Vol.2011-MBL-57 No.22
Vol.2011-UBI-29 No.22
2011/3/8

隣接ノード集合と残存電力量を考慮したセンサネットワーク・クラスタリング方式の提案

豊田慎之介† 佐藤文明†

現在，無線技術を応用した無線センサネットワークの研究が盛んに行われている．様々な分野に応用できると言われているが，センサの電源の多くは電池であり，外部から電源を供給することも困難な場合も多く，電池の交換にもコストがかかる．よって，特に大規模なセンサネットワークにおいて，省電力を考慮したルーティング方式は重要である．本論文では，クラスタヘッドを選ぶ際に隣接ノード集合の重複度と残存電力を評価することで，ノードの残存電力の偏りを抑えたクラスタリング方式を提案する．また，シミュレーションにより従来方式との比較を行い，提案方式の有効性を検証する．

Energy-Effective Clustering Algorithm Considering Adjacent Nodes and Remaining Electric Power for Wireless Sensor Networks

Shinnosuke Toyoda† and Fumiaki Sato†

Nowadays many researchers are researching wireless sensor networks (WSN). WSN can be applied to the various fields. But, many sensors work by a battery and, it is difficult to supply energy from the outside in most case. Besides, to exchange the battery causes a cost. Therefore, in a particularly large-scale WSN, the routing protocol that considered energy-saving is important. In this paper, we propose energy-effective clustering algorithm considering adjacent nodes and remaining electric power. In addition, we inspect effectiveness of our method by comparing our method with the traditional method by the simulation.

1. はじめに

近年，センサ端末の小型化・低価格化が進み，無線センサネットワークが注目されている．無線センサネットワークは搭載するセンサの種類や，ネットワークの規模などにより幅広い応用が可能で，様々な分野から期待されている．

一方で，センサ端末の電源は主に電池に依存するため，有限である．さらに，外部から電源を供給することも困難な場合が多く，電池を交換するのにもコストがかかってしまう．このため，特に大規模な無線センサネットワークの運用には，省電力に考慮したルーティング方式は重要であり，盛んに研究が行われている．センサネットワークに適した代表的なルーティング方式に LEACH[1]がある．LEACH はクラスタリングを用いたルーティング方式である．LEACH の主な特徴として，クラスタメンバからクラスタヘッド，クラスタヘッドからシンクまでがそれぞれ 1 ホップで構成されることと，負担のかかるクラスタヘッドを一定期間ごとに変更することが挙げられる．しかし，LEACH はノードの電力消費が偏りやすく，シンクと直接通信できないノードが存在するネットワークには適応できないという課題がある．そこで本論文では，クラスタ間マルチホップによってシンクと通信できないノードが存在するネットワークでも適応可能であり，隣接ノード集合の重複度と残存電力の情報を用いることで，残存電力の均等化とさらなる省電力化を目指した．

2. 従来研究

2.1 LEACH

センサネットワークに適した代表的なルーティングプロトコルとして LEACH[1]が挙げられる．LEACH(Low-Energy Adaptive Clustering Hierarchy)はクラスタリングを用いたルーティング方式で，ノードからクラスタヘッド，クラスタヘッドからシンクまで，それぞれ 1 ホップでデータを送信する．

LEACH の最大の特徴は，負荷の集中するクラスタヘッドを一定期間ごとに交代することである．これにより，負荷を分散させ，ネットワーク全体の寿命をのばすことができる．ここで，各ノードからシンクへデータを 1 回送信する周期をサイクル，クラスタヘッドを交代してから次のクラスタヘッドを交代するまでのサイクル周期をラウンドと呼ぶ．

LEACH では，前提として，すべてのノードがシンクと通信可能であり，各ノードが 1 ラウンド目にクラスタヘッドに立候補する基本確率 P を予め知っている必要があ

† 東邦大学理学部情報科学科
Department of Information Science, Faculty of Science, Toho University

Vol.2011-MBL-57 No.22
Vol.2011-UBI-29 No.22
2011/3/8

る．ラウンドが変わると，ノード n は自身がクラスタヘッドに立候補するかどうかを判定する．このとき，0～1 の乱数を作り，以下の式(2.1)の計算結果がその数以下であった場合， n はクラスタヘッドに立候補する．

$$T(n)=\begin{cases}\dfrac{P}{1-P*(r \bmod \frac{1}{P})} & \text{if} \quad n \in G \\ 0 & \text{otherwise}\end{cases} \qquad (2.1)$$

