

無線センサネットワークにおける 集約型自己組織化ノード位置推定方式の実装

北之馬 貴正¹ 高島 優斗¹ 安達 直世² 滝沢 泰久²

概要: Internet of Things や, Machine to Machine などの物理情報処理をインターネットに取り込む試みにおいて, 無線センサネットワークは必須技術であり, そのセンサノードの位置は重要な情報である. 我々は, センサノードの位置を近傍トポロジ情報のみから高精度に推定する自己組織化ノード位置推定方式 (SOL) を提案している. SOL は高い位置推定精度を有するが, ノード間通信量が多くなる問題がある. 本稿では, これを解決するための集約型 SOL の仮想トポロジを用いた実装手法およびその実機評価結果を示し, その有用性を議論する.

1. はじめに

無線センサネットワークは, Internet of Things や, Machine to Machine などの物理情報処理をインターネットに取り込む試みにおいて, 必須技術であり, そのセンサノードの位置は重要な情報である. これらを利用したサービスとしては, 広範囲に大量のセンシングデバイスを散布することを想定した, 環境モニタリングや気象観測, 災害対策等が考えられている.

現在, 利用もしくは研究されている無線ノード位置推定方式は, ノード間測距デバイスの使用の観点から分類すると, Range-Based と Range-Free に大別できる. 前者は測距デバイスを用いることにより高精度な位置推定が可能であるが, 特別なデバイスを必要とするためコストが高くなり無線センサノードに不適である. 後者は測距デバイスが不要であるが, 高精度な位置推定をするためには, 十分な数のアンカーノードにより構成された空間を必要とし, 任意の広範囲な空間に適用することは困難である.

上記問題を解決するため, 我々は自己組織化マップを用いたノード位置推定方式 [1] (SOL) を提案している. SOL では測距デバイスを用いず, 近傍トポロジ情報のみから, 高い位置推定が可能であり, その有効性が確認されている. SOL は分散型位置推定方式であり, 繰返し処理によるノード間通信が増大する問題がある. この問題を解消するために, クラウド環境に各センサノードの隣接ノード情報を集

約して構成した仮想無線センサネットワーク (仮想 WSN) に SOL を適用し (集約型 SOL), さらに仮想 WSN を細分化する仮想トポロジにより位置推定精度向上を図った. 本稿では, 集約型 SOL の仮想トポロジを用いた実装手法および実機評価結果を示し, その有用性を議論する.

2. 関連研究

現在, 利用もしくは研究されている位置推定方式は, ノード間測距デバイスの使用の観点から分類すると, Range-Based と Range-Free に分類できる.

2.1 Range-Based

Range-Based は位置推定処理にノード間の距離情報を利用するため, センサノードにノード間通信機能の他に測距デバイスを持たせる必要がある方式である. ノード間距離の測距には, TDOA (Time Difference Of Arrival), TOA (Time Of Arrival) が利用されている. TOA 方式は, 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, 伝送媒体の伝送速度からノード間の距離を計算する方式である. TOA 方式を利用した位置推定方式として最も一般的なものは GPS である. TDOA 方式は, 異なる 2 つの伝送媒体を用いて通信を行い, それらの到着時間の差からノード間の距離を計算する方式である. TDOA 方式を利用した位置推定方式としては, Active Bat [2] や Cricket [3] がある.

Range-Based はこれらの測距デバイスで得られた情報を使用し, 三辺測量を用いて位置推定を行う. しかし, TOA 方式や TDOA 方式を用いた位置推定技術は精度が高いが, 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,

¹ 関西大学 理工学研究科
Graduate School of Engineering, Kansai University

² 関西大学 環境都市工学部
Faculty of Environmental and Urban Engineering, Kansai University

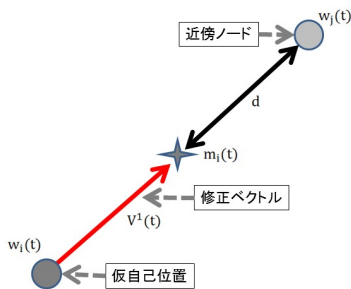


図 1 1次近傍ノードによる位置修正

センサノードの消費電力やコストの面において WSN での利用は不向きであると考えられる。

2.2 Range-Free

RangeFree 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。Centroid 方式 [4], APIT 方式 [5] や DV-Hop 方式 [6] 等がある。

