

東邦大学学術リポジトリ



OPAC

東邦大学メディアセンター

タイトル	無線センサネットワークにおける省電力クラスタリング方式の提案と評価
作成者（著者）	豊田, 慎之介
公開者	東邦大学
発行日	2013.3
掲載情報	東邦大学理学修士論文(情報科学専攻)平成24年度. 62. p.1 51.
資料種別	学位論文
内容記述	学位所得年月: 2013年3月 / 指導教員: 佐藤文明
著者版フラグ	author
メタデータのURL	https://mylibrary.toho u.ac.jp/webopac/TD02425043

無線センサネットワークにおける

省電力クラスタリング方式の提案と評価

学籍番号 6511013 氏名 豊田 慎之介

要旨

近年、センサ同士を無線通信で接続し、ネットワークを構築する無線センサネットワークが注目を集めている。しかし、センサ端末の電源は電池に依存しており、有限である。このため、無線センサネットワークにおいて、省電力に考慮したルーティングプロトコルは非常に重要であり、盛んに研究が行われている。本稿では、ノード同士で Hello メッセージを交換し、ノードの隣接ノードと電力残量の情報を同時に評価し、経路を構築する新たな省電力クラスタリング方式を提案する。提案手法では、効率的に、かつ電力にも配慮した経路を構築しつつ、クラスタヘッドにならなかつた一部のノードをスリープさせることにより、さらなる省電力化を図った。本稿では、シミュレーションにより従来研究と提案手法の比較を行い、提案手法の有効性を検証した。

目次

1	序論	1
2	研究背景	3
2.1	無線センサネットワーク	
2.2	LEACH	
2.2.1	LEACH のアルゴリズム	
2.3	HEED	
2.3.1	HEED のアルゴリズム	
2.4	HIT	
2.4.1	HIT のアルゴリズム	
2.5	MR-LEACH	
2.5.1	MR-LEACH のアルゴリズム	
2.6	その他の従来研究	
3	提案手法	19
3.1	提案手法における経路構築のアルゴリズム	
3.2	ノード密度を考慮した動的スリープ制御	
3.2.1	提案するスリープ制御方式のアルゴリズム	
4	評価	35
4.1	シミュレーション概要	
4.1.1	シミュレーション想定環境	
4.1.2	電力モデル	
4.1.3	シミュレーションパラメータ	
4.2	サイクル数ごとの生存ノード数の推移	
4.3	サイクル数ごとの領域カバー率の推移	
4.4	ノードの全消費電力に対する制御メッセージの割合	
4.5	スリープ制御の有無による提案手法の性能比較	
5	結論	49
	謝辞	50
	参考文献	51

第1章 序論

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を集めている．無線センサネットワークとは，温度センサ，照度センサ，加速度センサなどのセンサを，無線で接続し，ネットワークを構成するものである．無線センサネットワークは搭載するセンサの種類，ネットワークの規模などにより幅広い分野への応用が可能であり，期待が高まっている．

一方，センサ端末の電源は主に電池に依存しており，有限である．さらに，外部からの電源の供給が困難である場合も想定され，電池の交換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を長期間運用するためには，省電力化は必要不可欠であり，盛んに研究が行われている．

本稿では，ルーティング方式に着目し，省電力ルーティングの実現を図った．従来研究として，無線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に LEACH^[1]がある．LEACH では，観測領域を細かい領域（クラスタ）に分割する．そして，クラスタの代表ノードであるクラスタヘッドが，クラスタメンバからデータを収集し，それらを集約してシンクへ送信する．これにより，クラスタヘッドでないノードはシンクへ直接データを送信するのではなく，より近いクラスタヘッドへデータを送信できるようになる．通信の消費電力は通信距離に大きく依存しているため，負担の大きいシンクとの通信をクラスタヘッドのみが行うことにより，全体の消費電力の削減が可能になる．この他にも，LEACH を改良したアルゴリズムで，ノードがクラスタヘッドに立候補する際に，自身の電力残量を考慮することで全体の消費電力の偏りを解消する HEED^[2]，マルチホップ通信を用いることで電波干渉，通信電力を削減する HIT^[3]，クラスタを階層的に構築し，電波干渉を抑えるプロトコル MR-LEACH^[4] など様々なルーティング方式が提案されている．

本稿では，隣接ノード集合とノードの電力残量の情報を考慮して，経路を構築することにより，各ノードの電力残量の偏りを解消しつつ，無線センサネットワークの寿命を延ばすことができる新たなルーティング方式を提案する．さらに，提案手法ではノードの密度が高くなっている領域においては，クラスタヘッドでないノードの一部をスリープさせることにより，領域のカバー率を大きく損なうことなく，ネットワークの寿命のさらなる延長を図った．

さらに，本稿では，シミュレーションを用いて従来研究と提案手法との間で，ネットワークの寿命，及び領域のカバー率それぞれについて比較を行い，提案手法の有効性を検証した．

以下に本稿の構成を示す．第 2 章では，無線センサネットワークの概要と，従来研究の特徴，アルゴリズム，問題点について述べる．第 3 章では，隣接ノード集合と電力残量を考慮したクラスタリング方式を提案し，その特徴と詳細なアルゴリズムについて述べる．第 4 章では，シミュレーションの詳細について述べ，提案手法の評価を行う．最後に，第 5 章では本研究の結論と今後の課題について述べる．

第2章 研究背景

2.1 無線センサネットワーク

無線センサネットワークとは、無線通信装置を内蔵したセンサ端末によって構成されたネットワークである。以後、センサネットワークにおいてネットワークを構成するセンサ端末をノードと呼ぶ。

センサネットワークは、単にノード同士を接続するわけではなく、各ノードが自身の周辺にあるノードを自動的に見つけ出し接続し、ネットワークを構成している。また、複数のノードが状況によって自律的にネットワークを構成することも可能である。例えば、ノードの1つが故障してしまったとしても、他のノードはネットワークを再構成して通信を維持することが可能である。

ノードに搭載するセンサは目的によって様々である。例えば、気温を観測するためには温度センサ、日照量を観測するためには照度センサといった具合である。



図 2.1 無線センサネットワーク

センサネットワークの用途は様々であり、幅広い分野からの応用が期待されている。例えば、畜産業に応用することを考えると、家畜の体温や行動を、センサを使って監視することで、病気の早期発見や、繁殖期の到来を正確に知ることが可能になる。他にも、医療分野では、患者の血圧や体温などの情報を、センサを使って自動的に収集し、ネットワークを介して病院に送信することが出来れば、自宅で医師の診断を受けることも可能になる。このように、センサネットワークは搭載するセンサにより様々なことが実現できる。

表 2.1 センサネットワークの応用分野とその例

分野	応用例
医療	脈拍や血圧などの生体データを医師に送信することによる健康チェック
環境	温度や湿度などのモニタリングや、自然災害の監視
防災	自然環境にセンサを設置することによる自然災害の予知
防犯	侵入防止・盗難防止
交通	車々間通信によるデータ相互通信や衝突回避
建物	人の動きをモニタリング
物流	荷物の位置を管理
農業	農作物の管理
住宅	空調・照明・遠隔コントロールなど統合管理システム

実際にセンサネットワークが社会基盤として展開されている例に、スマートグリッドがある。スマートグリッドとは、発電機で生成される電力、太陽電池や風力発電などのクリーンエネルギーといった電力供給と、家庭内で利用される電力を、センサを使ってリアルタイムに取得し、電力の発電量・蓄積量・消費量を最適化するシステムのことである。すでに欧州では電力メータのセンサネットワーク化を実現する、スマートメータが導入されている。

このように、センサネットワークは各分野から、今後の普及を大いに期待されている。その一方で、センサネットワークには課題もある。センサネットワークを構成するセンサは、電源を電池に依存するものがほとんどで、電源は有限である場合が多い。その上、特にノード数が多く、広範囲に及ぶような大規模なセンサネットワークでは外部から電源を供給することも困難で、電池を交換するのもコストがかかってしまう。そのため、特にこのような場合では、ネットワークを長期間稼働させるために省電力化は必要不可欠である。

無線センサネットワークの省電力化を実現する 1 つのアプローチとして、省電力ルーティングプロトコルの導入があげられる。無線センサネットワークでは、センサ端末の消費電力の多くは通信によるものであるため、省電力に配慮したルーティングプロトコルは、ネットワークの長寿命化を実現するためには非常に重要である。

従来研究として、無線センサネットワークに適した省電力ルーティングプロトコルをいくつか紹介する。

2.2 LEACH

LEACH(Low-Energy Adaptive Clustering Hierarchy)^[1]は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である。LEACHはクラスタリングを用いたルーティング方式で、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、それぞれが1ホップで構成されるという特徴を持つ。また、LEACHで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させる。これにより、多数のノードに負荷を分散させ、クラスタヘッドになったノードがいち早く電力を使い果たしてしまう事態を防ぐことができる。これが、LEACHの最大の特徴である。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。

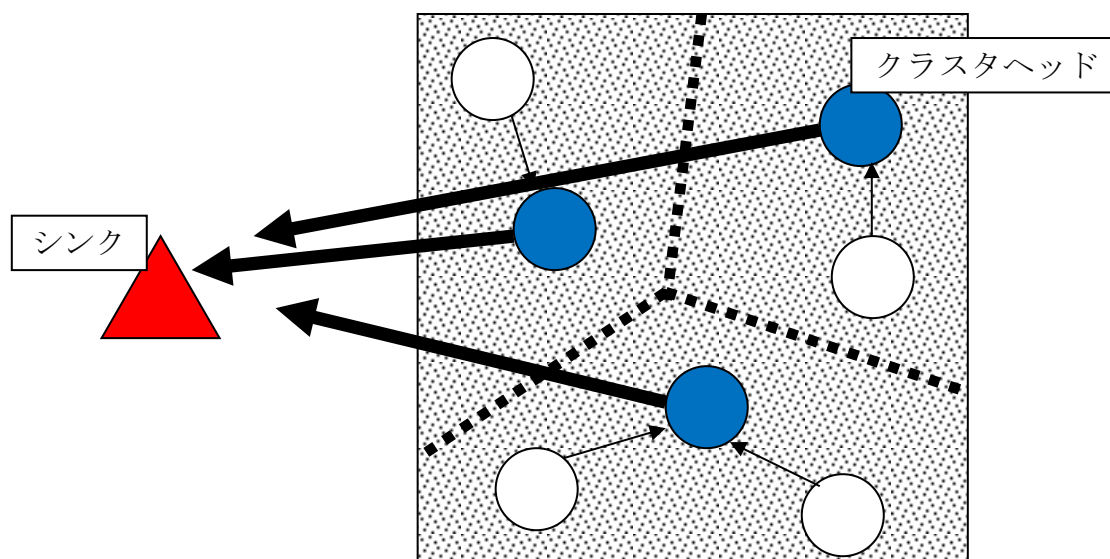


図 2.2 LEACH

2.2.1 LEACH のアルゴリズム

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る。

■クラスタヘッドへの立候補

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、 $0 \sim 1$ の乱数を作り、以下の式(2.1)で計算したしきい値 $T(n)$ を下回った場合、 n はクラスタヘッドに立候補する。

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

ここで、 r は現在のラウンド数、 G は過去 $1/P$ 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。

例として、 $P=0.2$ のとき、ノード $n1$ が 2 ラウンド目にクラスタヘッドに立候補する場合を考える。ここで $n1$ は 1 ラウンド目にクラスタヘッドに立候補しなかったとすると、 $n1$ は G に含まれるため、 $T(n1) = 0.33$ となる。よって、このとき $n1$ がクラスタヘッドに立候補する確率は、およそ 33%となる。

■クラスタの構成

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。

一方、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に入る。クラスタヘッド広告を受信すると、受信ノードは送信ノードのノード ID、及び受信電波強度を記憶する。

受信待ちを終えると、クラスタヘッドでないノードは、記憶したノードのなか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。このとき、受信電波強度の強いクラスタヘッドを選ぶのは、データ送信時の消費電力は送信距離に大きく依存するため、より近いクラスタヘッドを選ぶためである。また、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、直接シンクヘデータを送信する。