ここで，r は現在のラウンド数，G は過去 ラウンド以内にクラスタヘッドになっていないノードの集合を表す．つまり，すべてのノードは 1 / P ラウンド以内にクラスタヘッドに立候補することになる．クラスタヘッドに立候補したノードは自分の周囲のノードにクラスタヘッド広告をブロードキャストし，クラスタヘッドに立候補したことを知らせる．クラスタヘッドに立候補しなかったノードは，一定時間クラスタヘッド広告の受信待ちを行う．クラスタヘッド広告を受け取ったノードはクラスタヘッド広告を受信したノードのリストに送信してきたノードを加え，受信電波強度を記憶する．受信待ちを終えると，非クラスタヘッドノードは，リストの中から受信電波強度の最も強かったクラスタヘッドを選び，参加要求を送信する．クラスタヘッド広告を受信したノードが 1 つもなかった場合は，クラスタには所属せずに，直接シンクへデータを送信する．一方，クラスタヘッド広告を送信し終えたノードは，参加要求待ちを行う．すべての参加要求を受信したら，クラスタヘッドはクラスタメンバの TDMA 送信スケジュールを作成し，メンバに送信する．メンバは，送信スケジュールを受け取ると，クラスタヘッドが交代するまで自分の送信順を記憶しておく．ここまでの処理をラウンドが変わるごとに実行する．

スケジュールが決まったあとは，メンバはスケジュール通りの順番にセンサデータをクラスタヘッドに送信し，クラスタヘッドはすべてのメンバからデータを受け取ってからシンクへ圧縮したデータを送信する．これが LEACH における 1 ラウンド 1 サイクルの流れである．

LEACH には問題点もいくつか存在する．第一に，ノードの電力消費が偏りやすいという問題がある．これは，ノードがクラスタヘッドに立候補するかどうかを単純に確率とクラスタヘッドになった回数で判定しているため，最終的にどのノードも同じ頻度でクラスタヘッドになっていることが原因である．データ送信時の消費電力は送信距離に依存するため，シンクから離れたノードと近くにあるノードで同じ回数送信を行うと，離れたノードの電力のほうが早く枯渇してしまうのである．また，クラスタメンバを一切持たないクラスタヘッドが出現したり，どのノードもクラスタヘッドに立候補しないラウンドが存在したりするなど無駄が多いことも原因であると考えられる．第二に，LEACH はマルチホップをサポートしていない点である．すべてのノードがクラスタヘッドになれる必要があるため，シンクと直接通信できないノードが存在するネットワークでは，そのようなノードがクラスタヘッドになったときにデータをシンクへ転送することができなくなってしまう孤立したクラスタが生まれてしまう．このため，LEACH を適用することができない．

2.2 HEED

LEACH を改良したクラスタリング方式も提案されている．その一つに HEED(Hybrid, Energy-Efficient Distributed clustering)[2]がある．HEED では，各ノードがクラスタヘッドに立候補する確率を，初期電力 E_{max} とその時点の残存電力 $E_{residual}$ の比とすることで，より残存電力の大きいノードがクラスタヘッドに立候補しやすくしている．また，HEED ではクラスタヘッドの状態に，暫定クラスタヘッド状態を示す *tentative_CH* とクラスタヘッド決定状態を示す *final_CH* の二種類を設定している．*final_CH* を知らせるメッセージをブロードキャストしたノードはそのラウンドでクラスタヘッドを担うことが決定づけられる．一方，*tentative_CH* をブロードキャストしたノードは，後から自分よりも全体の通信コストを低くするノードがクラスタヘッド立候補をブロードキャストしてきた場合に，自身の立候補を破棄して他のクラスタへ所属することができるのである．

HEED では，ノードがクラスタヘッドに立候補する確率 CH_{prob} を以下の式(2.2)で計算する．

$$CH_{prob}=\max\left(C_{prob}*\frac{E_{residual}}{E_{\max}}, p_{\min}\right) \qquad (2.2)$$

ここで，C_{prob} は予め定められたクラスタヘッドの割合，$E_{residual}$ をノードの残存電力，E_{max} をノードの初期電力，p_{min} は E_{max} に反比例して決められる CH_{prob} の最小値をそれぞれ表している．

Vol.2011-MBL-57 No.22
Vol.2011-UBI-29 No.22
2011/3/8

各ノードは式(2.2)で CH_{prob} を計算した後に，以下を繰り返す．

- 自身のものを含め，クラスタヘッド立候補を 1 つ以上受信している場合
  - 通信コストが最小になるノードを自身のクラスタヘッドに選ぶ
  - 選んだノードが自分自身であった場合
    - CH_{prob} が 1 のとき
    - *final_CH* メッセージをブロードキャストする
  - CH_{prob} が 1 未満のとき
    - *tentative_CH* メッセージをブロードキャストする
- クラスヘッド立候補メッセージを 1 つも受信していない場合
  - CH_{prob} が 1 のとき
    - *final_CH* メッセージをブロードキャストする
  - CH_{prob} が 1 未満のとき
    - 0～1 の乱数 ≦ CH_{prob} のとき
      - *tentative_CH* メッセージをブロードキャストする
- CH_{prob} が 1 のとき
  - 繰り返しを終了する
- CH_{prob} が 1 未満のとき
  - $CH_{prob} \leftarrow min(CH_{prob} * 2, 1)$
  - 最初から繰り返す

