

# センサネットワークにおける 位置情報を用いないルーティング方式の検討

守田可南子<sup>†</sup> 佐藤雄亮<sup>\*\*</sup> 岡崎直宣<sup>\*\*</sup> 朴 美娘<sup>\*\*\*</sup>

<sup>†</sup>宮崎大学大学院工学研究科 <sup>\*\*</sup>宮崎大学工学部

<sup>\*\*\*</sup>三菱電機株式会社情報技術総合研究所

バッテリー駆動のセンサ端末で構成されるセンサネットワークでは、省電力化が重要な課題となる。特に、通信は最もエネルギー消費の高い動作だと言われており、通信量を抑制することは、電力資源が乏しいセンサネットワークにとって有効である。そこで、本論文では通信量を増加させずに、かつ位置情報を必要としない簡易なルーティング方式である HVGF (Hop-Vector based Greedy Forwarding) を提案する。提案方式では、 $n$  個の基準点を設け、基準点からのホップ数を要素とするベクトルを全てのノードに作成し、その  $n$  次元ベクトル間の距離に基づいて greedy forwarding を行う。

## Hop-Vector based Greedy Forwarding for Sensor Networks

Kanako MORITA<sup>†</sup>, Yusuke SATO<sup>\*\*</sup>, Naonobu OKAZAKI<sup>\*\*</sup>, Mi Rang PARK<sup>\*\*\*</sup>

<sup>†</sup>Graduate School of Engineering, University of Miyazaki

<sup>\*\*</sup>Faculty of Engineering, University of Miyazaki

<sup>\*\*\*</sup>Information Technology R&D Center, Mitsubishi Electric Corporation

For sensor networks consisting of battery-powered sensor nodes, power saving is significant challenge. It is effective to reduce the amount of communication for sensor networks where power resource is scarce, because sensor nodes consume energy most in communication. We propose a routing protocol, HVGF (Hop-Vector based Greedy Forwarding), which reduces the amount of communication and doesn't need geographic information. In this method, we introduce  $n$  base points, and assign  $n$ -dimensional hop-vector consist of number of hops from each base point to each nodes. Then, each node forwards packets greedily based on distance between hop-vectors of the node and destination.

### 1. はじめに

近年、無線通信技術の発展や、無線端末の普及により、無線移動端末などを用いて既存のインフラのない場所に一時的にネットワークを構築できるアドホックネットワークが盛んに研究されている。中でも、ネットワークに接続されたコンピュータが身の周りの「いたるところ」に存在し、「いつでも、どこでも、だれでも」利便性の高いサービスやさまざまな情報へアクセスできるユビキタス環境を実現させる重要な技術として、通信機能を持つ超小型センサによるセンサネットワークが注目されている。センサネットワークを構成するセンサ端末は設置の容易性や設置場所の制約を少なくするために、バッテリー駆動であることが多く、小型軽量化が求められる。それに伴い、各センサ端末の電力、メモリ、通信帯域といったリソースにも大きな制約がある。それ故、センサネッ

トワークでは従来の通信に求められていた高スループットや低遅延を犠牲にしても効率的な電力消費（省電力化）が重要になる。一般に無線通信機能を持つ情報通信端末における通信にかかる消費電力は大きく、特にセンサネットワークでは通信は最もエネルギー消費量の高い動作だと言われている<sup>[1]</sup>。そのため、通信量を抑制することは電力資源が乏しいセンサネットワークにとって有効である。また、小型軽量化が求められるセンサ端末は計算資源にも制約があるため、計算量を減らす必要がある。さらに、ネットワーク全体の稼働可能期間を延ばすためには、一部のノードに通信の集中が起らないようなルーティング方法が望まれる。そこで、本論文ではセンサネットワークにおける位置情報を必要としないルーティング方式を対象とし、既存の位置情報を用いた手法に対して通信量を増加させずに、かつ計算量が少ない簡易なルーティング方式、HVGF

(Hop-Vector based Greedy forwarding)を提案する。提案方式では  $n$  個の基準点を設け、基準点からのホップ数を要素とするようなベクトルを全ノードに作成し、その  $n$  次元ベクトル間の距離に基づいて greedy forwarding を行う。

## 2. 関連研究

ここでは対象とするセンサネットワークについて述べ、次にセンサネットワークを対象とした既存のルーティング方式を位置情報を用いる手法と位置情報を必要としない手法に分けて整理する。

