

大規模センサネットワークにおける 階層型ソフトウェア更新手法の検討

橋詰葵[†] 峰野博史[†] 水野忠則[†]

[†] 静岡大学情報学部

1 はじめに

近年、MEMS 技術や低消費電力無線通信の発展により、無線センサネットワーク (WSN) を実現する環境が整いつつある。WSN は通信機能を付加した小型センサ (センサノード) により構成される自律的ネットワークであり、センシングデータをもとに様々なアプリケーションやサービスを提供する。比較的新しい技術である WSN は発展途上の技術を多く含んでおり、これらの技術は運用中に変更が生じる可能性が高い。そのため、センサノード上のソフトウェアを更新する技術は WSN において重要なアプリケーションの一つとなりうる。現在、無数のセンサノードへ一括してソフトウェアを配送する手法として、無線マルチホップ通信を利用したものが研究されており [1]、こうしたソフトウェア更新手法はセンサノード上にリプログラミングプロトコルとして実装されている。リプログラミングプロトコルはソフトウェアデータをネットワーク全体へ拡散させることを目的としており、完全なデータは基地局から配送される。既存研究の多くは基地局を一つに限定しており、複数の基地局を用いたソフトウェア更新は考慮されていない。単一基地局環境ではネットワーク規模の拡大に伴いソフトウェア更新時間が増加すると考えられる。この課題を解決するために、複数の (疑似的な) 基地局を用いてネットワークを階層的に扱うことで高速にソフトウェアを更新する手法 (階層型ソフトウェア更新) が研究されている。階層型ソフトウェア更新では基地局の配置がソフトウェア更新性能に影響を与える。しかし既存の階層型ソフトウェア更新プロトコルでは基地局の配置について議論されていない。

そこで、本稿では基地局の数と配置の観点から階層型ソフトウェア更新について検討を行う。提案手法に基づきシミュレーションを行い、ソフトウェア更新時間、ネットワークトラフィック、消費電力について評価する。

以下、第 2 章ではソフトウェア更新や基地局の配置手法に関連した研究について述べ、第 3 章では提案手法について述べる。第 4 章でシミュレーション評価を行い、最後にまとめと今後の検討について述べ、本稿を終わる。

2 関連研究

2.1 階層型ソフトウェア更新

階層的にソフトウェア更新を行うプロトコルとして、Firecracker[2] や Sprinkler[3] が知られている。Firecracker では、まず基地局からネットワーク上の任意のノード (スーパーノード) へとソフトウェアデータを配送する。スーパーノードへの配送が終了した後、スーパーノードは自身の周囲のノードへとソフトウェアデータを配送する。Sprinkler ではネットワークをクラスタ化し、基地局は各クラスタヘッドへとソフトウェアデータを配送する。クラスタヘッドはソフトウェアデータを受信し終えると自身のクラスタに属するノードへとソフトウェアデータを配送する。

これらのプロトコルはネットワークを階層的に扱うことでソフトウェア更新の高速化を実現している。しかし、スーパーノードやクラスタヘッドといった疑似的な基地局の配置については議

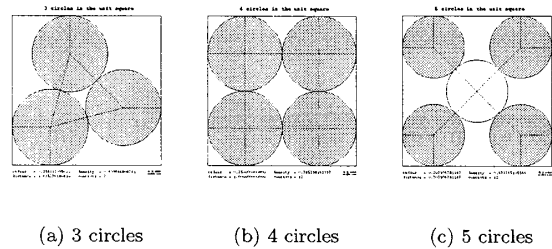


図 1: 円の詰め込み問題

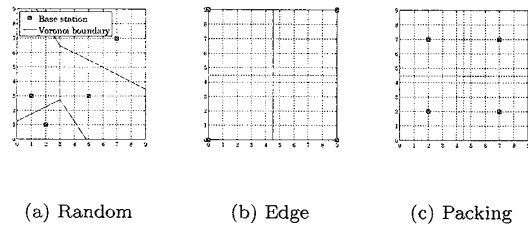


図 2: 基地局数 4 の場合の各トポロジ

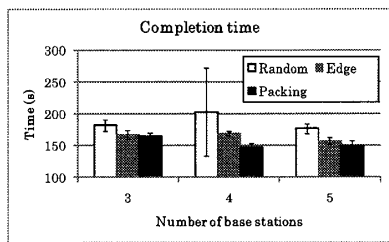
論されていない。これらの配置を考慮することでソフトウェア更新性能を向上させられると考えられる。

