

# ユビキタスセンシングシステム GlocalGrid における並列処理について

塚本昌彦<sup>†</sup> 藤田直生<sup>†</sup>

筆者らはローカル通信とグローバル通信を併せ持つユビキタスセンシングシステム GlocalGrid を提案・実装している。本稿では GlocalGrid を用いて、グローバル通信とローカル通信を組み合わせることで高度な並列計算を可能にするいくつかの手法について述べる。特に 2 近傍情報、4 近傍情報、8 近傍情報をネットワーク全体で同期をとりながら集めるステップ数について、ノードがデータ送信中の受信ができる場合とできない場合に分けて考察を行う。提案手法を用いて非常に大規模なネットワークにおいて画像処理のようなセンサデータの分散並列処理が行えるようになる。

## On Parallel Processing on a Ubiquitous Sensing System ‘GlocalGrid’

Masahiko Tsukamoto<sup>†</sup> and Naotaka Fujita<sup>†</sup>

The authors proposed and implemented a ubiquitous sensing system GlocalGrid, which provides local and global communication methods. In this paper, we show several algorithms of advanced parallel processing by using these two types of communication methods on GlocalGrid. We especially focus on the step number for collecting 2 neighbor, 4 neighbor and 8 neighbor information by synchronizing the whole network for the case when each node can receive a message while it sends a message, and the case when it cannot. By using our proposing methods, we can achieve distributed parallel processing of sensor data in a similar manner to image processing in a very large scaled network.

### 1. はじめに

人々の生活環境や海や山などの大規模フィールドなどの実世界にセンサを大量に配置し、それを環境センシングやモニタリング、安全性の管理などに統合的に利用するユビキタスコンピューティングが浸透している。近い将来には、人々の暮らしの中で多数の超小型コンピュータが広く使われることが見込まれている。しかしこれらをよく分散協調動作させるための技術は現在のところ不十分であり、これまでにない複雑かつ高度なプログラミングが求められる。効率よく多くのコンピュータを全体として制御したり、コンピュータ同士が自律的に連携を取り合ったりすることができるになれば、便利・快適、安全・安心、豊か・楽しさ、エコ、健康などといった全体的な目標の達成が容易になるものと考えられる<sup>1)</sup>。

筆者らの研究グループでは、これまでに多数のユビキタスデバイスを全体としてプログラムする『全体プログラミング』の研究を行っており、さまざまな手法やシステムを開発してきた。ユビキタスデバイスのマクロプログラミングに関する研究がなされている<sup>2)~6)</sup>が、従来のアプローチは個々のデバイスのプログラミングの延長であり、個別の ID を指定する必要があるなどデバイスの数に対するスケーラビリティがないのが現状である。筆者らはユビキタス環境において多数のユビキタスデバイスによる群コンピューティングを実現することを目標として、これまでにローカルとグローバルのトポロジの対応が比較的自明な格子状有線ユビキタスネットワーク GlocalGrid を提案し、実装を行った<sup>7)</sup>。格子状ネットワークではホップをカウントすれば座標値が得られるなど分散コンピューティングを行う上で様々な優位性を有するため、ルーティングや形状検出などに関するさまざまな研究が行われている<sup>8)~10)</sup> (ただし Zhang らのもの<sup>8)</sup>は若干トポロジが異なる)。一方、大規模並列計算にはブロードキャストが有効であり<sup>11)~13)</sup>、GlocalGrid では、グローバルで全体の同期と指示を行いながら、ローカルに情報交換することを可能にしている。これまでの研究では、GlocalGrid におけるトポロジ情報の設定や形状不良の検出方法などのアルゴリズムについて議論した。

本稿では GlocalGrid の形状を縦横一定数が並んだ長方形の形状に限定して、効率的な並列計算の方法について議論する。特に 4 近傍情報や 8 近傍情報を集めることで、平均やエッジ検出などの画像処理と同じ手法がセンサネットワークに対して適用でき、ローカル計算だけで実空間の特徴情報を抽出できることになる。LED やアクチュエータなどを用いて、エッジ部などの特徴点に対してローカルな措置が施せることになる。特に、各ノードがデータ送信時に受信ができるかどうかについて二つの場合に分けて、近傍情報の交換にかかるコストを考える。結果として、効率的に近傍情報を集める方法を示す。

