

受信ノード主導型 MAC プロトコルを利用した 管理コストが低い長寿命な無線センサネットワーク

小島 祥平^{1,a)} 吉廣 卓哉^{†1,b)}

概要: 近年, センサネットワークを実用化するために, IEEE802.15.4 等の通信方式が開発されている. これらの通信規格では, 低消費電力な動作により, バッテリ駆動のノードでも長期間にわたり動作が可能になるように設計されている. 一方, センサネットワークが社会を支えるネットワークとして動作するためには, 10 年以上の安定した運用を実現したいが, 現在の通信規格では 10 年の長い動作寿命の実現は難しい. このため, 定期的にバッテリーを交換する等の管理コストが必要になり, さらなる低消費電力化により, 管理コストの低減が望まれる. 本研究では, 受信ノード主導型の低消費電力な MAC プロトコルである RC-MAC を拡張することで, 10 年以上の長期運用を比較的低い管理コストで実現するセンサネットワーク実現技術を提案する. 提案手法では, データ配送木の葉ノードは受信処理が不要であることを利用して, 葉ノードの寿命を 10 年以上に延ばす RC-MAC の拡張を提案する. また, ブロードキャストをサポートしない RC-MAC でも効率良く配送木を構築するプロトコルを提案し, バッテリ交換を要する中継ノード数ができるだけ少ない配送木を自律的に構築する. これらの技術により, バッテリ交換コストをできるだけ低減した長期運用に耐える現実的な無線センサネットワークの実現を目指す.

1. はじめに

近年, 無線通信技術の高性能化や CPU の小型化また低価格化が進んでいる. これにより無線通信機能を持った小型センサを用いてネットワークを構成して, 各センサによって得られたデータを無線マルチホップ通信により収集する WSN (Wireless Sensor Network) の研究が進められている. WSN では, 他の無線マルチホップネットワークにはない特徴がある. まず, ネットワークの通信スループットは低くても良い一方で, データ収集の信頼性は向上したい要求がある. また, 各センサノードは必ずしも電源に接続できず, 多くの場合にバッテリー駆動が前提となる. センサノードをバッテリー駆動にすることで, センサの設置場所を選ばないセンサの配置が可能となるが, その一方で, 通信あるいはデータ処理に要する消費電力を低く抑えた長寿命なネットワークの実現が求められる.

無線センサネットワークを実用化するために, 近年では, IEEE802.15.4[19] 等の通信方式が開発されてきた. これらの通信規格では, 低消費電力な動作により, バッテリ駆動

のノードでも長期間にわたり動作が可能になるように設計されている. しかし, センサネットワークが社会を支えるネットワークとして動作するためには, 10 年以上の安定した継続運用が望ましく, 一方で現在の通信規格では, 一般的な乾電池やボタン電池による 10 年の長い動作寿命の実現は難しい. このため, 長期の運用を考えると, 定期的にバッテリーを交換する等の管理コストが必要であり, さらなる低消費電力化により, 管理コストの低減が望まれる.

実用的な無線センサネットワークの実現に向けた省電力なセンサネットワーク技術の研究は数多く行われている. 最も有効な方法の一つに, MAC プロトコルの省電力化がある [1]-[7]. これらは, 低いスループットで良いセンサネットワークにおいて, 受信可能状態で待機する代わりに定期的にスリープすることで, 待機電力を削減してセンサノードの消費電力を削減する. これらの MAC プロトコルには, 送信ノード主導型と受信ノード主導型の 2 種類があり, 一般には, RC-MAC[6] 等の受信ノード主導型 MAC プロトコルの方が消費電力の点では優れている. しかし, これらはブロードキャスト送信をサポートしないため, シンクノードまでデータを送信するための配送木 (配送経路) を構築するためには工夫が必要である.

そこで本研究では, RC-MAC を拡張した新たな低消費電力 MAC プロトコルと, この上位層で自律的に配送木を

¹ 和歌山大学大学院システム工学研究科
和歌山市栄谷 930 番地

^{†1} 現在, 和歌山大学システム工学部
Presently with 和歌山市栄谷 930 番地

a) s151020@sys.wakayama-u.ac.jp

b) tac@sys.wakayama-u.ac.jp

構築する経路制御プロトコルを提案することで、低い管理コストで長期運用可能な現実的なセンサネットワークの実現を目指す。具体的には、提案手法では、RC-MAC を拡張して動作状態を導入することで、効率的な配送木の構築と、低消費電力なデータ送信を両立する。「配送木構築状態」ではブロードキャストを可能にすることで効率的に配送木を構築し、「定常状態」では RC-MAC に基づいた消費電力の低い効率的なデータ集約を実現する。定常状態においては、葉ノードはデータフレームの受信が不要であることから、さらに消費電力を抑えて 10 年以上の長いノード寿命を実現する。複数の状態をシームレスに遷移させることで、RC-MAC による効率的なデータ通信性能を維持したままで、ノード故障などによるトポロジ変化時にも自律的に配送木を再構築できる。また、配送木の構築においては、バッテリー交換が必要な中継ノード数を低く抑えることで、管理コストの低いセンサネットワークを実現する。提案手法により、10 年以上にわたって、比較的少数のバッテリーを年に 1 度程度の頻度で交換することで維持できる、管理コストの低いセンサネットワークが実現できる。

本論文は、以下のように構成される。第 2 章では、関連研究について述べる。第 3 章では、提案手法における MAC プロトコルとして、RC-MAC の拡張について述べる。第 4 章では、配送木を構築する経路制御プロトコルについて述べる。第 5 章では提案手法の消費電力及び配送木の評価を行う。第 6 章で提案手法の実用化にあたっての課題について議論した後、第 7 章で本論文をまとめる。

2. 関連研究

本研究では、管理コスト最小化の観点から、ネットワーク全体の消費電力を低く抑えると同時に、定期的なバッテリー交換が必要な中継ノードの数を最小化するようなデータ配送木を自律的に構築し、維持するようなセンサネットワークの実現を目指している。有力な消費電力の低減手法として、センサネットワークを対象とした MAC 層プロトコルが活発に研究されている [1]-[7]。これらは、ノードを定期的にスリープさせながら、タイミングを合わせて隣接ノード間のデータ通信を行うことで、通信データ量に見合った低い消費電力で信頼性のある通信を実現する。S-MAC[1]、T-MAC[2]、B-MAC[3]、X-MAC[4] は、送信ノード主導の MAC プロトコルである。例えば B-MAC は、センサノードが一定時間毎にスリープとウェイクを繰り返し、送信時にはその時間間隔よりも長いプリアンプルを送ることでデータの受信を可能にする。これに対して、RI-MAC[5]、RC-MAC[6] は、受信ノード主導の MAC 層プロトコルである。受信ノードが定期的にビーコンを送信し、送信したいフレームを持つノード(送信ノード)は、ビーコンを受信した時に受信ノードにフレームを送信することで、受信待受電力を低く抑えた低消費電力なネットワークを実現す

る。しかし、受信ノード主導の MAC プロトコルはブロードキャストをサポートしておらず、これを用いて自律分散的に配送木を構築する方法は提案されていない。なお、無線センサネットワークの MAC 層プロトコルについては、文献 [7] のサーベイを参考にされたい。

