彩色アルゴリズムを用いたマルチシンクセンサネットワークにおける分散データ集約手法

川野 亮平† 小野寺 克美† 宮崎 敏明††

会津大学大学院コンピュータ理工学研究科† 会津大学コンピュータ理工学部††

1 はじめに

マルチシンクセンサネットワークでは，各センサノードが最近傍のシンクにデータを集約するようにすることが求められる．また，パケット衝突を回避し，効率的なデータを集約するために TDMA(Time Division Multiple Access)通信は有効な手段である．本稿では，マルチシンクおよび TDMA を採用するセンサネットワークにおいて，効率的なデータ集約を実現する彩色アルゴリズムに基づいた手法を提案する．また，シミュレーションによる実験結果を示し，提案手法の有効性を示す．

2 センサネットワークモデル

本稿では，災害現場や山中などのインフラ整備が困難な場所に多数のセンサノード(以下，ノードと略す)をばら撒き，ノードの協調作業により自律分散ネットワークを構築するワイヤレスセンサネットワーク(WSN)システムを想定する[1]．図 1 に我々が想定する WSN を示す．WSN の主要機能である気温や湿度などの環境情報の集約を効率的に行なうために，複数のシンクを設け，各ノードは近傍のシンクにデータ転送を行うことで省電力を図り，ネットワーク全体の稼働時間を増加させることを目標とする．また，パケット衝突を回避するために TDMA 通信を用いることにするが，そのための動的タイムスロット割り当て法も併せて提案する．

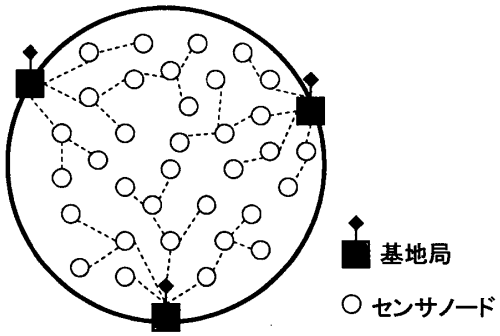

図 1 ：マルチシンクセンサネットワーク

3 提案アルゴリズム

図 2 に使用する変数の定義を示し，図 3 に提案アルゴリズムの概要を示す．提案アルゴリズムでは，次の 2 つの処理を行なう．すなわち，(1)マルチシンクセンサネットワーク構築と(2)パケット衝突回避のためのタイムスロット割り当て，である．これら 2 つの処理は，一定時間パケット受信を待つ処理 *Listen(T)* と共に，交互に繰り返し実行される．また，パケットを受信した場合は現在の処理を中断しリスト *P* に受信したパケットの情報を追加する．

3.1 マルチシンクセンサネットワーク

複数のシンクを導入することで，1 つのシンクが故障した場合でも他のシンクにデータを集約することが可能な頑健なデータ集約システムを構築することができる．

注記:
self: **処理を行なうノード自身.**
|x|: **リストxの要素数**
V: **近隣ノード情報のリスト**(*v*[0], *v*[1], ... , *v*[|*V*| − 1])
S: **シンク情報のリスト**(*s*[0], *s*[1], ..., *s*[|*S*| − 1])
P: **受信したパケットのリスト. パケットの受信時に更新される.**
(*p*[0], *p*[1], ..., *p*[|*P*| − 1])
ID : **各ノードまたはシンクを識別する数字**. この値は重複しない.
$0 \le$ シンクの*ID* $< |S| \le$ ノードの*ID*
parent: 親ノード
level: ノードのレベル
M: **彩色数**, i.e., **使用可能な色数.**
color: 色の数. **この数は整数である.** $0 \le color < M$.
stress: 彩色アルゴリズムで使用する整数
threshold: 彩色の失敗数. 本稿では１０.
T: **リスニング時間. 初期値が与えられる**

図 2： 提案手法で使用する変数と定数

```
procedure algorithm(M, T, S){
  int rand = generateRandomNumber(); //  0≤rand<M
  self->color = rand;
  self->level = ∞;
  self->parent = ∞;
  self->stress = 0;
  P.clear(); // list P の初期化

  for(i = 0; i < |S|; i++) {
    self->s[i]->ID =  ∞;
    self->s[i]->level = ∞;
 }

 // もしパケットを受信したら，メインの処理を中断し
 // List Pに受信したパケットの情報を追加する
 setupPacketReceiveHandler();

  repeat{
    (親と最近傍シンクの更新処理)
    listen(T); // 時間Tの間パケット受信待ち
    changeColor(M);
   T = (割り当てられた色に基づいた時間を決定する);
  }
}
```

図 3： 提案アルゴリズムメインルーチン

また、シンクおよびパケット中継ノードの負荷が分散できるためネットワークの長寿命化につながる．本稿では各ノードが最近傍のシンクにデータを伝達することでマルチシンクセンサネットワークを構築する．つまり，木の深さを示す *level* と親ノードを示す *parent* を設定し，図 1 のような木構造を構築する．

3.2 TDMA 通信のためのタイムスロット割り当て

WSN ではノードがパケット衝突を検知した場合，その

Distributed Data Aggregation in Multi-sink Sensor Networks using a Graph Coloring Algorithm

† Ryouhei Kawano, † Katsumi Onodera, †† Toshiaki Miyazaki
† Graduate School of Computer Science and Engineering, The University of Aizu
†† School of Computer Science and Engineering, The University of Aizu