<sup>†</sup> 神戸大学  
Kobe University

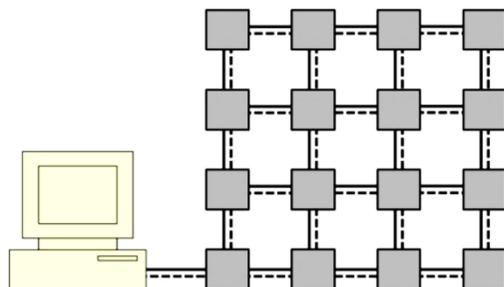


図 1. GlocalGrid のトポロジ例

## 2. ユビキタスセンシングシステム GlocalGrid

### 2.1 概要

GlocalGrid はグローバル通信とローカル通信の両者を備えるネットワークである。典型的なトポロジを図 1 に示す。図中、16 個の四角形がノードを表し、マイコンやセンサ、LED などを備えたユビキタスコンピュータを意味する。各ノードは縦横方向に規則正しく並んでいるものとする。本稿ではトポロジとして、縦  $M$  個、横  $N$  個の長方形のもののみを想定する。また、図では実線と破線で表される二種類の回線につながれているが、実線はグローバル通信を表しネットワーク全体でつながっているもの、破線はローカル通信を表し近傍間のみでつながっているものである。二種類の回線に対して必要に応じてホストコンピュータからコマンドを送信したり、情報を収集したりできるようになっている。通信の方式としては、グローバル、ローカルともに、一般的なシリアル通信を想定する。図 2 に実装したデバイスの外観を示す。デバイスにはマイコン、センサ、LED などが備えられている。

### 2.2 ノードの構成

各ノードは、上下左右の 4 方向に対して、ローカル通信およびグローバル通信でケーブルをつなげるコネクタを有する。ローカル通信とグローバル通信は、各ノードがいずれかを選んで送信ができるが、ノード A からローカルにメッセージ  $M$  を送出することを、

A,L: M

と記す。特定のローカル通信は上下左右の 4 方向に対して同時に送出されるものとし、メッセージが衝突しない限り、A に隣接するすべてのノードが  $M$  を受信する。各ノード



図 2 実装したデバイスの外観

は受信したメッセージがグローバルメッセージかローカルメッセージか、ローカル通信受信時には、上下左右のどちらの方向から受信したかが受信時に判別できるものとする。

ホストコンピュータからグローバルにメッセージ  $M$  を送出する場合は、

G: M

と記す。接続されたすべてのノードはメッセージ  $M$  を受信する。本稿でホストコンピュータから送信されるメッセージはすべてコマンドである。各ノードが出すメッセージはセンサ情報や近傍情報である。

図中で左上を原点、1 ホップごとに横方向に  $X$  軸、縦方向に  $Y$  軸とする座標系が導入されており、各ノードには自局の  $X, Y$  座標値が設定されているものとする。これらの設定の方法については筆者らの前論文で議論を行っている。

ホストコンピュータから特定のホストにメッセージ  $M$  を出すには、メッセージの先頭に条件部を付与する。このような動作を

G: [C] M

L: [C] M

と記す。C はホストを特定する条件であり、ID や座標のリスト、式などで表す。各ノードにおいて  $x$  は  $X$  座標値、 $y$  は  $Y$  座標値を示すものとする。例えば、

G: [(5,4),(2,1),(3,3)] M

は (5,4),(2,1),(3,3) の三つのノードに対してメッセージ  $M$  を送信すること、

G: [x % 3 = 1 && y % 3 = 2] M

は  $x$  座標値を 3 で割った余りが 1 で  $y$  座標値を 3 で割った余りが 2 であるようなノードにメッセージ  $M$  を送信することを意味する。本稿の後の部分では後者に類似したケースが何度か出てくるので、括弧内を  $x_3y_3$  と略記することとする。

ノードが送信時にはメッセージの受信ができる場合とできない場合が考えられる。送受信線が別で受信バッファを持っていれば送信時受信ができるものと想定できるし、そうでない場合は受信できないことが想定できる。前者を**送信時受信可能**、後者を**送**

**信時受信不能**と呼ぶことにする。本稿では両方の場合を想定して、二つの場合に分けて議論を行う。またいずれの場合も、送信時受信した場合でも、送信は継続できるものとする。

以下、GlocalGrid の並列計算問題を扱うが、問題自体には普遍性があるものと考えている。基本的には扱う問題は一般的な格子状のメッシュトポロジネットワーク上の情報交換の問題と類似しており、全体で同期さえ取れていれば適用できる手法もある。送信時受信に関しては、一般に単純な通信手法を用いている場合には難しいが、スペクトル拡散などで送信ノードごとにホッピングパターンや拡散方法を変えられる場合には送信時受信が可能なケースが想定できる。いずれにせよ GlocalGrid 以外のメッシュネットワークに議論が拡大できる可能性があるという意味で、議論には一般性があるものと考えられる。

