パイプライニングを用いた WSN におけるソフトウェア配送の効率化に関する検討

橋詰葵† 宮丸卓也†† 峰野博史† 寺島美昭‡ 徳永雄一‡ 水野忠則†

†静岡大学情報学部 ††静岡大学大学院情報学研究科 ‡三菱電機株式会社

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ワイヤレス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 では通信機能を付加した小型センサ（センサノード）を利用しており，発展途上の技術を含む WSN ではこのセンサノード上のソフトウェアを更新するリプログラミングが重要なサービスの 1 つとなる．リプログラミング手法として，近年では無線マルチホップ通信（図 1）を用いたワイヤレスリプログラミングという技術が注目されており，省電力かつ高速にソフトウェアを配送することが課題となっている．

リプログラミングのソフトウェア配送では，サイズの大きなソフトウェアデータを基地局から末端センサノードまで拡散させることを目標としている．そのため，大容量のデータを高速に配送するためにパイプライニングという技術が研究されている (Deluge[1], MNP[2])．パイプライニングではソフトウェアデータを複数のセグメントと呼ばれる単位に分割し，セグメントを並列に配送することで高速化を実現している．パイプライニングではセグメントを細かく分割すればするほど並列度が増すため，データの配送を高速に行うことができる．しかし同時に各セグメントに付随するコントロールメッセージが増加し，消費電力の増加や信頼性の低下を引き起こすという問題がある．

そこで本論文ではパイプライニングの性能評価を行い，セグメント分割によるソフトウェア配送の高速化を確認する．また，コントロールメッセージの増加により引き起こされる問題についても確認する．そして問題を解決する手法として，ローカルパイプライニングを提案する．

以下，第 2 章ではパイプライニングについて述べる．第 3 章ではシミュレーションによりパイプライニングの評価を行い，セグメント分割による影響を確認する．そして現状のパイプライニングにおける問題点や課題を整理する．第 4 章では課題に対してローカルパイプライニングという手法を提案する．そして最後にまとめと今後の検討について述べ，本論文を終わる．

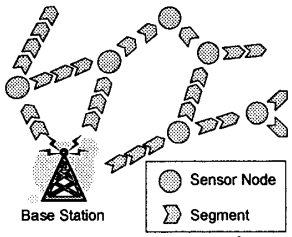

図 1: 無線マルチホップ通信

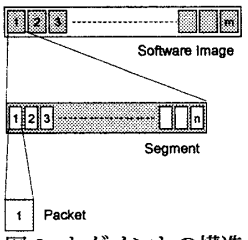

図 2: セグメントの構造

2 パイプライニング

2.1 概要

ワイヤレスリプログラミングでは，信頼性，電力効率性，配送時間という 3 つの設計目標が重要視されており，パイプライニングは配送時間の短縮のために提案された．パイプライニングでは，図 2 のようにソフトウェアデータを複数のセグメント単位に分割する．さらにセグメントは複数のパケットと呼ばれる配送単位で構成される．図 3 はセグメントに分割した場合とそうでない場合の配送を比較した図であるが，分割することでセグメントをオーバーラップさせて配送することが可能である．図 3 の配送において一定の間隔を空けているのは，隠れ端末問題を避けるためにセグメントの送信間隔を少なくとも 3 ホップ空けておく必要があるからである．

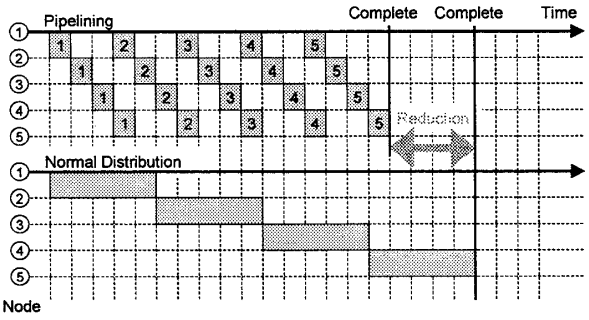

図 3: パイプライニング

2.2 ネゴシエーション

パイプライニングでは複数のセグメントを扱うため，必要なセグメントの情報を管理しなくてはならない．そこで，必要なセグメントのみをやり取りするためのネゴシエーションが必要である．ネゴシエーションでは ADV, REQ, DATA という 3 種類のメッセージを持つ 3 ウェイハンドシェイクを利用する．最初にソースノードは，自身が保持しているセグメント番号を ADV メッセージを通じて近隣ノードへ告知する．ADV メッセージを受信したノードは，メッセージ中のセグメント番号と自身のセグメント番号を比較し，もし持っていないセグメントがあれば REQ メッセージによりセグメントを要求する．ソースノードが REQ メッセージを受信したら初めて DATA メッセージを送信する．こうした 3 ウェイハンドシェイクの採用により，不必要なセグメントの送受信がなくなり，冗長なトラフィックの増加を抑えることが可能となる．