センサノードが環境から電力を得る(充電する)技術も存在し、太陽光パネルだけでなく、人や車、電波等による振動を電力に変えることで、センサノードの長寿命化を実現できる可能性がある [8][9][10]。しかし、現時点では、それらの技術によって得られる電力は僅かであり、また、天候等の環境要因に左右されるため、生成される電力に頼ってセンサネットワークを長期的に維持することは困難である。太陽光のみを用いてバッテリーレスなセンサネットワークを構築する試みも現れているが、その通信能力や安定性は限定的であり、用途が大きく限定される [11]。

一方、経路制御に関しては、センサネットワークに適した配送木を構築する手法が多数提案されている。Luo らは、複数存在する最短路木の中から、センサネットワークの生存時間を最大化するような配送木を構築するアルゴリズムを提案した [12]。Kuo らは、屋内の電源が接続されたセンサネットワーク等を対象として、ネットワーク全体の消費電力を最小化するような配送木を構築するアルゴリズムを提案した [13]。しかし、これらは最適化問題を定式化して解く集中制御に基づいた方法を提案しており、故障等によるトポロジの変化に対応して自律分散的に配送木を計算することはできない。また、提案手法は、管理コストに着目し、中継ノードの数を最小化することを目的としており、既存の配送木の構築法とは最適化指標が異なる。

Gnawali らは、ノードの故障等によるトポロジの変化に柔軟に対応し、パケットループを防ぐ動的な経路制御プロトコル CTP を提案した [14]。この手法は、センサネットワークを対象として、動的なトポロジ変更に対応できる経路制御法を提供する点で貢献がある。しかし、本研究が目指すような管理コストの低減を考慮していないだけでなく、この手法はブロードキャスト可能な MAC プロトコルを想定しており、RC-MAC のような受信ノード主導型の MAC プロトコルを用いる場合には適用が難しい。本研究で目指すような、受信ノード主導の MAC プロトコルを想定した自律的な配送木の構築手法は、これまでには見られない。

3. 提案手法 : MAC プロトコル

3.1 想定する無線センサネットワーク

本研究では、無線通信機能と何らかのセンサを搭載した多数のセンサノードから構成されるセンサネットワークを想定する。各センサノードは一定時間毎にセンシングにより値を生成し、ネットワーク内に複数存在するシンクノードのいずれかにマルチホップ通信を用いて転送する。ノー

ドの送信電力は一定であり、シンクノードまで複数の経路でデータを届けられる程度にはセンサが密に配置されていると仮定する。複数のシンクノードが存在しても良く、提案手法はその状況に対応しているが、以後は簡単のため、シンクノードが1つの場合を例に説明を進める。

3.2 提案手法の概要

提案手法は、RC-MAC を拡張し、状態遷移を導入することにより、効率的な配送木の構築と消費電力を抑えたデータ集約を兼ね備えた無線センサネットワークを実現する。RC-MAC を選択する理由としては以下が挙げられる

- 受信ノード主導のためデータ受信を行わない葉ノードは動作を省くことが可能であり、電力消費を抑えることができる
 - 受信ノードが送信ノードの送信順序を管理できるため複数の送信ノードが、同じ受信ノードへ同時に送信を行うことで発生する、電波の干渉を防ぐことができる
- このように、RC-MAC を用いることで、葉ノードの電力消費を抑えつつ、電波の干渉を防ぐことができる。

各ノードは、効率的に配送木を構築する「配送木構築状態」と、配送木に沿って低い消費電力でデータを集約する「定常状態」の2状態を切り替えて動作する。配送木構築状態では、RC-MAC を拡張してブロードキャスト通信を実現し、4章で述べる経路制御プロトコルを用いて配送木を効率的に構築する。定常状態では、RC-MAC の特徴を生かした消費電力の低い MAC プロトコルを動作させる。特に、配送木の葉ノードではセンシングデータの受信が不要であることを利用して、通常の RC-MAC よりもさらに消費電力を低減した MAC プロトコルを実現する。

各ノードは、ネットワークの構築時だけでなく、ノード故障等によるトポロジの変化時にも、配送木構築状態になり配送木を再構築する。あるノードがトポロジの変化を検出すると、これを隣接ノードに順に伝えることで、ネットワーク上の全てのノードが配送木構築状態になり、協調して配送木を再構築したあと、配送木が収束すると定常状態に遷移する。このような動作の中では、ネットワーク上に異なる状態のノードが混在する時間帯があるが、このときにも RC-MAC による効率的な通信が維持されることが肝要である。提案プロトコルでは、いずれの状態でも RC-MAC のデータ通信方式をサポートすることで、状態が異なるノードが混在する場合にも、データ集約の効率が維持されるようなプロトコル設計とした。

3.3 RC-MAC

提案 MAC プロトコルは、受信ノード主導型 MAC プロトコルの1つである RC-MAC の拡張として設計する。本節では、設計の基礎となった RC-MAC の動作を述べる。

RC-MAC[6] では、各ノード n_r が定期的にビーコンをブ

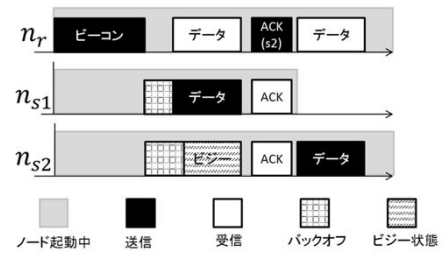


図1 RC-MAC

ロードキャストし、これを受信したノード n_s が n_r への送信フレームを持っていれば、 n_r へのフレーム送信を行う。各ノードは定期的にウェイクとスリープを繰り返し、ウェイクしたタイミングでビーコンを送信する。しばらく受信を待ち受け、フレームが受信されなければ、再びスリープする。受信待受時間を削減できるため、送信データ量が少ない場合には大きな消費電力の削減を見込める。また、競合を受信ノードが中心となって解決するため、隠れ端末問題も効率良く解決できる。

図1の例を用いて、動作を説明する。受信ノード n_r がビーコンを送信すると、全ての送信ノードは、CW (Contention Window) の範囲内でランダムに決めたバックオフ時間だけ待機した後に、データフレームを送信する。 n_r は、データフレームを受信すると、Ack フレームを返し、送信ノードによる次のデータ送信が始まる。

受信ノードは、送信ノードを学習し記憶しており、Ack フレームにより次に送信するノードを指定できる。このため、時間が経ち定常状態になると、受信ノードは全ての送信ノードを順番に指定することで、バックオフ時間なしに、順番にデータを送信させることができる。つまり、定常状態では、フレーム間の競合が発生せず、安定した通信が可能になる。また、近隣のノード間でビーコンの送信タイミングが近い場合にも、フレーム間の干渉が発生する。これに対して RC-MAC は、近隣とのビーコン送信タイミングを自動的に調整し、フレームが衝突しないようにする機能を持ち、定常状態では効率的な安定通信が可能になる。

3.4 提案 MAC プロトコル

提案 MAC プロトコルを具体的に説明する。提案法は配送木構築状態と定常状態の2状態を持ち、各ノードが状況に応じて状態を切替えながら動作する。定常状態の動作は、配送木における位置（中継ノードか葉ノードか）によって少し異なる。下記に、各状態の動作を述べる。

3.4.1 配送木構築状態

ネットワークの起動時やトポロジ変化時に、ノードは配送木構築状態になり、データを効率的に転送する配送木を構築する。配送木の構築には、隣接ノード間で制御メッセージの交換が必要であり、隣接ノード数が多いセンサネットワークでは、ブロードキャスト通信が必須となる。