### 2.3 コマンド

GlocalGrid は、センサネットワーク全体で分散並列のセンサデータ処理を行うことを意図している。典型的な利用パターンとしては、グローバル通信でホストからコマンドを送り、ローカル通信で近傍情報を交換しあって計算を行う。センサ情報の処理は画像処理のアルゴリズムを参考にし、4 近傍や 8 近傍の情報からフィルタやモルフォロジの演算を行い、その結果から何か動作を行っていく。動作としてここでは LED の点灯を取り上げ、例えばセンサ値のエッジ抽出をして、センサネットワーク全体を利用者が眺めることでフィールドを把握できるようにする。LED の点灯はラジコンヘリの誘導に使える可能性があるし、LED 以外にもスプリングラで水捲きをするようなアクションが考えられる。森林に張り巡らされたネットワークで野生動物を追跡するような応用例も考えられる。

グローバルおよびローカルで送るコマンドとしては、基本的な計算とデータの送受信、LED の点灯などのアクションを想定する。基本的な計算は適宜言葉や擬似プログラムとして記述する。例えば次のような書き方を考える。

G: 全センサデータを取得してセンサ ID と値のリストとして保持せよ。

G: [x<sub>3</sub>0] センサ 1 の値を取得してその値に比例して LED を点灯せよ。

下のコマンドは X 座標値が 3 の倍数であるノードに対して実行される。また、データ送受信を促すコマンドが用意される。

G: データを受信して c に格納せよ。

G: [x<sub>3</sub>0] 自局 ID と変数 c の値を送信せよ。

データ受信は別のコマンドが入ってくるとキャンセルされる。順次近傍情報を付加しながら情報交換を行える仕組みになっている。

本稿では、特定のデータの送信が 1 ステップの時間単位を要するものとし、計算等は瞬時に行えるものとする。例えば、基本的に X 軸方向、Y 軸方向ともに送信ノード

の隣接ノードが送信を止めると受信の妨害が起こらないので、X 座標、Y 座標の値を 3 で割った余りによって 9 回に分けて該当ノードが送信を行えば、全ノードが隣接情報を集めることができるようになる。つまり、

G:[x<sub>3</sub>0y<sub>3</sub>0] 自局情報を送信せよ。 G: 受信情報を保持せよ。

G:[x<sub>3</sub>1y<sub>3</sub>0] 自局情報を送信せよ。 G: 受信情報を保持せよ。

G:[x<sub>3</sub>2y<sub>3</sub>0] 自局情報を送信せよ。 G: 受信情報を保持せよ。

G:[x<sub>3</sub>0y<sub>3</sub>1] 自局情報を送信せよ。 G: 受信情報を保持せよ。

G:[x<sub>3</sub>1y<sub>3</sub>1] 自局情報を送信せよ。 G: 受信情報を保持せよ。

G:[x<sub>3</sub>2y<sub>3</sub>1] 自局情報を送信せよ。 G: 受信情報を保持せよ。

G:[x<sub>3</sub>0y<sub>3</sub>2] 自局情報を送信せよ。 G: 受信情報を保持せよ。

G:[x<sub>3</sub>1y<sub>3</sub>2] 自局情報を送信せよ。 G: 受信情報を保持せよ。

G:[x<sub>3</sub>2y<sub>3</sub>2] 自局情報を送信せよ。 G: 受信情報を保持せよ。

を順に一定時間あけて実行することにより、9 ステップで全ノードが隣接情報を得られることになる。さらに集めた近傍情報をもう一度同じ順序により全ノード間で交換すれば 2 ホップ先までの情報を交換できるようになる。つまり、

G:[x<sub>3</sub>0y<sub>3</sub>0] 得られた近傍情報を送信せよ。 G: 受信情報を保持せよ。

G:[x<sub>3</sub>1y<sub>3</sub>0] 得られた近傍情報を送信せよ。 G: 受信情報を保持せよ。

...