Centroid 方式は、通信可能な複数のアンカーノードの位置情報を位置推定を行うノードが取得し、それらの重心を利用して自身の位置を推定する方式である。APIT 方式は、複数個のアンカーノードの組み合わせから作成可能な全ての三角形に対して、位置推定を行うノードが外側にあるか内側にあるかを判定することで自身の位置を推定する方式である。DV-Hop 方式は、3つ以上のアンカーノードからのホップ数とアンカーノード間の距離から算出された1ホップの平均距離を利用して、アンカーノードとの距離を見積もることにより自身の位置を推定する方式である。

これらの方式は少なくとも3つ以上のアンカーノードが必要であり、精度の向上には多量なアンカーノードが必要のため広範囲な空間への適用には十分な事前準備が必要である。そのため、適用可能な環境は限定的となる。

3. SOL(Self-Organizing Localization)

SOL は、ごく少数のアンカーノードと各ノードの近傍ノード情報のみからノードの位置を推定する方式である。SOL は、次の特徴を有する。

- アンカーノードへの依存性が極めて低く、アンカーノードなしで相対位置を、3点で絶対位置を推定可能
- 測距デバイスなどの特別なデバイスを必要としない

3.1 SOL 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] 自己位置をランダムに生成する。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ブロードキャスト送信する。

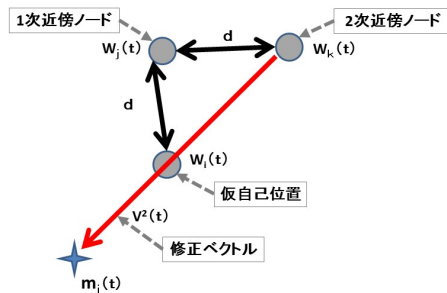


図 2 2次近傍ノードによる位置修正

[Step.2] 1次近傍ノード j から仮位置情報 $w_j(t)$ を受信したノードはその仮位置情報と測距デバイスを用いないため相対的なノード間距離として使用するホップ数 $d = 1$ により推定されるノード i の位置を入力ベクトル $m_i(t)$ とする。その後、ノード i の仮自己位置と入力ベクトル $m_i(t)$ の距離 $|m_i(t) - w_i(t)|$ が最小となるような修正ベクトル $V_i^{\{1\}}(t)$ を生成することにより、ノード i の仮自己位置を入力ベクトル $m_i(t)$ に近づける (図 1)。

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

また、修正処理の初期段階 (繰り返し回数が少ない) では、近傍ノード j の近傍ノード集合のうち、ノード i から2ホップにあたるノード (以降、2次近傍ノード) k の仮位置とホップ数 $2d = 2$ により推定される位置を入力ベクトルとする。ここでの入力ベクトルは、ノード i と2次近傍ノード k とのホップ数 $2d$ として生成する。その後、ノード i の仮位置をこの入力ベクトルに近づけるため、次のような修正ベクトルを生成する (図 2)。

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

ノードの仮自己位置が1次近傍ノードより2次近傍ノードの近くに位置する場合 (以後、トポロジの矛盾)、すなわち、 $|w_i(t) - w_j(t)| \geq |w_i(t) - w_k(t)|$ の場合、ネットワークトポロジの形成が誤っていると判断し、 $V_i^{\{1\}}(t)$ と $V_i^{\{2\}}(t)$ の両方により位置修正を行う。トポロジ矛盾がない場合は $V_i^{\{1\}}(t)$ のみで位置修正を行う。以上を定式化すると次のようになる。

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

$\alpha_i(t)$ 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。

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

ただし、 η は正の減衰定数である。

[Step.3] 前回の近傍ノードへの仮位置情報配信から一定時間経過後、修正された仮自己位置と自身の任意の1次近傍ノードの仮自己位置を1次近傍ノードへブロードキャスト送信する。この情報を得たノードがStep.2を実施する。

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ではホップ数をノード間の距離情報として使用しているが、これは相対的なものであり、かつ解像度が低いためネットワークトポロジが形成されていた場合においてもトポロジの矛盾を発生させる原因となりうる。従って、位置推定精度の向上と相対位置から絶対位置への変換のため、次の3つの処理を行う。

- 推定トポロジの矛盾判定とノード間距離の変更
- 推定再試行の実施
- 推定トポロジを絶対座標へ変換

3.2.1 推定トポロジの矛盾判定とノード間距離の変更

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。また上記のトポロジの矛盾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、トポロジの矛盾の判定を行っているノード*i*における矛盾の発生回数、 $N_i^{(2)}$ は、ノード*i*において矛盾の判定を実施した2次近傍ノード数、 θ は、誤再現判定閾値である。