一方で、RC-MAC に準じた通信を行う定常状態のノードが混在しても、シームレスにデータ通信ができなければならない。

これを実現するために、配送木構築状態では、基本的には RC-MAC と同様に動作するが、スリープすることなく常に受信待受を行う点が異なる。隣接ノードからの通信がいつでも受信可能になるため、配送木構築状態にある隣接ノードの間では、ブロードキャストフレームの送受信が可能になる。ブロードキャストは、4 章で述べる経路制御プロトコルにより、一定時間毎に制御メッセージを送信するために利用される。ブロードキャストフレームの送信タイミングは既定しないが、定常的な通信との衝突を防ぐために、RC-MAC のフレーム送信タイミングに準ずることが望ましい。

また、配送木構築状態になると、ビーコン、送信データフレーム、及び Ack フレームにフラグを立て、これらを受信したノードは、配送木構築状態でなければ配送木構築状態に遷移する。これにより、あるノードが配送木構築状態になれば、その近隣ノードも連鎖的に配送木構築状態になり、協調して配送木を構築する。ノードが配送木構築状態になって一定時間が経過すると、定常状態に遷移する。ノードが配送木構築状態にある期間の長さとして、配送木の収束にかかる時間と比較して十分に長い値を予め設定しておく。この期間が多少長くなったとしても、ネットワークの運用期間と比較すると十分に短いので、安定したネットワークでは消費電力への影響は僅かである。

3.4.2 定常状態

定常状態は、配送木の葉ノードであるか、中継ノードであるかによって、少し動作が異なる。

中継ノード: 中継ノードである場合は、RC-MAC と同じ動作をする。配送木が確定しており、RC-MAC が配送木に依存した通信パターンを学習するため、干渉のない安定したデータ通信が行われる。トポロジの変化を検出すると、配送木構築状態に遷移する。具体的には、配送木における親ノードにデータフレームを送信できない、親ノードからフラグが立った Ack フレームを受信する、あるいは、子ノードからフラグが立ったデータフレームを受信する、のいずれかの条件により、配送木構築状態に遷移する。

葉ノード: 葉ノードである場合は、中継ノードとは異なりデータフレームの受信を行わないため、その処理を省くことで、さらなる省電力化を行う。中継ノードとの相違は、定期的に送信するビーコンを止めることである。これにより、ビーコンの送信後にしばらく受信待受をすることもなくなり、省電力化の効果は大きい。中継ノードと同様に、配送木における親ノードへパケットを送信できないか、親ノードからフラグが立った Ack フレームを受信すると、配送木構築状態に遷移する。

4. 提案手法：経路制御プロトコル

4.1 概要

配送木の構築は、配送木構築状態において実行される。この間に、各ノードは定期的にブロードキャストメッセージを送信することで隣接ノードと必要な情報を交換し、自律分散的に配送木を構築する。配送木が構築されると定常状態に遷移するが、その後でも、ノードの故障等によりトポロジが変化すると、第 3.4 節で説明したように、提案手法ではネットワーク上の全てのノードが配送木構築状態になり、改めて適切な配送木が構築される。

配送木の構築にあたっては、バッテリー交換にかかる管理コストの最小化を目指す。ある配送木が構築されたとき、葉ノードはデータの受信が不要であるため、受信ノード主導型の MAC プロトコルを用いると 10 年以上の寿命を確保でき得るのに対して、中継ノードは長くても 1~2 年程度の寿命であることを想定する。このため、中継ノードは定期的なバッテリーの交換が必要であり、その分の管理コストがかかる。本研究では、中継ノードの数をできるだけ少なくするような配送木の構築プロトコルを設計することで、バッテリー交換に要する管理コストを低減する。

4.2 シンクノードまでの距離の把握

本研究で提案する配送木構築プロトコルでは、まず、全てのノードが、シンクノードまでの距離（ホップ数）を把握する。このために、定期的にブロードキャストする制御メッセージに、自分からシンクノードまでの距離を含める。この情報を隣接ノードと繰り返し交換することにより、全てのノードが、シンクノードまでの正しい距離を把握する。

初期状態では、シンクノードだけが、距離を 0 として持っている。シンクノードまでの距離を把握するために、各ノードは一定時間毎に制御メッセージを送信する。制御メッセージには、そのノード n_s が把握している、 n_s のシンクまでの距離 $dist(n_s)$ が格納される。そのうえで、制御メッセージを受信したノード n_r は以下のように動作する。受信したメッセージに格納されている距離 $dist(n_s)$ と自分が保持する距離 $dist(n_r)$ を比較する。 $dist(n_r)$ がまだ設定されていないか、 $dist(n_s) + 1 < dist(n_r)$ であれば、 $dist(n_r) = dist(n_s) + 1$ とする。そうでなければ、何もしない。全てのノードがこの手順を繰り返すことにより、最終的には、全てのノードが自分の正しい距離を把握できる。

図 2 は、初期状態から、シンクノードのみが制御メッセージを送信し、その隣接ノードが距離を 1 とした状態である。制御メッセージの送信を繰り返すことで、図 3 のように、全てのノードが自分の距離を把握できる。なお、ノードが故障する等の理由で正しい距離が変わることがある。これには、自分の距離を設定する元となったノードからの制御

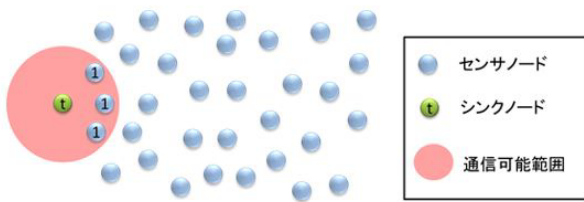


図 2 シンクノードからのメッセージ送信

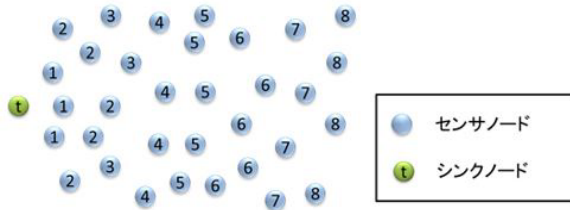


図 3 シンクノードからの距離の把握

メッセージが一定期間届かなければ、自分の距離を初期値（無設定）に戻す、等の処理で対応できる。

4.3 次ホップの決定

次に、シンクノードに至るための次ホップを決定する。本研究では、中継ノードの数を低減するために、通信を特定のノードに集中させる。このために、「優先度」という概念を導入する。優先度とはどの程度次ホップに選ばれるべきかを表し、各ノードは、自分よりも距離が1だけ小さいノードの中で、優先度が最大のノードを次ホップに選ぶ。優先度としては、そのノードの、配送木における子孫の数を用い、周囲の通信パターンに応じて値を調整する。子孫数は、そのノードが中継しなければならぬデータ量と比例するため、子孫数が大きいノードを次ホップに選ぶことで、通信を特定のノードに集中させ、中継ノードの数を低減できる。なお、ノードによって生成するデータのサイズや頻度が異なることも有り得るが、制御メッセージの拡張により容易に対応できる。

各ノード x が自分の子孫数 d_x を把握するために、送信する制御メッセージに、自分が把握しているその時点の子孫数と次ホップノードを含める。各ノードは、自分の子ノード（自分を次ホップに選んでいるノード）を管理し、子ノードの子孫数の合計を自分の子孫数とする。これにより、十分な時間が経過すると、全てのノード x が自分の子孫数 d_x を把握できる。