G:[x<sub>3</sub>2y<sub>3</sub>2] 得られた情報を送信せよ。 G: 受信情報を保持せよ。

というシーケンスを実行することで 2 ホップ先までの情報が得られる。これは 8 近傍情報を含むものであり、全ノードが 8 近傍情報を集めるには多くとも 18 ステップあればよいことが分かる。

4 近傍情報、8 近傍情報を集めると、方向微分、ソベルフィルタ、メディアン、ラプシアン、平均、オープニング、クロージングなどさまざまな計算が行え、例えば LED の点灯により値を出力することで、ビジュアルに利用者はセンサ情報の傾向を知ることができるようになる。

## 3. 送信時受信不能な場合のスケジューリング

### 3.1 基本設定

図 3 に示す 3×3 グリッドネットワークを例にとり、図 4 に情報交換の例を示す。a~i が各ノードの ID で、各ノードが近隣ノードとセンサ値を、①~⑤の 5 ステップで交換する。図 4 は a~i の各ノードの情報送信タイミング①~⑤と各ステップにおける送受信情報、最終的に得られた情報を示す。ノードの対応する位置にスケジュールが書かれている。この形式のダイアグラムをスケジュールダイアグラムと呼ぶ（送信タイミン

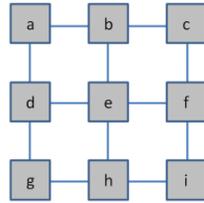


図 3. 3×3 グリッドネットワークの例

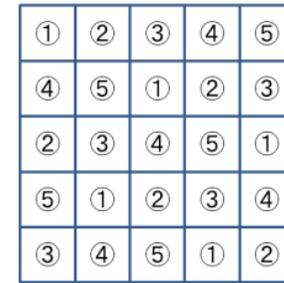
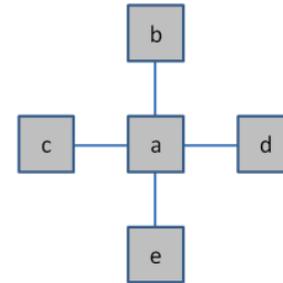


図 5. あるノードとその近傍ノード      図 6. 送信時受信不能な場合の排他的スケジュールの例

<b>a ①</b> ① (a) ② ab ③ - ④ adg ⑤ - abdg	<b>b ②</b> ① a ② (ab) ③ abcf ④ - ⑤ abcdefgh abcdefgh	<b>c ③</b> ① f ② ab ③ (abcf) ④ - ⑤ abcdefgh abcf
<b>d ④</b> ① * ② g ③ - ④ (dg) ⑤ abcdefgh abcdefgh	<b>e ⑤</b> ① f ② ab ③ gh ④ dg ⑤ (abcdefgh) abcdefgh	<b>f ①</b> ① (f) ② - ③ abcf ④ fghi ⑤ abcdefghi
<b>g ①②</b> ① (g) ② (g) ③ gh ④ adg ⑤ - adgh	<b>h ③</b> ① g ② g ③ (gh) ④ fghi ⑤ abcdefghi abcdefghi	<b>i ④</b> ① f ② - ③ gh ④ (fghi) ⑤ - fghi

図 4. 3×3 グリッドネットワーク上の情報交換の例

グ以外には必要に応じて省略する)。各ノードは、自局の送信タイミングにおいて、自局のセンサ情報に加えて自局がそれまでに受信している近隣ノードの情報を送信するものとする。丸カッコで囲まれた部分は情報を送信することを表す。太字は新規の有用な情報を表す。例えば、ノード a はステップ①で自局情報を送信する。同じタイミングで f も情報送信しており、ノード c,e,i が受信している。ステップ②では a は b から a,b の情報を受信する。a の情報は自局がステップ①で出した情報であり、a にとっては有用ではない。ステップ③では何も受信していない。送受信のない状況は「-」で示

している。ステップ④では d から a,d,g の情報を受信する。d はステップ②において g から情報を得ているため、この情報を送信している。ステップ⑤では何も受信していない。最終的にこの 5 ステップの情報交換で a,b,d,g の情報を得ることになる。ノード g のステップ①ではノード a とノード g が同時に送信を行うため、正しく受信が行えない。正しく受信できない状況を「\*」で示している。これは次のようなコマンド列によって実現できる。

- G: [a,f,g] 自局センサ情報を送信せよ。
  - G: [b,g] 自局センサ情報および保持情報を送信せよ。 G: 受信情報を保持せよ。
  - G: [c,h] 自局センサ情報および保持情報を送信せよ。 G: 受信情報を保持せよ。
  - G: [d,i] 自局センサ情報および保持情報を送信せよ。 G: 受信情報を保持せよ。
  - G: [e] 自局センサ情報および保持情報を送信せよ。 G: 受信情報を保持せよ。
- ここで以下の定理が成立する。