### 2.1 対象とするセンサネットワーク

センサネットワークは取得したセンシングデータの格納場所により、主に External Storage (以下, ES), Local Storage (以下, LS), Data-Centric Storage (以下, DCS) の3つに分類される。まず, ES ではセンシングデータをセンサネットワークの外部(シンクノードなど)に格納する。センシングの度にデータをネットワークの外部にある格納場所まで転送する必要がある、格納場所に近いセンサ端末に負荷が集中するという問題がある。しかし、ネットワークの外部にデータが格納されているため、観察者からのクエリに対するコストがないという利点がある。

次に, LS ではデータはセンシングしたセンサ端末自身が保持する。各センサ端末はセンシングと同時にデータを格納するので、格納場所までの通信コストがないという利点がある。しかし、観察者からのクエリを送る際、どのノードがどのようなデータを保持しているかわからず、クエリをフラッディングする必要があるため、クエリの通信コストが非常に大きいという課題がある。

これらに対し DCS はセンシングデータの属性に基づきデータの格納場所を決める方法である。センシングデータは属性に基づいてネットワーク内のセンサ端末に対応付けられ、センサ端末は対応付けられた属性のデータを保持する役割を担う。DCS では属性に基づいてデータの格納場所を特定可能であり、観察者からのクエリをユニキャストで送れるので、通信量を削減することができる。

本提案手法は DCS を対象とした、任意の2点間の転送(ルーティング)について考察する。DCS では、ネットワークをセンシングデータの属性数に分割し、分割された区画に属性を対応付け、区画ごとにデータを保存するノードを

配置する。各センサでデータが発生すると、その属性に対応するノードをあて先としてデータを転送する。センサが属性に1対1に対応するように規則正しく配置されている場合(以下、計画配置ネットワークと呼ぶ)には、属性ごとにデータを保持するノードが一意に決まる。一方、属性に対応する区画とは独立にセンサの配置が行われた場合は、その属性のデータを保存するノードを一意に決定する手段が必要となる。

### 2.2 既存のルーティング方式

センサネットワークでのルーティング方式は大きく分類すると GPS などにより取得した位置情報を用いる手法と、位置情報を必要としない手法に分けられる。位置情報を用いるルーティング方式の一つに、隣接ノードとあて先の位置情報のみでルーティング可能な GPSR<sup>[2]</sup>がある。この手法は自分の通信範囲内で一番あて先に近いノードに転送する greedy forwarding と、greedy forwarding が適用できない場合に用いられる perimeter forwarding の2つから成る。perimeter forwarding では greedy forwarding が適用できなくなった時点のノードとあて先を結んだ直線が横切っている面に沿って、左回りに転送する(right-hand rule)。perimeter forwarding は greedy forwarding が適用できるノードに到達するまで繰り返される。

位置情報を用いるルーティング方式では GPS や超音波、無線電波強度などを用いて位置情報が取得できることを前提としている。しかし、現時点で、屋外・屋内・地下を問わず利用でき、低コストで精度の良い測定が行える手法はまだ確立されていない<sup>[3]</sup>。しかも、位置情報を取得するためには、各センサ端末に GPS 受信モジュールのような位置を特定するための特別なデバイスを付与する必要がある。これは、各センサ端末を低コスト化、小型軽量化する上でネックになる。さらに、位置情報を取得する際にも電力を消費してしまう。そこで、位置情報を必要としないルーティング方式も提案されている<sup>[4]</sup>。

[4]ではセンサネットワーク上に構築したツリー状の経路を用いたルーティング方式を提案している。この手法では、各ノードにサブツリーの大きさに比例した範囲の値(ラベル)を割り当てる。ここで、そのラベルはツリーの横方向に値が単調増加(または単調減少)するように割り振られる。ルーティングの際は、あて先ノードが共通のサブツリーとなるようなノードを経由して転送される。ここで、ルートか

らのホップ数(深さ)が同じノード同士がリンクを構築し、横方向への転送を可能にすることで、ルート付近のノードへの負荷の集中を緩和するとともに、あて先までのホップ数を削減する工夫を行っている。ラベル作成時に必要な情報量・通信回数が少ないのでエネルギーコストは低いという利点があるが、2点を基準とするホップ数から各ノードの位置を求めているため、十分な精度が得られず、結果的にルート付近のノードへ負荷が集中するケースが多いと考えられる。

センサネットワークに限定せず、一般のアドホックネットワークを対象とした位置情報を用いないルーティング方式として[5]がある。[5]ではすべてのノードについて、ノード間のホップ数から求まる仮想的な位置(virtual coordinate)を求め、その仮想的な位置情報を基に greedy forwarding を行う。ここでは、周辺ノードの位置から順次、ノードの相対的位置を求めていく。シミュレーションにより、パケット転送成功率において正確な座標を用いた場合と同程度の、または、空洞や障害物がある場合にはそれを上回る性能があることが示されている。しかし、仮想的な位置を求める過程で、通信や計算のオーバーヘッドが非常に大きいという課題がある。従って、センサネットワークへの適用は現実的ではない。