ノード x の優先度 P_x は、次式により定義される。

$$P_x = d_x + L \quad (1)$$

ここで L は、周囲の通信パターンにより定まる調整値である。 L は、特定の通信パターンによく当てはまるほど大きな値を取り、自分に通信を集中させる。これにより、最適性の低い配送木に収束することを防ぎ、より最適性の高い配送木に変化するように誘導する。ここで、最適性の高い

配送木とは、中継ノード数が少ない配送木のことを指す。最適性の高い配送木では、シンクノードからの距離が同じノードの集合内では中継ノードは2ホップほど距離を置いて配置されていることが望ましい。そのような最適性の高い配送木を構築するために、 L が大きな値をとるべき通信パターンを3種類定義した。以下にこれらを説明する。

パターン (i): パターン (i) を図 4(a) に示す。このパターンでは、ノード X の隣接ノードである B と C が中継ノードとなり、比較的近い距離にある2ノードに通信が集中している。この場合には、図 4(b) のように、ノード X が次ホップとなり、 B と C の通信を引き受けることで中継ノードを削減し、より最適性の高い配送木に遷移する。このために、このような通信パターンに合致する場合に、 L が大きな値をとるようにする。

パターン (ii): このパターンでは、ノード X の隣接ノード B と、 B の隣接ノード A がいずれも中継ノードとなり、隣接ノードに通信が集中している（図 5(a)）。この場合には、図 5(b) のように、ノード X が次ホップとなり、ノード B の通信を引き受けることで、中継ノード間の距離を離すことができる。これにより最適性の高い配送木に推移する。

パターン (iii): このパターンでは、ノード X を含めて、周囲2ホップ以内に、通信が集中しているノードが存在せず、通信が分散している（図 6(a)）。この場合には、図 6(b) のように、ノード X が優先度を上げ、周囲の通信を引き受けることで、中継ノードの数が減り、より最適性の高い配送木に推移する。

各パターンへの合致度合い L は、周囲のノードの子孫数を表すベクトルを定義し、パターンを表すベクトルとの差異を数値化して求める。まず、子孫数を表すベクトルを定義する。パターン (i) の場合には、自分を x とおき、自分から1ホップの距離にあるノードのうち、子孫数が最大のノードを n_1^1 、2番目のノードを n_2^1 と定義する。同様に、2ホップの距離にあるノードのうち、子孫数が最大のノードを n_1^2 、2番目のノードを n_2^2 とする。ノード x の子孫数ベクトルは、各ノードの子孫数を用いて、 $D = (d_x, d_{n_1^1}, d_{n_2^1}, d_{n_1^2}, d_{n_2^2})$ と定義する。一方、パターンベクトルは、 $P = (0, 1, 1, 0, 0)$ と定義する。これらの差異を数値化することで L を求める。パターン (ii) の場合には、ノード n_1^2 の子孫数を報告した隣接ノードのうち最も子孫数が多いものを r_1 で表し、隣接ノードのうち、 r_1 を除いて、最も子孫数が多いものを n_1^1 として、子孫数ベクトルを $D = (d_x, d_{r_1}, d_{n_1^1}, d_{n_2^1}, d_{n_2^2})$ 、パターンベクトルを $P = (0, 1, 0, 1, 0)$ と定義すれば良い。

L は、2つのベクトルの差異を用いて、次のように計算する。子孫数ベクトルを $D = (d_1, d_2, \dots, d_n)$ 、パターンベクトルを $P = (p_1, p_2, \dots, p_n)$ とおく。変数 a を用いて、 aD と P の各要素の差の二乗和が最小になるように、すなわち、 $\sum_{i=1}^n (ad_i - p_i)^2$ が最小になるように a を定めたうえで、

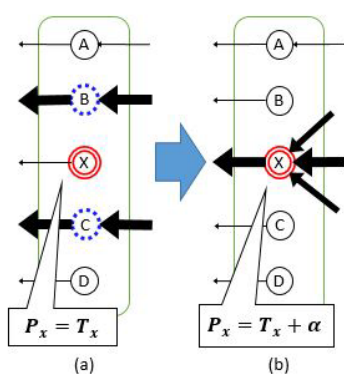


図 4 パターン (i) の経路変更

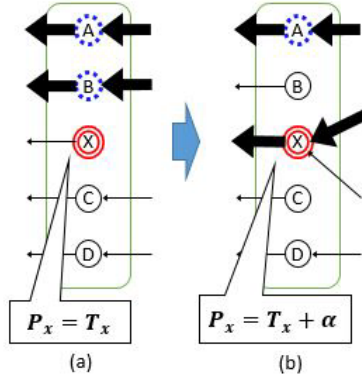


図 5 パターン (ii) の経路変更

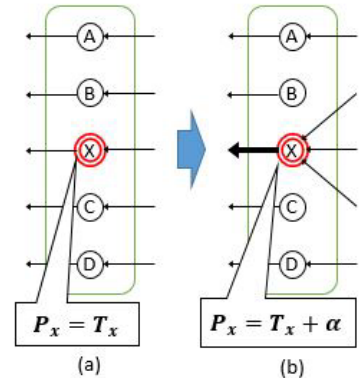


図 6 パターン (iii) の経路変更

$L = \beta \mu \frac{1}{\sum_{i=1}^n (ad_i - p_i)^2}$ とする．ここで、 μ は $\mu = \sum_{i=1}^n \frac{d_i}{n}$ により計算される近隣ノードの子孫数の平均であり、式 (1) において、根ノードからの距離によらず d_x と L の重みのバランスを一定に保つ役割を持つ．また、 β は両者の重みのバランスを決める定数である．

なお、パターン (iii) については、パターンベクトルの値が全て 0 になるため、上記の方法では合致度を測れない．そこで、 $x, n_1^1, n_1^2, n_2^1, n_2^2$ の子孫数がすべて平均的であれば、つまり、 $\mu = \frac{d_x + d_{n_1^1} + d_{n_1^2} + d_{n_2^1} + d_{n_2^2}}{5}$ において、適切な定数 $c (1 \leq c)$ に対して $d_x, d_{n_1^1}, d_{n_1^2}, d_{n_2^1}, d_{n_2^2}$ の全てが $c\mu$ 以下であれば、パターンに合致したと見なし、 L を決められた値に設定する．

最終的に、ノードの優先度は、各パターンに対して計算した L の値（それぞれ、 $L^{(i)}, L^{(ii)}, L^{(iii)}$ と書く）の和 $L = L^{(i)} + L^{(ii)} + L^{(iii)}$ を計算し、これを式 (1) に適用して求める．

上記の方法により、子孫数と通信パターンからノードの優先度を求められる．優先度が決まり、制御メッセージにより周知されると、一部のノードの次ホップが変更され、これにより子孫数と通信パターンが変化し、再び優先度が変更される．このようなフィードバックを経て、最終的に配送木が収束することが期待される．しかし、優先度が急激に変化すると、それにより次ホップも急激に変化し、この繰り返しにより配送木が振動し、収束しない問題が発生し得る．そこで提案手法では、適切な値 α をパラメータとして設定し、各ノードは、制御メッセージの送信 1 回ごとの優先度の変化幅を $[-\alpha, \alpha]$ の範囲に抑える．これにより、優先度は常に緩やかに変更され、バランスがとれた効率的な配送木に収束する．