トポロジの矛盾が発生する原因は、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ためと推定できる。従って、トポロジの矛盾と判定された場合は、ホップ数1(正の整数値)を減衰して、これをホップ度(1より小さい正の実数値)として、これによりトポロジ矛盾を解消して位置推定を行う(図3)。ノード間ホップ度は式(6)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。

$$d_{T+1} = \frac{d_T}{T+1} \quad (6)$$

3.2.2 推定再試行の実施

トポロジの矛盾が発生していると判断された場合は以下の処理を行う。

- 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ノードは、自身の学習係数 $\alpha_i(t)$ を初期値の1に戻し、位置推定処理を再試行する。

3.2.3 推定トポロジを絶対座標への変換

3点のアンカーノードの真位置 r_i と3点のアンカーノードの推定位置 p_i を使用し、アフィン変換行列 A を求める。

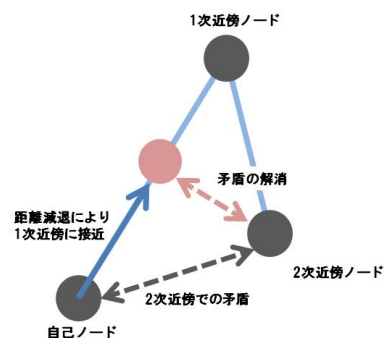


図3 ノード間トポロジ矛盾を解消するノード間ホップ度

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(7)によって求められる。

$$\begin{pmatrix} r_{ix} \\ r_{iy} \end{pmatrix} = \begin{pmatrix} a & b & c \\ d & e & f \end{pmatrix} \begin{pmatrix} p_{ix} \\ p_{iy} \\ 1 \end{pmatrix} \quad (7)$$

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

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分散型位置推定方式である。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。

- 各ノードは2次近傍ノードまでの不完全なトポロジ情報から位置推定するため、誤差が発生しやすい。
- 繰り返し処理によりノード間通信トラフィックが増大する。

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の実装を行う。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し、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取得し、これをクラウド環境へ転送する。
- クラウド環境で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ある。集約型SOLは、この完全なトポロジ情報を持つWSNにSOLを適用する。これにより、下記の効果が見込まれる。

- 完全なトポロジ情報を用いた*n*次近傍ノードとの位置更新が可能であり、高精度化できる。
- センサノードは隣接ノー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 仮想 WSN 構成方法

各センサノードは次の処理を行う。

- 自身のノード 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。
- 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シンクノードに転送する。
- シンクノード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のノードの隣接ノード情報を取得し、保持する。クラウド環境の SOL は上記の取得された隣接ノード情報を用いて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 j 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 x を中継する $(n+1)$ 次近傍ノード i 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。

4.3 n 次近傍ノードを利用した位置推定

集約型 SOL では、仮想 WSN を構成することから、完全なトポロジ情報を用いて n 次近傍ノードで、かつ他のノードの $(n-1)$ 次以下の近傍ノードではないノードを選択する。このノードはすべての $(n-1)$ 次近傍より遠方に位置するノードある。従って、 n 次近傍ノードとの距離を $n \times d$ として式 (8) 示すように n 次近傍ノードによる修正ベクトルを生成する。以上を定式化すると式 (9) のようになる。

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

$$w_i(t+1) = \begin{cases} w_i(t) + \alpha_i(t) \cdot (V_i^{\{1\}}(t) + V_i^{\{N\}}(t)) & (t < \tau_N) \\ w_i(t) + \alpha_i(t) \cdot (V_i^{\{1\}}(t) + V_i^{\{N-1\}}(t)) & (\tau_N < t < \tau_{N-1}) \\ \vdots & \\ w_i(t) + \alpha_i(t) \cdot (V_i^{\{1\}}(t) + V_i^{\{3\}}(t)) & (\tau_4 < t < \tau_3) \\ w_i(t) + \alpha_i(t) \cdot (V_i^{\{1\}}(t) + V_i^{\{2\}}(t)) & (otherwise) \end{cases} \quad (9)$$

