

NLOS 混在環境における無線センサネットワークの 集約型自己組織化ノード位置推定方式

高島 優斗¹ 北之馬 貴正¹ 安達 直世² 滝沢 泰久²

概要: 無線センサネットワークにおいて、センシングデータの取得位置は重要な情報である。そこで我々は自己組織化マップ (SOM) を用いたセンサノード位置推定方式 (Self-Organizing Localization, SOL) を提案している。SOL は、極少数のアンカノードを使用し、測距デバイスを用いずに、高精度な位置推定が可能であり、障害物による見通し内 (LOS: Line-Of-Sight) と見通し外 (NLOS: Non-Line-Of-Sight) が混在する環境においても従来方式と比較して位置精度の劣化が少ない。しかし、NLOS 混在環境での位置精度は十分ではない。また、SOL はセンサノード間通信が増大する課題がある。本稿では、NLOS 環境での位置推定精度の向上を行い、かつノード間通信数の削減を図るため、クラウドコンピューティングを前提とする集約型自己組織化ノード位置推定方式を提案し、その有用性を示す。

1. はじめに

無線端末とセンサを組み合わせた無線センサネットワーク (WSN) は、リアルワールドなサービスを目指すユビキタスネットワーク社会において、重要な技術要素である。リアルワールドなサービスの実現には、物理的な位置情報が重要な要素となる。そのため、WSN においてそのノード位置を正確に特定するための研究が数多く行われている。ノード位置推定技術は測距デバイスの有無から Range-Based と Range-Free に大別できる。Range-Based は位置推定処理に TDOA (Time Difference Of Arrival) や TOA (Time Of Arrival) といった測距デバイスを利用する方式である。代表例として GPS (Global Positioning System) が挙げられる。Range-Based は高精度な位置推定が可能であるが、特別な測距デバイスを必要とするため、コストや電力消費においてセンサノードには適さない。一方、Range-Free は、Centroid [4] や DV-Hop [5] などがあり、アンカノードとの相対関係から位置推定を行うため測距デバイスを必要としない。しかし、Range-Free が高精度な位置推定するためには、アンカノードを大量に配置する必要がある。そのため、使用環境が制限され、拡張性、柔軟性に欠ける問題点がある。以上のことから、我々は、測距デバイスを必要とせず、極少数のアンカノードにおいてセンサノード位置を高精度に推定するため、自己組織化マップ (SOM) を用いたセンサノード位置推定方式 (Self-Organizing Localization,

SOL) [1] を提案している。WSN は、長期の計測目的に利用されるため、各ノードの消費電力は重要な要素となる。しかし、SOL は各ノード間で相互に位置情報を交換するため、各ノードの通信回数が多くなりセンサノードの電力消費が大きくなる課題がある。また、SOL は各ノードにおいて適当数の近傍ノードを必要とするが、NLOS 混在環境では一部のノードにおいて十分な近傍ノード数が確保されず、精度劣化する場合がある。本研究は、以上の問題を解決するため、クラウドコンピューティングを前提とする集約型 SOL を提案する。集約型 SOL は、各センサノードの隣接ノード情報をクラウド環境に集約し、これにより構成される仮想 WSN に SOL を適用する方式である。

2. 関連研究

2.1 Range-Based 位置推定方式

Range-Based 位置推定方式は測距デバイスを用いる方式である。無線端末で用いられている測距デバイスは TDOA (Time Difference Of Arrival), TOA (Time Of Arrival) のいずれかの方式である。

TOA 方式を用いた代表的な位置推定方式は GPS である。GPS は 4 機以上の GPS 衛星と通信を行い、送信側から受信側に信号が到着するまでの時間を測定し、ノード間の距離を計算して三辺測量を行い、自身の位置を推定する方式である。

TDOA 方式を用いた位置推定技術としては Active Bat [2] と Cricket [3] などが挙げられる。TDOA 方式は送信側と受信側で異なる伝送媒体による通信を行い、伝搬到着時間の

¹ 関西大学 理工学研究科
² 関西大学 環境都市工学部

差からノード間の距離を計算する方式である。これらも測定した距離を用いた三辺測量から位置推定を行う。

TOA 方式や TDOA 方式を用いた Range-Based 方式は位置推定の精度が高いが、センサノードに特別な測距デバイスを用いる必要があり、センサノードの消費電力やコストの面において WSN での利用には適さない。

2.2 Range-Free 位置推定方式

Range-Free 位置推定方式は、位置推定に測距デバイスを用いない方式である。例としては、Centroid 方式 [4]、DV-Hop 方式 [5] や APIT 方式 [6] を説明する。