### 3. 提案方式 HVGF

本論文ではセンサネットワークにおける位置情報を必要としないルーティング方式を対象とし、既存の位置情報を用いた手法に対して通信量を増加させずに、かつ計算量が少ない簡易な方式である HVGF (Hop-Vector based Greedy Forwarding) を提案する。

提案手法では  $n$  個 ( $n > 2$ ) の基準点を設け、全てのノードに基準点からのホップ数を要素とするベクトルを作成する。ここで、ルーティングの精度を上げるために、基準点をそれぞれが互いになるべく離れるように指定する必要がある。そこで、基準点をネットワークの周辺に配置されているノード(周辺ノード)の中から選択することとした。そのため、まず周辺ノードを判別し、それらの中からそれぞれができるだけ離れるように  $n$  個の基準点を選択する。そして、基準点からのホップ数を要素とするベクトルを全てのノードに作成し、その  $n$  次元ベクトルをアドレスとして用いた greedy forwarding を行う。ここで、greedy forwarding とは、各ノードが隣接ノードの情報を保持しており、自分の隣接ノードの中で一

番あて先に近いノードへ転送することをあて先に届くまで繰り返す転送方法[4]のことである。

提案手法の詳細を以下に述べる。まず、3.1から3.3でルーティングの基準になる各ノードのアドレスの作成方法について述べる。次に3.4で計画配置ネットワークにおいて、あて先ノードが一意に指定されることを前提とした場合のルーティング手法について提案する。さらに3.5で、一般のネットワークに適用するための拡張について述べる。

#### 3.1 周辺ノード判別

ここでは周辺ノードの判別手順を述べる。この手順により、各ノードは自律的に自分が周辺ノードであるかどうかを判定する。

##### 【周辺ノード判別手順】

- 1) ネットワークの中央付近の1つのノード (*bootstrap node*) を指定する。ここで、中央付近とは厳密にネットワークの中心である必要はなく、周辺からある程度はなれていれば良い。
- 2) *bootstrap node* は、ネットワーク内に HELLO メッセージをブロードキャストする。この HELLO メッセージには、ノードを1つ経由するごとに1加算されるカウンタが含まれており、HELLO メッセージを受け取ったノードは、*bootstrap node* までのホップ数 ( $H_b$ ) を知る。
- 3) 各ノードは  $H_b$  を自分の通信範囲内だけにブロードキャスト(1ホップブロードキャスト(1-HBC))する。これにより、各ノードは1ホップ隣接ノードの  $H_b$  のリストを得る。
- 4) 各ノードは、自分とその1ホップ隣接ノードの  $H_b$  のリストを1-HBCする。これにより、各ノードは2ホップ以内の隣接ノードの  $H_b$  のリストを得る。
- 5) 各ノードは、この  $H_b$  のリストを参照し、2ホップ以内の隣接ノードの中で自分より *bootstrap node* から遠い(ホップ数が多い)ノードがない場合は、自律的に周辺ノードとなる。 □

#### 3.2 基準点選択

3.1で判別した周辺ノードの中から、それぞれができるだけ離れるように基準点を選択する。以下に基準点選択手順を示す。

まず、全周辺ノードにランダムな値のタイマをセットする。このタイマが一番最初に発火したノードが最初の基準点となり、自身のIDを HELLO メッセージに含めてブロードキャスト

トする。その際、HELLO メッセージにはノードを1つ経由するごとに1加算されるカウンタが含まれており、カウントアップしながら転送される。以降、これを受信した周辺ノードの中から、すでに選ばれている基準点から遠いノードほど小さな上限値のタイマをセットするなどの方法により、互いにできるだけ離れるように  $n$  個の基準点を順次選択し、そのたびに HELLO メッセージをブロードキャストする。これにより、各ノードは各基準点までのホップ数を知ることができる。なお、ここで用いられるタイマが発火するタイミングは選択された基準点からのブロードキャストがネットワーク全体にいきわたった後になるように設定する必要がある。

