

省電力化のためのマルチホップ通信制御の設計と実装

岩崎 央[†] 廣津 登志夫[†]法政大学情報科学部[†]

1. はじめに

センサネットワークは、環境を観測するセンサが搭載された大量の小型ノードにより構築されるネットワークである。センサネットワークでは様々な事象の観測が行われ、防犯・防災、環境保全、健康・医療、家電機器の保守など広い範囲に応用されている。無線センサノードは無線通信でデータの伝達をするため設置場所の制約が少ないが、多くの場合バッテリー駆動であるためその稼働時間は有限である。一方、多数のノードのバッテリーを交換するコストは大きいのでバッテリーの駆動時間を延ばすことが求められる。これまでの研究[1][2]で、各ノードが取得した観測データの時間的な変動やノード相互の観測データの空間的な多様性に着目し、センシング頻度を抑制する手法が提案されてきた。しかし、これらの手法ではシングルホップを前提としており、マルチホップ通信の中継を考慮した中継ノードの起動タイミングの制御は考慮されていなかった。そこで、本研究ではマルチホップ通信のためのクラスタ内通信における集約ノードのスリープ制御及び通信タイミング制御の設計と実装を行う。

2. 既存研究

既存研究[1]では、ノード取得した観測データから空間的多様性に着目したセンシング頻度抑制による省電力化手法を提案している。この手法では、ノード同士で観測データを交換し、観測データに相関がある場合にクラスタを形成する。クラスタ中のクラスタヘッドは各々の子ノードに対して同期パケットを送信する。同期パケットにはクラスタの所属ノード数、センシング間隔、所属ノードIDが含まれており、子ノードは所属ノードIDから自身がクラスタの何番目に所属するか把握することができる。同期パケットを受信したタイミングからクラスタの所属番目とセンシング間隔で乗算しただけでなくしてセンシングを行い、その後所属ノード数とセンシング間隔を乗算した間隔でセンシングを行う。この手法では、すべてのノードがサンプリングした情報をシンクノードにシングルホップで通信することを想定していた。しかし、実環境でのノード配置の自由度を高めることを考えると、直接シンクノードに通信できなくても構わないマルチホップの通信を実現する必要がある。

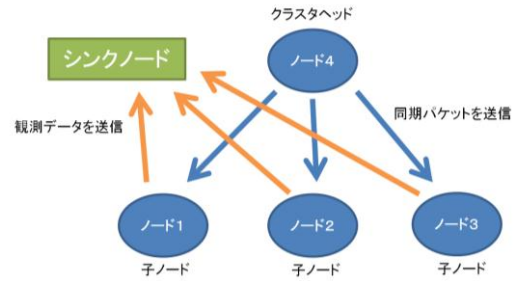


図1 既存研究

3. スリープ制御

マルチホップを考慮した場合、子ノードがデータを送信する際にクラスタヘッドが中継ノードとなり通信を中継する。そのためクラスタは起動していなければならない。ノードの消費電力を抑制するためにノードが通信とセンシング以外ではスリープ状態にするスリープ制御を行う。また、スリープ制御の起動タイミングに合わせて通信する。

3.1. 子ノード

クラスタヘッドから子ノードへの同期パケットが送信される。同期パケットを受信する前では、子ノードは常に起動している。受信した後、子ノードはセンシングと送信以外ではスリープ状態となる。同期パケットの情報からセンシングを行うタイミングを把握し、そのタイミングでスリープ状態から起動する。同期パケットを受信した子ノードはパケットの所属ノードIDから自身のノードIDを見つけ、自身のノードがクラスタの何番目に所属しているのかを把握する。同期パケットを受信してから（センシング間隔×クラスタ内の順番）秒後にスリープ状態から起動してセンシングと送信を開始する。その後、センシング間隔とクラスタサイズを掛けたもの（秒）がセンシング間隔となり、その間隔でスリープ状態から繰り返し起動して、センシングと送信を行う。例えば、センシング間隔10秒、所属ノードが2番目、クラスタサイズが3である場合には同期パケットを受信してから 10×2 秒後に起動してセンシングと送信を行う。その後 10×3 秒間隔でスリープ状態から起動してセンシングと送信を行うことになる。

3.2. クラスタヘッド

クラスタヘッドは同期パケットを送信後、クラスタに所属するそれぞれの子ノードからのパケットを受信するまで常に起動している。各子ノードからの最初のパケットを受信したタイミングでマージン分早く起動するようにタイマーをスタートしスリープ状態となる。このタイマーによる制御においては、再起動のタイミングのずれや、起動後に受信可能になる時間を考慮して、ある程度のマージンを設定する。同期パケットで指定したクラスタの全ノードにわたるサンプリング間隔のタ

A Design and Implementation of Power-saving Communication Control on Multi-hop Sensor Network

[†]Hiroshi Iwasaki and Toshio Hirotsu

[†]Hosei University

イミングがマージンの中央に来る用に再起動のタイマーを設定する. 図2に, 全ノードでみたサンプリング間隔10s, マージン800msの場合のタイミング制御を図示する.