**定理 1** : 3×3 以上のグリッドネットワークにおいて、送信の衝突なしに全ノードが情報を送信するのに 5 ステップで行える。これが最小である。

**証明** : 端点でない一つのノードに着目する (ノード a)。ノード a の上下左右には 4 つのノード b,e,c,d がある (図 5 参照)。a が送信するとき、b,c,d,e が受信するがその時にはこれらは送信ができない。逆に、b が送信するとき、c,d,e のいずれかが送信すると a は受信できなくなる。c,d,e が送信するときも同様。以上のことから全員が送信するためには少なくとも 5 ステップかかる。逆に図 6 のスケジュールダイアグラムに示すようなスケジュール (一部分あるいは繰り返し) で送信すると、5 ステップですべてのノードが妨害しあうことなく送信が行える。 □



図 7. 2 近傍情報交換のノード設定例

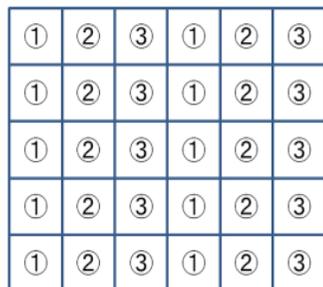


図 8. 送信時受信不能な場合の 2 近傍情報交換のためのスケジューリング

実行コマンドは以下ようになる.

G:[x<sub>5</sub>0y<sub>5</sub>0||x<sub>5</sub>2y<sub>5</sub>1||x<sub>5</sub>4y<sub>5</sub>2||x<sub>5</sub>1y<sub>5</sub>3||x<sub>5</sub>3y<sub>5</sub>4] 自局・近傍情報送信, G: 受信情報保持

G:[x<sub>5</sub>1y<sub>5</sub>0||x<sub>5</sub>3y<sub>5</sub>1||x<sub>5</sub>0y<sub>5</sub>2||x<sub>5</sub>2y<sub>5</sub>3||x<sub>5</sub>4y<sub>5</sub>4] 自局・近傍情報送信, G: 受信情報保持

...

G:[x<sub>5</sub>4y<sub>5</sub>0||x<sub>5</sub>1y<sub>5</sub>1||x<sub>5</sub>3y<sub>5</sub>2||x<sub>5</sub>0y<sub>5</sub>3||x<sub>5</sub>2y<sub>5</sub>4] 自局, 近傍情報送信, G: 受信情報保持

衝突のないような送信を**排他的な送信**, 排他的送信のみからなるスケジューリングを**排他的スケジューリング**と呼ぶ. 実はこれは格子状無線メッシュネットワークの TDMA アルゴリズム<sup>14)15)</sup>と類似している. TDMA のタイムスロットが本稿のステップ割り当てに相当しており, カラリングアルゴリズムはこの定理に相当する.

### 3.2 2 近傍情報交換

全ノードが横方向の両隣のノードと情報を交換する方法を考える. これを 2 近傍情報交換と呼ぶ. 図 5 でいうと a は c,d のセンサ情報を得たいものとする. ただし, 端点のノードに関しては存在する側だけでよいものとする.

**定理 2:** 3×1 以上の送信時受信不能なグリッドネットワークにおいて, 2 近傍情報交換には 3 ステップかかる. これが最小である.

**証明:** 端点ではない一つのノードに着目する (ノード a). 図 7 に示すように, a の両側には 2 つのノード b,c がある. a がデータを送信しているときに b,c が受信するには b,c は送信できない. b が送信しているときに a が受信するには c は送信できないし, c が送信しているときに a が受信するには b は送信できない. ここで, b,c が a からの情報を受信するのに直接受信するのではない, あるいは a が b,c の情報を受信するのに直接受信するのではない場合は少なくとも 3 ステップかかるため除外してよい. この

ことから 2 近傍情報交換には少なくとも 3 ステップかかる. 一方, 図 8 のようなスケジューリング (この繰り返しあるいはその一部) によって 3 ステップで近傍情報が交換できる. □

排他的スケジューリングでは, 全ノードが衝突せずに送信を行うのに 5 ステップかかるが, 全ノードの 2 近傍情報の交換には 3 ステップしかかからない点が重要である. 基本的に上下方向には通信は衝突している (排他的ではない) が, 左右方向には衝突しておらずデータの交換が行えることになる.

### 3.3 4 近傍情報交換

各ノードが上下左右の 4 近傍ノードから情報を得るためのステップを考える. 端点のノードに関しては 2 近傍情報交換の場合と同様存在するノードだけとする.

**系 1:** 3×3 以上の送信時受信不能なグリッドネットワークにおいて, グリッドネットワークにおいて, 4 近傍情報交換は 5 ステップで行える.

**証明:** 3.1 節での排他的スケジューリングで 4 近傍情報が交換できる. □

排他的スケジューリングにおいては, 3.2 節の証明と同様の方法で, 少なくとも 5 ステップ必要であることが簡単に示せるため, これが最小である. 非排他的なケースも含めた場合の最小性については示していない.

