

## MANETでのグループモビリティを考慮した 安定なクラスタリング

黒岩潤平<sup>†1</sup> 山内由紀子<sup>†1</sup>  
孫為華<sup>†1</sup> 伊藤実<sup>†1</sup>

クラスタリングはネットワークの階層的な管理を実現する手法であり、大規模なMANET運用の要素技術である。Johnenらは、MANETを頂点重み付きグラフと見なした自己安定クラスタリングアルゴリズムを提案した。このアルゴリズムはトポロジ変化や頂点重みの変化に対する自律適応性を有しているが、クラスタの安定性を考慮していない。本稿では、Johnenらが提案したクラスタリングアルゴリズムにおけるクラスタの安定性を改善するために、ノードの移動特性に基づいたノードの重み割り当て手法を提案する。シミュレーション評価実験により、提案手法はクラスタヘッドの変化回数に関して、従来手法と比較して59%改善することを確認した。

### Stable Clustering for Group Mobility in Mobile Ad-Hoc Networks

JUNPEI KUROIWA,<sup>†1</sup> YUKIKO YAMAUCHI,<sup>†1</sup>  
WEIHUA SUN<sup>†1</sup> and MINORU ITO<sup>†1</sup>

Clustering is a method for hierarchical management of a network. Especially, clustering is often used in MANETs. Johnen et al. proposed a self-stabilizing clustering algorithm which treats a MANET as a weighted graph. The algorithm has autonomous adaptability against topology changes and weight changes, etc., while it does not consider the stability of clusters. In this paper, we present a weight assignment method for Johnen's algorithm that reflects the mobility of each node to its weight so that the stability of clusters should be improved. Simulation results show that the proposed method has improved the number of changes in clusterheads of the former method by 59%.

### 1. はじめに

近年、携帯電話やPDAなどの無線端末が広く普及し、モバイルアドホックネットワーク(MANET)への注目が高まっている。MANETを対象としたシステムは、ネットワークトポロジの変化に自律的に対する自律適応性や、ユーザに安定した機能を提供し続ける安定性を必要とする。この2つの性質を提供し、ネットワークを効率良く管理する手法として、**クラスタリング**がよく用いられる。クラスタリングとは、ネットワークをクラスタと呼ばれる複数のグループに分割する手法である。各クラスタは**クラスタヘッド**と呼ばれる1つのノードと、**通常ノード**と呼ばれるいくつかのノードから成り、クラスタ内のノードの管理、クラスタ間の通信などをクラスタヘッドが一括管理する。クラスタリングはネットワークの階層的な管理を実現し、自律分散的なネットワーク管理の基盤として利用されている。

ネットワークの変化やノードの故障に対する自律適応性を備えた分散アルゴリズムの設計手法として、**自己安定**という概念<sup>5)</sup>がよく知られている。自己安定アルゴリズムは、どのような初期状況からもシステムがやがて目的のふるまいを行うことを保証する。このような性質から、MANETを対象とした自己安定アルゴリズムが近年よく研究されている<sup>8),10)</sup>。

本稿では、MANET環境におけるノードのモビリティを考慮した自己安定クラスタリング手法を提案する。ノードの移動により、クラスタヘッドの交代や、通常ノードが所属するクラスタの変化が起こると、通信のオーバーヘッドが増える可能性がある。そこで、MANETにおけるノードのモビリティの特徴に着目し、長期間安定してクラスタヘッドの役目を担うノードを発見することにより、クラスタの安定性を実現する手法を提案する。

ノードのモビリティに関して、本稿ではイベント会場や駅周辺などをフィールドとして想定する。このような環境では、歩行者はしばしばある程度のグループを形成し、ある一定期間同じような移動を行うため、グループごとにクラスタを構成すれば、安定したクラスタになると考えられる。また、ノードの移動によって異なるグループ同士が一時的に近付き、再び離れるといった、グループが交差するような状況でも、交差の前後でクラスタ構造が維持されることが望ましい。提案手法では、このようなモビリティグループを検出し、モビリティグループの重心、ノード数、移動ベクトルをもとにクラスタヘッドを計算することにより、安定したクラスタリングを実現する。

提案手法の性能を評価するため、理論解析とシミュレーション実験を行った。一般的な理論解析を行うことは非常に困難なため、本稿ではノードが単純な移動を行うような状況を想定し、クラスタの安定性を示すクラスタヘッド変化回数、クラスタの質を示す平均クラスタ数、計算時間を示す収束に必要なラウンド数についてそれぞれ解析を行った。その結果、従来の自己安定クラスタリング手法<sup>11)</sup>と比較して、提案手法はノード数 $N$ 、経過時間 $T$ に対し、平均クラスタ数を $\lfloor N/2 \rfloor$ で維持しつつ、クラスタヘッドの変化回数を $O(NT)$ から $O(T)$ に削減し、収束に必要なラウンド数を $O(NT)$ から $O(T)$ に削減した。シミュレーション実験では、ノードがグループで移動するようなMANET環境を想定し、提案手法によるクラスタの安定性を検証した。従来手法<sup>11)</sup>と比較した結果、平均クラスタ数とメッセー