つまり、LEACH では次の図 2.3 のように、ラウンドごとにクラスタの構造も変化する。

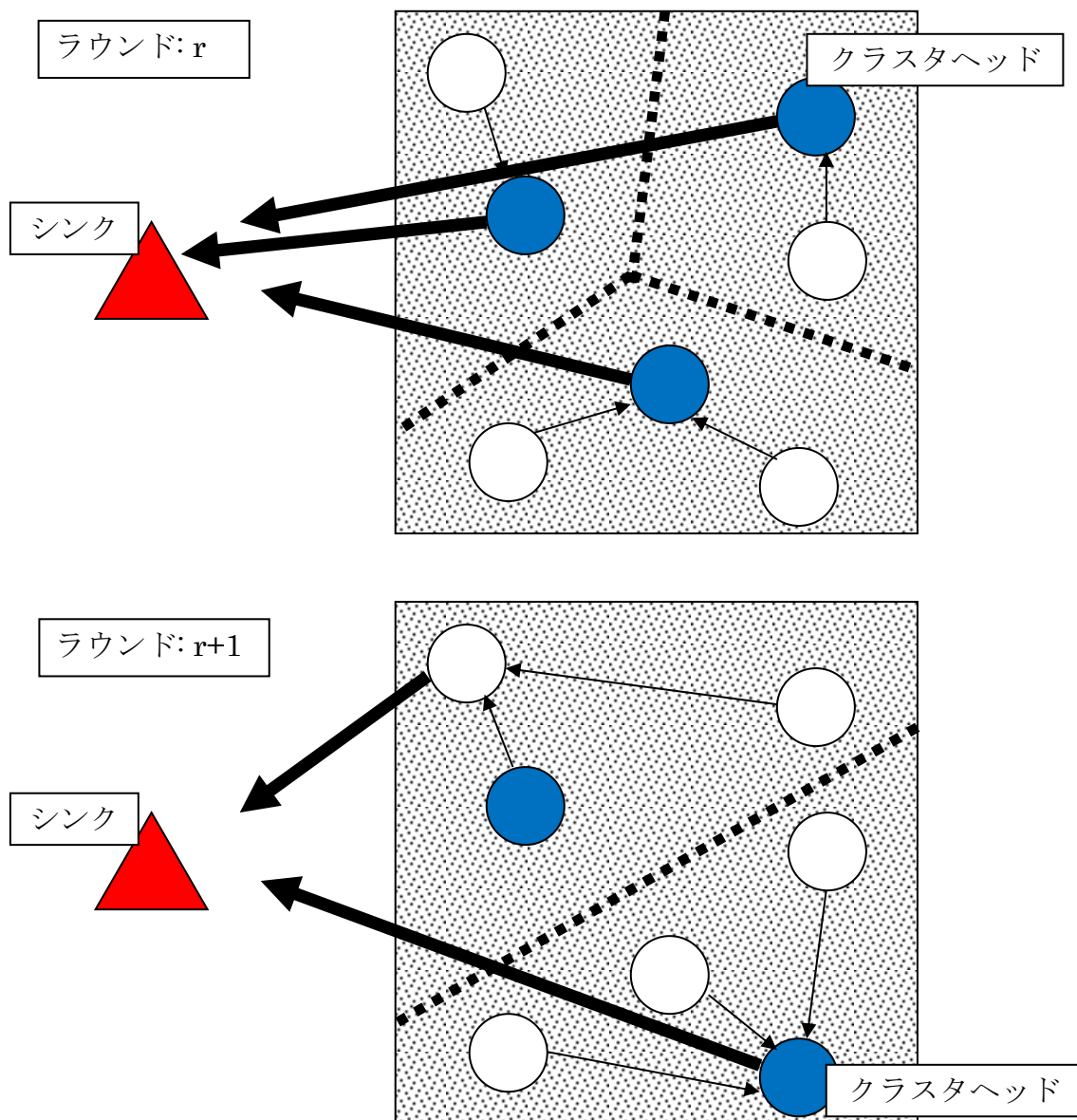


図 2.3 ラウンドの変更に伴うクラスタ構造の変化

■送信スケジュールの作成

クラスタヘッドに立候補したノードは、クラスタヘッド広告を送信し終わると一定時間参加要求待ち状態に入る。この参加要求待ち時間のうちに参加要求を送信してきたノードをクラスタメンバとみなす。

すべてのメンバから参加要求を受信し、参加要求待ち時間が終了すると、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、メンバにブロードキャストする。

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。

■センサデータの送信

クラスタのセットアップが完了すると、データ収集フェーズに入る。クラスタヘッドでないノードは送信スケジュールに従ってクラスタヘッドにデータを送信する。クラスタヘッドはすべてのメンバからデータを受信すると、それらと自身のセンサデータを圧縮してシンクに転送する。

このサイクルを一定数繰り返し、次のラウンドに移行する。

2.3 HEED

LEACH ではノードがクラスタヘッドに立候補する際に、単純な確率とクラスタヘッドになった回数で判定を行なっていたため、最終的にどのノードも同じ頻度でクラスタヘッドに立候補する。前述のとおり、データ送信時の消費電力は送信距離に大きく依存するため、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。このため、LEACH ではノードの電力が偏りやすいという問題がある。また、クラスタヘッドの出現位置が偏り、メンバを持たないクラスタヘッドや、クラスタに所属できないノードが発生しやすくなるという問題もある。

例として、下の図 2.4 では、ノード A はクラスタヘッドに立候補したもののメンバが存在しない。また、ノード B はいずれのクラスタヘッドのクラスタヘッド広告を受信できず、直接シンクへ送信している。

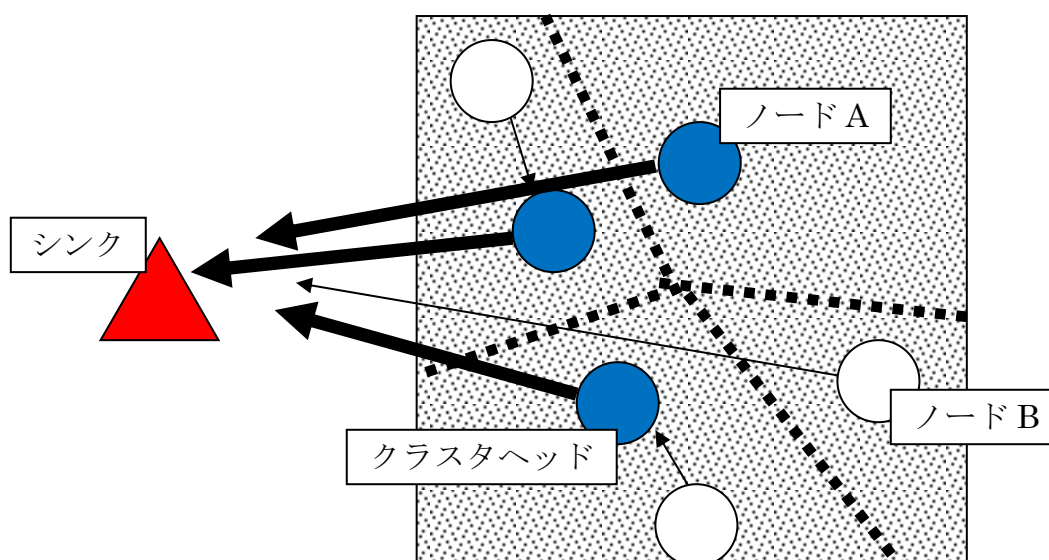


図 2.4 LEACH におけるクラスタの偏りの例

これらの問題を解消する LEACH の改良プロトコルに 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、より消費電力の小さいノードがクラスタヘッドに立候補しやすくしている。これにより、LEACH におけるシンクから離れたノードの電力が枯渇しやすい問題を解消し、ノードの電力残量の均等化が可能となる。

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。これにより、メンバを持たないクラスタヘッドや、クラスタに所属できないノードの発生を抑え、無駄が少ないクラスタの構成を実現している。

2.3.1 HEED のアルゴリズム

HEED では、ラウンドの開始時にノードがクラスタヘッドに立候補する確率 CH_{prob} を次の式(2.2)で計算する。

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

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。

■ 繰り返しフェーズ

各ノードは式(2.2)で CH_{prob} を計算すると、以下の処理を繰り返し行う。

- ・ 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
- ・ クラスタ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
 - ・ $0 \sim 1$ の乱数 $\leq CH_{prob}$ 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。

2.4 HIT

クラスタ内で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 HIT(Hybrid Indirect Transmissions)^[3]がある。HIT では、クラスタヘッドを含むすべてのノードが、自身の上位ノードを 1 つだけ保持する。これによって、下流から上位へデータを転送していくことにより、マルチホップ通信を実現している。また、各ノードの上位ノードの情報を交換することで、電波の干渉を発生させずに、同時通信数を最大化させる送信スケジュールをすべてのノードで共有するという特徴を持つ。

2.4.1 HIT のアルゴリズム

HIT は次の 7 つのフェーズから成る。

1) クラスタヘッド選出フェーズ

このフェーズでは、1 つあるいは複数のクラスタヘッドが選出される。各クラスタはクラスタヘッドを 1 つ持つ。クラスタヘッドが 1 つの場合、ノード ID の順序によってクラスタヘッドの役割をローテーションさせる。すなわち、現在のラウンド数を全体のノード数で割った余りが、自身のノード ID に等しい場合、クラスタヘッドに立候補する。

2) クラスタヘッド広告フェーズ

このフェーズでは、クラスタヘッドが自身のノード情報をクラスタヘッド広告としてブロードキャストする。クラスタヘッドでないノード i がクラスタヘッド広告を受信すると、クラスタヘッドとの距離を計算する。そして、自身に最も近いクラスタに所属する。 i は所属するクラスタヘッド H までの距離を $d(H, i)$ として保持する。

3) クラスタセットアップフェーズ

このフェーズでは、クラスタが形成され、上位、下流の関係が生まれる。まず、クラスタヘッドでないノード i は Member メッセージをブロードキャストする。Member メッセージには i のクラスタヘッドのノード ID、及び i からクラスタヘッドまでの距離が格納される。Member メッセージを交換することにより、各ノード i は他のすべてのノード j について、 i から j までの距離 $d(i, j)$ 、及び j からクラスタヘッド H までの距離 $d(j, H)$ を保持する。これらの情報を用いて、ノード i は、以下の 2 つの条件を満たすノードのうち、 i に最も近いノード

ドを i の上流ノード u_i として選出する.

- $d(u_i, H) < d(i, H)$
- $d(i, u_i) < d(i, H)$

これらの条件は, u_i が i よりもクラスタヘッドに近く, また i がクラスタヘッドと直接通信するよりも u_i と通信するほうが, コストが小さいことを示している.

図 2.5 にクラスタセットアップフェーズにおける, 上流ノード選出の例を示す.

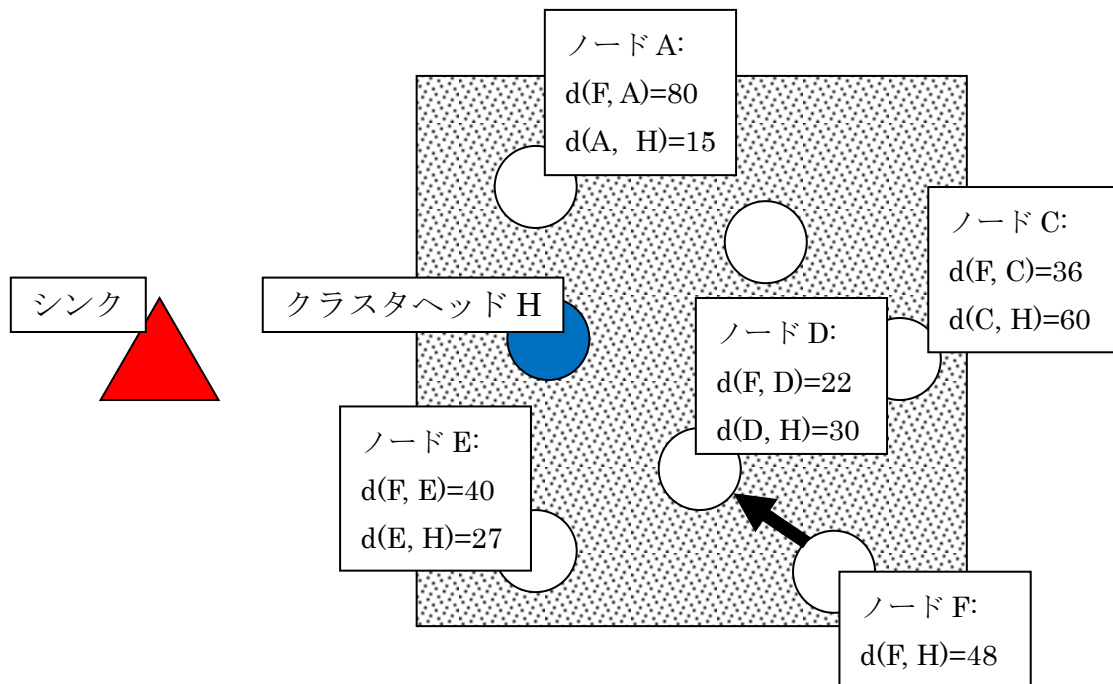


図 2.5 HIT における上流ノード選出の例

ここでノード F の上流ノード選出に注目する. ノード F からクラスタヘッド H までの距離は 48 である. よって, 1 つ目の条件 $d(u_F, H) < d(F, H)$ を満たすノードは, ノード A, D, E の 3 つである. このうち, 2 つ目の条件 $d(F, u_F) < d(F, H)$ を満たすノードは, D, E の 2 つとなる. ここで, ノード F により近いノードは D なので, F はノード D を上流ノードとして選出する.

4) ルートセットアップフェーズ

各ノードがフェーズ 3 で上流ノードを決定した後, すべてのノードは, 上流ノードまでの推定距離を含んだ Upstream メッセージをブロードキャストする. これにより, すべてのノードが他のすべてのノード j の上流ノード u_j , 及びノード j から j の上流ノード u_j までの距離 $d(j, u_j)$ をそれぞれ知ることが可能になる. これらの情報を元に, 各ノードはノード j の下流ノード集合 $\text{DOWN}(j)$ を計算することができる.