Centroid 方式では、まず、アンカノードが自身の位置情報を含んだパケットを一定の時間間隔でブロードキャスト送信する。位置推定処理を行うノードは、このパケットを受信することで自身と通信可能なアンカノードの位置情報を取得し、それらの重心から自身の位置を推定する。

DV-Hop 方式は、アンカノード間の距離とホップ数から 1 ホップの距離を見積り、この見積もった 1 ホップ距離とノードとアンカノード間のホップ数からノードのアンカノードまでの距離を算出する。この距離を 3 つ以上を用いた三辺測量から位置推定する。

APIT 方式は Centroid 方式と同様にアンカノードが位置情報を含んだパケットを定期的送信する。各ノードは受信したパケットから 3 つのアンカノードの組み合わせで作成可能な三角形を導き出す。この三角形すべてに対して、自分が内側か外側にいるのかの計算を行い、自分の位置を絞り込んでいく方式である。

これらの方式は少なくとも 3 つ以上のアンカノードが必要であり、精度の向上には多量なアンカノードを有するため広範囲な空間への適用は困難である。

3. Self-Organizing Localization (SOL)

SOL は、次の特徴を有する。

- アンカノードへの依存性が極めて低い。具体的には、アンカノード数が 2 点で相対位置を、3 点で絶対位置を推定可能である。
- 測距デバイスを必要としない。
- WSN のノード数が増えるに従い、位置推定精度は高精度化し、従来方式の精度を凌駕する。
- 従来方式と比較して、NLOS 混在環境において位置推定精度の劣化を抑制する。

3.1 位置推定のアルゴリズム

SOL は、3 つのステップにより位置推定を行う。

[Step.1] 各ノードは、自己位置をランダムに生成し、これを仮自己位置とする。以降、ノード i の修正 t 回目の仮自己位置を $w_i(t)$ と表記する。各ノードはこの仮自己位置を 1 次近傍 (1 ホップ) ノードにブロードキャスト送信

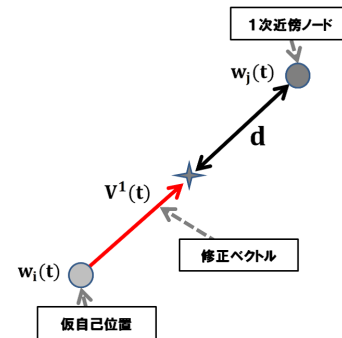


図 1 1 次近傍ノードによる位置修正

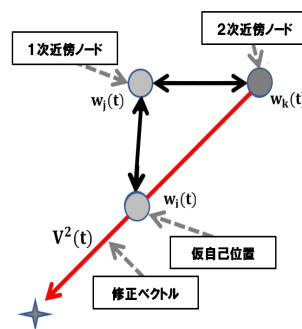


図 2 2 次近傍ノードによる位置修正

する。

[Step.2] 1 次近傍ノード j から仮自己位置 $w_j(t)$ を受信したノード i は、仮自己位置 $w_i(t)$ と 1 次近傍ノード j の仮自己位置 $w_j(t)$ の直線上においてノード j から 1 ホップ距離 d に修正する 1 次近傍修正ベクトル $V_i^1(t)$ を次式のように算出する。(図 1) 1 ホップ距離 d は絶対距離ではなく相対距離として、ホップ数とする。

$$V_i^{\{1\}}(t) = \frac{d - |w_i(t) - w_j(t)|}{|w_i(t) - w_j(t)|} (w_i(t) - w_j(t)) \quad (1)$$

また、1 次近傍ノードによる修正と同様に、ノード i において 2 次近傍ノード k から 2 ホップ距離とする 2 次近傍修正ベクトル $V_i^2(t)$ を次式のように算出する。(図 2)

$$V_i^{\{2\}}(t) = \frac{d + d - |w_i(t) - w_k(t)|}{|w_i(t) - w_k(t)|} (w_i(t) - w_k(t)) \quad (2)$$

この 2 つの修正ベクトルを用いて次のように仮自己位置を修正する。

- ノード i が 1 次近傍ノード j より 2 次近傍ノード k から遠い仮自己位置の場合、2 次近傍ノードとの相対関係に誤りが無いことから 1 次近傍修正ベクトルのみで修正する。式 (1)
- ノード i が 1 次近傍ノード j より 2 次近傍ノード k に近い仮自己位置 ($|w_i(t) - w_j(t)| \geq |w_i(t) - w_k(t)|$) (図 3) の場合 (以下、トポロジ矛盾)), 1 次近傍ノードとの相対関係に誤りがあるため、1 次近傍修正ベクトルと