3 評価

3.1 シミュレーション環境

TinyOS[3] ネットワークシミュレータ (TOSSIM[4]) を用いてパイプライニングをシミュレーション評価する．本シミュレーションではパイプライニングによる配送の高速化を確認する．また，現状のパイプライニングの問題点についても確認する．シミュレーションにはパイプライニングを実装しているリプログラミングプロトコルである MNP を用いる．

本シミュレーションにおいて，センサノードは半径 50 feet の通信範囲を持つものとする．ノードは直線状に 100 台配置されており，各ノード間の距離は 40 feet である．配送するデータ量は 160 パケットであり，このデータのセグメント分割数を 4 通りに変化させて（分割なし，5 分割，10 分割，20 分割）調査する．以上の環境で，完全なデータを持つ基地局からデータが配送されることになる．

3.2 メッセージ量

図 4 に (a) 総トラフィック量，(b) データ配送量を示す．このグラフから，セグメント分割数が増せば増すほど時間当たりの総トラフィック量やデータ配送量が増加していることが分かる．そ

An Examination of the Efficient Software Distribution with Pipelining for Wireless Sensor Network

Aoi Hashizume†, Takuya Miyamaru††, Hiroshi Mineno†,
Yoshiaki Terashima‡, Yuichi Tokunaga‡, Tadanori Mizuno†
†Faculty of Informatics, Shizuoka University,
††Graduate School of Informatics, Shizuoka University,
‡Mitsubishi Electric Corporation

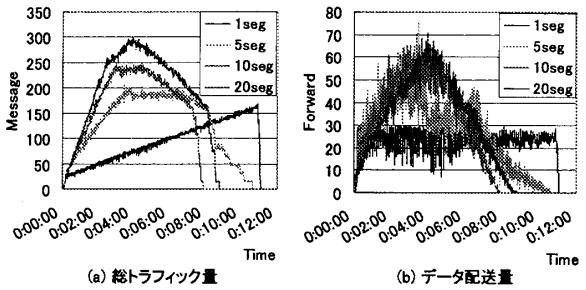

図 4: メッセージ量

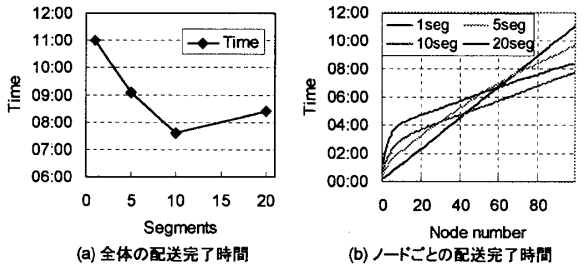

図 5: ソフトウェア配送完了時間

の結果，分割なしの場合に比べて高速に配送が完了している．

3.3 ソフトウェア配送完了時間

図 5 に (a) 全体の配送完了時間，(b) ノードごとの配送完了時間を示す．図 5 (a) から，セグメントを分割しない場合に比べて分割した場合の方が高速に配送が完了していることが分かる．

図 5 (b) から，セグメントを分割しない場合は終始一定の間隔で配送が完了していることが分かる．これと比較して，分割した場合では配送開始直後は多くの時間を要しているが，ある一定の時間が経過した後は短い間隔で配送が完了していることが分かる．また，分割数が大きければ大きいほどこのパイプライニングの特性が顕著に表れているということも分かる．前節と本節の結果から，パイプライニングによりソフトウェア配送完了時間が短縮できることが確認できた．

3.4 現状の問題点と課題

パイプライニングでは分割数が増せば増すほど配送が高速で行われることに対し，図 5 (a) では 20 分割の場合よりも 10 分割の場合の方が高速となっている．これは，20 分割の場合では総トラフィック量に対するコントロールパケット数が増大し，結果として単位時間当たりのソフトウェアデータ配送量が 10 分割の場合に比べて少なくなったためである．この結果から，パイプライニングを用いた配送において，単純にセグメント分割数を増加させることが高速化へと繋がるわけではないということが分かる．そのため，ノード数やデータ量といった環境条件から，最適なセグメント分割数を見つける必要があると考えられる．また，セグメント分割に伴うコントロールパケットの増加は，電力消費量の増加や信頼性の低下を招く．以上の理由から，コントロールパケットをできる限り増やさずに効率的な配送を実現できる仕組みが必要だといえる．