5) ブロッキングセット作成フェーズ

このフェーズでは、各ノードはフェーズ 4 で収集した情報を利用し、下流ノードのためのブロッキングセットを作成する。ノード i は必要十分条件、

- $d(i, u_i) > d(i, u_j)$

を満たすノード j をブロックする。このフェーズの最後に、すべてのノードは、ブロックした下流ノードのリストをブロードキャストする。

6) TDMA スケジュール作成フェーズ

このフェーズでは、各ノードは衝突を回避しつつ、同時通信数が最大になるように TDMA スケジュールを作成する。

7) データ伝送フェーズ

このフェーズでは、各ノードは環境のセンシングを行い、センシングしたデータを TDMA スケジュールに従い、上流ノードに転送する。

2.5 MR-LEACH

MR-LEACH(Multi-hop Routing with Low Energy Adaptive Clustering Hierarchy)^[4]は、シンクからクラスタヘッドを階層的につないでいくことで、マルチホップを実現し、通信の消費電力を削減可能にしたプロトコルである。また、MR-LEACHではノードの電力残量も考慮されており、電力残量の大きいノードがクラスタヘッドに立候補するという特徴も持つ。

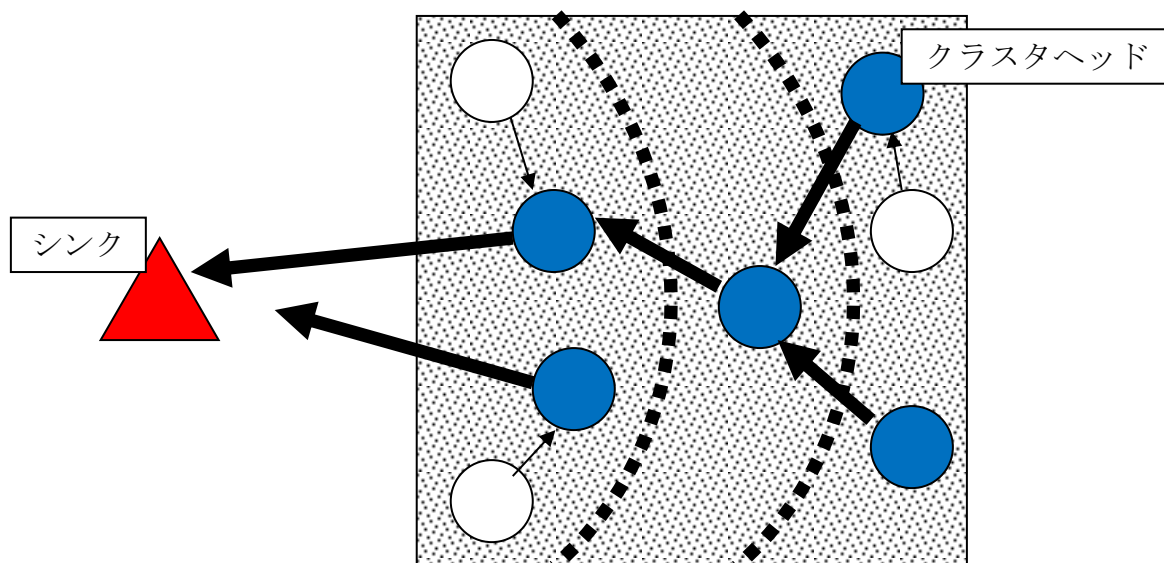


図 2.6 MR-LEACH

2.5.1 MR-LEACH のアルゴリズム

MR-LEACHでは、まずクラスタを作成し、クラスタが形成された後に階層化による経路構築を行う。

■クラスタ作成フェーズ

ラウンドが変更され、このフェーズが開始するとすべてのノードは HELLO メッセージを半径 r の範囲にブロードキャストする。この HELLO メッセージには以下の情報が含まれる。

- 送信ノードのノード ID
- 送信ノードの電力残量

各ノードは、周辺ノードからこの HELLO メッセージを受信すると、送信ノードのノード ID、および電力残量の情報を保存する。

すべてのノードが HELLO メッセージの送受信を終えると、各ノードは保存

した周辺ノードの電力残量の情報と自身の電力残量を比較する。そして、自身の電力残量が最も大きかった場合、クラスタヘッドに立候補し、クラスタヘッド広告を半径 r の範囲にブロードキャストする。このクラスタヘッド広告にはクラスタヘッドのノード ID を含む。

クラスタヘッド広告を受信したノードは、送信ノード（クラスタヘッド）のノード ID を保存する。

すべてのクラスタヘッド広告を受信し終わると、各ノードは最も受信電波強度が強かったクラスタヘッドに対して、参加要求を送信し、そのクラスタに所属する。

ここまでの流れで、MR-LEACH におけるクラスタが作成される。

■経路構築フェーズ

このフェーズの開始時に、まずシンクが自身のノード ID を含んだ確認メッセージをすべてのノードに対してブロードキャストする。

確認メッセージを受信したすべてのクラスタヘッドは、自身のノード ID を含んだ応答メッセージを半径 r' の範囲にブロードキャストする。ここで、半径 $r' > r$ である。

シンクは応答メッセージを受信すると、送信してきたクラスタヘッドのノード ID をレイヤ 1 のノードリストに保存する。すなわち、レイヤ 1 のノードリストは、シンクから半径 r' の範囲内に存在するクラスタヘッドのリストである。

次に、シンクはレイヤ 1 のノードリストを付加した確認メッセージをすべてのノードにもう一度ブロードキャストする。

クラスタヘッドはこの確認メッセージを受信すると、自身のノード ID がレイヤ 1 のノードリストに含まれているか確認する。含まれていた場合、自身をレイヤ 1 のクラスタヘッドとして記憶する。含まれていなかった場合、自身のノード ID を含んだ応答メッセージを再度ブロードキャストする。

レイヤ 1 のクラスタヘッドは、応答メッセージを受信したノードのノード ID のリストをシンクへ通知する。

シンクがこれを受信すると、受信したノード ID のリストをレイヤ 2 のノードリストに保存する。

すべてのレイヤ 2 のノードリストを受信し終わると、レイヤ 1、レイヤ 2 のノードリストを含んだ確認メッセージを再度ノード全体に対しブロードキャストし、リストに含まれていなかったノードが応答メッセージをブロードキャストする。

このような流れをすべてのクラスタヘッドが応答しなくなるまで繰り返し、経路の構築が完了する。次の図 2.7 に MR-LEACH の経路構築フェーズのメッ

メッセージシーケンス図を示す。

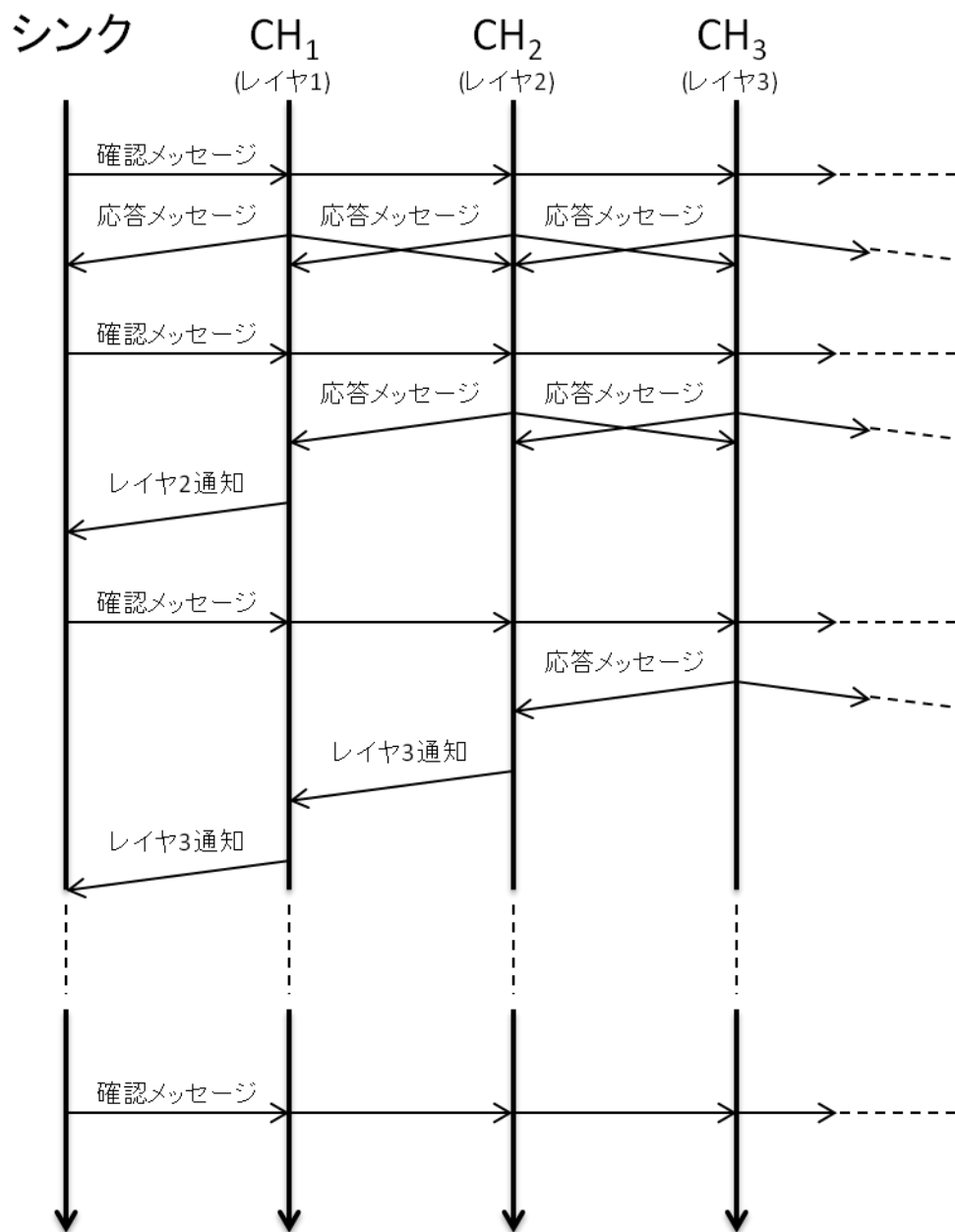


図 2.7 MR-LEACH の経路構築におけるメッセージシーケンス図

2.6 その他の従来研究

この他にも、クラスタではなく最も近いノード同士を結ぶチェーンを構成する PEGASIS^[5], クラスタ内でこのチェーンを構成する TPC^[6]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. これらは, 位置情報を用いているため, 本研究の比較対象としていない.

第3章 提案手法

本稿では、無線センサネットワークにおける隣接ノード集合とノードの電力残量を考慮した省電力ルーティング方式を提案する。提案手法では、各ノードが Hello メッセージを 2 回フラッディングにより交換することで、隣接ノード集合と周辺ノードの電力残量の情報を得る。クラスタヘッドを選出する際には、これらの情報を用いて各ノードに対し評価値を計算し、評価値の高いノードを中継ノードとして採用することで、通信のコストを削減し、ネットワーク全体の長寿命化を図った。また、提案手法では、クラスタヘッドにならなかったノードの一部をスリープさせることにより、領域カバー率を大きく損なうことなく、ノードのさらなる省電力化を実現した。

3.1 提案手法における経路構築のアルゴリズム

本節では、提案手法での経路の構築アルゴリズムを詳細に解説する。提案手法における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^[7]を参考にしている。

提案手法では、各ノード n は以下の情報を保持する。

- 隣接ノード集合: $\text{nbr}(n)$
- 2 ホップ隣接ノード集合: $\text{twoNbr}(n)$
- 未到達ノード集合: $\text{unreached}(n)$

$\text{nbr}(n)$ は、 n と直接隣接しているノードの集合である。 $\text{nbr}(n)$ に含まれるノード i について、以下の情報を保持する。

- i の電力残量
- $\text{nbr}(i)$

ただし、 $\text{nbr}(i)$ はノード ID のみのリストである。よって、 $\text{nbr}(i)$ に含まれる各ノードについては、電力、隣接ノード集合の情報を保持しない。

$\text{twoNbr}(n)$ は、 n から 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の集合である。すなわち、 $\text{nbr}(n)$ に含まれるすべてのノード i の隣接ノード集合 $\text{nbr}(i)$ の和集合から $\text{nbr}(n)$ に含まれるノードを除いた集合である。

$\text{unreached}(n)$ は、 $\text{twoNbr}(n)$ に含まれるノードのうち、隣接するクラスタヘッドを経由して 2 ホップまたは 3 ホップで到達できないノードの集合である。

■ Hello メッセージ交換フェーズ

新しいラウンドが開始され、経路構築がはじまると、まずこのフェーズが実行される。

このフェーズでは、各ノードは周辺ノードと Hello メッセージをフラッディングを用いて交換することにより、2 ホップ先までのノードの隣接状況、および隣接ノードの電力残量の情報を得る。このとき、フラッディングの電波到達距離は狭いものから必要に応じて徐々に広げていく。これにより、通信の消費電力の削減を図る。また、各ノードの電力残量の情報を収集し、代表ノードを決定する。以下、詳細なアルゴリズムを述べる。

1. Hello(1)のフラッディング

まず、シンクがすべてのノードに対して Hello(1)交換要求メッセージを送信する。このメッセージには、フラッディングに用いる電波到達距離、及びフラッディング開始ノードの情報を含む。フラッディング開始ノードには基本的には前のラウンドの代表ノードが指定される。

各ノードは、このメッセージを受信すると、シンクから指定された電波到達距離を保存し、Hello(1)の受信待ちに入る。シンクから指定されたフラッディング開始ノードは、このメッセージを受信すると、電波到達範囲内のノードに Hello(1)をブロードキャストする。また、Hello(1)には、送信ノードの電力残量の情報を含む。

ノード j の Hello(1)を受信したノード i は、隣接ノード集合 $\text{nbr}(j)$ に、 j のノード ID、及び電力残量を保存する。このとき、ノード i が Hello(1)未送信だった場合、 j を仮親に設定し、Hello(1)をブロードキャストする。この Hello(1)には i の電力残量、及び仮親の情報が含まれる。ノード j が i の Hello(1)を受信すると、 j は i を仮メンバとしてリストに保存する。