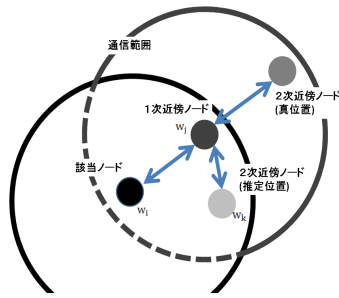


図3 トポロジ矛盾

2次近傍修正ベクトルの両方から仮自己位置 $w_i(t)$ を修正する。

以上の修正を次式に示す。

$$w_i(t+1) =$$

$$\begin{cases} w_i(t) + \alpha_i(t) \cdot V_i^{\{1\}}(t) & (|w_i(t) - w_j(t)| < |w_i(t) - w_k(t)|) \\ w_i(t) + \alpha_i(t) \cdot (V_i^{\{1\}}(t) + V_i^{\{2\}}(t)) & (|w_i(t) - w_j(t)| \geq |w_i(t) - w_k(t)|) \end{cases} \quad (3)$$

$\alpha_i(t)$ は t 回目の修正時のノード i の学習関数であり、次のようになる。

$$\alpha_i(t) = \eta \alpha_i(t-1) \quad (0 < \eta < 1) \quad (4)$$

η は正の減衰定数である。

[Step.3] 自身の仮自己位置 $w_i(t)$ に加えて、ランダムに1次近傍ノードの仮自己位置 $w_j(t)$ を選択し、これら2つのノードの仮自己位置を近傍ノードへブロードキャストする。

以上の Step.2 および Step.3 を繰り返し、各ノードは自己位置を推定し、ネットワークトポロジを再現する。

3.2 位置推定補正処理

SOL ではノード間は距離はホップ数を用いているため、推定位置は相対位置であり、絶対位置としては精度が低い。従って、絶対位置を算出し、位置推定精度を向上させるため、次の処理を行う。

- 推定トポロジにおけるトポロジ矛盾の判定と推定再試行の実施
- 推定トポロジを絶対座標への変換

3.2.1 推定トポロジにおけるトポロジ矛盾の判定と推定再試行の実施

- 位置推定処理が収束状態 (学習関数 $\alpha_i(t)$ が一定の閾値以下) になった段階でトポロジの矛盾の判定を開始する
- トポロジの矛盾が発生していると判定された場合、全ノードに対して位置推定処理の再試行を通知するメッセージを送信する。
- メッセージを受信した1次近傍ノードは、自身の学習

関数 $\alpha_i(t)$ を初期値の1に戻し、位置推定処理を再試行する。

また、上記の再試行通知の判定には以下の式を用いる。

$$\frac{I_i^{\{2\}}}{N_i^{\{2\}}} < \theta \quad (5)$$

$I_i^{\{2\}}$ は、ノード i におけるトポロジ矛盾の発生回数、 $N_i^{\{2\}}$ は、ノード i において認識している2次近傍ノード数、 θ は、誤再現判定閾値である。

3.2.2 推定トポロジを絶対座標への変換

3点のアンカノードの真位置 r_i と3点のアンカノードの推定位置 p_i を使用し、アフィン変換行列 A を求める。求めたアフィン変換行列 A をすべてのノードに与えることによって、推定トポロジを絶対座標へと変換する。 A は下記の式 (6) によって求められる。

$$\begin{pmatrix} r_{ix} \\ r_{iy} \end{pmatrix} = \begin{pmatrix} a & b & c \\ d & e & f \end{pmatrix} \begin{pmatrix} p_{ix} \\ p_{iy} \\ 1 \end{pmatrix} \quad (6)$$

$$A = \begin{pmatrix} a & b & c \\ d & e & f \end{pmatrix}$$

3.3 NLOS 混在環境における位置推定

SOL は、NLOS 混在環境において位置推定を行うため、下記の改良を行っている。

- ノード間トポロジ矛盾を解消するノード間ホップ度 (正の実数値)
- 3次近傍ノードでの位置修正の追加

3.4 ノード間トポロジ矛盾を解消するノード間ホップ度 (正の実数値)