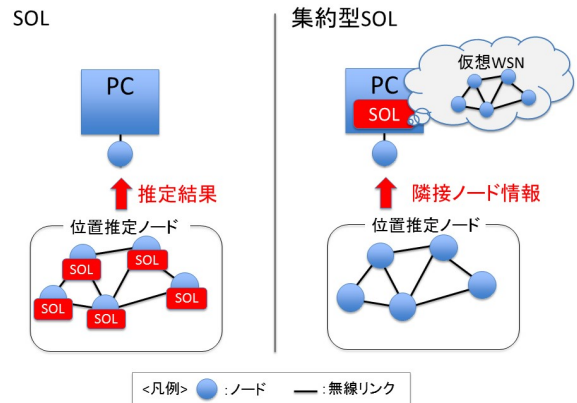


図 4 SOL と集約型 SOL

5. 仮想トポロジを用いた集約型 SOL

既存方式は分散型であるため、各位置推定ノードにおいて SOL が実行される (図 4 左)。一方、集約型 SOL は、クラウド環境にすべてのセンサノードの隣接ノード情報を集約し、これにより構成される仮想 WSN において SOL が実行される (図 4 右)。従って、集約型 SOL は

- クラウド環境の仮想 WSN へ適用した SOL
- 各センサノードにおける隣接ノード情報の取得とこの情報のクラウド環境への転送

から構成される。この構成に基づき、仮想トポロジを用いて高精度化を図る集約型 SOL を次のように実装する。

- クラウド環境において、IEEE802.15.4 でサポートしている Link Quality Indicator(LQI) により仮想 WSN を細分化する仮想トポロジ
- WSN において、シンクノード制御によるセンサノードの隣接ノード取得と上記 LQI の取得およびそれらのシンクノードへの転送

5.1 集約型 SOL における仮想トポロジ

WSN の構成は必ずしも多ホップとは限らない。SOL はホップ数を近傍ノードとの相対関係としているため、近傍ノードとのホップ数が少ない WSN では近傍ノードの相対関係が不十分となり、位置推定精度劣化が起こる。この問題を解決するため、仮想トポロジを実装する。

仮想トポロジは WSN の無線リンク品質に用いてその無線リンク長を仮想的に短くし、実際の 1 次近傍ノード群を複数ホップノード群として扱う。これにより、近傍ノードの相対関係がより細分化された仮想 WSN を構成する (図 5)。この仮想トポロジによる仮想 WSN へ SOL を適用し、SOL の位置推定精度の向上を図る。

仮想トポロジ実装において、無線リンク品質は、WSN を ZigBee 準拠と想定し、ZigBee の MAC レイヤ仕様の IEEE802.15.4 でサポートしている LQI を用いる。LQI を用いるため、距離おける LQI 値の依存性を予備実験した。

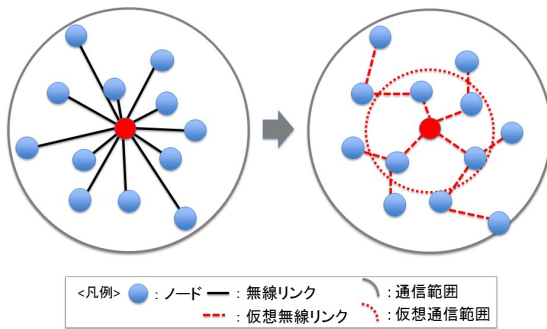


図 5 仮想トポロジ構成例

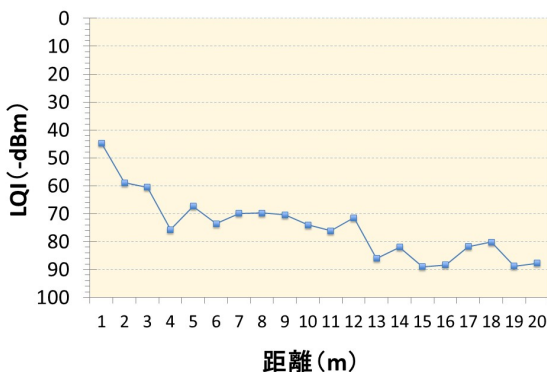


図 6 距離と LQI の相関関係

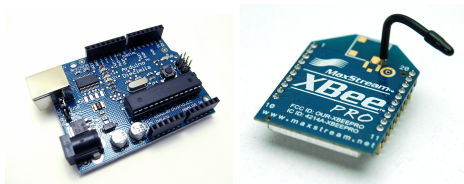


図 7 Arduino と Xbee

図 6 に示すように、LQI は絶対距離を求めるには適さない。しかし、近傍の範囲として用いることは可能と考える。すなわち、LQI から 1 次近傍ノードをフィルタリングして、無線リンク長を短く扱う。