このようにして、フラッディングの開始ノードを根とするツリーが一時的に構築される。

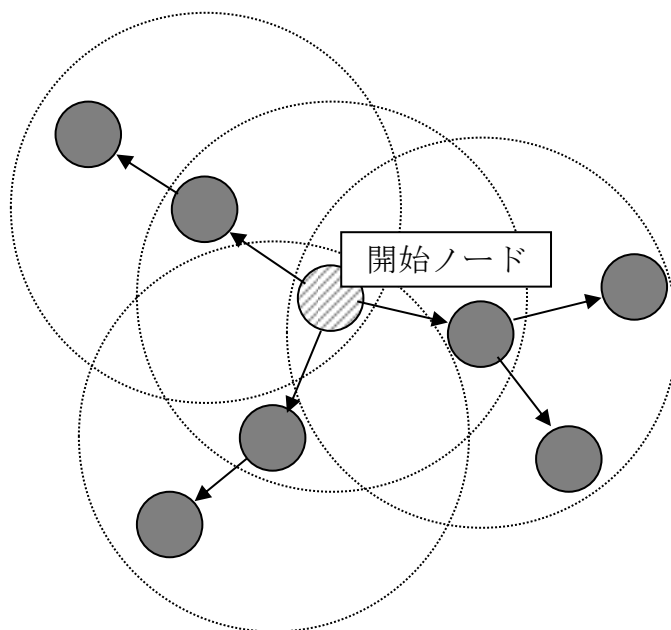


図 3.1 Hello(1)のフラディングにより構築される一時的なツリー

2. Hello(2)メッセージの交換

シンクは、Hello(1)交換要求メッセージ送信後、一定時間が経つと、全ノードに対してHello(2)交換要求メッセージを送信する。

仮メンバがないノード、すなわちHello(1)で構築されたツリーの葉ノードがこのメッセージを受信すると、Hello(2)をブロードキャストする。ここで、送信ノード s のHello(2)には以下の情報を含む。

- s の隣接ノード集合 $\text{nbr}(s)$
- s の電力残量
- s のノード ID

送信ノード s のHello(2)を受信した受信ノード r は、 $\text{nbr}(r)$ の s の情報に、 $\text{nbr}(s)$ の情報を付け加える。また、 $\text{nbr}(s)$ のうち、 $\text{nbr}(r)$ に含まれないノードを $\text{twoNbr}(r)$ に追加する。さらに、 s が r の仮メンバに含まれていた場合、自身の電力残量と s の電力残量を比較し、より電力残量大きい方のノード ID と電力を保存する。他の仮メンバからHello(2)を受信した場合も、同様にして電力残量の大小を確認し、最大値を保持する。

すべての仮メンバからHello(2)を受信したノードは、自身もHello(2)をブロードキャストする。このとき、仮親に渡される電力残量が最大のノードの情報は、送信ノードを根とする部分木における最大値を保持するノードの情報となる。

このようにして、木の末端からHello(1)の開始ノードまでHello(2)が伝搬される。Hello(1)の開始ノードがすべての仮メンバからHello(2)の受信を終えると、開始ノードはノードすべての中で電力残量が最大のノードの情報を得ることが

できる。

また、シンクから Hello(2)交換要求メッセージを受信した際に、いずれのノードからも Hello(1)を受信しなかった場合、このノードはシンクに対して経路構築エラーを報告する。

シンクは経路構築エラーを受信すると、フラッディングの電波到達距離を一段階広いものに更新し、Hello(1)交換要求メッセージを再度送信する。

このプロセスにより、いずれのクラスタにも所属できないノードの発生を防ぐことができる。

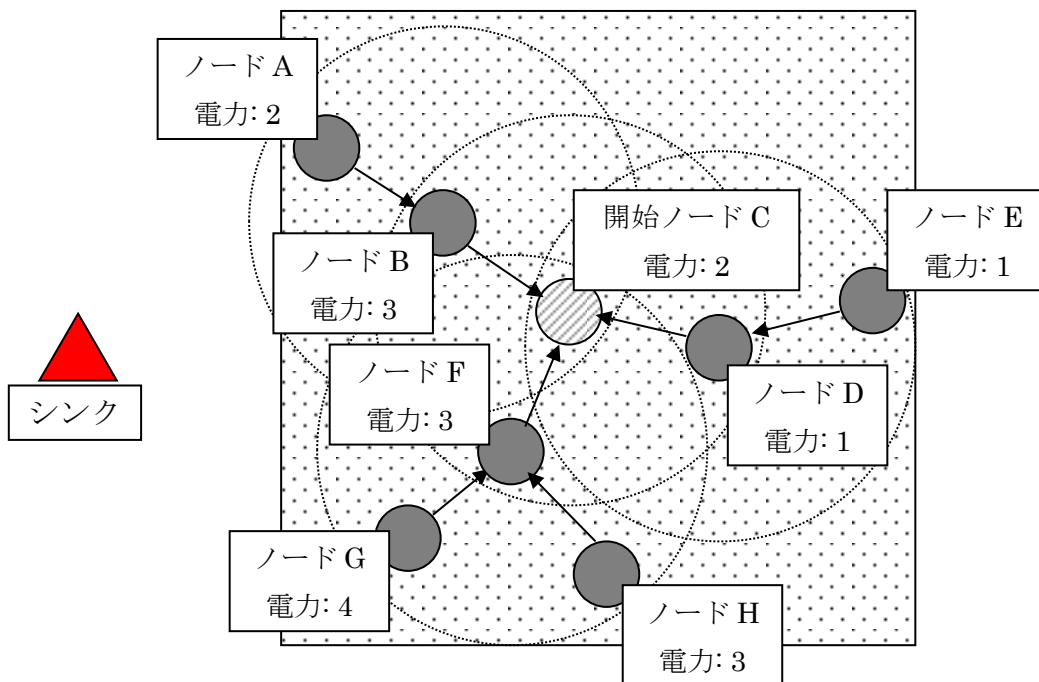


図 3.2 Hello(2)の交換の例

図 3.2 における Hello(2)の交換の様子を解説する。まず、シンクが Hello(2)交換要求メッセージを全ノードに対しブロードキャストする。

すべてのノードのうち、ノード A, E, G, H はそれぞれ仮メンバを持っていないため、Hello(2)のブロードキャストを行う。

ここで、ノード F に注目する。ノード F は、仮メンバに G, H を持っている。よって、この 2 つのノードの Hello(2)を受信した後に Hello(2)をブロードキャストすることができる。ノード B, D についても同様に仮メンバからの Hello(2)を受信後、Hello(2)をブロードキャストする。

開始ノード C は、仮メンバ B, D, F からの Hello(2)を受信すると、自身の Hello(2)を送信し、ノード G の電力残量とノード ID の情報を最大値として保存する。

3. 代表ノード通知

Hello(1)のフラッディング開始ノードがすべての仮メンバから Hello(2)を受信すると、最終的に最も電力残量が大きいノードを代表ノードに決定し、シンクへ通知する。

シンクは、これを受信すると、代表ノードの情報を保存する。これで、このフェーズは終了となる。

■ クラスタヘッド選出フェーズ

このフェーズでは、先の Hello メッセージ交換フェーズで選出された代表ノードからクラスタヘッドを選択していき、マルチホップ経路を構築する。クラスタヘッドの選出を行う際には、ノードの隣接ノードに占める未到達ノードの数、及びノードの電力残量を同時に評価することで、効率のよい、かつノードの電力にも配慮した経路の構築を可能にした。また、未到達ノードをすべて 2 ホップで到達させるのではなく、一部を 3 ホップで到達させることにより、クラスタヘッドの過剰な出現の防止を図った。以下に、このフェーズのアルゴリズムの詳細を述べる。

1. クラスタ構成要求のブロードキャスト

シンクは、代表ノードの通知を受けると、クラスタ構成要求をブロードキャストする。このメッセージには代表ノードの情報が含まれる。

代表ノードに選ばれたノードがこのメッセージを受信すると、クラスタヘッドの選出を開始する。

2. 代表ノードによるクラスタヘッドの選出

代表ノード L は、隣接ノード集合 $\text{nbr}(L)$ に含まれるすべてのノード i について、以下の式(3.1)を用いて評価値 v_i を計算する。

$$v_i = c_i * \left(\frac{e_i}{e_{\text{ave}}} \right)^W \quad (3.1)$$

ここで、 c_i は L の未到達ノード集合 $\text{unreached}(L)$ と $\text{nbr}(i)$ との間のノード重複個数、 e_i は i の電力残量、 e_{ave} は $\text{nbr}(L)$ に含まれるすべてのノードの電力残量の平均、 W は電力の重みを示す定数である。すなわちこの式では、 L の未到達ノードを隣接ノードにより多く含み、かつ電力残量が大きいノードほど評価値が高くなる。

代表ノード L は評価値が最も高くなったノード n_l を最初のクラスタヘッドに選出する。 L は n_l をクラスタヘッドに選ぶと、 $\text{unreached}(L)$ から $\text{nbr}(n_l)$ に含

まれるノードを削除する。

次に代表ノード L はクラスタヘッドの選出を更に続けるのかどうかの判定を行う。判定には、以下の式(3.2)で計算される 2 ホップカバー率を用いる。

$$2 \text{ ホップカバー率} = \frac{\text{twoNbr}(L) - \text{unreached}(L)}{\text{twoNbr}(L)} \quad (3.2)$$

ここで、式(3.2)における $\text{twoNbr}(L)$ 、及び $\text{unreached}(L)$ はそれぞれの集合に含まれるノードの数を示している。よって 2 ホップカバー率とは、 $\text{twoNbr}(L)$ におけるカバー済みノードの割合を示すものである。ここで、 L の 2 ホップカバー率が予め定められたしきい値以上になった場合、 L はクラスタヘッドの選出を終了する。

一方、しきい値を下回っていた場合は L は次のクラスタヘッドの選出を行う。 L は n_1 を除いた $\text{nbr}(L)$ の各ノード i に対して、式(3.1)を用いて評価値 v_i を再計算する。ここで、評価値の再計算を行うことで、 n_1 と近い隣接ノード集合を持つノードの評価値が高くなることを防いでいる。 L は新たに計算した評価値が最も高くなったノード n_2 を次のクラスタヘッドに選出し、 n_1 のときと同様に、 $\text{unreached}(L)$ から $\text{nbr}(n_2)$ に含まれるノードを削除する。その後、2 ホップカバー率を再度計算し、しきい値以上であればクラスタヘッドの選出を終了する。しきい値を下回った場合は、同様の手順で 2 ホップカバー率がしきい値以上になるまでクラスタヘッドの選出を繰り返す。

次の図 3.3 を用いて代表ノード L がクラスタヘッドの選出を行う流れを具体的に説明する。

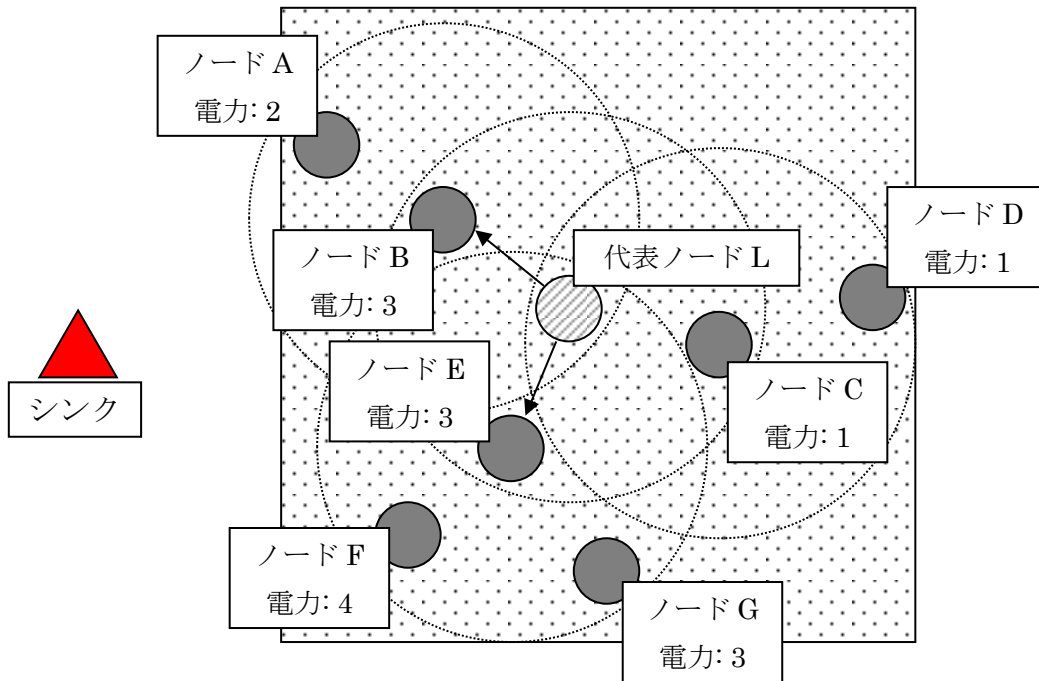


図 3.3 代表ノードによるクラスタヘッド選出の例

ここで、この例における 2 ホップカバー率のしきい値は 0.6, 電力の重み W は 2 であるとする。

まず, L は $\text{nbr}(L)$ に含まれるすべてのノードの評価値を計算する。ここで, $\text{nbr}(L)$ に含まれるノードはノード B, C, E である。これらのノードの電力残量の平均 e_{ave} は約 2.33 である。

ノード B は, $\text{unreached}(L)$ に含まれるノード A を隣接ノードに持つため, cb は 1 である。また, B の電力残量は 3 なので, 式(3.1)にもとづいて v_B を計算すると, v_B は約 1.65 となる。同様に, ノード C の評価値 v_C は約 0.18, ノード E の評価値 v_E は約 3.31 となる。よって, L は評価値の最も高いノード E を最初のクラスタヘッドに選出し, $\text{unreached}(L)$ からノード F, G を削除する。

ここで, L は 2 ホップカバー率を計算する。 $\text{twoNbr}(L)$ のノード数は 4, $\text{unreached}(L)$ のノード数は 2 なので, 式(3.2)に従い計算すると, 2 ホップカバー率は 0.5 となる。これは, しきい値 0.6 に満たない値であるため, L は次のクラスタヘッドを選出する。