SOL はホップ数をノード間距離として用いる。しかし、ホップ数は距離情報としては相対的でかつ解像度が低いためトポロジの矛盾が発生する場合がある。トポロジの矛盾の発生する原因は、1次近傍ノード間のホップ数1が距離情報として大きな値であると仮定できる。この仮定に従いホップ数1(正の整数値)を減衰して、1より小さい正の実数値とすることによりトポロジ矛盾を解消して位置推定を行う。(図4) この減衰した値をホップ度と呼ぶ。ノード間ホップ度は式 (7) のように算出する。 T は再試行回数であり、 d_T は再試行時のホップ度 (初期値 d_0 は1) である。

$$d_{T+1} = \frac{d_T}{T+1} \quad (7)$$

3.5 3次近傍ノードでの位置更新の追加

SOL は近傍ノードの位置情報が推定に重要な要素となる。しかし障害物の存在する環境においては、ノード配置によって近傍ノードが少数になる。そこで近傍ノード情報

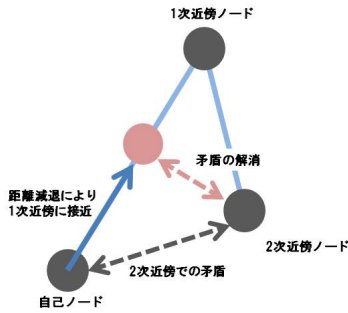


図4 ノード間トポロジ矛盾を解消するノード間ホップ度（正の実数値）

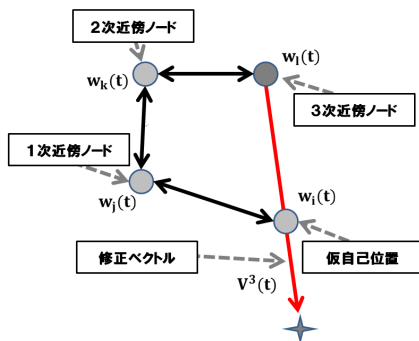


図5 3次近傍ノードによる位置修正

Fig. 5 Location update with 3 hop neighborhood node.

を増加させるため、3次近傍ノードの位置情報 $w_l(t)$ を取得させた。その際、ここでの入力ベクトルは、ノード i と3次近傍ノード k との距離は、少なくとも、1次近傍ノードより遠く位置することから2次近傍ノードと同様に次のように算出する。

$$V_i^{\{3\}}(t) = \frac{2d - |w_i(t) - w_l(t)|}{|w_i(t) - w_l(t)|} (w_i(t) - w_l(t)) \quad (8)$$

近傍ノードの位置情報は受信した1次近傍ノードからの2次近傍ノード、3次近傍ノード情報をそれぞれ任意に1つ近傍ノードを選択する。(図5)

また、修正の段階を初期段階、以降の段階の2つに分ける。初期段階では3次近傍ノードと1次近傍ノードによる位置修正を行い、初期段階以降は2次近傍ノードと1次近傍ノードにより位置更新を行う。すなわち、位置推定初期段階では比較的遠いノード位置情報を用いて大域的な処理を行い、その後比較的近いノード位置情報から、局所的な処理を行う。

4. 集約型 SOL

4.1 SOL の問題点

SOL は、各ノードが自律的に自己位置を推定する方式である。しかし、各ノードが個別に推定を行うため、下記の問題が発生すると考えられる。

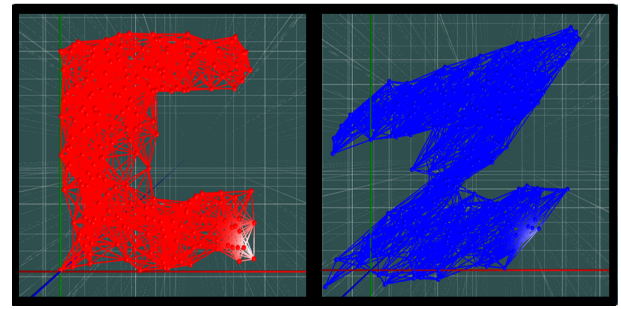


図6 折れ曲がりトポロジ (左:オリジナルトポロジ 右:推定トポロジ)

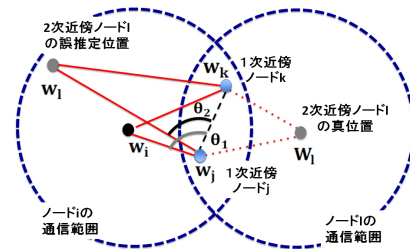


図7 折れ曲がり推定トポロジと角度（方向）による判定

- NLOS 混在環境における近傍ノード数の不足による精度劣化が起こり、推定トポロジが折れ曲がりトポロジとなり (図6) となる場合があり、位置精度の劣化の主要因となる。
- 各ノードの通信回数が多くなり、電力消費が大きくなる。