繰り返しの中でノードが *final_CH* をブロードキャストしなかった場合，クラスタヘッド立候補メッセージを受信したノードから通信コストが最小のノードを自身のクラスタヘッドに選び，参加要求を送信する．また，クラスタヘッド立候補メッセージを受信したノードがなかった場合，*final_CH* メッセージをブロードキャストし，クラスタヘッドとなる．

2.3 その他の従来研究

この他にも，マルチホップ通信を用いることで電波干渉を抑え，できるだけ多くの通信を同時並行的に行うプロトコル HIT[3]や，クラスタではなく最も近いノード同士を結ぶチェインを構成する PEGASIS[4]，クラスタ内でこのチェインを構成する TPC[5]など，センサネットワークに適した様々な省電力ルーティングプロトコルが提案されている．

3. 隣接ノード集合と残存電力を考慮したセンサネットワーク・クラスタリング方式

3.1 概要

本論文では，隣接ノード集合と残存電力を考慮したクラスタリング方式を提案する．提案方式では，シンクを含む全てのノードが，自身の残存電力と隣接ノード集合の情報を含んだ Hello メッセージをラウンドごとに数回交換する．これにより，各ノードは自身に直接隣接するノードに関して，その隣接ノード集合と，残存電力の情報を保持することができる．クラスタヘッドの選出は，まずシンクが全ての隣接ノードについて，残存電力と隣接ノード集合から評価値を計算し，評価値をもとに次のクラスタヘッドを選出し，通知する．クラスタヘッドに選ばれたノードも同様にして次のクラスタヘッドを選出する．これにより，クラスタヘッドの出現位置の偏りをなくし，ノード全体の電力消費の均等化を図る．センサデータの収集は，クラスタヘッド間でマルチホップを行うことで，シンクと直接通信できないノードが存在するネットワークでもデータを収集することが可能になる．

3.2 クラスタヘッドの選出

ここでは，1 ラウンドにおけるクラスタヘッドの選出を終えるまでの流れを解説する．なお，クラスタヘッドの選出方法は，高密度アドホックネットワークにおける中継用ランドマークノードの選出方法[3]を参考にしている．

(1) 隣接ノード情報の交換

新しいラウンドが開始すると，シンクノード *s* は自身の残存電力と隣接ノード集合 nbr(*s*)の情報を含んだ Hello メッセージを周囲のノードにブロードキャストする．なお，このとき Hello メッセージに含める nbr(*s*)の情報は接続情報のみで，隣接ノードの残存電力や隣接ノードの情報は含めない．ただし，シンクが最初にこの Hello メッセージを送信するとき，隣接ノード集合は空である．これを受け取ったノード *n* は，隣接ノード集合 nbr(*n*)に *s* の残存電力と nbs(*s*)の情報を加える．また，nbr(*s*)に含まれるノードうち，nbr(*n*)に含まれていないノード，言い換えると，1 ホップでは到達できないが 2 ホップで到達可能なノードを 2 ホップリスト twoNbr(*n*)に追加する．そして，ノード *n* はシンクと同様に，自身の残存電力と nbr(*n*)の情報を含んだ Hello メッセージを周囲のノードにブロードキャストする．さらにこれを受け取った別のノード *n2* は，先のノード *n* と同様に，nbr(*n2*)と twoNbr(*n2*)を更新する．ただし，*n* の隣接ノード情報に *n2* が含まれていた場合，これを除外して更新を行う．そして，この Hello メッセージを受け取る以前に，他のノードからの Hello メッセージを受け取っていなければ，ノード *n2* も Hello メッセージをブロードキャストする．このようにして，ネットワーク全

Vol.2011-MBL-57 No.22
Vol.2011-UBI-29 No.22
2011/3/8

体でノード情報の交換が行われる．また，1 度目の Hello メッセージをブロードキャストしたシンクノード *s* は，一定時間後に 2 度目の Hello メッセージをブロードキャストする．なお，ここでの一定時間は，周囲の全ノードからの Hello メッセージを受信するのに十分な時間である．他のノードは，1 度目の Hello メッセージを最初に受け取ったノードから 2 度目の Hello メッセージを受信すると，2 度目の Hello メッセージをブロードキャストする．これによって，全てのノードの twoNbr が正しく更新される．