L はノード B, C に対して再度評価値の計算を行う。ここで, e_{ave} は B, C の平均の 2 に更新される。よってノード B の評価値 v_B は 2.25, C の評価値 v_C は 0.25 となり, L は評価値の高い B を次のクラスタヘッドに選出し, $\text{unreached}(L)$ からノード A を削除する。

ここで, L は再び 2 ホップカバー率を計算する。 $\text{unreached}(L)$ のノード数は 1

となっているので、2 ホップカバー率は 0.75 となり、しきい値を超える。よって、 L はここでクラスタヘッドの選出を終了する。

3. 代表ノードによるクラスタヘッド広告の送信

代表ノード L はクラスタヘッドの選出を終えると、クラスタヘッド広告を電波到達範囲のノードに対してブロードキャストする。このクラスタヘッド広告には以下の情報を含む。

- L が選出したクラスタヘッドのリスト
- 3 ホップチェック要求リスト

ここで、3 ホップチェック要求リストには最終的な $\text{unreached}(L)$ の情報が含まれる。

このクラスタヘッド広告を受信したノード r は、隣接するクラスタヘッドのリストに受信したノードのリストに L のノード ID と受信電波強度を保存する。また、 $\text{unreached}(r)$ から $\text{nrb}(L)$ 及び L が選出したクラスタヘッドの隣接ノードに含まれるノードを削除する。さらに、 r が L が選出したクラスタヘッドのリストに含まれていた場合、 r は L を親とするクラスタヘッドとなり、次のクラスタヘッド選出を開始する。

4. クラスタヘッドによるクラスタヘッドの選出

クラスタヘッド h は、新たにクラスタヘッドを選出する前に、2 ホップカバー率を計算する。この時点で、2 ホップカバー率がしきい値以上であれば、 h は次のクラスタヘッドを選出しない。2 ホップカバー率がしきい値を下回っていた場合は、代表ノード同様に、 $\text{nbr}(h)$ の各ノードに対し評価値を計算し、最も評価値が高いノードをクラスタヘッドに選出する。代表ノード同様に 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出し、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完了すると、親ノードの 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を作成する。3 ホップチェック応答リストは、親ノードの 3 ホップチェック要求リストに含まれるノードのうち、 h が新たに選出したクラスタヘッドの隣接ノードに含まれているノードのリストである。すなわち、親ノードからクラスタヘッドを通して 3 ホップで到達できることになる。

ここまですべてを終えて、 h はクラスタヘッド広告をブロードキャストする。 h のクラスタヘッド広告には以下の情報が含まれる。

- h が新たに選出したクラスタヘッドのリスト
- 親ノードのノード ID

- 親ノードへの3ホップチェック応答リスト
- h の3ホップチェック要求リスト

h の3ホップチェック要求リストは代表ノードの場合と同様に、クラスタヘッド選出を終えての $\text{unreached}(h)$ の情報が付加される。

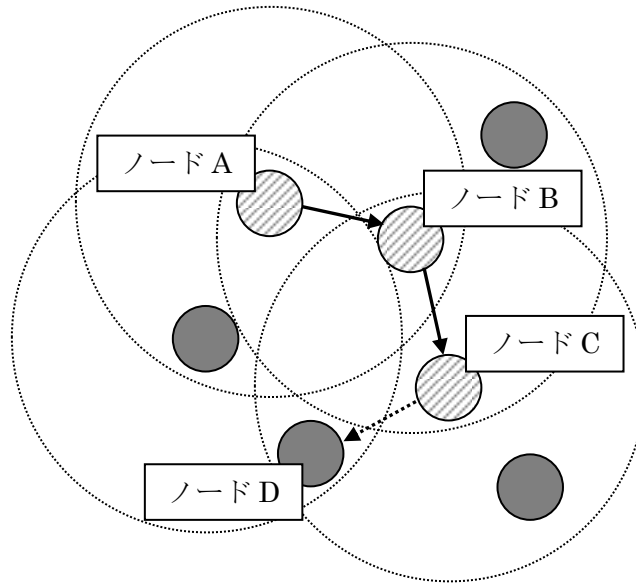


図 3.4 3ホップチェック要求への応答例

図 3.4 を用いて親ノードへの3ホップチェックに対する応答リストの作成を具体的に説明する。まず、クラスタヘッド A はノード B をクラスタヘッドに選び、クラスタヘッドの選出を終了したとする。すると、 $\text{unreached}(A)$ はノード D が残り、3ホップチェック要求リストに格納され、クラスタヘッド広告とともに周囲のノードに通知される。

ノード B がこれを受信すると、まずクラスタヘッドの選出を行う。ここで、B はノード C を新たなクラスタヘッドに選び、クラスタヘッドの選出を終了したとする。ここで B は親ノードである A 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を作成する。ノード D は B が新たに選んだクラスタヘッド C を経由して到達可能であるため、3ホップチェック応答リストにノード D を格納する。そして、クラスタヘッド広告とともに周囲にノードにブロードキャストする。

A はこれを受信することにより、ノード D は到達済みとみなし $\text{unreached}(A)$ から削除する。

5. クラスタヘッド広告の受信処理

クラスタヘッド h のクラスタヘッド広告を受信したノード r は、 $\text{unreached}(r)$

から、 $\text{nbr}(h)$ 及び h が選出したクラスタヘッドの隣接ノード集合に含まれるノードを削除する。

ここで、 r が h の親ノードであった場合、 h を子クラスタヘッドのリストに追加し、 $\text{unreached}(r)$ から h の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。

r がクラスタヘッドであった場合（親ノードの場合を含む）、 h 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を作成する。ここでの3ホップチェック応答リストは r と隣接しているクラスタヘッドの隣接ノード集合に含まれているノードのリストとなる。3ホップチェック応答リストを作成すると、ユニキャストで h に送信する。

r がクラスタヘッドでない場合、かつ h から新たにクラスタヘッドに選ばれていた場合は、上記クラスタヘッドによるクラスタヘッドの選出の流れに従い、クラスタヘッドの選出、及び3ホップチェック応答処理を行う。

6. 3ホップチェック応答の受信処理

クラスタヘッド h は、3ホップチェック応答をユニキャスト、またはクラスタヘッド広告を通して受信すると、 $\text{unreached}(h)$ から3ホップチェック応答リストに含まれていたノードを削除する。隣接するすべてのクラスタヘッドから3ホップチェック応答リストを受信し、かつ $\text{unreached}(h)$ が空にならなかった場合、 h は $\text{nbr}(h)$ のクラスタヘッドでないノードに対し、式(3.1)で評価値を計算し、最も評価値の高いノードに対してクラスタヘッド要請メッセージをユニキャストで送信する。

クラスタヘッド要請を受けたノードは、クラスタヘッドによるクラスタヘッド選出のアルゴリズムに従い、次のクラスタヘッドの選出、及び h の3ホップチェック要求に対する応答処理を行う。

このノードの3ホップチェック応答リストを受信しても $\text{unreached}(h)$ が空にならなければ、さらに同様の手順でクラスタヘッドを選び、クラスタヘッド要請を送信する。

こうして、 $\text{unreached}(h)$ が空になる、すなわち、すべての $\text{twoNbr}(h)$ のノードに対して3ホップ以内で通信可能になるまでこれらの処理を繰り返す。

■ 終了報告送信フェーズ

このフェーズでは、クラスタヘッド選出を完全に終えたクラスタヘッドが親ノードに対し終了報告を転送する。ネットワークの末端のクラスタヘッドからその親ノードに送信され、最終的に代表ノードまで終了報告が転送され、経路の構築が完了する。また、各ノードはいずれかのクラスタヘッドに参加要求を送信し、クラスタが構成される。

提案手法では、クラスタヘッド h が終了報告を送信するためには、以下の 2 つの条件を満たす必要がある。

- $\text{twoNbr}(h)$ に含まれるすべてのノードに他クラスタヘッドを経由して 3 ホップ以内に到達可能
- すべての子クラスタヘッドから終了報告を受信済み、または子クラスタヘッドが存在しない

2 つ目の条件は、 h を根とする部分木上に存在する他のすべてのクラスタヘッドが終了報告を送信済みであることを意味している。これにより、代表ノードがこの条件を満たした時に、経路構築の完了を知ることができる。

1. 末端ノードによる終了報告の送信

上記 2 つの条件を最初に満たすことができるのは、子クラスタヘッドを持たないネットワークの末端に位置するクラスタヘッドである。

子クラスタヘッドを持たないクラスタヘッド h は、クラスタヘッド広告をブロードキャストする時点で、 $\text{unreached}(h)$ が空になっていた場合、クラスタヘッド広告に終了報告を含めてブロードキャストする。

一方、3 ホップチェック応答メッセージを受信して $\text{unreached}(h)$ が空になった場合は、親ノードに対してユニキャストで終了報告を送信する。

2. 終了報告の送信

子クラスタヘッドを 1 つ以上持つクラスタヘッド h は、上記の 2 条件を満たした段階で終了報告を親ノードにユニキャストで送信する。

すなわち、 $\text{unreached}(h)$ が空になっていた場合、すべての子クラスタヘッドから終了報告を受信したときに終了報告を送信する。

一方、 $\text{unreached}(h)$ が空になる前にすべての子クラスタヘッドから終了報告を受信し終わった場合には、3 ホップチェック応答メッセージを受信し、 $\text{unreached}(h)$ が空になった時に終了報告を送信する。

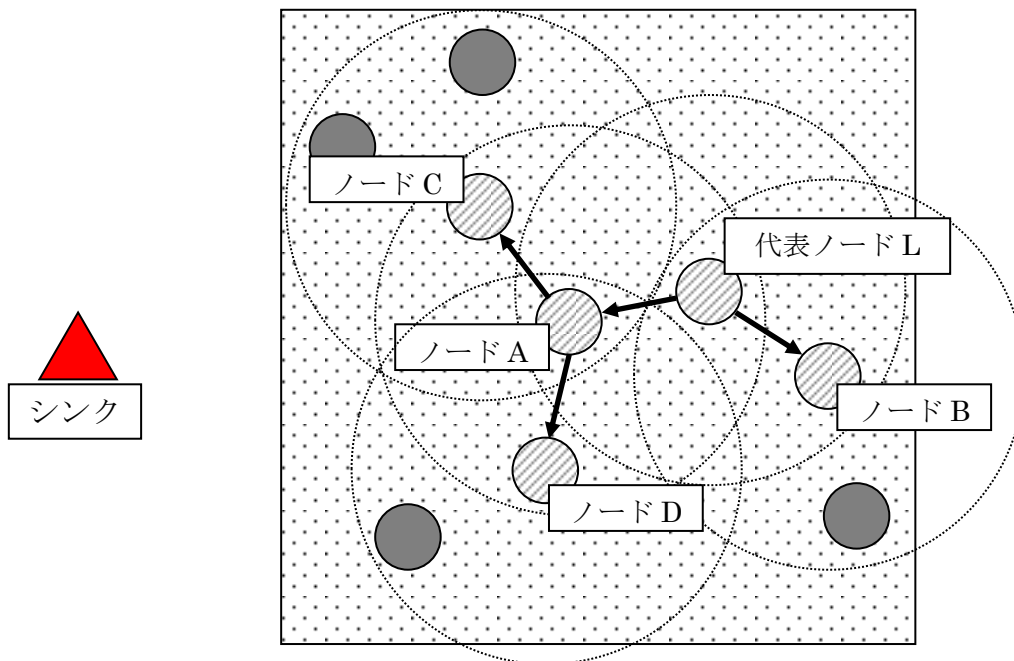


図 3.5 終了報告の送信

図 3.5 のネットワークでは、ノード B, C, D はクラスタヘッドになった時点で、未到達ノード集合は空になっている。このため、新たにクラスタヘッドを選出する必要もなく、終了報告を送信するための 2 つの条件を満たす。よって、これら 3 つのノードはクラスタヘッド広告に終了報告をのせてブロードキャストする。

一方、ノード A, 及び代表ノード L はそれぞれ子クラスタヘッドからの終了報告を受信しなければ、自身の終了報告を送信できない。

ノード A は、C, D からの終了報告を受信すると、L に自身の終了報告を送信する。

代表ノード L はノード B から終了報告を受信し、さらにノード A からの終了報告も受信すると、シンクに終了報告を送信する。

3. クラスタの構成

代表ノードが上記の条件を満たすと、シンクに終了報告をユニキャストで送信する。

シンクはこれを受信するとすべてのノードに対し、クラスタ所属要請メッセージをブロードキャストする。

クラスタヘッドでないノードは、このメッセージを受信すると、隣接するクラスタヘッドのリストの中から、クラスタヘッド広告の受信電波強度が最も大きかったノードを選び、クラスタ参加要求をユニキャストで送信する。