上記問題を解決するため、クラウドコンピューティングを前提とする集約型 SOL を提案する。集約型 SOL は、各センサノードからの隣接ノード情報をクラウド環境へ集約する。これにより構成する仮想 WSN へ SOL を適用した方式である。集約型 SOL は次のようなクラウド環境を想定する。

- WSN のセンサノードは自己の隣接ノード情報をセンシングし、これをクラウドサーバへ転送する。
- クラウドサーバは各センサノードの隣接ノード情報を集約し、仮想 WSN を構成する。

仮想 WSN はすべてのノードの隣接ノード情報を集約するため、完全なトポロジ情報を有する。一方、そのトポロジのジオメトリ（形状）は全くのランダム（でたらめ）である。集約型 SOL は、この完全なトポロジ情報をもつ WSN に SOL を適用（クラウド化）し、前述の問題解決するため次の処理を行う。

- 精度劣化の原因は折れ曲がりトポロジの発生である。しかし、距離判定ではトポロジ矛盾 (図7) を検出できないため角度判定を追加する。
- 完全なトポロジ情報を用いた N 次近傍ノード更新により大域的情報から全体形状の再現精度向上を図る。
- クラウドクライアントであるセンサノードは隣接ノード

ドの探索のみを行うため通信回数を大幅に削減できる。

4.2 クラウド環境におけるトポロジ情報を用いた位置推定アルゴリズム

4.2.1 仮想 WSN の構成

各センサノードは次のように近傍ノード情報を収集し、サーバ内で仮想的なネットワーク (仮想 WSN) を作成する。

- 自己ノード ID をデータとしてブロードキャスト (広告ブロードキャスト) 送信する。
- 各センサノードは受信した広告ブロードキャストから隣接ノードの ID を取得し、取得したノード ID リストを隣接ノード情報としてサーバに転送する。
- サーバはこれをクラウド環境へ転送する。

以上により、クラウド環境は全てノードの隣接ノード情報を取得し、保持する。集約型 SOL は上記の取得されて隣接ノードを用いてクラウド上で次のように近傍トポロジを構成する。

- ノード i の隣接ノード情報に含まれるノードをノード i の 1 次近傍ノードとする。
- 上記 1 次近傍ノード j の隣接ノード情報に含まれるノードで、ノード i およびノード i の隣接ノード情報に含まれないノードをノード i のノード j を中継する 2 次近傍ノードとする。
- 同様に、 n 次近傍ノード x の隣接ノード情報に含まれて、 $(n-1)$ 次までの近傍ノード群の隣接ノード情報に含まれないノードをノード i のノード x を中継する $(n+1)$ 次近傍ノードとする。
- 上記処理を再帰的に繰り返し、ノード i の近傍トポロジを拡大し、ネットワーク全体を通してノード i の多次近傍ノードを設定する。

以上の処理をクラウド上で全てのノードに実施し、個々のノード毎に多次近傍ノードを設定し、これを仮想 WSN とする。

4.2.2 n 次近傍ノードによる位置推定

集約型 SOL はを用いてホップ数に応じてノード間距離が増加する。ノードを選択する。すなわち、選択されるノードは、ノード i から n 次近傍ノード以下であり、かつノード i からの $(n-1)$ 次近傍ノードのいずれのノードよりもノード i から遠方に位置する条件を満たす。従って、一定距離 $d \times$ ホップ数 n により位置更新を行う。多ホップによる n 次近傍修正ベクトルは式 (9) から求める。 $w_n(t)$ は n 次近傍ノードの仮自己位置である。さらに、位置修正の初期段階は広い範囲の近傍ノードを用いて大域的なトポロジを形成し、修正段階が進行すると位置修正に使用する近傍ノード情報を減少させ、局所的なトポロジを形成し、収束させる。従って、次式 (10) のように仮自己位置情報の更新を行う。 τ_N は推定の段階を分ける閾値である。

$$V_i^n(t) = \frac{nd - |w_i(t) - w_n(t)|}{|w_i(t) - w_n(t)|} (w_i(t) - w_n(t)) \quad (9)$$

$$w_i(t+1) = \begin{cases} w_i(t) + \alpha_i(t) \cdot (V_i^{\{1\}}(t) + V_i^{\{n\}}(t)) & (t < \tau_n) \\ w_i(t) + \alpha_i(t) \cdot (V_i^{\{1\}}(t) + V_i^{\{n-1\}}(t)) & (\tau_n < t < \tau_{n-1}) \\ \vdots & \\ w_i(t) + \alpha_i(t) \cdot (V_i^{\{1\}}(t) + V_i^{\{3\}}(t)) & (\tau_4 < t < \tau_3) \\ w_i(t) + \alpha_i(t) \cdot (V_i^{\{1\}}(t) + V_i^{\{2\}}(t)) & (otherwise) \end{cases} \quad (10)$$