### 3.3 $n$ 次元ベクトル作成

次に、各ノードに  $n$  個の基準点からのホップ数を要素とする  $n$  次元ベクトルを作成する手順について述べる。

各ノードは 3.2 節で選択された基準点からフラッディングされた HELLO メッセージを受信することで、各基準点までのホップ数を知りベクトルとして保持する。

基準点の数を  $n$  とすると、ノード  $i$  のベクトル  $\vec{v}_i$  は

$$\vec{v}_i = (h_1^i \cdots h_n^i)^T$$

と表される。

基準点や周辺ノードを含め、全てのノードに  $n$  次元ベクトルを作成する。図 1 に  $n=4$  のときの例を示す。

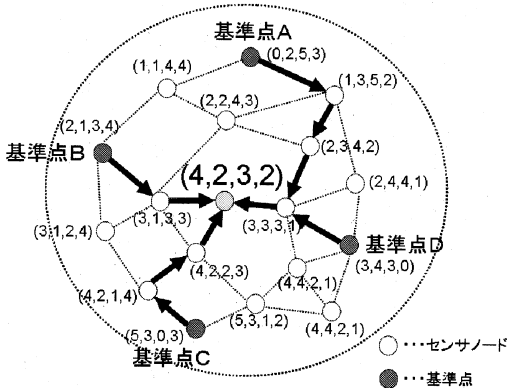


図 1 :  $n$ 次元ベクトル作成例 ( $n = 4$  の場合)

### 3.4 HVGF : Hop-Vector based Greedy Forwarding

本節では、3.3 で作成した  $n$  次元ベクトルを各ノードやあて先のアドレスとして用いた

greedy forwarding (HVGF) について述べる。この手法は計画配置ネットワークを対象とし、あて先アドレスが与えられたときに、データを保持すべきノードが一意に決まる場合を想定した手法である。

以下、本論文では各ノードやあて先のアドレスを表す  $n$  次元ベクトルを単に「アドレス」と呼ぶ。また、送信されるパケットのヘッダにはあて先アドレスとして  $n$  次元ベクトルが含まれていることとする。さらに、各ノードはノード ID により一意に識別できるものとする。

まず、各ノードは自分のアドレス情報を 1-HBC する。これにより、各ノードは自分の通信範囲内にあるノード (隣接ノード) のアドレス情報を得る。次に得られたアドレス情報を基に、各ノードは隣接ノードの中でアドレスがあて先ノードのアドレスと一番「近い (距離が小さい)」ノードに転送する。これを受信したノードは同様に、自分の隣接ノードの中で一番あて先アドレスと近いアドレスを持つノードに転送する。基本的に、この動作をあて先に届くまで繰り返す。途中のノードにおいて、隣接ノードの中であて先アドレスとの距離が小さいものが見つからなかった場合はそのノードで停止し、データを保持する。

ここで  $\vec{v}_k$  と  $\vec{v}_l$  の間の距離  $D(\vec{v}_k, \vec{v}_l)$  はベクトル  $\vec{v}_{kl} = (\vec{v}_k - \vec{v}_l)$  のノルムとして定義する。

$$D(\vec{v}_k, \vec{v}_l) = \left\| \vec{v}_{kl} \right\|_p = \left( \sum_{i=1}^n (h_i^k - h_i^l)^p \right)^{\frac{1}{p}}$$

以下に記号の定義を示し、続いて HVGF の手順の詳細を述べる。

#### 【記号の定義】

$V$  : ノードの集合

$R \subseteq V \times V$  : 隣接しているノード間の関係

$N_x = \{y \mid (x, y) \in R, y \in V\}$  : ノード  $x$  の隣接ノードの集合

$\vec{v}_d$  : あて先アドレス

$\vec{v}_x$  : ノード  $x$  のアドレス

#### 【Hop-Vector based Greedy Forwarding 手順】

- 1) あて先アドレスを指定されたノード  $s$  はそのアドレスが自分のアドレスと一致するかどうか比較する。



- 2) 一致した場合にはデータを保持する。
  - 3) 一致しなかった場合、ノード  $s$  は全てのノード  $y \in N_s$  のアドレス  $v_y$  とあて先アドレス  $v_d$  との距離  $D(v_d, v_y)$  を計算する。
  - 4) ノード  $s$  は  $D(v_d, v_{y_k}) = \min_i \{D(v_d, v_{y_i}) \mid y_i \in N_s\}$  かつ  $D(v_d, v_s) > D(v_d, v_{y_k})$  なるノード  $y_k$  が存在するかどうか調べる。
  - 5)  $y_k$  が存在しなかった場合には転送を停止し、データを保持する。
  - 6)  $y_k$  が存在した場合には  $y_k$  へデータを転送する。
  - 7) ノード  $s$  からデータを受け取ったノード  $y_k$  はあて先アドレスと自分のアドレスを比較し、以降、2) から 7) を繰り返す。□
- ここで、2つのノードが「隣接している」とは互いの通信範囲内に入っており、双方向のリンクでつながっている状態と定義し、互いに隣接しているノードを他方の「隣接ノード」と呼ぶ。