5. 評価

5.1 消費電力の評価

5.1.1 評価方法

提案手法と既存手法の消費電力を見積もり、ネットワー

クの寿命を比較する．本比較では、各ノードの動作から定常状態における消費電力をモデル化し、モデルに基づいて解析的に消費電力を見積もる．なお、センシングの時間間隔が十分に長いセンサネットワークにおいては、RC-MAC の機能により、定常状態ではフレームの送信スケジュールは安定し、フレームが衝突する確率は低いことから、フレームの衝突は考慮しない．

既存手法としては、低消費電力で動作する MAC プロトコルである RC-MAC[6] を用いる．RC-MAC は配送木の構築機能を持たないが、評価にあたっては、提案手法の配送木を用いることを仮定する．

5.1.2 フレームサイズ

提案手法では MAC フレームの構造や各フィールドのサイズを規定することはないが、評価のために用いるフレームサイズの想定について述べておく．

各ノードは、 r [秒] 毎にセンシングを行い、得られた値は 2[Bytes] の値とする．各ノードは 4[Bytes] で表現されるノード ID を持つ．各ノードが r [秒] 毎に送信するデータフレームは、ヘッダに送信元ノード ID、次ホップノード ID、シーケンス番号 (2[Bytes]) に加えて、プロトコルバージョンやチェックサム等の制御用フィールドを加えて 24[Bytes] とする．

ビーコンフレームは、定常状態では、送信元ノード ID に少しの制御用フィールドを含み、6[Bytes] とする．

配送木構築状態では、ビーコンと同じ頻度で制御メッセージを送信することにする．制御メッセージに含まれるのは、送信元ノードの ID (4[Bytes])、優先度値 (2[Bytes])、子孫の数 (4[Bytes])、次ホップノードの ID (4[Bytes])、および、隣接ノードのノード ID と優先度値 (1 ノードあたり 6[Bytes]) である．ただし、隣接ノードの情報については、子孫数が大きいもののみが重要であるため、子孫数の上位 k 個の情報のみで十分である．本研究では、 $k = 2$ とした．その結果、フレームサイズは、 $20 + 6k = 32$ [Bytes] となる．

表 1 消費電力諸元

操作	時間 (秒)		電流 (mA)	
センサからのデータ取得	t_{data}	1.1	c_{data}	20
データ送信 (1 バイト)	t_{txb}	416E-6	c_{txb}	20
データ受信 (1 バイト)	t_{rxb}	416E-6	c_{rxb}	15
受信待受	-	-	c_{listen}	7
スリープ	-	-	c_{sleep}	0.030

表 2 設定パラメータ

変数	意味	値
C_{batt}	バッテリー容量	2500[mAh]
V	動作電力	3[V]
b	ビーコン送信間隔 (秒)	20[秒]
p	制御メッセージ送信間隔 (秒)	20[秒]
r	センシング間隔 (秒)	300[秒]
l_{data}	データフレームサイズ	25[Bytes]
l_{ack}	Ack フレームサイズ	6[Bytes]
l_{beacon}	ビーコンサイズ (定常状態)	6[Bytes]
$l_{control}$	制御メッセージサイズ	28[Bytes]
$t_{lifetime}$	ノードの寿命 (秒)	-
$t_{converge}$	経路収束までの時間 (秒)	800[秒]
t_{txwait}	送信時のビーコン待受時間 (秒)	1[秒]
t_{CW}	ビーコン送信後の受信待ち時間 (秒)	0.3[秒]

5.1.3 電力モデル

消費電力は、無線センサとして頻繁に利用される MICA2[18] の消費電力をモデル化し、これを基に計算する。MICA2 の種々の動作に対する消費電力は、文献 [3][15][16] に掲載されている値を用いた (表 1)。MICA2 は単三電池 2 本で動作し、その容量を $C_{batt} = 2500[\text{mAh}]$ とする。また、動作電圧は $V = 3[\text{V}]$ である。

消費電力モデルは、文献 [3] に用いられているものを用いた。すなわち、一秒あたりの消費電力を E とおくと、 E は次式で表される。

$$E = E_d + E_{tx} + E_{rx} + E_{listen} + E_{sleep}$$

ここで、 E_{rx} は受信電力、 E_{tx} は送信電力、 E_{listen} は受信待受電力、 E_d はセンサからデータを取得する電力、 E_{sleep} はスリープに必要となる電力である。個々の値は、用いる MAC プロトコルやノードにより異なる。以下では、ビーコン送信間隔を b (秒) センサからのデータ取得間隔を r (秒)、送信するデータフレームのサイズを l_{data} とする。これらを含めて、消費電力諸元、及び設定パラメータを、表 1 と 2 に示す。

定常状態の消費電力

(a) 提案手法の葉ノード

E_d : センシング一回あたりの消費電力は $t_{data}c_{data}V$ であり、これが r 秒に一度必要であるから、一秒あたりの消費電力は、 $E_d = \frac{t_{data}c_{data}V}{r}$ である。センシングにかかる消費電力は、全てのノード、全ての動作モードに共通である。

E_{tx} : データの送信も r 秒に一度であり、1 フレームの送

信に必要な電力は $l_{data}t_{txb}c_{txb}V$ であるから、一秒あたりの消費電力は、 $E_{tx} = \frac{l_{data}t_{txb}c_{txb}V}{r}$ である。

E_{rx} : 葉ノードは、データフレームの送信時に親ノードからビーコンと Ack フレームを受信する。よって、 $E_{rx} = \frac{(l_{beacon} + l_{ack})t_{rxb}c_{rxb}V}{b}$ である。

E_{listen} : 受信待受は、提案手法では、データ送信のたびに発生する。送信データの発生タイミングが親ノードのビーコンと無関係であれば、一回あたりの待受時間の平均が $\frac{b}{2}$ であるが、ビーコンの送信は一定時間毎であるため、送信ノードは、送信のためにウェイクするタイミングを学習できる。ここでは、学習の結果、送信のためのビーコン受信待受は、 $t_{txwait} = 1[\text{秒}]$ に短縮されたと仮定する。よって、一秒あたりの消費電力は、 $E_{listen} = \frac{t_{txwait}c_{listen}V}{r}$ である。

E_{sleep} : スリープの時間は、上記いずれにも含まれない時間全てであるから、 $t_{sleep} = 1 - \frac{t_{data}}{r} - \frac{l_{data}t_{txb}}{r} - \frac{l_{ack}t_{rxb}}{r} - \frac{t_{txwait}}{r}$ であり、その消費電力は、 $E_{sleep} = t_{sleep}c_{sleep}V$ である。

(b) 提案手法の中継ノード

中継ノードでは、葉ノードの処理に、データフレームを送受信する処理が加わる。また、送受信すべきデータの量は、そのノードの子孫の数に比例し、この分の電力が必要になる。このため、送信電力、受信電力、待受電力が葉ノードとは異なる。

E_{tx} : 送信するデータ量は配送木における子孫の数に比例し、その分の送信電力が必要になる。これと同数のデータフレームを受信するが、この際に Ack フレームを送信する。また、データを受信するために、定期的にビーコンを送信する必要がある。配送木における子孫の数を $n_{descendant}$ とすると、必要な送信電力は、 $E_{tx} = \frac{n_{descendant}(l_{data} + l_{ack})t_{txb}c_{txb}V}{r} + \frac{l_{beacon}t_{txb}c_{txb}V}{b}$ である。