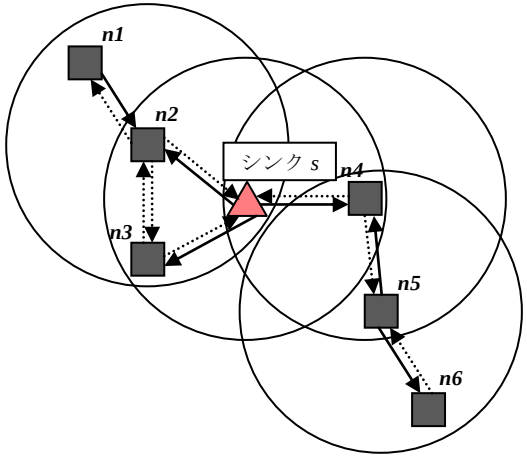

図 3.1 Hello メッセージの交換

図 3.1 は Hello メッセージのやりとりの様子を示している．まず，シンクが自身の残存電力の情報を含めた Hello メッセージを周囲のノードにブロードキャストする．ノード *n2* はこれを受け取ると，隣接ノード集合 nbr(*n2*)にシンクと，シンクの残存電力を記憶する．その後，自身の残存電力と，nbr(*n2*)に含まれるノードの情報（この場合はシンクのみ）を含んだ Hello メッセージをブロードキャストする．ただし，シンクの情報は隣接情報のみで，残存電力や，シンクの隣接ノードなどは含まれない．また，ノード *n3*, *n4* も同様の処理を行う．

次に，ノード *n2* の Hello メッセージを受信するノードはノード *n1*, *n3* とシンクである．*n1* は nbr(n1)に *n2* を加え，*n2* の残存電力と隣接ノードも記憶する．また，twoNbr(*n1*)にシンクを加える．その後，自身の残存電力と隣接ノード情報（*n2* のみ）を Hello メッセージに含め，ブロードキャストする．*n3* も同様に，nbr(*n3*)に *n2* を追加するが，*n2* の隣接ノードであるシンクは *n3* から 1 ホップで到達可能なため，twoNbr(*n3*)は更新されない．また，先にシンクからの Hello メッセージを受けているため，Hello メッセージのブロードキャストも行わない．一方，シンクは nbr(*s*)に *n2* を加えるが，*n2* の隣接ノード情報にはシンクが含まれているため，これを除外する．よってシンクが保持する *n2* の隣接ノードは空のままとなる．また，シンクも *n3* 同様，Hello メッセージのブロードキャストは行わない．

他のノードも同様にして Hello メッセージの交換を行い，2 回目の Hello メッセージも同様に処理される．最終的にシンクの隣接ノード集合 nbr(*s*)は，*n2*, *n3*, *n4* となり，twoNbr(*s*)は，*n1*, *n5* となる．

(2) シンクのクラスタヘッドの選出

シンク *s* は 2 度目の Hello メッセージを送信し終えてから一定時間後，クラスタヘッドの選出を始める．*s* は隣接ノード集合 nbr(*s*)に含まれるすべてのノードに対して，評価値 *v* を計算する．ここで，ノード $n \in$ nbr(s)について，評価値 v_n は，nbr(*s*)と nbr(*n*)の間のノードの重複数 c_n と *n* の残存電力 e_n，nbr(*s*)のすべてのノードの残存電力の平均値 e_{ave} を用いて，以下の式(3.1)で求められる．

$$v_n = \frac{1}{c_n + 1} * \left(\frac{e_n}{e_{ave}} \right)^W \qquad (3.1)$$

なお，*W* は残存電力の重みを示す正の定数である．つまり，nbr(*s*)と nbr(*n*)のノード重複個数が少なく，残存電力が多いノードほど評価値が高くなる．*s* は評価値が最も大きかったノード $n1 \in$ nbr(*s*)を次のクラスタヘッドに選ぶ．

さらに，twoNbr(*s*)全体のノード数に対する twoNbr(*s*)と nbr(*n1*)のノードの重複数の割合がしきい値以上でなければ，次のクラスタヘッドを選出する．なお，この割合を以後 2 ホップカバー率と呼ぶ．s は *n1* を除いたノード $n \in$ nbr(*s*)の評価値 v_n を再計算する．このとき，v_n は twoNbr(*s*)から nbr(*n1*)との共通部分を除いたノードの集合と，nbr(*n*)とのノードの重複数 c_n，*n* の残存電力 e_n，nbr(s)の *n1* を除いたノードの残存電力の平均 e_{ave} を用いて，以下の式(3.2)で計算する．*n1* と同様に，評価値が最も大きかったノード *n2* を次のクラスタヘッドに選ぶ．twoNbr(s)と，nbr(*n1*)と nbr(*n2*)の和集合とのノードの重複数が 2 ホップカバー率のしきい値を超えていなければ，同様の処理を繰り返し，2 ホップカバー率のしきい値を超えるまでクラスタヘッドを選ぶ．

$$v_n = \frac{1}{c_n + 1} * \left(\frac{e_n}{e_{ave}} \right)^W \qquad (3.2)$$

こちらは，twoNbr(*s*)の未到達ノードと nbr(*s*)の重複個数，*n* の残存電力がそれぞれ多いノードの評価値が高くなる．

Vol.2011-MBL-57 No.22
Vol.2011-UBI-29 No.22
2011/3/8

図 3.2 は，2 ホップカバー率のしきい値が 0.7 のときに，シンクがクラスタヘッドを選ぶ様子を示している．

シンク *s* の隣接ノード集合 nbr(*s*)と nbr(*n4*)の重複ノードは *n5* のみで，重複個数は 1 である．同様に，nbr(*n5*), nbr(*n6*), nbr(*n7*)との重複個数も 1 である．