### 3.5 HVGF の拡張

次に、属性に対応する区画とは独立にセンサの配置が行われた場合に適用するために HVGF を拡張する。計画配置ネットワークでは属性ごとにデータを保存すべきノードが一意に決定する。これに対し、一般にはノードの配置とネットワークの分割が必ずしも対応するわけではない。そのような場合に DCS に対応するためには、以下の3つの点について考慮する必要がある。

#### 1) アドレス決定

上述のように、計画配置ネットワークでは属性ごとにデータを保存すべきノードが一意に決定するため、このノードのアドレスをあて先アドレスとして用いることができる。しかし、ノードの配置とネットワークの分割が対応しない場合には、1つのノードのアドレスをその区画を表すアドレスとして使用できない。そのため、それぞれの区画に対して、その区画を表す一つのアドレスを決める必要がある。そこで、ここでは、仮に配置した基準点(仮基準点)から各区画までのホップ数の概算を、ネットワークサイズ、ノード密度、各ノードの通信範囲より求め、それらを要素とする  $n$  次元ベクトルをその区画を示すアドレスとする。

#### 2) データを保存するノードの決定

一般に前述のように定めたアドレスとノ

ードとは1対1に対応しない。すなわち、必ずしも指定されたアドレスを持つノードが存在するとは限らず、反対に複数のノードが同じアドレスを持つ場合もありうる。

そのような場合でも前述のように定めたアドレスをあて先として指定したときに、データを保存すべきノード(担当ノード)を一意に決定する必要がある。なぜならば、本来なら1つのノードに保存されるべきデータが複数のノードに分散して保存された場合、観察者からその属性にクエリが送られたときに、その属性の全てのデータを取得することができなくなるからである。また、クエリ自体が担当ノードに届かない場合も考えられる。

そこで、ここでは、あて先として指定されたアドレスを持つノードが存在しない場合には、あて先として指定されたアドレスと一番近いアドレスを持つノードを担当ノードとする。ここで、その候補が複数ある場合には、ノードIDが一番若いものを担当ノードとする。また、あて先として指定されたアドレスを持つノードそのものが複数存在する場合には、ノードIDが一番若いノードを担当ノードとする。

#### 3) 保存先ノードへのデータ転送

対応付けられた属性の担当ノードがその属性のデータを全て保持するためには、担当ノードあてに転送されたセンシングデータが担当ノードまで必ず到達する必要がある。なぜなら、担当ノードをあて先として転送されたときに、途中で転送が停止した場合、その停止したノードにデータが保存されてしまうからである。

センシングデータやクエリがそのデータの属性に割り当てられた担当ノードに到達することを完全に保障するためには、転送が停止した際に、そのノードは他の全てのノードに、よりあて先アドレスに近いアドレスをもつノードがないか、問い合わせれば良い。しかし、このことは、通信量、消費電力の増加を招く。これは、電力資源が少ないセンサネットワークでは大きな問題となる。

そこで、ここでは探索時のクエリの送信を1ホップに限ることで、通信量、消費電力を軽減した最近接ノード探索手法を提案する。最近接探索手法では2ホップ先の隣接ノードに、転送が停止したノードよりあて先に近いノードが存在しないかどうかを問い合わせ、存在した場合にはその中で一番近いノードに転送する。存在しない場合には転送を停止し、そのデータを保持する。この手法はセンシングデータやクエリが必ず担当ノードに到達することを保障しているわけではないが、実用的な精度で探索で

きると考える。詳しい手順を以下に述べる。  
以下において  $I_x$  はノード  $x$  の ID を表す。その他の記号は 3.4 と同様である。

**【最近接ノード探索手順】**

- 1) ノード  $k$  で転送が停止した場合、ノード  $k$  は全ての  $y \in N_k$  に対して  

$$D(\vec{v}_d, \vec{v}_{z_l}) = \min\{D(\vec{v}_d, \vec{v}_{z_l}) \mid \forall y \in N_k, z_l \in N_y\}$$
 なるノード  $z_l$  の ID  $I_{z_l}$  と、ノード  $z_l$  のアドレスとあて先アドレスとの距離  $D(\vec{v}_d, \vec{v}_{z_l})$  を問い合わせるクエリを 1-HBC で送信する。
- 2) ノード  $k$  からクエリを受け取った全ての  $y \in N_k$  は、  