クラスタヘッドは、クラスタ参加要求を送信してきたノードをメンバに追加する。

ここまでの処理で、提案手法における経路及びクラスタの構成が完了する。

次の図 3.6 に提案手法の経路構築におけるメッセージシーケンス図を示す。

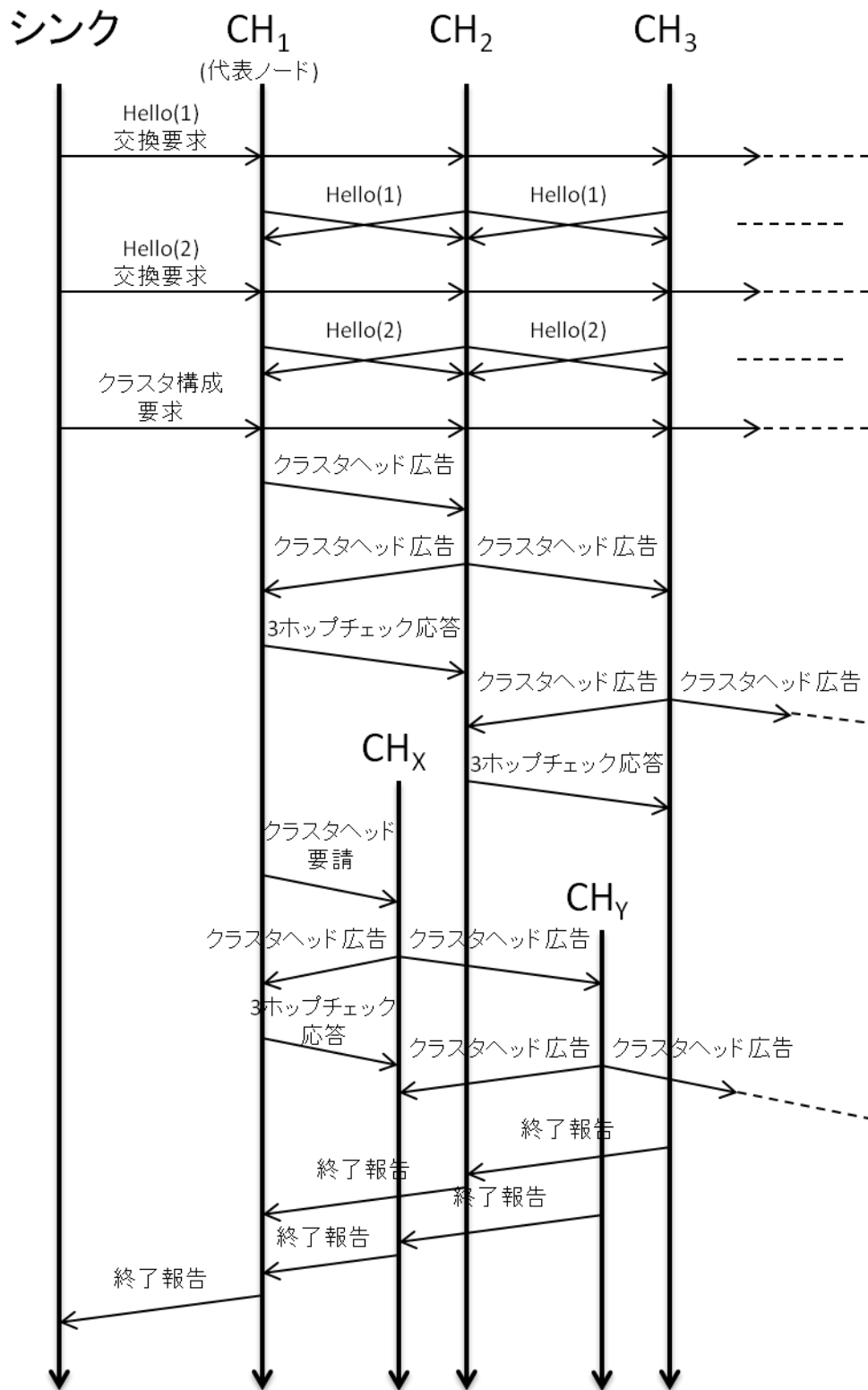


図 3.6 提案手法の経路構築におけるメッセージシーケンス図

3.2 ノード密度を考慮した動的スリープ制御

提案手法では、クラスタヘッドにならなかったノードの一部をスリープさせることにより、ノードの電力を温存させ、ネットワーク全体の寿命の向上を図った。また、周辺ノードの密度が高いノードをスリープ対象に選ぶことにより、スリープによって領域カバー率を損なってしまう問題の抑制を図った。さらに、提案するスリープ制御方式では、ノードの密度に応じてスリープするノードの数が増減するという特徴を持つ。

3.2.1 提案するスリープ制御方式のアルゴリズム

クラスタヘッドでないノード n は、クラスタ参加要求を送信後、このラウンドをスリープするかどうかの判定を行う。

まず、ノード n は、 $\text{nbr}(n)$ に含まれるノードのうち、 n との推定距離がセンシング半径以下であるノードの数 t_n を調べる。 n との推定距離がセンシング半径以下であるかを判定することにより、そのノードが n のセンシング範囲内に存在しているかを判断することができる。この時点で、 t_n がしきい値 $k(k>0)$ を下回っていた場合、ノード n はスリープには移行しない。

t_n がしきい値 k 以上であった場合は、自身を含む n のセンシング範囲内のノードについて、最も電力残量が小さいノードから t_n/k 個までのノードを調べる。

この中に n が含まれていると、 n はこのラウンドをスリープする。

つまり、ノード n がスリープに移行するためには、以下の2つの条件を満たす必要がある。

- t_n がしきい値 k 以上
- n の電力残量がセンシング範囲内のノードの中で小さい順で t_n/k 位以内

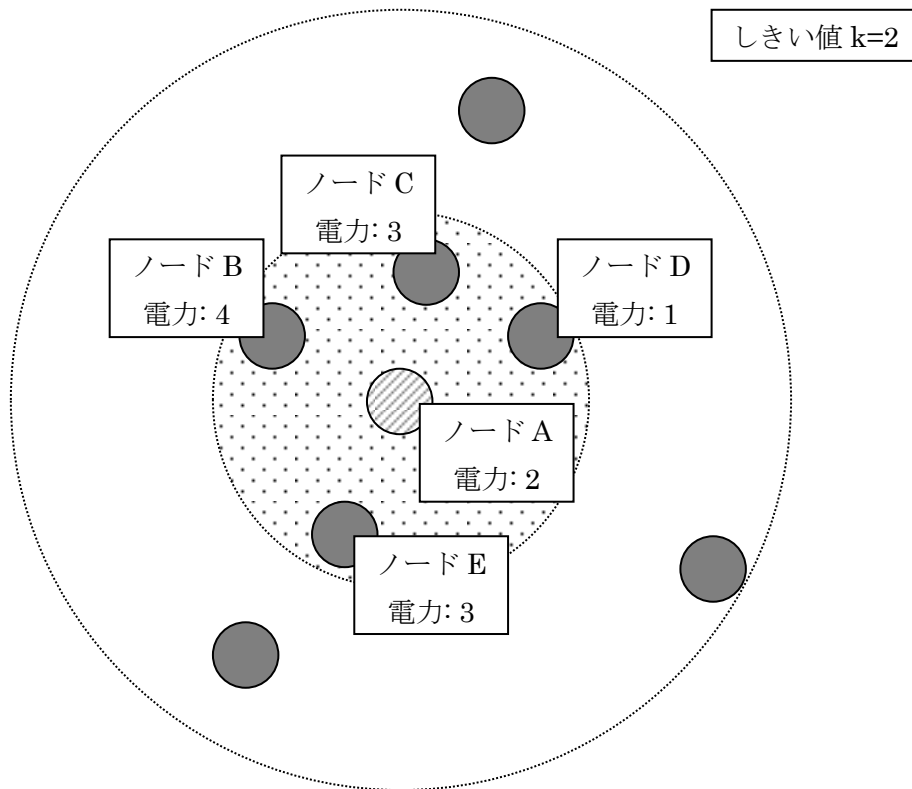


図 3.7 提案手法におけるノードのスリープ移行判定

図 3.7 において、ノード A がスリープ移行判定を行う様子を解説する。

まず、A は $\text{nbr}(A)$ の中で A との距離がセンシング半径以下であるノードの数 t_A を調べる。この条件を満たすノードは、ノード B、ノード C、ノード D、ノード E の 4 つなので、 t_A は 4 となる。ここで、この例におけるしきい値 k は 2 なので、 $t_A > k$ となり、スリープに移行するための 1 つ目の条件はクリアとなる。

次に、A はセンシング範囲内のノードについて、電力残量が小さい順に t_A/k 位までの順位を調べる。ここで、 $t_A=4$ 、 $k=2$ なので、 $t_A/k=2$ である。よって、ノード A は電力残量の小さい順に 2 位までのノードを調べる。すると、最も電力残量が小さいのはノード D、2 番目はノード A となる。

よって、スリープに移行するための 2 つ目の条件も満たしているため、ノード A はスリープに移行する。

第4章 評価

4.1 シミュレーション概要

提案手法の評価は、Java 言語により提案手法及び、LEACH, HEED, HIT, MR-LEACH のシミュレーションプログラムを作成し、シミュレーションを行い、結果を比較することで行った。

シミュレーションは観測領域上にノードを一定個数ランダムに配置し、すべてのノードが電力を使い果たすまで行った。

比較は、サイクル数ごとの生存ノード数の推移、サイクル数ごとの領域カバー率の推移、ノードの全電力消費量における制御メッセージの割合についてそれぞれ行った。

また、提案手法のスリープ制御の有無による性能比較も行い、提案手法におけるスリープ制御の有効性を検証した。

4.1.1 シミュレーション想定環境

シミュレーションの想定環境として、観測領域は $100\text{m} \times 100\text{m}$ とし、シンクは観測領域の中心から西に 75m の位置に配置する。すなわち、想定環境では、シンクとノードは最低でも 25m は離れていることになる。ノードは観測領域上のランダムな位置に配置し、移動はしないものとする。

図 4.1 に想定するシミュレーション環境の概略図を示す。

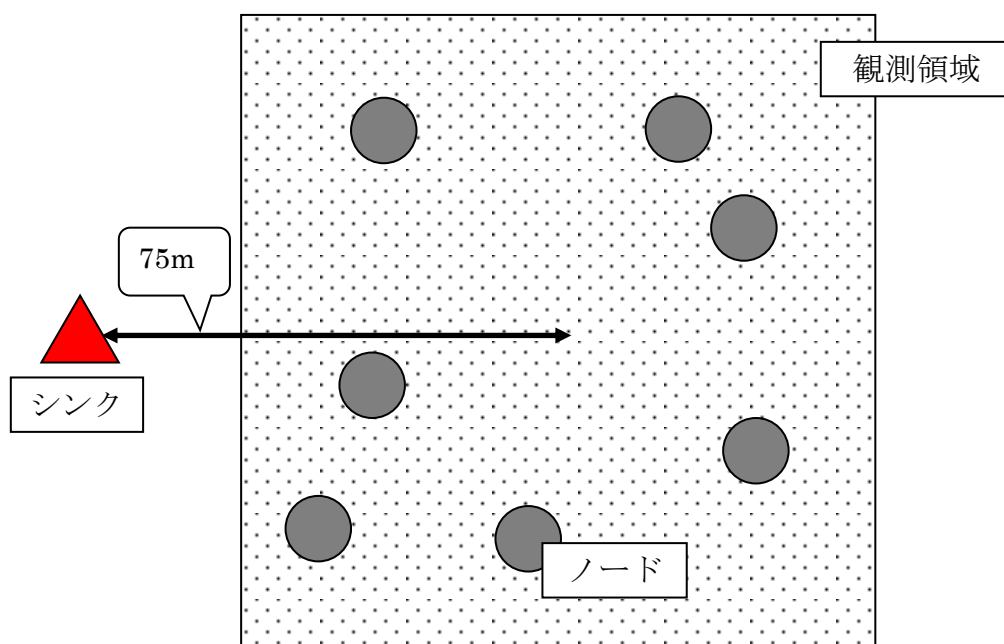


図 4.1 シミュレーション想定環境

4.1.2 電力モデル

無線通信による電力の消費モデルは, [1]と同じ物を用いた. k bits のメッセージを 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 のメッセージを受信するときの消費電力 E_R は式(4.2)でそれぞれ与えられる.

$$E_T = E_{\text{elec}}k + \epsilon_{\text{amp}}kd^2 \quad (4.1)$$

$$E_R = E_{\text{elec}}k \quad (4.1)$$

ここで, E_{elec} は 1bit 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.

4.1.3 シミュレーションパラメータ

ここでは、今回のシミュレーションに用いたパラメータについて述べる。まず、シミュレーションに用いる共通パラメータを表 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
初期電力	4 J
制御メッセージ	500 bits
データサイズ	2000 bits

また、ノード数は 50, 100, 200 とし、それぞれのノード数で比較を行った。

■LEACH のシミュレーションパラメータ

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 50, 100, 200 それぞれの場合について各パラメータを変化させてシミュレーションを行い、最適値を求めた。表 4.2 に LEACH のシミュレーションパラメータを示す。

表 4.2 LEACH のシミュレーションパラメータ

パラメータ	ノード 50	ノード 100	ノード 200
1 ラウンドの サイクル数	100	100	100
基本確率 P	0.1	0.075	0.05

■HEED のシミュレーションパラメータ

HEED 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、ノードがクラスタヘッドに立候補する基本確率 C_{prob} 、及び CH_{prob} の最小値 P_{min} をそれぞれ定める必要がある。LEACH 同様のシミュレーションを行い、最適値を求めた。次の表 4.3 に HEED のシミュレーションパラメータを示す。

表 4.3 HEED のシミュレーションパラメータ

パラメータ	ノード 50	ノード 100	ノード 200
1 ラウンドの サイクル数	100	100	50
基本確率 C_{prob}	0.025	0.025	0.025
CH_{prob} の最小値 P_{min}	0.0025	0.0025	0.0025

■HIT のシミュレーションパラメータ

HIT 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を定める必要がある。LEACH, HEED 同様のシミュレーションを行い、その結果に従い定めた HIT のシミュレーションパラメータを表 4.4 に示す。

表 4.4 HIT のシミュレーションパラメータ

パラメータ	ノード 50	ノード 100	ノード 200
1 ラウンドの サイクル数	700	1000	1000

■MR-LEACH のシミュレーションパラメータ

MR-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、電波到達範囲 R 、及び R' をそれぞれ定める必要がある。シミュレーションの結果に従い、以下の値に決定した。

表 4.5 MR-LEACH のシミュレーションパラメータ

パラメータ	ノード 50	ノード 100	ノード 200
1 ラウンドの サイクル数	500	400	450
電波到達範囲 R	60	60	70
電波到達範囲 R'	80	90	90

■提案手法のシミュレーションパラメータ

提案手法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、2 ホップカバー率のしきい値、電力の重み W 、及びスリープに必要な周囲ノード数のしきい値 k をそれぞれ定める必要がある。従来手法同様にシミュレーションを行い、シミュレーションパラメータを決定した。

表 4.6 提案手法のシミュレーションパラメータ

パラメータ	ノード 50	ノード 100	ノード 200
1 ラウンドの サイクル数	1000	1000	1000
2 ホップカバー率 のしきい値	0.5	0.5	0.5
電力の重み W	3	3	2
周辺ノードの しきい値 k	2	2	2

4.2 サイクル数ごとの生存ノード数の推移

ノード数 50, 100, 200 それぞれの場合のサイクル数ごとのノード数の推移のグラフをそれぞれ, 図 4.2, 図 4.3, 図 4.4 に示す.