4.2.3 推定トポロジの正誤判定

トポロジ矛盾判定は各ノードの 2 次近傍ノードまでにおいてトポロジ矛盾の有無を判定する。判定方式として、以下の 2 つを使用する。

- 距離によるトポロジ矛盾判定
- 角度 (方向) による判定

距離によるトポロジ判定は 3.4 節と同様であるが、この判定では、図 7 に示すような推定トポロジでは $(|w_i - w_l| \geq |w_i - w_j|)$ を満たすが、折れ曲がりによるトポロジ矛盾を検出できない。そのため、角度を用いた判定を行い、推定ノードの方向を判定する。

図 7 に示すような共通する 1 次近傍ノード j, k を持ち、かつノード i から 2 次近傍ノードに存在する l を選択する。その後、ベクトル V_{jk} から見たベクトル V_{ji} との角度 θ_1 とベクトル V_{jl} との角度 θ_2 を算出する。図 7 角度 θ_1 と角度 θ_2 の積が $\theta_1 \times \theta_2 \leq 0$ となる場合、トポロジーが正しく推定されていると判定を行う。

上記 2 つの方式で推定トポロジーの判定を行い、式 (11) を満たす場合、推定を終了する。式 (11) を満たさない (折れ曲がりトポロジ) 場合、ノード位置推定を再試行する。 B は距離を用いた判定を行った回数であり、 b は距離矛盾の発生回数である。 A は角度 (方向) を用いた判定を行った回数であり、 a は方向矛盾の発生回数である。 $\eta^{\{1\}}$ と $\eta^{\{2\}}$ は判定閾値である。

$$\frac{b}{B} \leq \eta^{\{1\}} \cap \frac{a}{A} \leq \eta^{\{2\}} \quad (11)$$

5. 評価

5.1 シミュレーション方法

シミュレーション諸元は表 1 に示す。障害物とフィールドについては図 8 に示す。障害物がある場合、ノード間通信は障害物によりのノード間が NLOS となる場合、通信不可とする。

5.2 通信回数

図 9 に、SOL と集約型 SOL のそれぞれの位置推定まで

表 1 シミュレーション諸元

フィールド範囲	1.0 × 1.0
障害物座標	(0.5, 0.35), (0.5, 0.75) (1.0, 0.75), (1.0, 0.35)
通信半径	0.2
アンカノード数	3
位置修正実行回数	300
評価トポロジ数	50

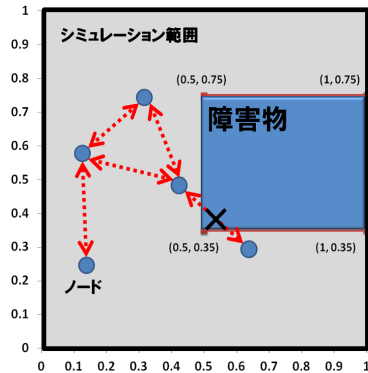


図 8 シミュレーションフィールドにおける障害物と NLOS

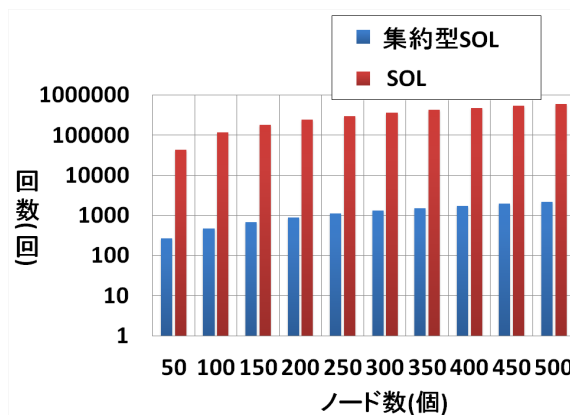


図 9 通信回数比較グラフ

の全ノード送信回数を示す。

図 9 から分かるように、集約型 SOL は大幅に送信回数を削減する。SOL は位置更新を各ノードで実施するため、ノード間で更新仮自己位置を頻繁に交換する。そのため通信量は大きくなる。一方、集約型 SOL は位置更新をクラウド環境の仮想 WSN で実施するため、各ノードは位置更新時の仮自己位置の交換は不要であり、仮想 WSN の構成のために自身の隣接ノード情報をクラウドサーバへ転送するだけである。