E_{rx} : 受信電力は、親ノードからのビーコンの受信と、子ノードからのデータの受信、及び、親ノードへデータを送信した際の Ack フレームの受信にかかる電力の和である。よって、 $E_{rx} = \frac{l_{beacon}t_{rxb}c_{rxb}V}{b} + \frac{n_{descendant}(l_{ack} + l_{data})t_{rxb}c_{rxb}V}{r}$ である。

E_{listen} : 待受電力は、データ送信待ちにかかる電力と、ビーコンを送信した後の受信待ちにかかる電力の和である。後者にかかる時間 t_{CW} は Contention Window (CW) の大きさに依存するが、通信速度から適切な値として 300[ms] を設定する。待受電力は、 $E_{listen} = \frac{t_{txwait}c_{listen}V}{r} + \frac{t_{CW}c_{listen}V}{b}$ である。

(c) 比較手法 RC-MAC

RC-MAC の消費電力は、中継ノードにおいては提案手法と同じである。葉ノードにおいては、フレーム受信のためにビーコンを送信し、 t_{CW} の間だけ受信待受をする電力が追加される。これにより、送信電力と待受電力が提案手法とは異なる。

E_{tx} : 送信電力は、ビーコンの送信電力を提案手法の葉ノードに追加して、 $E_{tx} = \frac{l_{data}t_{txb}C_{txb}V}{r} + \frac{l_{beacon}t_{txb}C_{txb}V}{b}$ である。

E_{listen} : 待受電力は、提案手法の中継ノードと同じであり、 $E_{listen} = \frac{t_{xwait}C_{listen}V}{r} + \frac{t_{CW}C_{listen}V}{b}$ である。

配送木構築状態の消費電力

(a) 提案手法の葉ノード

配送木構築状態では、提案手法は、送受信以外の全ての時間で受信待受を行うため、受信待受時間が大幅に増大する。また、このため、周囲のノードのビーコンと制御メッセージを全て受信することになり、受信電力も増大する。さらに、葉ノードも配送木構築状態ではビーコンと制御メッセージを送信するため、送信電力も増大する。

E_{rx} : 隣接ノードからのビーコンと制御メッセージを全て受信するため、その分の受信電力が提案手法の葉ノードに追加される。そのノードの隣接ノード数を n_{neigh} とすると、受信時間は、 $t_{rx} = \frac{l_{ack}t_{rxb}}{r} + \frac{n_{neigh}l_{beacon}t_{rxb}}{b} + \frac{n_{neigh}l_{control}t_{rxb}}{p}$ であり、受信電力は $E_{rx} = t_{rx}C_{rx}V$ である。

E_{tx} : 定期的なビーコン送信と制御メッセージ、及びデータの送信が必要になる。従って、送信時間は $t_{tx} = \frac{l_{data}t_{txb}}{r} + \frac{l_{beacon}t_{txb}}{b} + \frac{l_{control}t_{txb}}{p}$ であり、送信電力は $E_{tx} = t_{tx}C_{tx}V$ となる。

E_{listen} : 送受信していない時間は全て受信待受時間である。よって、1秒あたりの受信待受時間は、 $t_{listen} = 1 - t_{tx} - t_{rx}$ であり、受信待受電力は、 $E_{listen} = t_{listen}C_{listen}V$ となる。

E_{sleep} : スリープしないため、電力は $E_{sleep} = 0$ である。

(b) 提案手法の中継ノード

中継ノードでは、葉ノードの電力に加えて、データの送受信の電力が追加される。

E_{rx} : 葉ノードの受信電力に、子孫数のデータと Ack の受信が加わる。そのノードの子孫数を $n_{descendant}$ 、隣接ノード数を n_{neigh} とすると、受信時間は、 $t_{rx} = \frac{l_{ack}t_{rxb}}{r} + \frac{n_{neigh}l_{beacon}t_{rxb}}{b} + \frac{n_{neigh}l_{control}t_{rxb}}{p} + \frac{n_{descendant}(l_{data}+l_{ack})t_{rxb}}{r}$ であり、受信電力は $E_{rx} = t_{rx}C_{rx}V$ である。

E_{tx} : 葉ノードの送信電力に、子孫数のデータと Ack の送信が加わる。送信時間は、 $t_{tx} = \frac{l_{data}t_{txb}}{r} + \frac{l_{beacon}t_{txb}}{b} + \frac{l_{control}t_{txb}}{p} + \frac{n_{descendant}(l_{data}+l_{ack})t_{txb}}{r}$ であり、送信電力は $E_{tx} = t_{tx}C_{tx}V$ となる。

ノードの寿命

提案手法の消費電力は、先述の、一秒間あたりの消費電力を用いて、次式によりノードの寿命を求める。ここで、 E^s は定常状態の電力を表し、 E^c は配送木構築状態の電力を表す。 $t_{converge}$ は、配送木が収束するまでの時間を示す。

$$t_{lifetime} = \frac{(C_{batt} - \frac{E^c}{t_{converge}}) \times V}{E^s} \times 60 \times 60$$

5.1.4 評価結果

提案手法と RC-MAC の葉ノードのノード寿命をそれぞれ

図7と図8に示す。RC-MAC の場合には、ノードの寿命は概ね2年程度であるのに対して、提案手法は概ね5年以上の長寿命を実現できている。これは、葉ノードにおいては20秒に1回のビーコンの送信とこれに伴う受信待受の消費電力が支配的であり、これらの処理を省くことによって、大幅な消費電力の低減が可能であることを示している。また、これらの図には、提案手法において電力消費割合が大きかったスリープ電力を、MICA2の既定値である0.03mAから変更した値も示した。その結果、スリープ電流を低減することで、葉ノードは10年以上の長寿命化を実現できることがわかった。なお、MICA2は少し古い製品であるが、より新しいセンサノード端末の一つである MICAz ではスリープ電流を $9 \mu A$ にまで低減している [17]。

図9は、提案手法における、中継ノードの寿命を表している。提案手法の中継ノード（定常中継ノード状態）は RC-MAC と同じ動作をするため、RC-MAC とは比較していない。中継ノードでは、センシング間隔と配送木における子孫数が支配的であり、図9からも、そのことが見てとれる。図8との差分は、子孫ノードからのデータの送信であり、データ送信量によるノード寿命への影響の大きさがわかる。

中継ノードはバッテリー交換が必要になることから、管理コストを低減するためには、一定以上の寿命があることが望ましい。しかし、ハードウェアが進歩しても、送受信にかかる電力は、送受信の電波強度が変わらない限り、それほどは変わらない。本評価では、送信電力は10[dbm]を想定しているが、送信電力を低くすることで、バッテリーを交換するノードが増えるものの、ノードの寿命を延ばせると考えられる。また、今回は通信速度に合わせるために Contention Window (CW) を0.3[ms]としたが、通信速度が早くなれば、受信待受時間を低減でき、ノードの寿命を延ばせる可能性がある。