$$D(\vec{v}_d, \vec{v}_{z_l}) = \min\{D(\vec{v}_d, \vec{v}_{z_l}) \mid \forall y \in N_k, z_l \in N_y\}$$
 なるノード  $z_l$  の ID  $I_{z_l}$  と、ノード  $z_l$  のアドレスとあて先アドレスとの距離  $D(\vec{v}_d, \vec{v}_{z_l})$  をノード  $k$  へ知らせる。
- 3) 全ての  $y \in N_k$  からクエリに対する返信を受け取った後、ノード  $k$  は  

$$D(\vec{v}_d, \vec{v}_{z_l}) = \min\{D(\vec{v}_d, \vec{v}_{z_l}) \mid \forall y \in N_k, z_l \in N_y\}$$
 なるノード  $z_l$  とあて先との距離  $D(\vec{v}_d, \vec{v}_{z_l})$  と、ノード  $k$  のアドレスとあて先アドレスとの距離  $D(\vec{v}_d, \vec{v}_k)$  を比較する。
- 4)  $D(\vec{v}_d, \vec{v}_{z_l}) < D(\vec{v}_d, \vec{v}_k)$  の場合は、 $z_l$  にパケットを転送する。
- 5)  $D(\vec{v}_d, \vec{v}_{z_l}) > D(\vec{v}_d, \vec{v}_k)$  の場合は転送を停止し、ノード  $k$  がデータを保持する。
- 6)  $D(\vec{v}_d, \vec{v}_{z_l}) = D(\vec{v}_d, \vec{v}_k)$  の場合には、  

$$I_{z_l} < I_k$$
 の場合のみノード  $z_l$  に転送する。 □

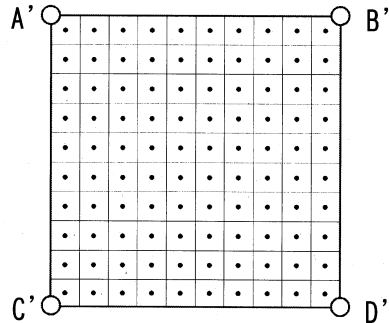
**4. 評価**

提案手法を評価するためにシミュレーションを行い、提案手法である HVGF (Hop-Vector based Greedy Forwarding) と HVGF に最近接ノード探索手法を適用した手法 (HVGF-S)、既存の実際の位置情報を用いた greedy forwarding (以下、TPGF (True Position Greedy Forwarding) と呼ぶ。) の 3 つを比較した。表 1 にシミュレーション環境を示す。

表 1: シミュレーション環境

ネットワークサイズ	25m×25m 350m×350m
センサノードの数	50-9800 個
センサノードの通信範囲	8m
基準点の数 ( $n$ )	4
シミュレーション回数	100回×100セット

以下の評価では、属性に対応する領域として  $m \times m$  の格子状に分割した区画を考える。図 2 は  $m=10$  の場合を示す。同図において  $A' \sim D'$  は仮基準点を表す。計画配置ネットワークでは、各区画の中心に 1 つのノードが存在し、隣の区画 (斜め含む) のノードが互いに隣接しているものとする。



○: 仮基準点  
 ・: 計画配置ネットワークにおけるノードの位置

図 2: 属性に対応付けられる区画

**4.1 評価基準**

本論文では、提案手法の評価に以下の二つの基準を用いる。

**i) 成功率**

ルーティングの際の経路構築の基本的な性能を表す指標として成功率を用いる。送信ノードから送られたパケットが担当ノードに到着した場合を成功とし、その割合を成功率と定義する。

TPGF では、あて先アドレスを 2 次元座標で指定する。ランダムにノードを配置する一般のネットワークでは、指定したあて先アドレスに一番近いノードが担当ノードとする。

HVGF, HVGF-S では、仮基準点から各区画の中心までのホップ数の概算をネットワークサイズやノード密度、通信範囲から求め、それらを要素とする 4 次元ベクトルであて先を

指定する。ランダムにノードを配置する一般のネットワークでは、指定したアドレスと一番近いアドレスを持つノードを担当ノードとする。

なお、担当ノードの候補が複数ある場合は、ノード ID が一番若いものを担当ノードとする。

#### ii) 平均ホップ数

通信量の指標として平均ホップ数を用いる。ここで、送信ノードからあて先ノードまでに経由するノード数の平均値を平均ホップ数と定義する。

### 4.2 シミュレーション結果

TPGF, HVGF, HVGF-S それぞれ、100 回異なる送信元から異なるあて先へ転送することを 1 セットとして、100 セットのシミュレーションを行い、その平均値を結果として用いた。なお、平均ホップ数は転送が成功した場合のみを計測した。