5.3 相対位置評価と絶対位置評価

以下の方式での比較評価を行う。

- 集約型 SOL
- DV-hop 方式
- SOL

評価は、相対位置評価と絶対位置評価の 2 通り行う。相対位置評価は推定されたノード位置により構成されるネットワークの形状 (推定ネットワーク形状) とオリジナルネットワークの形状の相似性を、次の式により評価する。

$$r_{ij} = \frac{\tilde{d}_{ij}}{d_{ij}} \quad (12)$$

$$\bar{r} = \frac{1}{|N|} \sum_{i=1}^{|N|} r_{ij} \quad (13)$$

$$V[r_{ij}] = \frac{1}{|N|C_2} \sum_{i=1}^{|N|-1} \sum_{j=i+1}^{|N|} (1 - r_{ij}/\bar{r})^2 \quad (14)$$

$\tilde{d}_{ij}$ は推定位置におけるノード i とノード j の距離 (推定距離), d_{ij} はオリジナルのネットワークにおけるノード i とノード j の距離, N は位置推定ノードの集合, $|N|$ はノード数 (集合 N の要素数) を示す。平均 $\bar{r}$ が 1 に近づけば推定ネットワーク形状はオリジナルのネットワーク形状と同じスケールとなり、分散 $V[r_{ij}]$ が 0 に近づけば推定ネットワーク形状はオリジナルのネットワーク形状と相似となる。すなわち、 $\bar{r}$ が 1, $V[r_{ij}]$ が 0 の場合、推定ネットワーク形状はオリジナルのネットワーク形状に一致する。

絶対位置評価は、推定された各ノードの位置と真位置のユークリッド距離の総和の平均である位置推定誤差 Err_{ave} を用いて評価する。 Err_{ave} は次の式 (15) のように求める。