2.2 施設配置問題

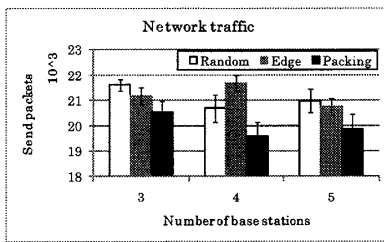
センサネットワークの課題の一つとして、効率的なネットワークの構築があげられる。効率には複数の意味が考えられるが、ここではソフトウェア配送の高速化やネットワークトラフィックの削減、センサノードの消費電力の削減に関する効率のこととし、またネットワークの構築とは基地局の配置のこととする。この基地局の配置問題は、施設配置問題の一種と考えることができる。施設配置問題とは、対象の地域に任意の施設 (学校、病院、デパートなど) を配置する際に、経済性や利便性といった観点から最適な配置を考える数理問題である。この問題を解決するために、図形の詰め込み問題や被覆問題、ポロノイ図などが利用されるが、本稿では正方形への円の詰め込み問題を応用した。この詰め込み問題の応用による基地局の配置手法について次章で述べる。

3 提案手法

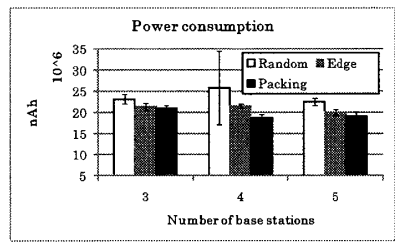
各基地局からのソフトウェア配送の波 (プロパゲーション・ウェーブ) が互いに干渉する際、その影響が最も小さくなる配置を考える。平面上のネットワークにおいて、基地局を中心として同心円状に等しい速さでソフトウェアが拡散していくと仮定すると、各基地局のプロパゲーション・ウェーブが互いに接したときに、それらの同心円の面積が最大となる配置が望ましいと考える。これは、円の詰め込み問題と置き換えることができる。例えば正方形の中に互いに面積が等しい円を詰め込む場合、図 1 に示すように円の個数によって最適な配置が知られている [4]。この円の数を基地局の数として、円の中心となる位置に基地局を配置する。以下ではこの配置手法を Packing と呼ぶ。



(a) ソフトウェア更新完了時間

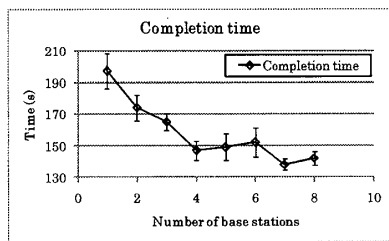


(b) ネットワークトラフィック

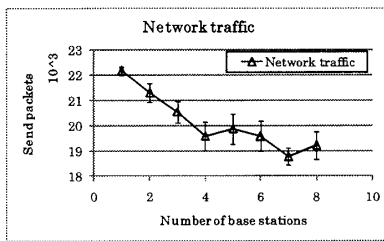


(c) 消費電力

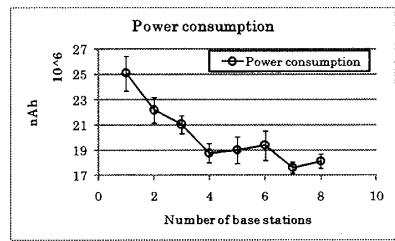
図 3: 基地局の配置手法の比較



(a) ソフトウェア更新完了時間



(b) ネットワークトラフィック



(c) 消費電力

図 4: Packing 手法による基地局数の比較

4 シミュレーション評価

4.1 シミュレーション環境

TinyOS[5] ネットワークシミュレータ TOSSIM[6] を用いて Packing をシミュレーション評価した。センサノード 100 台が 10×10 の格子状に配置された環境を想定した。センサノードは 40 feet 間隔で設置されており、ノードの通信半径は 50 feet となっている。配送するソフトウェアのサイズは 80 パケットであり、このソフトウェアデータを 10 個のセグメントに分割して配送する。プロトコルには MNP[7] を使用しており、パケットロス隠れ端末問題など MAC レベルで発生するもののみを想定している。