仮想トポロジを構成する場合、LQI によるフィルタリングは高い値（仮想リンク長が相当短い）から開始し適正な近傍ノード数となる LQI を用いる。

5.2 センサノードにおける隣接ノード情報取得と転送

5.2.1 センサノード構成

無線センサノードのハードウェアは CPU、無線デバイス、バッテリー、センサにより構成されるが、今回は位置推定の検証のためセンサは搭載しない。CPU として Arduino、無線デバイスとして XBee を使用する (図 7)。Arduino はハードウェア・ソフトウェアともに容易に開発できる柔軟性のあるオープンソース電子工作用プラットフォームである。XBee はセンサおよび計測制御ワイヤレス・ネットワークのための国際的な業界標準化技術で、ZigBee を搭載しており Arduino と接続するための I/F も用意されている。

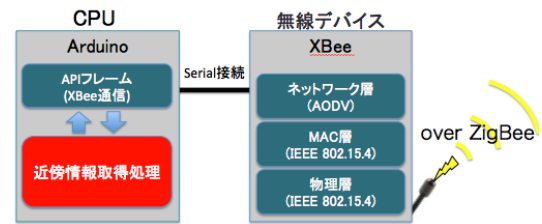


図 8 ソフトウェア構成

図 8 に無線ノードにおけるソフトウェア構成を示す。CPU では送受信におけるデータの格納、取得の処理と、隣接ノード情報取得処理を行う。XBee では ZigBee の MAC 層仕様に従い、IEEE802.15.4 により無線通信を行う。他ノードとのデータのやりとりや、LQI を取得する際の物理層とのやりとりは XBee で定義されている API(Application Programming Interface) を用いる。

5.2.2 隣接ノード情報取得と転送

各センサノードは自身の隣接ノードへの広告として自身の ID をデータとするブロードキャストを行う。この広告ブロードキャストにより各ノードは隣接ノード ID を取得する。同時に、隣接ノードのフィルタリングを行うため、XBee の物理層から提供される LQI も取得する。この広告ブロードキャスト送信を各ノードが任意に行くと広告ブロードキャストの競合が発生し、ID 取得および正確な LQI 取得が困難となる。従って、シンクノードからセンサノードの広告ブロードキャストをスケジューリングし、逐次送信をセンサノードへ指示する。この制御により、センサノード間で広告ブロードキャストの競合を排除し、確実な隣接ノード ID 取得と正確な LQI 取得を行う。この制御により各センサノードは取得した隣接ノード ID と LQI を隣接ノード情報としてシンクノードへ転送する。

6. 評価

6.1 評価方式

実験の評価は、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。

$$r_{ij} = \frac{\tilde{d}_{ij}}{d_{ij}} \quad (10)$$

$$\bar{r} = \frac{1}{|N|} \sum_{i=1}^{|N|} r_{ij} \quad (11)$$

$$V[r_{ij}] = \frac{1}{|N|C_2} \sum_{i=1}^{|N|-1} \sum_{j=i+1}^{|N|} (1 - r_{ij}/\bar{r})^2 \quad (12)$$

\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



図 9 実験風景

とノード j の真値距離, 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 (13) のように求める. W_i はノード i の真位置, w_i は推定位置を示す.

$$Err_{ave} = \frac{1}{|N|} \sum_{i=1}^{|N|} |W_i - w_i(t)| \quad (13)$$

評価実験は 図 9 のように屋内と屋外の 2 か所で行った. ネットワーク空間を $5.0\text{m} \times 5.0\text{m}$ の平面として定義し, この空間にノードをランダムに配置する. また, 比較対象として, アンカーノード 3 点で絶対位置が推定でき, かつ既存方式で最も精度が高いとされている DV-Hop を用いる, 屋外の実験で得た隣接ノード情報を DV-Hop に与えることで位置を推定し比較評価する. 実験では, PC と XBee で構成されたシンクノードを擬似的にクラウド環境として用いた.