ノードは再送を行なわなければならないため，冗長な電力消費が発生する．つまり，低消費電力が求められるWSN においてパケット衝突は大きな問題となる．提案手法ではこのパケット衝突の問題を TDMA(Time Division Multiple Access)通信を実現することで解決する．TDMA では一定の時間周期で多数のタイムスロットと呼ばれる時間単位に分割し，各ノードに近隣ノードと通信のタイミングが重ならないようにタイムスロットを割り当てる．各ノードは，割り当てられたタイムスロットに基づいてパケットの送信を行なう．図 4 にタイムスロット割り当ての例を示す．図 4 のノード A の 2 ホップ内のノードに同一のタイムスロットを割り当てているノードが存在しないため，衝突が無く通信が行える．

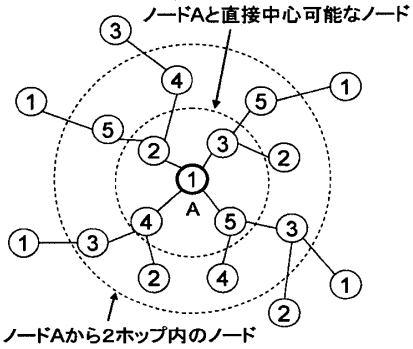

図４：タイムスロット割り当て．各数字はタイムスロットを示す

このタイムスロット割り当てを実現するために，著者等が以前提案した分散彩色アルゴリズム[2]に変更を加えて適用する．TDMA 通信を実現するためには全てのタイムスロットに別々の色を割り当てる必要がある．つまり，各ノードは 2 ホップ以内のノードとは異なるタイムスロットを割り当てるべきである．本アルゴリズムでは，ある一定値 *threshold* 回連続して衝突なしにタイムスロットを割り当てられないノードはパケットの送信を行なえないものとする．図 5 にタイムスロット割り当て手法の擬似コードを示す．

4　性能評価

提案アルゴリズムの効率性を評価するためにパケットの衝突数に関するシミュレーション実験を行なった．実験結果を図 6 に示す．図 6 の x 軸は時間ユニット数を示し，y 軸はパケットの衝突数を示す．実験は 9 種類mの異なるノード数を持つグラフに対して各 200 回試行し，その平均値を算出した．図6中，各線は，ノード総数の違いを表している．各試行では 4000 ユニット時間シミュレーションを行い，100 ユニットごとに評価を行なった．図 6 よりノード数が 200，300，400 のグラフでは，ほぼ全てのノードで衝突なしに通信が行えることが分かる．また，ノード数が 1000 の場合，多くのノードで通信が行えないが，初期状態に比べて約 1/70 にパケットの衝突数が減少していることがわかる．これより，提案手法は，パケット衝突に関して，効果的であることを示している．

5　終わりに

彩色アルゴリズムを使用したデータ集約手法を提案し，シミュレーション実験より提案手法の有効性を示した．今後は実機による評価を行ない，本手法の実用性を検証する．

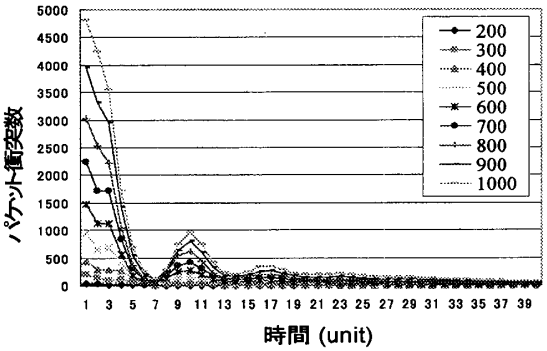

図６：パケット衝突数

```
procedure changeColor(M){
 int usedColor[M];
 int next, rand, conflict  =  0;

 if(stress < 0) {
   stress++;
   return;
 }
 create list V; // 2ホップ内の上位のレベルか同等の
                // レベルのノードのリストを作成する
 for(i=0; i<M; i++) usedColor[i]=0;  //初期化
 for(i=0; i<|V|; i++){
   usedColor[v[i]->color] = 1;
   if(self->color == v[i]->color) conflict = 1;
 }
 if(conflict==1){
   stress++;
   if(stress > threshold){
     rand = generateRandomNumber(); // 0 ≤ rand < threshold
     stress = -rand;
   }
   if(0<=stress && stress <= threshold){
     usedColor[self->color]++;
    next=(usedColor[]から最小のインデックスを
                                        取得する);
     if(stress != 0) self->color = next;
     if(usedColor[color] == 0){
       conflict = 0;
       stress = 0;
     }else{
       if(stress == 0){
         rand = generateRandomNumber();
                                  // 0 ≤ rand < threshold
         stress = -rand;
       }
     }
   }
   return;
 }
 (近隣にデータをブロードキャストする);
}
```

図５：タイムスロット割り当てアルゴリズム

謝辞

本研究の一部は，日本学術振興会科学研究費補助金(課題番号:18500060)およびテレコム先端研究支援センター(SCAT)研究助成金による．

参考文献

[1] I.F Akyldiz et al, "Wireless Sensor Network: A Survey", Comp. Networks J, vol.38, no.4, pp.393-422, Mar. 2002

[2] R. Kawano, and T. Miyazaki, "Distributed Coloring Algorithm for Wireless Sensor Networks and Its Applications," Proc. 7th IEEE International Conference on Computer and Information Technology (CIT2007), Oct. 2007.