しかし，残存電力を見ると，*n4* が最も多く，*n5*, *n7* は同程度で，*n6* が最も少ない．よって，この場合評価値が最も高くなるのは *n4* なので，*s* は *n4* を最初のクラスタヘッドに選ぶ．twoNbr(*s*)には，*n1*, *n2*, *n3*, *n8*, *n7* が含まれ，ノード数は 5 ある．twoNbr(*s*)と nbr(*n4*)の重複ノード *n1*, *n2* の 2 つなので，2 ホップカバー率は 2 / 5=0.4 となり，しきい値の 0.7 を下回っているので，シンクはさらにクラスタヘッドの選出を行う．ここで，twoNbr(*s*)と nbr(*n5*), nbr(*n6*), nbr(*n7*)との重複個数は，*n5* が 1, *n6* が 0, *n7* が 2 である．よって，ここで *s* は *n7* をクラスタヘッドに選ぶ．これによりさらに *n8*, *n9* の 2 ノードが 2 ホップで到達可能になるため，2 ホップカバー率は 4 / 5=0.8 となりしきい値の 0.7 を超えるので，*s* はクラスタヘッドの選出をここで終了する．

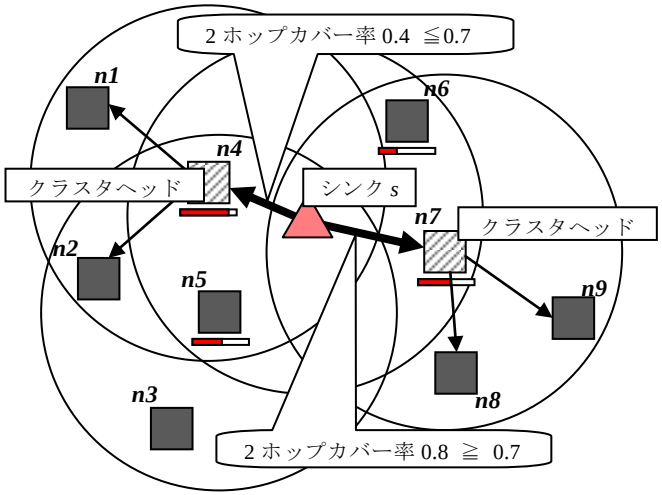

図 3.2 シンクのクラスタヘッド選出

シンク *s* は，クラスタヘッドの選出を終えると，周囲のノードにクラスタヘッド広告をブロードキャストする．クラスタヘッド広告にはシンクが選んだクラスタヘッドの情報と，twoNbr(*s*)に含まれるノードで，新たに選んだどのクラスタヘッドからも到達できないノードが存在するならば，それらのリストが含まれている．このリストのことを以後，3 ホップチェック要求リストと呼ぶ．

s のクラスタヘッド広告を受信したノード *n* は，自分の周囲に存在するクラスタヘッドのリストに *s* を加える．また，*n* がクラスタヘッドに選ばれていた場合は，*n* は *s* を親クラスタヘッドとして，クラスタヘッドになり，以下の処理を行う．

(3) ノード *n* のクラスタヘッド選出

n は周囲のクラスタヘッドの nbr を参照して，2 ホップカバー率を調べる．この時点でしきい値を超えていなければ，シンク同様に 2 ホップカバー率のしきい値を超えるまで式(3.2)の評価値を使いクラスタヘッドの選出を行う．

クラスタヘッドの選出を終えると，*n* は *s*, すなわち親クラスタヘッドから 3 ホップチェック要求リストに対する 3 ホップチェック応答リストを作成する．3 ホップチェック応答リストは，3 ホップチェック要求リストに含まれるノードのうち，*n* が新たに選んだクラスタヘッドを含んだ，*n* の隣接クラスタヘッドを経由して *n* から 2 ホップで到達可能なノードが格納される．つまり，親クラスタヘッドからは 3 ホップで到達可能なノードになる．

また，twoNbr(*n*)に含まれるノードのうち，隣接クラスタヘッド経由で，2 ホップで到達できないノードが存在する場合，それらのノードを 3 ホップチェック要求リストに加える．ただし，親クラスタヘッドの 3 ホップチェック要求リストに含まれているノードは除外する．