6.2 評価結果

比較対象である DV-Hop は最小二乗法が収束せず, 多数のノードがアンカーノード 3 点の重心に推定される. 平均誤差はノード数に関わらず, 約 1.9m (図 12) であるが, \bar{r} はノード数が増加するに従い, 1 を大きく上回り (図 10 左), $V[r_{ij}]$ は 0 を大きく上回る (図 11 左). すなわち, DV-Hop の推定ネットワーク形状は全くオリジナルネットワーク形状と異なり, 平均誤差が示した値よりネットワーク全体としての位置関係は精度が著しく低いことが分かる.

一方, 集約型 SOL は屋内と屋外のいずれの環境においても, ノード数が増えるに従い, \bar{r} は 1 に近づき (図 10), $V[r_{ij}]$ は 0 へ近づく (図 11). すなわち, 集約型 SOL はオリジナルネットワーク形状とほぼ同じとなるネットワーク形状を推定している. さらに, 平均誤差もノード数の増加に従い, 小さくなり (図 12), 50 ノードでは 50cm 以下となる.

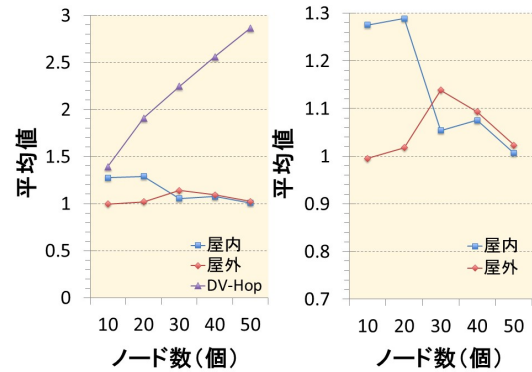


図 10 スケール評価

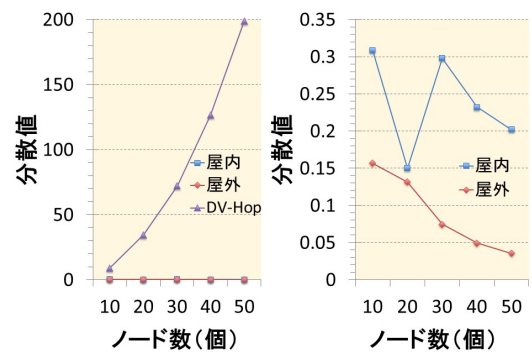


図 11 相対評価

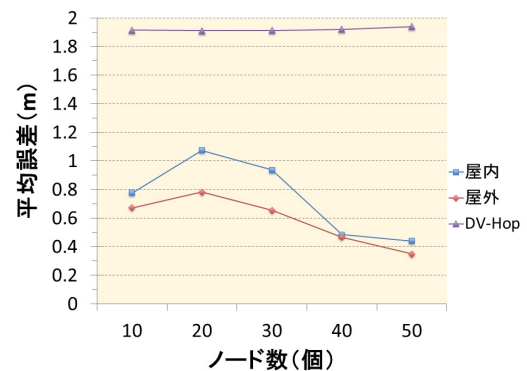


図 12 絶対評価

集約型 SOL の屋内と屋外との環境の違いを比較すると, 屋内においてはノード数が少ない場合, \bar{r} , $V[r_{ij}]$, および平均誤差のいずれも屋外に劣る. これは屋内環境はマルチパスフェージングの影響により LQI が変動するためと考えられる. しかし, ノード数の増加に従い, 屋内の各評価値も改善し, 屋外の評価値に近づく. すなわち, 集約型 SOL はノード数の増加に従い, 電波環境の影響を排除できると考えられる.

7. まとめ

本稿では, 仮想トポロジを用いた集約型 SOL の実装手

法を示し、さらに実機による実験表から次の有用性を確認した。

- ノード数の増加に従い、ノード位置を高い精度で推定できる。
- ノード数に従い、電波環境の影響を排除できる。

今後は、LQIに関するフィルタリングをより電波環境に非依存とする手法を検討し、さらなる集約型 OSL の位置推定精度の向上を図る予定である。

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.Heidemann, and D.Estrin : GPS-less low cost outdoor localization for very small devices, IEEE Personal Communications Magazine (2000).
- [5] Tian He, Chengdu Huang, Brian M.Blum, John A.Stankovic, and Tarek F.Abdelzaher : Range-free localization and its impact on large scale sensor networks, ACM Transactions on Embedded Computing Systems (TECS), v.4 n.4, p.877-906 (2005).
- [6] D.Niculescu and B.Nath : DV-based positioning in ad hoc networks, Telecommun.Syst, vol.22,pp267-280 (2003).