また、Packing の比較対象として Random と Edge という 2 つの手法を用意した。図 2 に基地局数が 4 の場合における各手法のトポロジを示す。着色された正方形は基地局を表し、実直線は基地局を母点とした場合のポロノイ境界を表している。Random は一様乱数により基地局の座標を決定したものである。Edge は基地局がネットワーク端に位置し、かつ各基地局を母点とした場合のポロノイ領域の面積が等しくなるような配置である。

以下ではまず Random, Edge, Packing の各手法を比較し、基地局の配置とソフトウェア更新性能の関係を調査する。次に Packing について、基地局数を 1 から 8 まで変化させ、基地局数とソフトウェア更新性能の関係を調査する。調査項目はソフトウェア更新時間、ネットワークトラフィック、消費電力の 3 項目とする。

4.2 基地局の配置

図 3 に基地局の配置手法を比較した結果を示す。全体の傾向として Packing, Edge, Random の順に優れているとわかる。また、すべての結果において Packing は最も優れた値を示している。このことから、提案手法は有効な基地局配置手法といえる。

4.3 基地局数

図 4 に Packing 手法を用いて基地局数を比較した結果を示す。基地局数が 1 から 4 へと増加する過程では基地局数の増加に伴いソフトウェア更新性能も向上しているとわかる。しかし基地局数が 5 以上になるとソフトウェア更新性能が単調増加することはなくなり、性能変化は不安定になっている。また、基地局数が 4 以下の場合と比較して、性能の変化が緩やかになってい

る。これらのことから、あるネットワークサイズに対して効率的にソフトウェア更新性能を向上させられる基地局数には上限があると考えられる。

5 終わりに

本稿では階層型ソフトウェア更新を行う際の基地局配置手法について検討を行った。またシミュレーション評価により、提案手法が有効なこと、及び基地局数の増加が必ずしもソフトウェア更新性能の向上に繋がるわけではないことを確認した。

今後の課題として、対象とするネットワーク規模に適した基地局数の導出が考えられる。本稿では 100 台の格子状ネットワークで評価を行い、望ましい基地局数は 4 台前後とした。しかしネットワーク規模が拡大した場合、そのネットワークに適した基地局数は本稿で示した値よりも大きくなると考えられる。様々な規模で評価を行うことで、ノード台数と最適な基地局数の関係の定式化が可能と考えられる。

参考文献

- [1] Qiang Wang, Yaoyao Zhu, Liang Cheng: Reprogramming wireless sensor networks: challenges and approaches. IEEE Network 20(3), pp. 48-55, 2006.
- [2] Philip Levis and David Culler: The Firecracker Protocol. Proc. 11th ACM SIGOPS European Workshop, 2004.
- [3] V. Naik et al.: Sprinkler: A Reliable and Energy Efficient Data Dissemination Service for Wireless Embedded Devices. 26th IEEE Real-Time Systems Symposium, pp. 277-286, 2005.
- [4] Packing of circles in the unit square: <http://hydra.nat.uni-magdeburg.de/packing/csq/csq.html>
- [5] TinyOS: <http://www.tinyos.net/>
- [6] Simulationg TinyOS Networks: <http://www.cs.berkeley.edu/pal/research/tossim.html>
- [7] S. S. Kulkarni and L. Wang: MNP: Multihop Network Reprogramming Service for Sensor Networks. In: Proc. IEEE ICDCS, pp. 7-16, 2005