ここまでを終えて，*n* はクラスタヘッド広告を周囲のノードにブロードキャストする．*n* のクラスタヘッド広告には，*n* が新たに選んだクラスタヘッドのリストの他，親クラスタヘッドの 3 ホップチェック要求リストに対する 3 ホップチェック応答リスト，*n* の 3 ホップチェック要求リストが含まれる．

n からクラスタヘッド広告を受信したノードは，*n* を隣接クラスタヘッドのリストに加える．さらに，受信したノードが *n* によってクラスタヘッドに選ばれたノードであれば，*n* 同様にクラスタヘッドの選出，*n* の 3 ホップチェック要求リストに対する 3 ホップチェック応答リストの作成, 3 ホップチェック要求リストの作成を行う．また，受信したノードがクラスタヘッドだった場合，3 ホップチェック要求の応答処理を行い，*n* に 3 ホップチェック応答リストを送信する．

(4) クラスタヘッド *n* の 3 ホップチェック応答の受信処理

クラスタヘッド *n* は，3 ホップチェック要求リストを送信する際に，自身に隣接するクラスタヘッドを，応答待ちリストに加える．これらの隣接クラスタヘッドから 3 ホップチェック応答があると，応答待ちリストからそのノードを削除し，3 ホップチェック要求リストから，3 ホップチェック応答リストに含まれるノードを削除する．応答待ちリストが空になると，すべての隣接クラスタヘッドから応答があったとみな

Vol.2011-MBL-57 No.22
Vol.2011-UBI-29 No.22
2011/3/8

すが，この時点で，まだ 3 ホップチェック要求リストが空になっていなかった場合，nbr(*n*)に含まれるノードの中で，隣接ノードに 3 ホップチェック要求リストとの重複ノード個数に，式 2.1 で残存電力の重みを加えた評価値が最も高いノード *n2* を，新たにクラスタヘッドに選び，*n2* に対して 3 ホップチェック要求リストを送信する．

n2 はこれを受け取ると，(3)の手順でクラスタヘッドの選出を行い，*n* に対する 3 ホップチェック応答リストを含めたクラスタヘッド広告をブロードキャストする．これを受け取ってもまだ 3 ホップチェック要求リストが空にならない場合は，3 ホップチェック要求リストが空になるまで，これらの処理を繰り返す．

(5) 終了報告の送信

クラスタヘッド *n* は，新たにクラスタヘッドを選出せずに，3 ホップチェック要求リストが空になった場合，親クラスタヘッドにクラスタヘッド選出の終了報告を送信する．親クラスタヘッドは，すべての子クラスタヘッドから終了報告を受け，3 ホップチェック要求リストも空になったら，さらに親クラスタヘッドへ終了報告を送信する．このようにして，終了報告を転送していき，シンクが終了報告を送信できる状態になると，ネットワークのすべてのノードがいずれかのクラスタヘッドに所属できることになる．よって，これをもってクラスタヘッドの選出を終了する．

クラスタヘッドの選出が終了すると，シンクはクラスタ所属要求をブロードキャストする．これを受信したノードは，隣接クラスタヘッドのリストのうち，クラスタヘッド広告の受信電波強度が最も強かったクラスタヘッドを親クラスタヘッドとして，参加要求を送信する．一方，クラスタ所属要求を受信したノードが送信したクラスタヘッドの子クラスタヘッドだった場合，クラスタ所属要求をブロードキャストする．これにより，すべてのノードがいずれかのクラスタヘッドへ所属することになる．

クラスタヘッドは，クラスタ所属要求をブロードキャストしてから一定時間後に，クラスタメンバのデータ送信スケジュールを作成し，子ノードに送信する．子ノードはスケジュールを受け取ると，ラウンドが変わるまで自身の送信順を保持する．

3.3 センサデータの収集

センサデータを収集するタイミングは，シンクがデータの送信要求をブロードキャストして各ノードに伝える．シンクの子ノードは，送信要求を受けて，シンクへセンサデータを送信する．子クラスタヘッドが受信すると，自身の周囲のノードへ送信要求をブロードキャストする．こうして，すべてのノードに送信要求が行き渡る．クラスタヘッドは,すべての子ノードと子クラスタヘッドからセンサデータを受け取ると，それらと自身のデータを圧縮して，親クラスタヘッドへ送信する．親クラスタヘッドも同様にして，すべての子ノード，子クラスタヘッドからのデータの受信を完了してから親ノードへデータを転送する．

このようにして，ネットワークの末端からすべてのノードが収集したセンサデータをシンクに収集する．シンクがすべてのデータを収集し終えると，1 サイクルが完了する．これを一定サイクル繰り返し，次のラウンドへ移る．

4. 評価

評価は，Java 言語により LEACH と提案方式のシミュレーションプログラムを作成し，シミュレーションにより行う．シミュレーションは，砂漠や汚染地帯など人が立ち入ることが困難な観測領域にセンサを空中から散布する場合を想定し行う．また，シンクは観測領域の外にあるものとする．以下に想定する環境の概略図を示す．

