 バイトの増加に抑えられると考えられ、実際の利用ではほとんど問題がない。また、センシングデータの集約を行ったため、平均値や最大値といったセンシングデータが多く集まることで得られる情報が容易に取得可能になった。今後はオーレイネットワークを構築するうえで問題となる、現実のネットワークの乖離を埋めていく必要がある。

参 考 文 献

- 1) Live E! Project: <http://live-e.org>
- 2) Bharambe, A.R., Agrawal, M. and Seshan, S.: Mercury: Supporting scalable multi-attribute range queries, *SIGCOMM '04: Proc. 2004 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, New York, NY, USA, pp.353-366, ACM Press (2004).
- 3) Harvey, N.J.A., Jones, M.B., Saroiu, S., Theimer, M. and Wolman, A.: Skipnet: A scalable overlay network with practical locality properties, *4th USENIX Symposium on Internet Technologies and Systems*, pp.113-126 (2003).
- 4) Matsuura, S., Fujikawa, K. and Sunahara, H.: Mill: An information management and retrieval method considering geographical location on ubiquitous environment, *Applications and the Internet Workshops, 2006. SAINT Workshops 2006 International Symposium on Applications and the Internet*, p.4 (2006).
- 5) Ratnasamy, S., Francis, P., Handley, M., Karp, R. and Schenker, S.: A scalable content-addressable network, *SIGCOMM '01: Proc. 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, New York, NY, USA, pp.161-172, ACM Press (2001).
- 6) Stoica, I., Morris, R., Liben-Nowell, D., Karger, D.R., Kaashoek, M.F., Dabek, F. and Balakrishnan, H.: Chord: A scalable peer-to-peer lookup protocol for internet applications, *IEEE/ACM Trans. Netw.*, Vol.11, No.1, pp.17-32 (2003).
- 7) 金子 雄, 春本 要, 福村真哉, 下條真司, 西尾章治郎: ユビキタス環境における端末の位置情報に基づく P2P ネットワーク, *情報処理学会論文誌: データベース*, Vol.46, No.SIG18(TOD28), pp.1-15 (2005).

(平成 19 年 5 月 18 日受付)

(平成 19 年 11 月 6 日採録)



洞井 晋一 (学生会員)

平成 17 年神戸大学海事科学科卒業。平成 19 年奈良先端科学技術大学院大学情報科学研究科修士課程修了。同年より同大学院情報科学研究科博士課程。オーレイネットワーク

ク、分散処理システムの研究に従事。



松浦 知史

平成 15 年立命館大学理工学部物理学科卒業。平成 17 年奈良先端科学技術大学院大学情報科学研究科修士課程修了。同年より同大学院情報科学研究科博士課程。オーバーレイネットワーク、センサネットワークの研究に従事。



藤川 和利 (正会員)

昭和 63 年大阪大学基礎工学部情報工学科卒業。平成 3 年同大学院基礎工学研究科博士後期課程退学後、同年大阪大学基礎工学部助手等を経て、平成 14 年奈良先端科学技術大学院大学情報科学センター助教授、平成 17 年同大学情報科学研究科助教授、平成 19 年同准教授、現在に至る。博士(工学)。分散処理システム、マルチメディアシステムの研究開発に従事。電子情報通信学会、IEEE、ACM 各会員。



砂原 秀樹 (正会員)

昭和 58 年慶應義塾大学工学部電気工学科卒業。昭和 63 年同大学院博士課程修了。同年電気通信大学情報学部助手。平成 6 年奈良先端科学技術大学院大学情報科学センター助教授。平成 13 年同大学情報科学センター教授。平成 17 年同大学情報科学研究科教授、現在に至る。工学博士。インターネット、大規模広域分散環境、ネットワーク、並列処理、オペレーティングシステム、電子図書館に関する研究に従事。電子情報通信学会、ACM、IEEE 各会員。