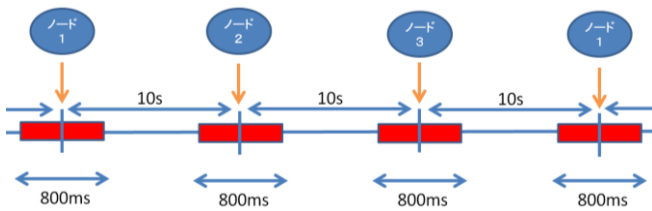


図2 クラスタヘッドの起動時間と受信

3.3. 同期の回復

クラスタヘッドと子ノードがそれぞれのタイマーにより制御されるため, 各ノード毎のタイマーの精度の違いに起因して, 長時間連続稼働すると同期が外れることが考えられる. そこで, 同期が外れた場合に回復の処理が必要となる. ここでは, 受信をできなかった場合に次の受信するタイミングのマージンを前後に長く取り起動する時間を一時的に長くする. また, クラスタヘッドが子ノードからの通信を受信するたびにその受信したタイミングでタイマーを再スタートさせる.

4. 実装

Crossbow社のセンサノード MICA mote を用いて実装を行った. スリープ制御については一度だけ起動するワンショットタイマーと繰り返し起動するリピータタイマーを用いて実装を行った. クラスタヘッドは, リピータタイマーで受信するタイミングの 400ms 早く起動し, 起動した後ワンショットタイマーをスタートさせ 800ms 後にスリープ状態にする. このようにして受信するタイミングの前後に 800ms 起動するマージンを取った. リピータタイマーを用いるので受信する間隔で繰り返し起動する. また, 受信をできなかった場合に次の受信するタイミングのマージンを通常の 5 倍の 4000ms を取り起動する時間を一時的に長くした.

5. 評価

クラスタヘッドと子ノードの 3 ノードを用いて, 180 分クラスタヘッドが各子ノードから受信するパケット数を観測して評価を行った. 各子ノードは 30 秒毎にクラスタヘッドに送信を行う. 各子ノードは 10 秒ずつずらしてサンプリングを行い, クラスタヘッドに対して送信を行う. 10 分毎に各ノードのパケット到達率を集計したグラフを図3に示す. node1 は 26 分後に通信ができなくなった. また, node2 は 176 分後に通信ができなくなった.

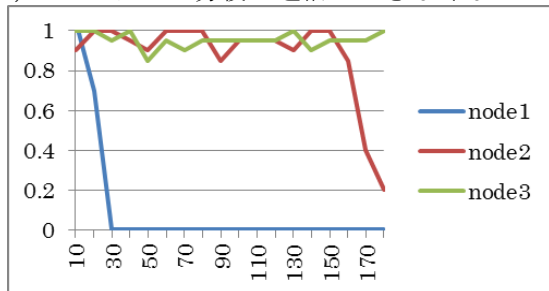


図3 10分毎に集計したパケット到達率

次に同期の回復を実装した結果を図4に示す. 図4のパケット到達率からわかるように今回の 180 分間の観測では通信ができなくなることはなかった. このことから, クラスタヘッドと子ノードの通信のタイミングがずれていき, 通信ができなくなる状態になることを防ぐことができた.

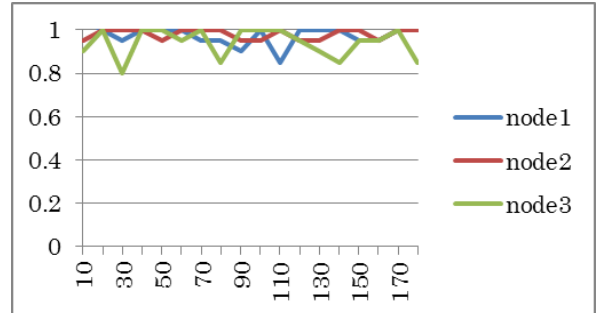


図4 10分毎に集計したパケット到達率(同期の回復)

6. 考察

本論文の手法では, 各ノードそれぞれの時計を用いてタイミング制御しており, ノード間での速度の調整は行っていない. そのため, 長時間運用すると精度の違いが積み重なってタイミングがずれてしまう可能性がある. これに対しては, 同期の回復のしくみが正しく機能し, タイミングの回復が行われることが確認できた. クラスタヘッドと子ノードとの同期を回復する方法としては本提案手法で用いたクラスタヘッドがタイミングをずらす方法以外にも, 子ノードが送信のタイミングを変える方法が考えられる. これにより多数ある子ノードの起動時間は最小限に抑えることができるが, 子ノードがタイミングの調整をしようとすると, 親ノードとタイミングが合っているかどうかの返戻を待つて確認する間起動し続けなければいけない. そこで本研究では, クラスタヘッドから子ノードへ通信を行わなくても同期を回復することができる後者の方法を取った. しかし, タイミングのずれをクラスタヘッドだけが吸収しているため, 長期にわたる運用ではサンプリングタイミング自体が非均質になる可能性がある.

7. まとめ

本研究ではスリープ・通信タイミング制御の実装を行った. センサネットワークの省電力化を行うためには, 通信やセンシング以外ではスリープを行うことで消費電力を抑えることが有効であると考え, スリープ制御・通信タイミング制御の実装を行った. 今後の課題として, 一定時間経過後クラスタ内で再同期を行いクラスタ内の子ノード同士の起動タイミングの調整が挙げられる.

8. 参考文献

- [1] 高橋 佑吾, 廣津 登志夫, 省電力センシングにおけるデータ均質性の向上, 情報処理学会第74回全国大会 p131-132
- [2] 西谷 信之介, 廣津 登志夫, 福田 健介, 菅原 俊治, 栗原 聡, 環境の変動に自律追従する省電力情報収集基盤, マルチメディア通信と分散処理ワークショップ論文集, 2008(14), 85-90, (2008)