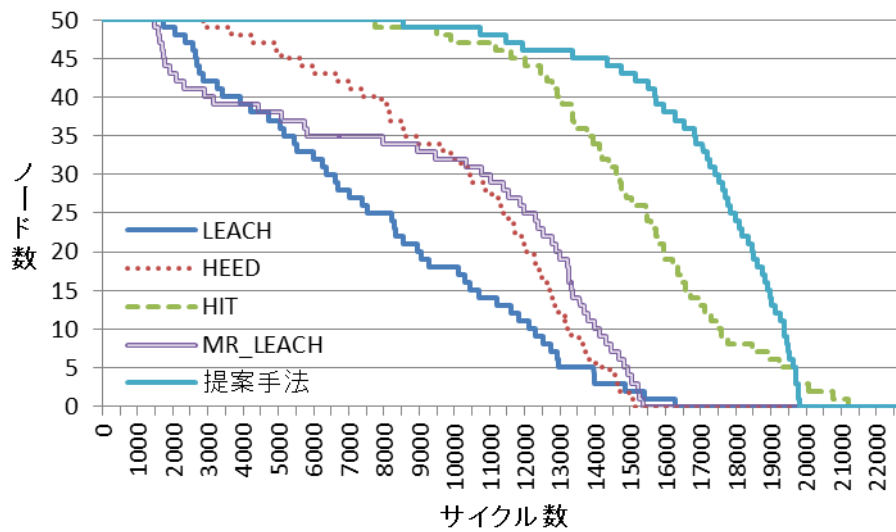


図 4.2 サイクル数ごとの生存ノード数の推移 (ノード 50)

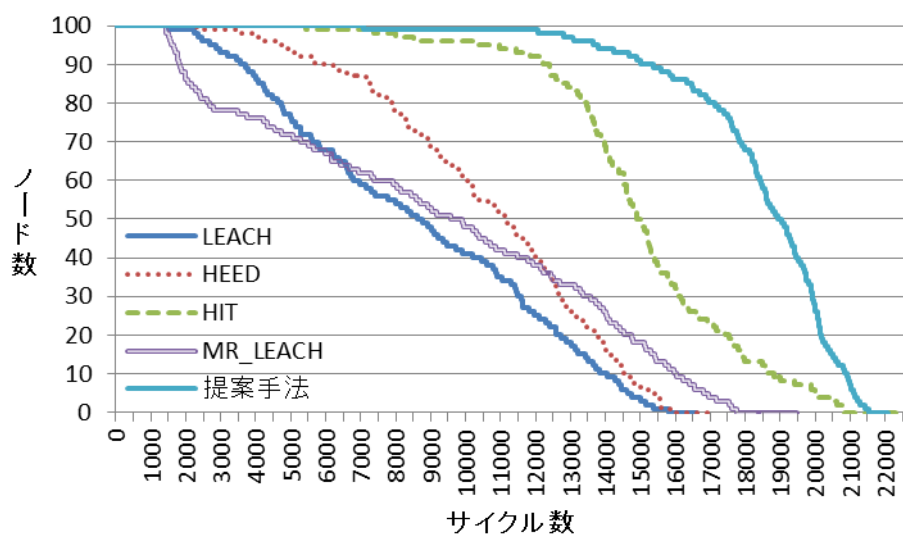


図 4.3 サイクル数ごとの生存ノード数の推移（ノード 100）

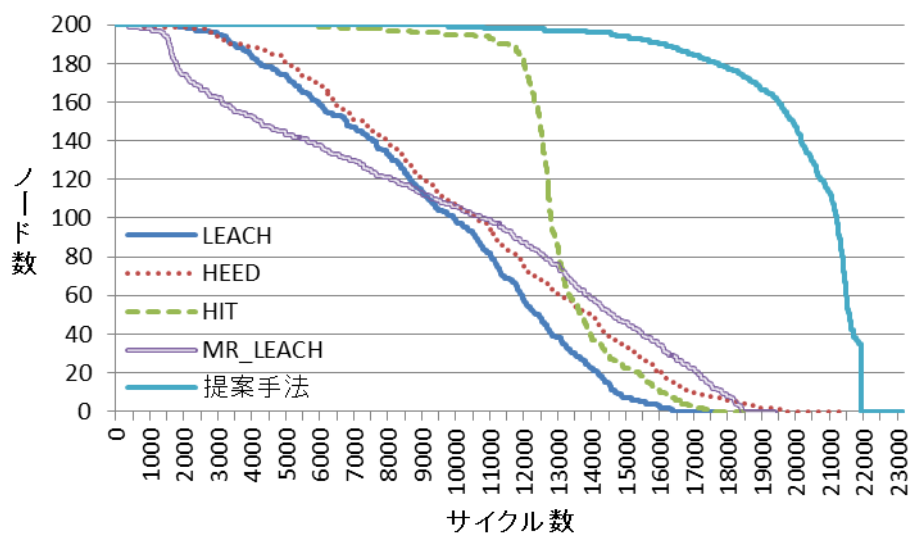


図 4.4 サイクル数ごとの生存ノード数の推移（ノード 200）

また, 図 4.5 にそれぞれのノード数ごとのノードの平均生存時間のグラフ示す.

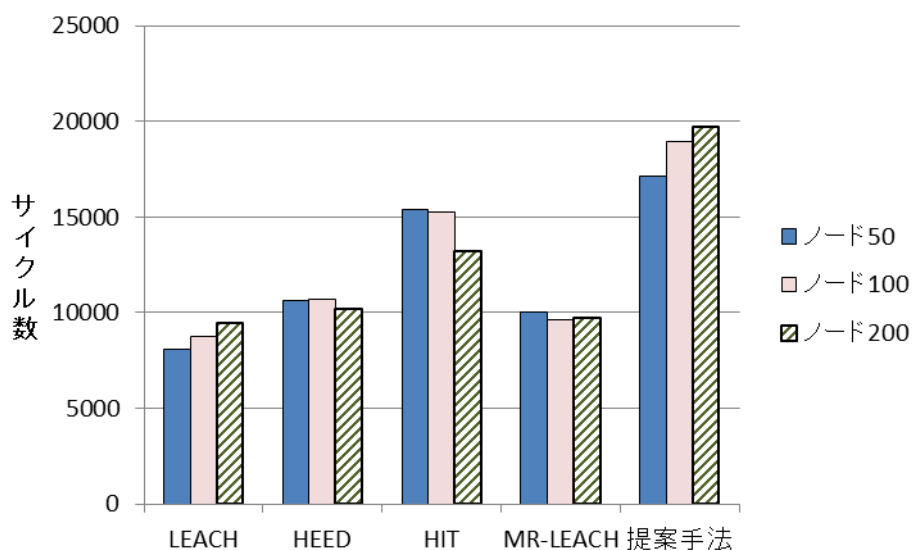


図 4.5 ノードの平均生存時間

これらの結果を見ると、提案手法ではいずれのノード数の場合でも従来手法よりノードの平均生存時間が伸びていることがわかる。ノード 50 では従来手法との差も小さいが、提案手法ではノード数が増えると、パフォーマンスが大幅に増加していることがわかる。これは、ノード数が増えると、ノード密度が大きい領域が増え、スリープできるノードが増えたため、パフォーマンスの向上につながったものと考えられる。また、提案手法ではノード数 200 のときに、生存ノード数 190 程度までは緩やかに減少し、その後一気にノードが減っていることがわかる。これは、提案手法ではより各ノードの電力をバランスよく消費しているためと考えられる。

これらの結果から、ノードの生存時間という観点で見ると、提案手法はノード数が多い場合ほど有効であるといえる。

4.3 サイクル数ごとの領域カバー率の推移

ここで、本稿における領域カバー率は、観測領域の $1\text{m} \times 1\text{m}$ の格子状にチェックポイントを設け、領域全体のチェックポイントのうち、有効なノードのセンシング範囲に含まれているチェックポイントの割合と定義する。

図 4.6, 図 4.7, 図 4.8 にそれぞれノード数 50, 100, 200 のときのサイクル数ごとの領域カバー率の推移のグラフを示す。

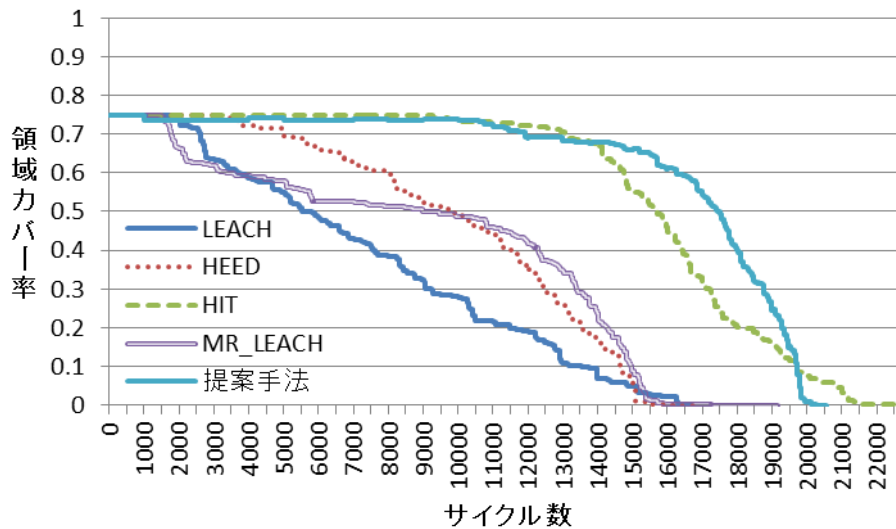


図 4.6 サイクル数ごとの領域カバー率の推移 (ノード 50)

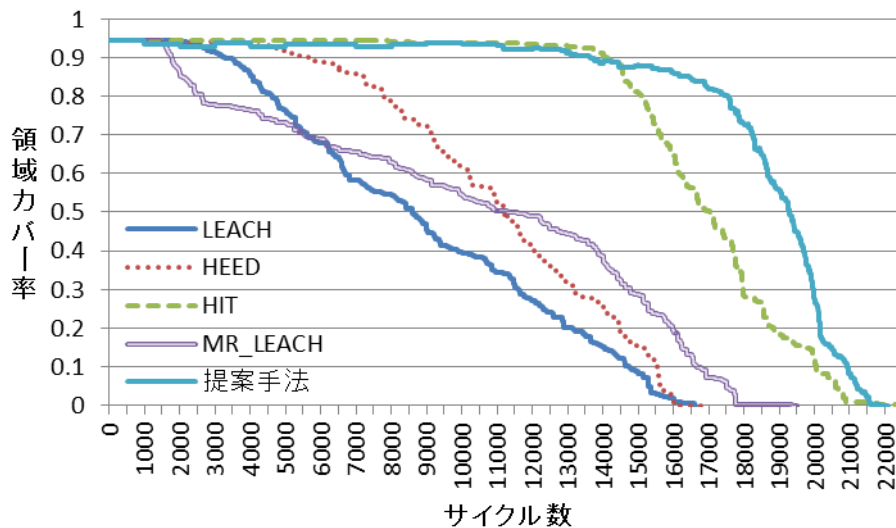


図 4.7 サイクル数ごとの領域カバー率の推移 (ノード 100)

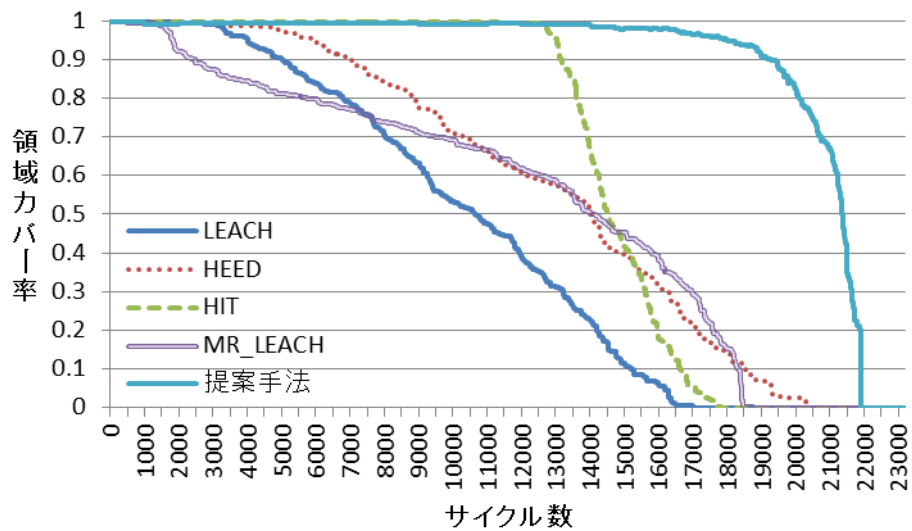


図 4.8 サイクル数ごとの領域カバー率の推移（ノード 200）

これらの結果を見ると、提案手法ではスリープによる領域カバー率の低下が見られるものの、おおよそ 5%以下程度には収まっていることがわかる。生存ノード数の推移と同様に、ノード数が増えるほどパフォーマンスが向上している。特にノード 200 のときには、領域カバー率はほとんど損なっていない。