ここではまず、4.2.1 でアドレスとノードが 1 対 1 に対応する計画配置ネットワークにおける成功率と平均ホップ数についての結果を述べる。4.2.2 以降は一般的なノード配置においてネットワークサイズやノード密度の観点から比較する。

#### 4.2.1 計画配置ネットワークにおける結果

330m×330m のネットワークにおいて、 $m=60$  としてシミュレーションを行った。その結果を表 2 に示す。同表より、TPGF, HVGF, HVGF-S のいずれも成功率は 100% となっている。提案手法の平均ホップ数は既存の TPGF に比べて約 18% 少なくなっており、それだけ転送の際に消費する電力が少ないと考えられる。

表 2：計画配置ネットワークにおける結果

	成功率 (%)	平均ホップ数 (hop)
TPGF	100	27.9
HVGF	100	22.9
HVGF-S	100	22.9

#### 4.2.2 一般のネットワークにおける結果

次に、ノードをランダムに配置してシミュレーションを行った。

図 3 はノード密度を固定(7 個/100m<sup>2</sup>)し、ノード数を 50 から 9800 まで増加させたときの、それぞれの成功率を示している。同図よりノード数の増加に伴い、HVGF の成功率は TPGF に比べて大幅に低下している。これは、転送の途中で自分の隣接ノードの中によりあて先に近いノードが存在しない場合があるが、ネットワークサイズが大きくなるに従い、送信

元とあて先の距離も長くなるため、自分よりあて先に近いノードが存在せず転送が停止する割合が増えたためだと考えられる。これに対し、HVGF-S の成功率は TPGF に比べてわずかなに上まわっている。これは、自分の隣接ノードの中によりあて先に近いノードが存在しない場合でも最近接ノード探索手法を用いて 2 ホップ先まで問い合わせることで、よりあて先に近いノードを見つけることができたためだと考えられる。

ネットワーク規模の拡大に伴う、HVGF-S の成功率の低下はわずかであり、大規模なネットワークに対して拡張性があると考えられる。また、最近接ノード探索手法を適用した HVGF-S は未適用の HVGF より成功率が高くなっており、最近接ノード探索手法の有効性が示された。

図 4 はノード密度を固定(7 個/100m<sup>2</sup>)し、ノード数を 50 から 9800 まで増加させたときの、送信元からあて先までの平均ホップ数を表している。同図より、ネットワークサイズにほぼ比例して平均ホップ数が増加している。TPGF-S に比べて HVGF-S は約 15% 平均ホップ数が増加している。これは、実際の位置情報を用いた TPGF に比べて簡易なホップ数を用いている提案手法はルーティングの精度が低いためだと考えられる。

図 5 はノード密度の異なるネットワークにおける成功率の違いを示している。横軸は 100m<sup>2</sup> あたりのノード数、縦軸は成功率を表している。同図より、8 個/100m<sup>2</sup> より粗なネットワークでの HVGF-S の成功率は、TPGF に比べて高い値となっている。これは、比較的粗なネットワークでは提案手法の方がネットワークポロジをより反映しているからだと考えられる。これに対し、8 個/100m<sup>2</sup> より密なネットワークでは、TPGF が 100% に近づいているのに対し、HVGF-S は 93% 程度にとどまっている。失敗した例を解析してみた結果、周辺ノード付近があて先になった場合が多かった。よって、属性に対応付ける区画から周辺ノード付近を除外することで改善できる可能性があると考えられる。

図 6 はノード密度の異なるネットワークにおける、送信元からあて先までの平均ホップ数を表している。横軸は 100m<sup>2</sup> あたりのノード数、縦軸は平均ホップ数を表している。同図より、HVGF, HVGF-S の平均ホップ数はノード密度にかかわらずほぼ一定であり、ノード密度に対する拡張性があると考えられる。