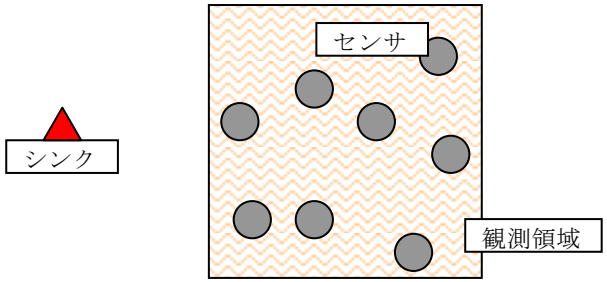

図 4.1 シミュレーション想定環境

4.1 シミュレーション環境

シミュレーション環境は，観測領域を 150m * 150m とし，観測領域の中心から西 100m の位置にシンクを配置する．電波到達範囲を LEACH200m, 提案方式 50m とし，ノードは 100～500 個の一定数を観測領域内のランダムな座標に配置する．ただし，電波到達範囲 75m で完全に孤立したノードは出現しないものとする．

また，送受信における電力の消費モデルは[1]と同じものを用いた．k bits を d m 離れたノードへ送信するときの消費電力 E_T は式(4.1)で，k bits を受信するときの消費電力 E_R は式(4.2)で与えられる．

$$E_T = E_{elec}k + \varepsilon_{amp}kd^2 \qquad (4.1)$$

$$E_R = E_{elec}k \qquad (4.2)$$

ここで，ここで，E_{elec} は 1 bit を送受信する際に消費する電力，ε_{amp} は送信時にかかる電力を示す．

Vol.2011-MBL-57 No.22
Vol.2011-UBI-29 No.22
2011/3/8

評価は，500～4000 の一定サイクル数シミュレーションを繰り返し，シミュレーション終了時におけるノードの残存電力の最小値と平均値を比較して行う．シミュレーションに用いる共通パラメータを以下の表 4.1 に示す．

表 4.1 共通シミュレーションパラメータ

パラメータ	値
E_{elec}	50 nJ / bit
ε_{amp}	100 pJ / bit / m^2
制御メッセージ	500 bits
データサイズ	2000 bits

LEACH のシミュレーションパラメータとして，1 ラウンドあたりのサイクル数とノードがクラスタヘッドに立候補する基本確率 P を定める必要がある．ノード数 200，サイクル数 1000 でシミュレーションを行った結果，1 ラウンドあたりのサイクル数 60，基本確率 P=0.1 のときにそれぞれパフォーマンスが最も優れる結果となった．よって，本論文では LEACH のシミュレーションパラメータにこれらの値を用いた．

提案方式のシミュレーションパラメータとしては，1 ラウンドあたりのサイクル数，2 ホップカバー率のしきい値，そして残存電力の重み W 定める必要がある．こちらも LEACH 同様に，ノード数 200，サイクル数 1000 でシミュレーションを行った結果，1 ラウンドあたりのサイクル数 90，2 ホップカバー率のしきい値 0.2，残存電力の重み W=4.0 をそれぞれ最適値とし，採用した．

4.2 評価・考察

上記のシミュレーションパラメータを用いて，LEACH と提案方式の比較を行い，提案方式の性能評価を行った．

ノードの初期電力 10J，サイクル数 1000 でノード数の違いによる性能の比較の結果を図 4.2 に，ノードの初期電力 20J，ノード数 200 でサイクル数による比較の結果を図 4.3 にそれぞれ示す．