### 3.4 8 近傍情報交換

全ノードが上下左右および左上, 右上, 左下, 右下の 8 ノードと情報を交換することを考える. 端点に関しては存在するものだけでよいものとする. 雑音除去や特徴抽出などに用いられるラプラシアンやモルフォロジなど画像処理手法の多くが 8 近傍情報に基づいているため重要である.

**定理 3:** 3×3 以上の送信時受信不能なグリッドネットワークにおいて, グリッドネットワークにおいて, 8 近傍情報交換は 6 ステップで行える.

**証明:** 図 9 のようなスケジューリングによって 6 ステップで行える. □

ここで, 各ステップにおける情報内容は図 10 のとおりである. なお, 6 ステップの最小性は示していない.

**定理 4:** 3×3 以上の送信時受信不能なグリッドネットワークにおいて, 排他的スケジューリングでは 8 近傍情報交換は 7 ステップで行える. これは最小である.

①	②	③
④	④	④
①	②	③
⑤	⑤	⑤
①	②	③
⑥	⑥	⑥

図 9. 送信時受信不能な場合の 8 近傍情報交換のためのスケジューリング例

<b>a</b> ①④ ① (a)* ② ab ③ - ④ (ab)* ⑤ abdegh ⑥ -	<b>b</b> ②④ ① a ② (ab)* ③ abc ④ (abc)* ⑤ abfghkl ⑥ -	<b>c</b> ③④ ① - ② ab ③ (abc)* ④ (abc)* ⑤ abcdef ⑥ -
<b>d</b> ①⑤ ① (d)* ② de ③ - ④ ab ⑤ (abde)* ⑥ abdegh	<b>e</b> ②⑤ ① d ② (de)* ③ def ④ abc ⑤ (abcdef)* ⑥ abcdefghi	<b>f</b> ③⑤ ① - ② de ③ (def)* ④ abc ⑤ (abcdef)* ⑥ abcdefghi
<b>g</b> ①⑥ ① (g)* ② gh ③ - ④ - ⑤ abde ⑥ (abdegh)*	<b>h</b> ②⑥ ① g ② (gh)* ③ ghi ④ - ⑤ abcdef ⑥ (abcdefghi)*	<b>i</b> ③⑥ ① - ② gh ③ (ghi)* ④ - ⑤ abcdef ⑥ (abcdefghi)*

図 10. 図 9 のスケジューリングによる情報交換の内容

**証明** : 図 11 のスケジューリングにより 7 ステップで実行できる. 次に, 最小性を示す. ノードがもれなく一度送信するのに少なくとも 5 ステップかかるが, ステップ⑤で送信された内容は, その時点では斜め 4 方向へは伝達されない. 二つの場合に分けられる.

- ・ステップ⑤で送信するノードの近傍ノードが 4 つある場合: ステップ⑥で斜め 4 方向へ伝達するためにはステップ⑤の送信ノードの近傍 4 ノードのうち少なくとも 2 ノードが転送を行わないといけない. その場合, ステップ⑤の送信ノードで衝突する.
- ・ステップ⑤で送信するノードの近傍ノードが 4 つない場合: そのノードから桂馬と

①	③	⑤	④	②
⑥	⑦	①	③	⑤
④	②	⑥	③	⑤
③	⑤	④	②	①
②	①	③	⑤	④
⑦	⑥	③	⑤	④
⑤	④	②	①	③
		⑦	⑥	

図 11. 送信時受信不能な場合の 8 近傍情報交換のための排他的スケジュールの例

⑤	a	b
c	d	⑤
	e	f

図 12. 定理 4 の証明の説明用コンフィギュレーション

び方向にあるノードの一つが同じくステップ⑤で送信するノードであることを容易に示すことができる. さらにステップ⑤で送信するこの二つのノードが斜め方向に情報伝達するために, 競合なしにいずれの近傍ノードも同時に送信できないことについてもまた, 容易に確認できる. □

図 12 に定理 4 の証明の二つ目のケースについて一例を示す. 図中傍線はネットワーク端辺を表し, ⑤と記す二つのノードはステップ⑤で初めて自局情報を送信する. 左の⑤の情報は a,c に, 右の⑤の情報は b,d,f に到達する. ステップ⑥で完了するには, 左の情報は d に, 右の情報は a,e に到達しないといけないが, c が送信すると b,c が送信しなければならず⑤で衝突する. a が送信すると a は受信できなくなる.