5 個/100m<sup>2</sup> より粗なネットワークでの

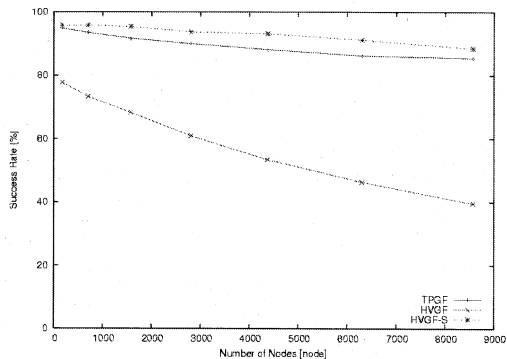


図 3：ネットワークサイズと成功率

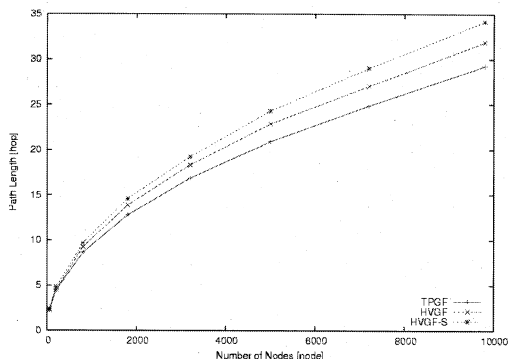


図 4：ネットワークサイズと平均ホップ数

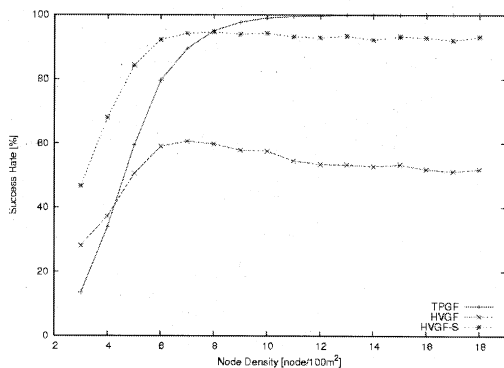


図 5：ノード密度と成功率

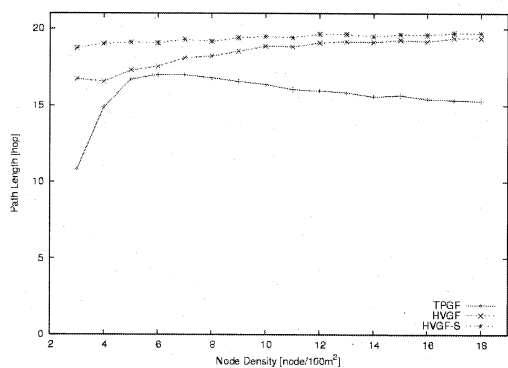


図 6：ノード密度と平均ホップ数

TPGF の平均ホップ数が少ないのは、成功率も低いことから送信元とあて先が近い場合のみ転送が成功したためだと考えられる。5 個/100m<sup>2</sup> より密なネットワークでの提案手法の平均ホップ数は、既存の手法の TPGF に対し約 18% 程度増加している。

以上より、提案手法は計画配置ネットワークやノード密度が低いネットワークに適した手法であることが分かる。また、ノード密度が高いネットワークでの成功率は十数% 下がるものの、GPS などを用いる必要がないため、センサネットワークにおいては有効性が高いと考えられる。

## 5. まとめ

本論文ではセンサネットワークにおける位置情報を必要としないルーティング方式を対象とし、既存の位置情報を用いた手法に比べて通信量を大きく増加させずに、かつ計算量が少ない簡易なルーティング手法を提案した。提案手法ではネットワークの周辺に  $n$  個の基準点を設け、基準点からのホップ数を要素とするベクトルを全ノードに作成し、 $n$  次元ベクトル間の距離に基づいて greedy forwarding を行う。

シミュレーションにより、既存の位置情報を

用いる greedy forwarding に比べ、ノード密度が低いネットワークにおいて有効なことが分かった。

今後は、障害物や空洞があるネットワークでのシミュレーションを行う予定である。

## 参考文献

- [1] 安藤 繁, 田村 陽介, 戸辺 義人, 南 正輝, “センサネットワーク技術 ユビキタス情報環境の構築に向けて”, 東京電機大学出版局, May 2005
- [2] Brad Karp and H.T.Kung, “GPSR: Greedy Perimeter Stateless Routing for Wireless Networks”, MobiCom 2000
- [3] 小野 昌之, 福井 潔, 柳原 健太郎, 福永 茂, 原 晋介, 北山 研一, “無線を使った位置検出”, 沖テクニカルレビュー, 第204号 Vol.72 No.4, P24-27, October 2005
- [4] James Newsome and Dawn Song, “GEM: Graph Embedding for Routing and Data-Centric Storage in Sensor Networks Without Geographic Information”, ACM SenSys November 2003
- [5] Ananth Rao, Sylvis Ratnasamy, Christos Papadimitriou, Scott Shenker and Ion stoica, “Geographic Routing without Location Information”, MobiCom 2003