5.2 配送木の評価

5.2.1 評価方法

本研究では、バッテリー交換にかかる管理コストを低減するような配送木を構築する。従って、配送木における中継ノードの数を評価した。

評価には C++ 言語で開発した自作のシミュレータを用いた。本シミュレータでは、各ノードが一定時間毎に制御メッセージを送信し、配送木が収束するまでを模倣した。データフレームの配送や MAC プロトコル等は扱わず、従って、制御メッセージの損失がないことを仮定した。1000 × 1000[m] の正方形領域に 1000 ノードをランダムに配置し、左辺中央にシンクノードを1つ設置した。各ノードの送信電力は10[dbm]を想定し、通信可能範囲を半径100[m]の円とした。送信電力10[dbm]の時の通信可能距離は、文献 [11] で用いられている値と同じである。

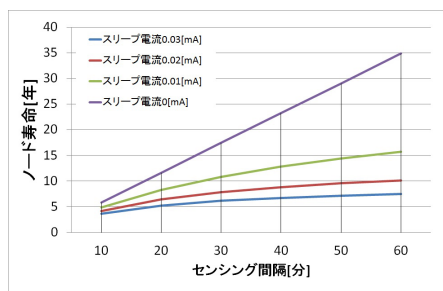


図 7 提案手法: 葉ノードの寿命

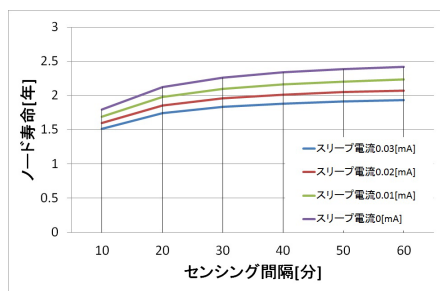


図 8 RC-MAC: 葉ノードの寿命

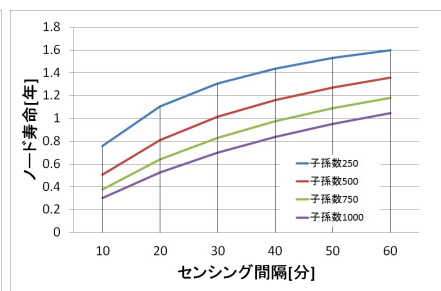


図 9 提案手法: 中継ノードの寿命

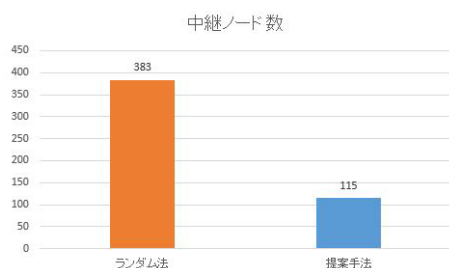


図 10 中継ノード数の比較

比較手法としては、既存研究の中に中継ノード数を最小化する方法が存在しないため、単純な方法を用いる。各ノードが自分よりもシンクまでの距離が1だけ小さいノードの中からランダムに次ホップを選ぶランダム法を実装し、提案手法と比較した。

5.2.2 結果と考察

提案手法とランダム法の中継ノード数を図 10 に示す。提案手法は、ランダム法と比較して、中継ノード数を大幅に削減できたことがわかる。また、中継ノード数は、ノード数には依存せず、フィールドの大きさと通信可能距離に依存するため、10[dbm]の送信電力で100[m]の通信が可能だとすれば、1000×1000[m]のフィールドでは100個程度のノードのバッテリー交換が必要である。

提案手法と比較手法の配送木を図 11 と図 12 にそれぞれ示す。配送木のノードは、シンクノードまでの距離によって色分けされており、単位時間あたりのパケット送信数が多いほどリンクを太く表示している。この図から、提案手法の配送木では、シンクノードからの距離が等しい(同じ色で表されている)ノードの集合を見ると、中継ノードがある程度の距離を離して配置されており、提案手法が比較手法より最適性の高い配送木を構築したことがわかる。ただし、中継ノード数が最小となるネットワークを求める問題は複雑であり、中継ノード数の最小値は求められていない。(著者らは、NP 完全問題ではないかと予想している。)このため、最適性の評価については今後の課題としたい。

6. 実用に関する議論

提案するセンサネットワークの実用化にあたっての議論を行う。まず、バッテリーの交換コストは、交換ノード数が

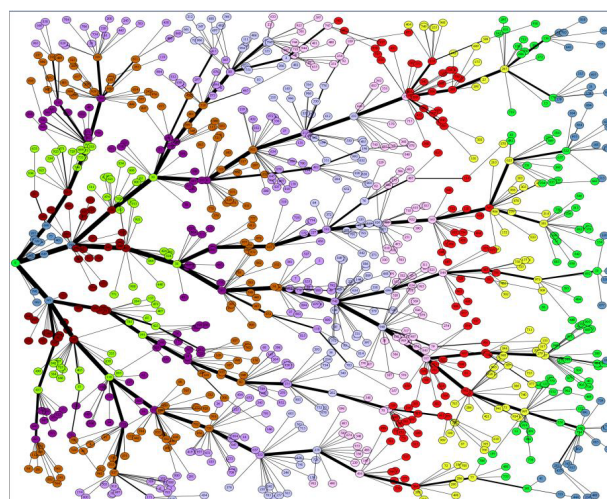


図 11 提案手法の配送木

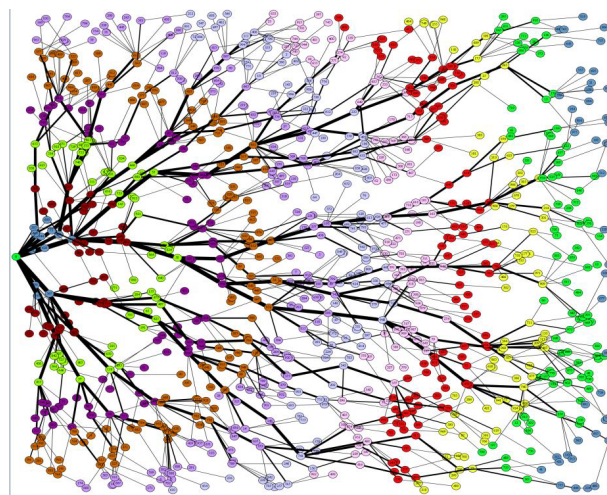


図 12 ランダム法の配送木

多数になるほど大きくなり、その削減効果も大きくなることを指摘しておく。例えば、農場などの広大な土地に土中温度や湿度を計測するセンサを設置する場合には比較的密な配置を必要とするため、センサ数も膨大になり、これを少数に抑えることは管理コストを大幅に削減する。本実験の結果からは、1000×1000[m]の領域においては交換ノード数を100個程度に抑えられることが示されており、定期的に全ノードの交換をすることと比較すると、大きな管理コストの削減になる。

また、市街地などにおいては、バッテリーを交換しにくい位置にセンサを配置する場合があります。提案手法はこのような場合にも便利である。交換しにくい位置のセンサを葉ノードにしてバッテリー交換を不要にすれば、管理コストの低減につながるからである。このためには、中継ノードになる優先度を設定しておき、できるだけ優先度の高いノードが中継ノードに選ばれるような仕組みを実装すれば良い。そのような仕組みの実装は容易である。