## 4. 送信時受信可能な場合のスケジューリング

### 4.1 基本設定

各ノードがデータ送信時にデータを受信できるというケースを考える. 実は前論

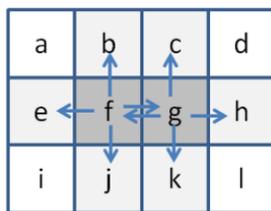


図 13. 送信時受信可能な場合の情報送信の例

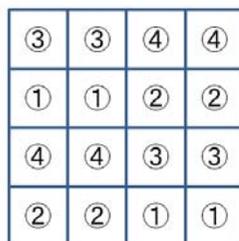


図 14. 送信時受信可能な場合の排他的スケジュールの例

文中で示した筆者らの実装 (図 2) ではこれが可能である. この場合でも衝突は起こりうる点に注意が必要である. すなわち, 図 13 のようなケースで, f,g の同時送信を行った場合, 衝突なしに f の送信情報を b,e,g,j が受信し, g の送信情報を c,g,h,k が受信できる.

これをふまえて以下のような定理が導かれる.

**定理 5 :**  $3 \times 3$  以上の送信時受信可能なグリッドネットワークにおいて, 排他的スケジュールで全ノードが情報を送信するには 4 ステップかかる. これが最小である.

**証明 :** あるノード a の上下左右 4 ノード b,e,c,d を考える (図 5 の状況). a がノード b,e,c,d の送信を受信するためにはこれらは排他的でなければならないため, 少なくとも 4 ステップかかる. 図 14 のスケジュールによって, 4 ステップで全ノードの送信が行える. □

#### 4.2 2 近傍情報交換

2 近傍情報交換に関しては, 一般的には送信時受信不能の場合と同様 3 ステップかかることを示すことができる. ただし, 1 行のみからなるグリッドという特殊な場合には 2 ステップで実行できる.



図 15. 送信時受信可能な場合においても衝突のあるスケジュールの例

**定理 6 :**  $3 \times 2$  以上の送信時受信可能なグリッドネットワークにおいて, 2 近傍情報交換には 3 ステップで実行できる. これが最小である.

**証明 :** 3.2 節と同じスケジュールで行える. 3 つ以上のノードに近傍するあるノードに着目すると, このノードと同時に送信でき, しかもその送信情報をもとのノードが受信できるものは最大 1 つである. 残り 2 ノードはこのノードが情報を受信できる状態でこのノードと同時に送信できない. さらにこの 2 ノードが同時に送信すると元のノードが受信できないので, 同時に送信することはできない. □

図 7 でノード a は b か c どちらかと同じタイミングで送信が行える. だからといって図 15 のようなスケジュールは, 上下の衝突があるためうまく情報交換が行えない.

**系 2.  $M \times 1$  ( $M \geq 3$ ) の送信時受信可能なグリッドネットワークにおいて, 2 近傍情報交換は 2 ステップで実行できる. これが最小である.**

**証明 :** 図 15 の 1 行目のスケジュールで実現できる. □

#### 4.3 4 近傍情報交換

4 近傍情報交換に関しては送信時受信不能の場合と比べると少ないステップで実現できる.

**系 3 :**  $3 \times 3$  以上の送信時受信可能なグリッドネットワークにおいて, 4 近傍情報交換は 4 ステップで行える.

**証明 :** 4.1 節で述べた排他的スケジュールで 4 近傍情報が交換できる. □

ただし, 4 ステップの最小性は示していない.

#### 4.4 8 近傍情報交換

8 近傍情報に関しては以下のような定理を得ている.

**定理 7 :**  $3 \times 3$  以上の送信時受信可能なグリッドネットワークにおいて, 8 近傍情報交換は排他的な方法では 6 ステップで行える.

③	③	④	④
①	①	②	②
⑤	⑤	⑥	⑥
④	④	③	③
②	②	①	①
⑥	⑥	⑤	⑤

図 16. 送信時受信可能な場合の排他的な 8 近傍情報交換のスケジュール例

証明: 図 16 のスケジュールで行える。 □

ここで、ステップ④で出された情報を斜め方向に伝搬するためにステップ⑤、⑥が必要である。もちろん 3.4 節に述べた非排他的な方法でも 6 ステップで行うこともできる。5 ステップで実現できるかどうかは分かっていない。