4 ローカルパイプライニングの提案

4.1 概要

前章で挙げた課題に対して，本論文ではローカルパイプライニングという配送手法を提案する．ローカルパイプライニングでは，まずノードを複数のグループに分ける．そしてグループごとにセグメント分割数を割り当てることによって，コントロールパケット数や消費電力をグループの状態に応じて柔軟に変更可能にするという仕組みである．Deluge や MNP で使われている従来のパイプライニングは，セグメント分割数をネットワーク全体で固定している．そのため，例えばバッテリ残量の少ない一部のノードのためにセグメント分割数を少なくすると，他のノードの配送パフォーマンスまで低下してしまう．これはネットワーク全体の配送効率を考えると適切とはいえない．こうした問題に対処

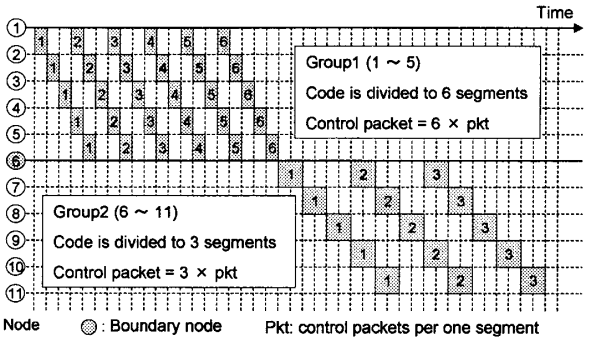

図 6: ローカルパイプライニング

すべく，ローカルパイプライニングでは配送パフォーマンスを維持したまま現実的な配送ができることを目標としている．

4.2 配送過程

図 6 はローカルパイプライニングの配送過程を示している．この図では，ノード 1 から 5 までを 1 グループ，ノード 6 から 11 までを 2 グループとしている．セグメント分割数は，グループ 1 では 6 分割，グループ 2 では 3 分割と仮定した．コントロールパケットについては，1 セグメントあたり pkt 個のメッセージが必要だとすると，グループ 1 では $6 \times pkt$ 個，グループ 2 では $3 \times pkt$ 個のコントロールメッセージを扱うこととなる．

ローカルパイプライニングでは異なるセグメント分割数を持った複数のサブネットワークを扱うため，セグメントの配送途中でセグメント分割数を変更しなければならない．この処理はグループの末端に位置する境界ノードによって行われる．例えば図 6 では境界ノードはノード 6 となる．境界ノードはすべてのセグメントを受信し終えてから，新たに自分のグループのセグメント分割数で配送を開始する．完全なデータを揃えてから配送を開始する理由は，セグメント分割数の変更によりメッセージ衝突などが起こり，他の配送に影響を与えることを避けるためである．また，境界ノードは完全なデータを保持しているため，基地局と等しい役割を果たすとも考えられる．

5 終わりに

本論文のまとめを述べる．パイプライニングのシミュレーション評価を行い，その効果を確認した．また，パイプライニングの問題点と課題を整理した，そしてグループごとに分割数を自由に調整可能な手法としてローカルパイプライニングを提案した．セグメント分割数を調整することで，電力消費量の大きいコントロールパケットの量を調整できる．よって，提案手法を用いることで電力効率のよい配送を実現できると考えられる．

最後に今後の検討について述べる．まず，パイプライニングの性能を多様な環境で調査する必要がある．ノードの配置を格子状やランダムに変化させる，配送するデータ量を増減させるなどしてシミュレーションを重ねることで，ノードの数や配置，配送するソフトウェアデータ量といった環境条件から最適なセグメント分割数を導き出すことが可能になると思われる．また，ローカルパイプライニングを実装し，評価しなければならない．さらに，セグメント分割数やグループ化の基準となる他の環境条件を検討する必要がある．

参考文献

[1] Hui, J.W., Culler, D.: The Dynamic Behavior of a Data Dissemination Protocol for Network Programming at Scale. In: Proc. ACM SenSys 2004, pp. 81-94. ACM, New York (2004)

[2] Kulkarni, S.S., Wang, L.: MNP: Multihop Network Reprogramming Service for Sensor Networks. In: Proc. IEEE ICDCS, pp. 7-16 (2005)

[3] TinyOS: http://www.tinyos.net/

[4] Simulationg TinyOS Networks: http://www.cs.berkeley.edu/ pal/research/tossim.html