よって、領域カバー率の観点からも提案手法はノード数が多い場合において特に有効であるといえる。

4.4 ノードの全消費電力に対する制御メッセージの割合

図 4.9 にそれぞれのノード数でのノードの全消費電力に対する制御メッセージの割合のグラフを示す。

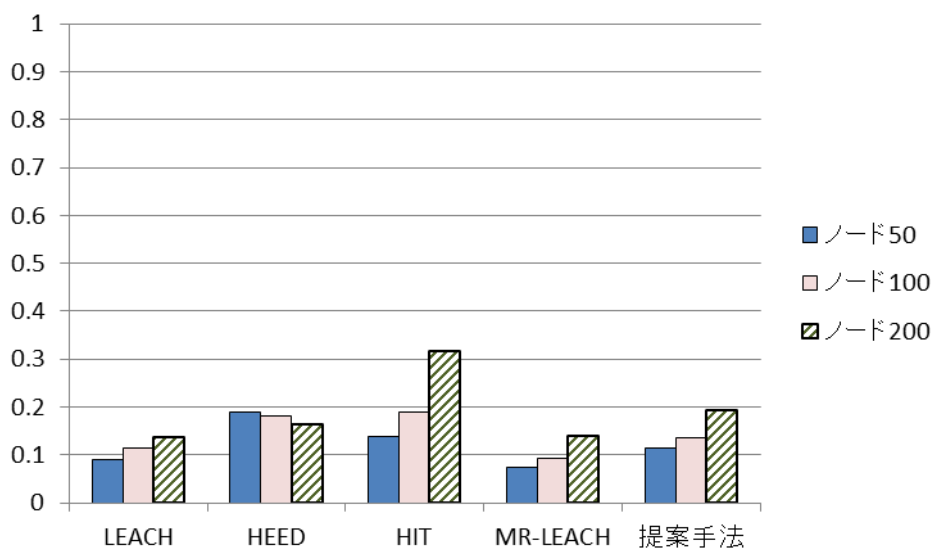


図 4.9 ノードの全消費電力の対する制御メッセージの割合

提案手法は、従来手法よりも経路構築に要する制御メッセージの送信回数は多いが、従来手法と比べ経路の再構築回数が少なく済むため、全体としての制御メッセージのオーバーヘッドは従来手法とあまり変わらなかったものと考えられる。

4.5 スリープ制御の有無による提案手法の性能比較

提案手法におけるスリープ制御が、ノードの平均生存時間、及び領域カバー率にどの程度影響をおよぼすのか比較を行った。

図 4.10 にノードの平均生存時間のグラフを示す。

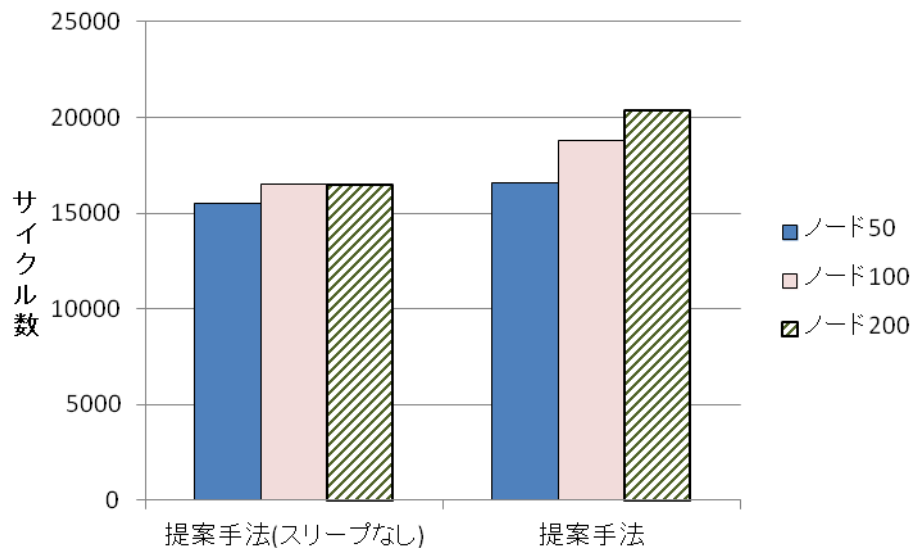


図 4.10 スリープの有無による提案手法のノード平均生存時間

このグラフを見ると、スリープの導入により、提案手法の各ノードの平均生存時間が伸びていることがわかる。

また、ノード 50, 100, 200 それぞれの場合の領域カバー率の推移のグラフを図 4.11, 4.12, 4.13 に示す。

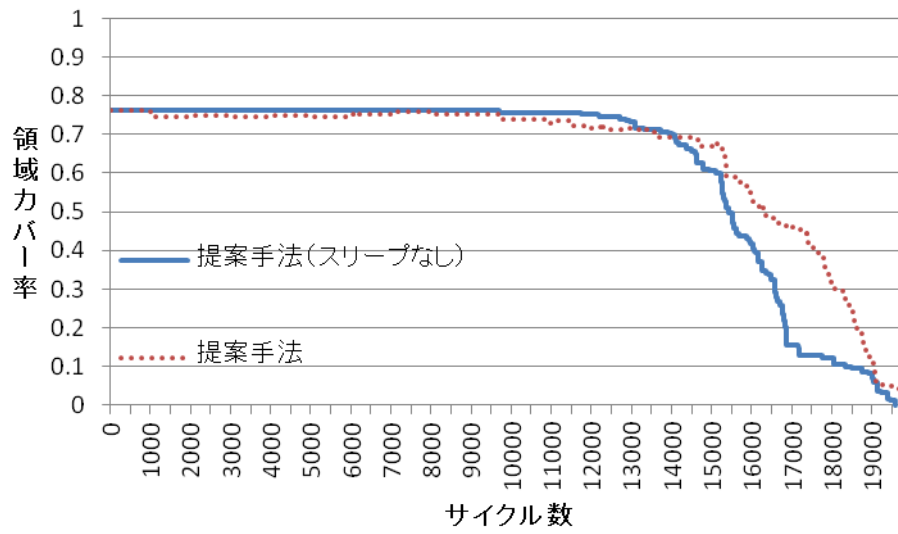


図 4.11 スリープの有無による領域カバー率の比較 (ノード 50)

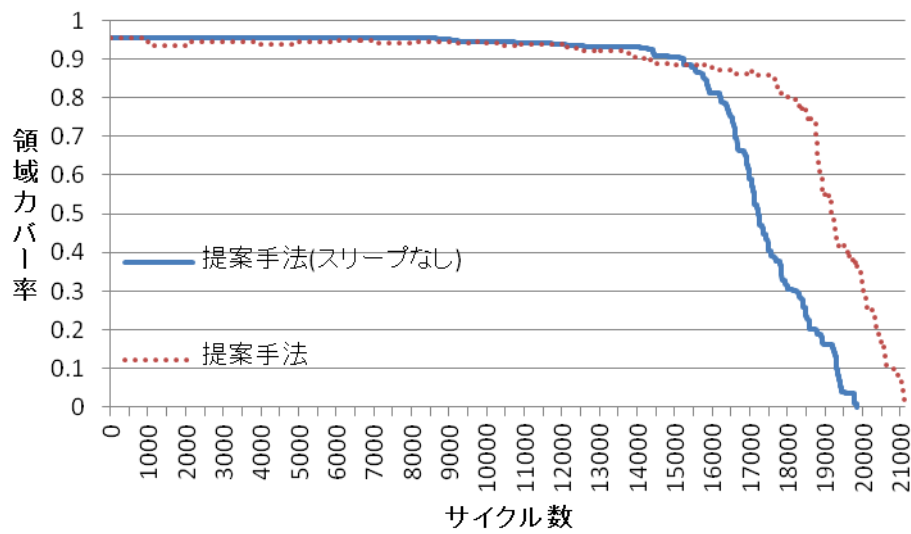


図 4.12 スリープの有無による領域カバー率の比較 (ノード 100)

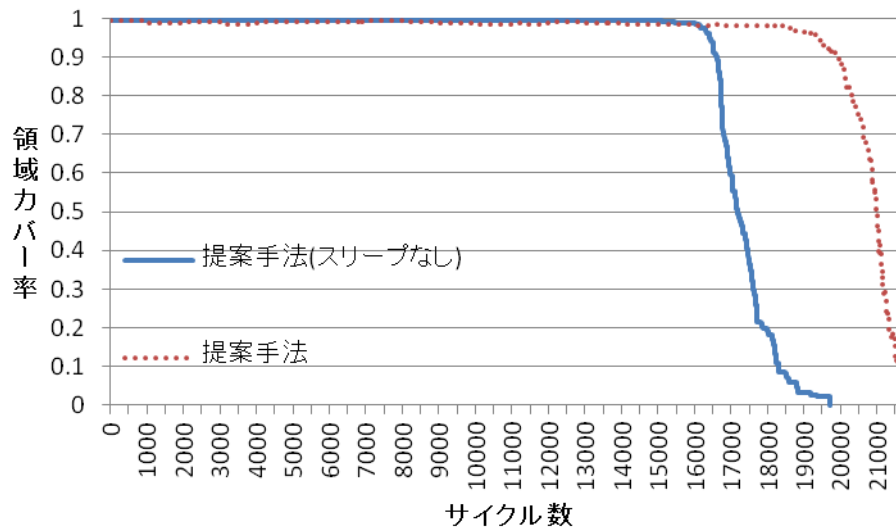


図 4.13 スリープの有無による領域カバー率の比較（ノード 200）

これらの結果を見ると、スリープを導入しても領域カバー率はほとんど損なっていないことが分かる。よって、提案手法ではスリープを導入することにより、領域カバー率を大きく損なうことなく、ノードの寿命をのばすことができたと言える。

第5章 結論

本稿では、無線センサネットワークにおけるノードの隣接ノード集合と電力残量を考慮した省電力クラスタリング方式、並びに領域カバー率を損なわない動的スリープ制御方式を提案した。無線センサネットワークでは、ネットワークを構成するノードの電力が有限であり、長期間の運用には省電力化が不可欠である。無線センサネットワークでは、ノードは通信に費やす電力が大きい。このため、省電力を考慮したルーティング方式を利用することはネットワークの長寿命化には非常に有用であり、様々な方式が提案されている。

LEACH は無線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であり、クラスタリングによってノードの負荷を分散している。**HEED** は **LEACH** を改良したアルゴリズムであり、ノードの消費電力や周囲のクラスタヘッドの立候補状況の情報を用いることで、**LEACH** で発生しやすいノード間の電力の偏りを解消している。**HIT** はクラスタ内部でマルチホップ通信を行うプロトコルであり、各ノードの送信距離を最小化し、スケジューリングにより電波干渉を回避しつつ、同時通信数を最大化するという特徴を持つ。

提案手法では、クラスタヘッドを選出する際に、ノードの隣接ノードに占める未到達ノードの数、及びノードの電力残量を同時に評価することにより、効率的、かつ電力にも配慮した経路の構築を可能にした。さらに、クラスタヘッドに選ばれなかったノードの一部をノードの密度に応じてスリープさせることにより、領域カバー率を損なわずにノードの生存時間のさらなる延長を図った。

そして、本稿ではシミュレーションにより提案手法と **LEACH**, **HEED**, **HIT**, **MR-LEACH** を比較し、提案手法の性能評価を行った。その結果、提案手法を用いることにより、従来手法よりも高いパフォーマンスを得られ、スリープによる領域カバー率の低下もほとんど見られないことがわかった。このことから、提案するルーティング手法、及び動的スリープ制御手法が無線センサネットワークにおいて有用であることを示した。また、ノード数が多い場合ほど、ノードの平均生存時間、領域カバー率ともによりよいパフォーマンスを得られることがわかった。

最後に今後の課題を述べる。まず、今回のシミュレーションでは **MAC** 層以下は深く作りこまなかったため、電波の干渉などの問題は考慮されていない。また、電力の消費は通信のみで発生させていたが、待機電力などその他の要因による電力消費も考慮する必要がある。

これらの課題に取り組むためにも、今後は **QualNet** などのシミュレータを用いてより現実に近い環境でのシミュレーションを行うことが必要である。

謝辞

本研究を進めるにあたって，研究全般にわたり適切なご指導ならびに御助言をいただきました東邦大学理学部情報科学科佐藤文明教授に厚く御礼申し上げます．最後に多大なるご協力をいただいた佐藤研究室のメンバーに，心から感謝の意を申し上げます．

参考文献

- [1] W. R. Heinzelman, A. Chandrakasan and H. Balakrishnan: “Energy-Efficient Communication Protocol for Wireless Microsensor Networks”, Proceedings of the 33rd Hawaii International Conference on System Sciences, pp. 1-10, Jan. 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, Oct. 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, Nov. 2004
- [4] M. O. Farooq, A. B. Dogar and G. A. Shah: “MR-LEACH: Multi-hop Routing with Low Energy Adaptive Clustering Hierarchy”, Fourth International Conference on Sensor Technologies and Applications, pp.262-268, 2010
- [5] 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, Sep. 2002
- [6] 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, Aug. 2004
- [7] 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, Jul. 2003