## 5. おわりに

ローカルトポロジとグローバルトポロジの関係が比較的自明な格子状の有線ネットワーク GlocalGrid において、ローカル通信とグローバル通信を用いたいくつかの並列計算手法について述べた。2 近傍、4 近傍、8 近傍の情報交換のステップ数について論じ、効率のよいアルゴリズムを示したが、いくつかについてその最小性がまだ証明できていない。またシステムソフトウェアの実装はまだ不完全であり、早急に完成させ、実環境での利用を進めていきたい。グローバルに指示を出し、ローカルにいろいろ調べたり、問題を解決したりしていくことはスケーラビリティの観点から非常に有効性が高い。本稿中で何度も述べたようにセンサネットワークに画像処理のアルゴリズムを適用することが GlocalGrid の重要な狙いの一つであり、今後その有効性を検証していきたい。GlocalGrid 上でデータを集約することやもっと複雑な計算を行うこと、個別のノードがブロードキャストを使うことなどに関してはまだまだ考察が不十分であるが、ツリー構造などを使う面白いやり方が考えられるのではないかと考えている。また、本稿で述べた枠組みの一部は、GlocalGrid だけでなく、一般の無線メッシュなどのネットワークに適用可能であるものと考えられる。今後さらに検討を行ってきたい。

**謝辞** 本研究は科研費基盤(A) 23240010 (研究代表 塚本昌彦) ならびに若手(B) 23700083 (研究代表 藤田直生) によるものである。ここに記して謝意を表す。

## 参考文献

- 1) 塚本, "ウェアラブル・ユビキタスコンピューティング - 超小型コンピュータと人, 物, 実世界のシンビオシス," 情報処理学会誌, Vol.47, No.8 (2006)
- 2) K. Whitehouse, C. Sharp, E. Brewer, and D. Culler, "Hood: a Neighborhood Abstraction for Sensor Networks," Proc. of the 2nd Int. Conf. on Mobile Systems, Applications, and Services (MOBISYS) (2004)
- 3) R. Gummadi, O. Gnawali, and R. Govindan, "Macroprogramming Wireless Sensor Networks using Kairos," Proc. of the 1st Int. Conf. on Distributed Computing in Sensor Systems (DCOSS) (2005)
- 4) L. Mottola and G. P. Picco, "Programming Wireless Sensor Networks with Logical Neighborhoods," Proc. of the 1st Int. Conf. on Integrated Internet Ad hoc and Sensor Networks (InterSense) (2006)
- 5) R. Newton and M. Welsh, "Region Streams: Functional Macroprogramming for Sensor Networks," Proc of the 1st Int. Workshop on Data Management for Sensor Networks (DMSN) (2004)
- 6) A. Pathak, L. Mottola, A. Bakshi, G. P. Picco, and V. K. Prasanna, "A Compilation Framework for Macroprogramming Networked Sensors," Proc. of the 3rd Int. Conf. on Distributed Computing on Sensor Systems (DCOSS) (2007)
- 7) 塚本, 藤田, "格子状ネットワークにおけるグローバル通信とローカル通信を組み合わせたユビキタスコンピューティング," 情報処理学会研究報告 2011-MBL-59(20) (Sept. 2011)
- 8) C. Zhang, T. Herman, "Localization in Wireless Sensor Grids," Computers and Their Applications 2006: 388-393 (2006)
- 9) J. M. Gutiérrez López, R. C. Rumín, T. M. Riaz, J. M. Pedersen, O. B. Madsen, "Characterization of Static/Dynamic Topological Routing for Grid Networks," ICN 2009: 367-373 (2009)
- 10) R. R. Brooks, P. Y. Govindaraju, M. Pirretti, N. Vijaykrishnan, M. T. Kandemir, "Clone Detection in Sensor Networks with Ad Hoc and Grid Topologies," IJDSN 5(3): 209-223 (2009)
- 11) J. C. Browne, K. Kane, H. Tian, "An Associative Broadcast Based Coordination Model for Distributed Processes," COORDINATION 2002: 96-110 (2002)
- 12) E. A. Varvarigos, D. P. Bertsekas, "Dynamic Broadcasting in Parallel Computing," IEEE Trans. Parallel Distrib. Syst. 6(2): 120-131 (1995)
- 13) S. Subramanian, S. Shakkottai, A. Arapostathis, "Broadcasting in Sensor Networks: the Role of Local Information," IEEE/ACM Trans. Netw. 16(5): 1133-1146 (2008)
- 14) R. Mangharam, R. Rajkumar, "MAX: A Maximal Transmission Concurrency MAC for Wireless Networks with Regular Structure," BROADNETS 2006 (2006)
- 15) I. Amdouni, P. Minet, C. Adjih, "Node Coloring for Dense Wireless Sensor Networks," CoRR abs/1104.1859: (2011)