$$Err_{ave} = \frac{1}{|N|} \sum_{i=1}^N |W_i - w_i| \quad (15)$$

W_i はノード i の真位置, w_i は推定位置を示す。

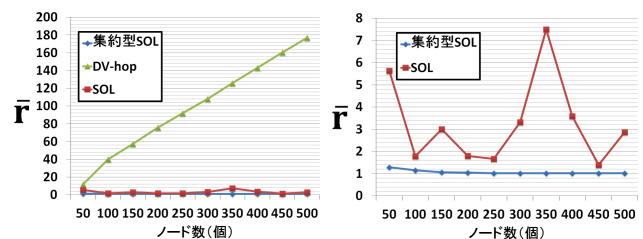


図 10 ノード数に応じた平均

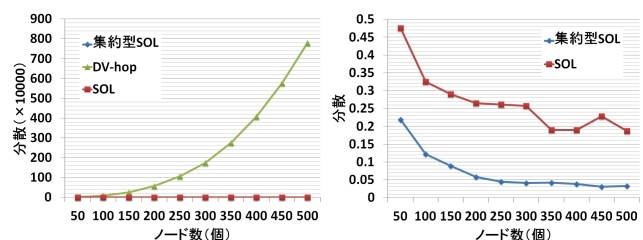


図 11 ノード数に応じた分散

図 12 から分かるように、SOL と比較して、集約型 SOL はその位置推定誤差が非常に小さく、位置推定精度が圧倒

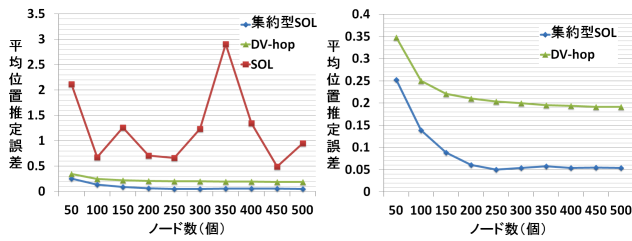


図 12 ノード数に応じた絶対位置評価

的に高い。SOL は少数のネットワークトポロジにおいて、部分的な形状は正しく推定されているが、折れ曲がりが発生し全体トポロジとして大きな誤差となる。数は少ないが誤差が非常に大きなケースがあり、これが全トポロジの平均誤差を大きく引き上げる。一方、集約型 SOL は完全に折れ曲がりトポロジを抑制し、安定して低い誤差を維持する。図 10 に示す集約型 SOL の $\bar{r}$ は SOL の $\bar{r}$ より低く、かつ 1 に極めて近い。図 11 の集約型 SOL の $V[r_{ij}]$ も SOL の $V[r_{ij}]$ より低く、かつ 0 に極めて近い。従って、集約型 SOL は推定ネットワーク形状においても優れ、その形状はオリジナルネットワーク形状にほぼ一致する (図 13)。一方、DV-Hop の $\bar{r}$ および $V[r_{ij}]$ のいずれも、SOL や集約型 SOL と比較すると、非常に大きい。従って、DV-Hop の推定ノード位置により構成されるネットワーク形状はオリジナルネットワークの形状と著しく異なる (図 14)。この原因は、NLOS 混在環境においては DV-Hop 方式では算出された 1 ホップの距離精度が低いいため、多角測量の最小二乗から解が得られず、その場合にはノード位置を 3 つのアンカノード重心として推定するためである、すなわち、NLOS 混在環境では DV-Hop はほぼ機能せず、図 12 の示す DV-Hop の位置推定誤差では現れない大きな誤差となる。以上のことから、集約型 SOL は NLOS 環境においてもネットワーク全体の高精度な形状再現と高精度なノード位置推定が維持可能であるといえる。

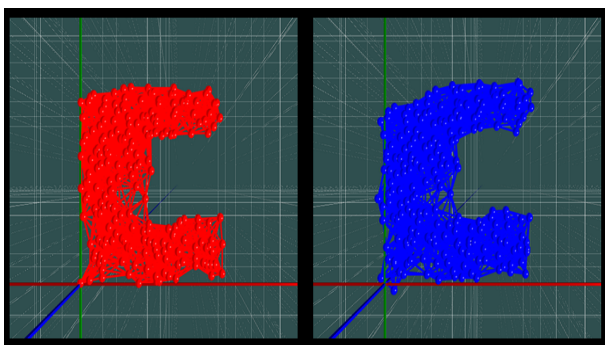


図 13 集約型 SOL での位置推定トポロジ (左:オリジナルトポロジ 右:推定トポロジ)

6. まとめ

本稿は、クラウドコンピューティングを前提とする集約

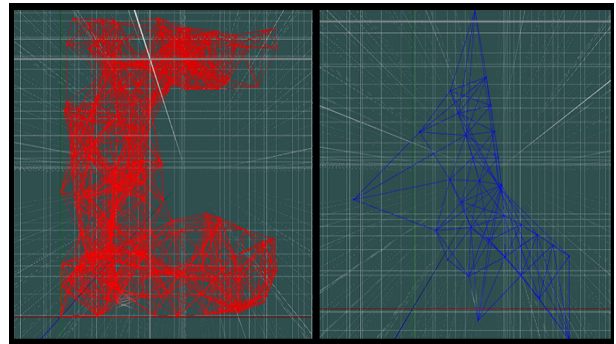


図 14 DV-hop での位置推定トポロジ (左:オリジナルトポロジ 右:推定トポロジ)

型 SOL を提案した、また、集約型 SOL の基本的な NLOS 環境のシミュレーション評価から次の優位性を確認した。

- NLOS 環境においても高い位置推定精度を維持可能である。
- 大幅に通信回数を削減可能である。

今後は多様な NLOS 環境で集約型 SOL の有効性の検証を進め、さらに集約型 SOL の 3 次元化を検討する。

参考文献

- [1] 大野, 安達, 滝沢: 無線センサネットワークにおける自己組織化位置推定方式の提案, 情報処理学会論文誌, Vol.53, No.7, pp.1774-1782, (2012).
- [2] A.Harter, A.Hopper, P.Steggles, A.Ward, and P.Webstar: The anatomy of a context-aware mobile applications, MOBICOM1999 (1999).
- [3] N.Priyantha, A.Miu, H.Balakrishman, and S.Teller: The cricket compass for context-aware mobile applications, MOBICOM2001 (2001).
- [4] N.Bulusu, J.Heidemmann, and D.Estrin: GPS-less low cost outdoor localization for very small devices, IEEE Personal Communications Magazine (2000).
- [5] D.Niculescu and B.Nath: DV-based positioning in ad hoc networks, Telecommun.Syst, vol.22,pp267-280 (2003).
- [6] D.Niculescu and B.Nath: DV-based positioning in ad hoc networks, Telecommun.Syst, vol.22,pp267-280 (2003).