本研究の評価結果から、提案手法では、葉ノードの寿命に対して中継ノードのノード寿命がかなり短くなることがわかった。中継ノードの寿命は、データのセンシング間隔と子孫数の影響を大きく受ける。シンク付近のノードの子孫数は急激に大きくなるため、これらの寿命が極端に短くなり、管理上のボトルネックになる可能性がある。この問題は、適切な対策で回避できる。例えば、直線状に電源に接続したノードを配置し、これら全てをシンクノードとすると、データが一点に集まることによる子孫数の偏りを避けられる。提案手法では、シンクノードを特別なノードとして扱わず、他のノードと同じ動作を行うため、このような複数のシンクノードが存在するネットワークにもそのまま適用が可能である。また、バッテリー残量が少ない中継ノードがある場合には、これが葉ノードになるように、自律的に配送木を再構築する仕組みを組み込んでも良い。後者のような、管理コストを低く抑えながらもネットワークの寿命を延伸するような経路制御の仕組みは、今後の課題の一つである。

最後に、実用を考えるならば、バッテリーを交換すべきノードをどうやって発見するかを考える必要がある。このためには、GPS等により各ノードが位置を測定してデータと一緒にシンクノードに収集しておく方法や、何らかの近距離通信機器を持ち歩いて近くのセンサノードとバッテリー残量を確認する方法などが考えられる。いずれにせよ、密に配置されたセンサノードを対象とする場合には、高精度な位置の特定技術が必要であるが、そのような技術も、今後の課題とする。

7. おわりに

本論文では、比較的少数のノードのバッテリーを定期的に交換することで10年以上の運用が可能な、インフラとしての無線センサネットワーク構築技術を提案した。提案手法は、消費電力性能に優れる受信ノード主導型MACプロトコルの一つであるRC-MACを拡張し、状態遷移を導入することで、自律的に管理コストの低い配送木を構築することと、消費電力の低い効率的なデータ集約を両立できる。代表的なセンサノードの一つであるMICA2を想定した電力モデルに基づいて消費電力を評価すると、配送木における葉ノードは5年～15年程度の長い寿命を達成できることが明らかとなった。また、提案手法が構築する配送木に

よって、バッテリー交換が必要な中継ノードの数を大幅に低減できることが明らかとなり、管理コストを低減できることを示した。これにより、10年以上の運用が可能なインフラとしての無線センサネットワークの実現に向けて、その実現可能性を示すことができたと考えている。

今後の課題は多数あるが、現時点では、ノードの故障などによるトポロジの変化に対して、自律的かつ柔軟に配送木を再構成する機能の評価を実施できていない。まずはこれを実施し、提案手法の経路制御の柔軟性を示したい。また、提案手法を発展させて、一部の中継ノードのバッテリー残量が少なくなった場合に、柔軟に配送木を再構築し、管理コストを抑えつつも、ネットワーク全体の寿命を延ばす方法を開発したいと考えている。

参考文献

- [1] W. Ye, J. Heidemann, and D. Estrin, "An Energy-efficient MAC Protocol for Wireless Sensor Networks," In Proc. of the 21st Conference on Computer Communications (INFOCOM'02), 2002.
- [2] T. van Dam and K. Langendoen, "An Adaptive Energy-efficient MAC Protocol for Wireless Sensor Networks," In Proc. of the 1st ACM Conference on Embedded Networked Sensor Systems (SenSys'03), pp. 171-180, 2003.
- [3] J. Polastre, J. Hill, and D. Culler, "Versatile Low Power Media Access for Wireless Sensor Networks," In Proc. of the 2nd ACM Conference on Embedded Networked Sensor Systems (SenSys'04), pp.95-107, 2004.
- [4] M. Buettner, G. Yee, E. Anderson, and R. Han, "X-MAC: A Short Preamble MAC Protocol for Duty-cycled Wireless Sensor Networks," In Proc. of the 4th ACM Conference on Embedded Networked Sensor Systems (SenSys'06), 2006.
- [5] Y. Sum, O. Gurewits, and D. B. Johnson, "RI-MAC: A Receiver-initiated Asynchronous Duty Cycle MAC Protocol for Dynamic Traffic Loads in Wireless Sensor Networks," In Proc. of the 6th ACM Conference on Embedded Networked Sensor Systems (SenSys'08), pp.1-14, 2008.
- [6] P. Huang, C. Wang, L. Xiao, "RC-MAC: A Receiver-Centric MAC Protocol for Event-Driven Wireless Sensor Networks," In Proc. of the 18th International Workshop on Quality of Service (IWQoS'10), 2010.
- [7] P. Huang, L. Xiao, M.W. Mutka, and N. Xi, "The Evolution of MAC Protocols in Wireless Sensor Networks: A Survey," IEEE Communications Surveys & Tutorials, Vol. 15, No.1, 2013.
- [8] C. Park and P.H. Chou, "Ambimax: Autonomous Energy Harvesting Platform for Multi-supply Wireless Sensor Nodes," In Proc. of the IEEE SECON'06, 2006.
- [9] P. Dutta, J. Hui, J. Jeong, S. Kim, C. Sharp, J. Taneja, G. Tolle, K. Whitehouse, and D. Culler, "Trio: Enabling Sustainable and Scalable Outdoor Wireless Sensor Network Deployments," In Proc. of the 4th ACM Conference on Embedded Networked Sensor Systems (SenSys'06), pp.407-415, 2006.
- [10] S. Sudevalayam and P. Kulkarni, "Energy Harvesting Sensor Nodes: Survey and Implications," IEEE Communications Surveys & Tutorials, Vol. 13, No. 3, pp443-461, 2011.
- [11] 猿渡俊介, 森戸貴, 南正輝, 森川博之, "バッテリーレス無

- 線センサネットワークと省メモリ型データ収集プロトコル,” 情報処理学会論文誌, Vol. 55, No.1, pp.1-14, 2014.
- [12] D. Luo, X. Zhu, X. Wu, and G. Chen, ”Maximizing Lifetime for the Shortest Aggregation Tree in Wireless Sensor Networks,” In Proc. of the 30th Conference on Computer Communications (INFOCOM’11), 2011.
 - [13] T.W. Kuo and M.J. Tsai, ”On the Construction of Data Aggregation Tree with Minimum Energy Cost in Wireless Sensor Networks: NP-Completeness and Approximation Algorithms,” In Proc. of the 31st Conference on Computer Communications (INFOCOM’12), 2012.
 - [14] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, ”Collection Tree Protocol,” In Proc. of the 7th ACM Conference on Embedded Networked Sensor Systems (SenSys’09), 2009.
 - [15] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, ”Wireless Sensor Networks for Habitat Monitoring,” In Proc. of the 1st ACM international workshop on Wireless sensor networks and applications (WSNA’02), pp.88–97, 2002.
 - [16] V. Shnayder, M. Hempstead, B. Chen, G.W. Allen, and M. Welsh, ”Simulating the Power Consumption of Large-Scale Sensor Network Applications,” In Proc. of the 2nd ACM Conference on Embedded Networked Sensor Systems (SenSys’04), 2004.
 - [17] E. Perla, A.Ó Catháin, R.S. Carbajo, M. Huggard, and C. Mc Goldrick, ”PowerTOSSIM z: realistic energy modelling for wireless sensor network environments,” In Proc. of the 3rd ACM workshop on Performance monitoring and measurement of heterogeneous wireless and wired networks (PM2HW2N’08), pp.35–42, 2008.
 - [18] J. L. Hill and D.E. Culler, ”MICA: A Wireless Platform for Deeply Embedded Networks,” IEEE Micro, Vol.22, No.6, pp.12–24, 2002.
 - [19] IEEE802.15.4 Standard, <http://www.ieee802.org/15/pub/TG4.html>, 2011.