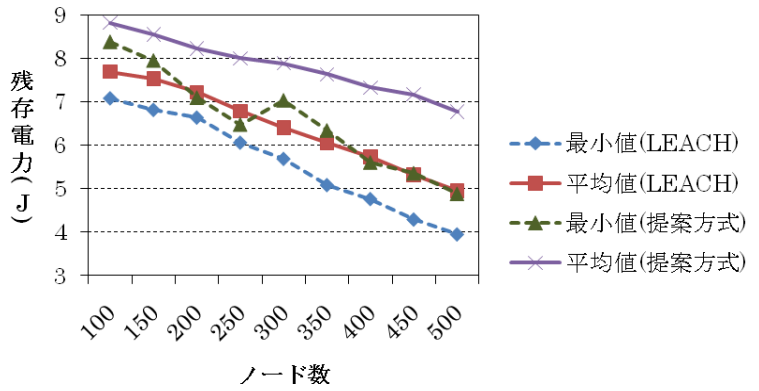

図 4.2 ノード数による LEACH と提案方式の性能比較

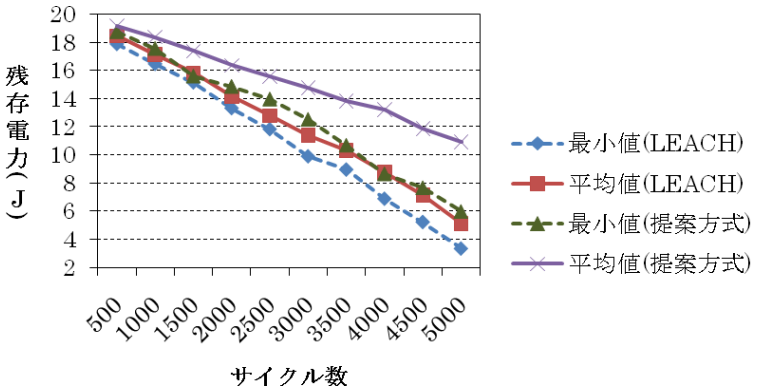

図 4.3 サイクル数による LEACH と提案方式の性能比較

図 4.2 を見ると，LEACH，提案方式ともにノード数の増加に伴い残存電力は減少する傾向にあるが，提案方式のほうがこの傾向が緩やかである．提案方式によるパフォーマンスの改善率は平均して，最小値で 17.34%，平均値で 22.05%となった．また，ノード数 450 のとき，最小値 24.85%，ノード数 500 のとき，平均値 36.70%となり，それぞれ最大の改善率となった．このことから，提案方式はノード数が多いほど LEACH より有効であると言える．

Vol.2011-MBL-57 No.22
Vol.2011-UBI-29 No.22
2011/3/8

また，図 4.3 を見ると，提案方式は LEACH よりサイクル数の増加によるパフォーマンスの低下率が小さくなっていることがわかる．特に平均値は，サイクル数の増加にともなって差が大きく開いていることがわかる．提案方式によるパフォーマンスの改善率は平均して，最小値で 16.06%，平均値で 25.17%となった．また，最大ではサイクル数 5000 のときで，平均値は 79.42%，最小値では 113.28%の改善率となった．このことから，提案方式は長期間の使用においても，LEACH より優れていると言える．

5. まとめ

本論文では，センサネットワークにおける隣接ノード集合と残存電力に着目し，それらを用いた省電力クラスタリング方式を提案した．そして，提案方式に基づいたシミュレーションプログラムを作成し，LEACH との性能比較を行った．その結果，提案手法を用いたことにより，LEACH よりも高いパフォーマンスを得られることがわかった．ノード数による比較では，最小値で 17.34%，平均値で 22.05%の改善が見られた．サイクル数による比較では，最小値で 16.06%，平均値で 22.17%の改善が見られた．

最後に今後の課題を述べる．

まず，提案方式はノードの電力がなくなってしまった場合や，電波の干渉を想定していない．このような場合でも，電力がなくなったノードを迂回してネットワークを維持するなどの対策を考え，より現実に近い環境での評価が必要である．

また，今回は孤立したノードが出現しないことを想定し，電波到達範囲を固定したが，ノードが孤立しない最低限の電波到達範囲を動的に検出し，使用するようにすればより柔軟で効率的なプロトコルとなると考えられる．

最後に，今回は残存電力に関して LEACH とのみの比較を行ったが，HEED やその他のプロトコルとの比較や，サイクルごとのデータ収集率の変化など残存電力以外の評価も行うことで,提案方式のさらなる改善案を見つけることができると考えられる．

今後は，上記のような課題の解決に向けて取り組んでいきたい．

謝辞　本研究の一部は日本学術振興会科学研究費基盤研究(c)(22500071)の助成を受けたものである．

参考文献

1)　W. R. Heinzelman, A. Chandrakasan and H. Balakrishnan: “Energy-Efficient Communication Protocol forWireless Microsensor Networks”, Proceedings of the 33rd Hawaii International Conference on System Sciences, pp. 1-10, (2000).

2)　O. Younis and S. Fahmy: “HEED: A Hybrid, Energy-Efficient, Distributed Clustering Approach for Ad Hoc Sensor Networks”, IEEE Transaction on Mobile Computing, vol.3, No.4, pp. 366-379, (2004).

3)　B. J. Culpepper, L. Dung and M. Moh: “Design and Analysis of Hybrid Indirect Transmissions (HIT) for Data Gathering in Wireless Micro Sensor Networks”, ACM Mobile Computing and Communications Review, vol.3, pp.61-83, (2004).

4)　S. Lindsey, C. Raghavenda and K. M. Sivalingam: “Data Gathering Algorithms in Sensor Networks Using Energy Metrics”, IEEE Transactions on Parallel and Distributed Systems, vol.13, No.9, pp.924-935, (2002).

5)　W. Choi, P. Shah and S. K. Das: “A framework for energy-saving data gathering using two-phase clustering in wireless sensor networks”, Proceedings of the First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, pp.203-212, (2004).

6)　牛島準一,沖野正宗,加藤聰彦,伊藤秀一: “高密度アドホックネットワークにおける中継用ランドマークノードの選出方法の提案と評価”, 電子情報通信学会技術研究報告, pp. 93-96, (2003).