<sup>†1</sup> 奈良先端科学技術大学院大学  
Nara Institute of Science and Technology

ジ数を保ちながら、クラスタヘッドの変化回数を 59% 削減できていることが確認できた。

## 2. 関連研究

MANET におけるノードのモビリティやバッテリー残量に着目したクラスタリング手法が多く提案されている。Chatterjee らは、ネットワークを頂点重み付きグラフとして扱うことによって MANET に対するクラスタリングアルゴリズムを提案した<sup>3)</sup>。この手法はクラスタリングの安定性を目的とし、全隣接ノードとの距離の総和とバッテリー容量、ノードの移動量からクラスタヘッドを選択する。Bazzal らの手法<sup>7)</sup>は、より隣接ノードの多いノードをクラスタヘッドとすることで、形成されるクラスタの数を抑制する。Chinara らは、クラスタの形成にかかるオーバーヘッドの削減やネットワークのライフタイムの延長を目的とし、バッテリー残量が多いノードをクラスタヘッドとして選ぶクラスタリング手法を提案した<sup>4)</sup>。以上の手法はいずれもノードのモビリティ及びバッテリー残量に着目してクラスタヘッドを決定するが、ノードの移動状況を考慮していない。クラスタヘッドの選択にノードの移動特性を反映することによって、より安定したクラスタ構造を形成できる可能性がある。

Johne らは、MANET を想定した頂点重み付きネットワークに対する自己安定クラスタリングアルゴリズム (RCSA) を提案した<sup>10)</sup>。RCSA はトポロジ変化が起きた場合にも、クラスタに所属しないノードは存在しないという性質を保証するアルゴリズムである。クラスタヘッドはノードの重みに基づいて選択され、重みが大きいほどクラスタヘッドに適したノードとなる。山内らは、重みが大きいノードの移動によって一時的なクラスタヘッドの交代が生じ、RCSA の出力するクラスタ構造が頻繁に変化することを指摘し、クラスタの安定性を改善する重み割り当て手法を提案した<sup>11)</sup>。この手法では、ノード間の相対位置の変化から各ノードが重みを計算し、相対位置の変化が少ないノードほど、大きな重みを持つ。しかし、この手法は現実的なノードの移動状況を想定していない。

MANET におけるノードのモビリティを考慮する際、現実の人間の動きを抽象化したモビリティモデルがよく用いられている<sup>1)</sup>。MANET のモビリティモデルを大別すると、各ノードが独立に移動するモデルと、複数のノードがグループを作り、一群となって移動するモデルが存在する。複数のノードがグループを作って移動するモデルは一般的に**グループモビリティモデル**と呼ばれる。各ノードが独立に移動するモビリティモデルとしてよく用いられているのが、**ランダムウェイポイントモデル (RWP)** である<sup>2)</sup>。グループモビリティモデルとしてよく利用されているのが**リファレンスポイントグループモビリティ (RPGM)** モデルである。RPGM では各グループの中央位置を仮想的に決め、中央位置の移動ベクトルとグループ内の各ノードの移動ベクトルが存在する。グループ全体の動きは中央位置の移動ベクトルによって決定し、各ノードは中央位置から一定距離内でランダムに移動する。本稿のシミュレーションでは、提案手法が有効であると考えられる RPGM と代表的なモビリティモデルである RWP を用いて比較実験を行った。

## 3. 諸定義

分散システムをグラフ  $G = (V, E)$  として表す。  $V = \{p_1, p_2, \dots, p_n\}$  はノードの集合であり、  $E$  はノード間の双方向通信リンクの集合である。MANET においてノードは無線端末を示し、ノード間の通信リンクは無線端末間の通信を示す。無線ネットワークでノードは通信範囲内の全てのノードとの間にリンクを持つ。ノード  $p_i$  と  $p_j$  に対して  $(p_i, p_j) \in E$  のとき、  $p_i$  と  $p_j$  は**隣接する**という。ノード  $p_i$  が隣接するノードの集合を  $N_i$  と表す。ノード  $p_i$  はユニークな識別子  $id_i$  と局所変数を持つ。ノードの**状態**をそのノードのすべての局所変数の値によって定義する。

分散システムの通信モデルとして**局所的な共有メモリモデル**<sup>\*1</sup>を用いる。各ノード  $p_i$  は、ノード  $p_j \in N_i \cup \{p_i\}$  の局所変数を遅延なく直接読むことは可能である。書き換えることができる局所変数は  $p_i$  自身の局所変数のみである。スケジューラとして  **$d$ -デーモン**を想定する。各計算ステップにおいて、 $d$ -デーモンは、ノードの空でない任意の部分集合を選択し、選択されたノードはアルゴリズムを実行する。 $d$ -デーモンによる実行は非同期実行である。

システムの状況を全ノードの状態の組として定義する。システムの実行を状況の無限系列  $e = e_0, e_1, e_2, \dots$  で表記する。ここで、 $e_{i+1}$  は  $e_i$  に 1 つの計算ステップを適用した結果、もしくは終端状況である。

**問題**はすべてのノードの出力変数に対する論理式で定義される。問題の仕様を満たす状況を**正当な状況**と呼ぶ。アルゴリズム  $A$  が収束性と閉包性の 2 つの性質を満たす時、かつその時に限り  $A$  は**自己安定**であるという。**収束性**とは任意の状況から始まる任意の実行に正当な状況が含まれること、**閉包性**とは正当な状況から始まるどのような実行も正当な状況のみから成ることである。自己安定アルゴリズムはシステムのトポロジ変化や**一時故障** (例えばメモリクラッシュ) が発生した後の任意の状況を初期状況と見なし、システムがやがて問題の仕様を満たすことを保証する。自己安定アルゴリズム  $A$  の収束時間はシステムが正当な状況に到達するまでの最大のラウンド数 (最悪ラウンド数) である。

**クラスタリング問題**はノードの集合を**クラスタヘッド**と**通常ノード**から成る**クラスタ**に分割する問題である。MANET では、各通常ノードが所属するクラスタのクラスタヘッドと直接通信できる必要がある。よって、以下の**支配性**をクラスタリング問題の制約とする。

### 条件 3.1 (クラスタリングの支配性)

すべての通常ノードは、所属するクラスタのクラスタヘッドに隣接する。

本稿では、自己安定クラスタリングアルゴリズム RCSA<sup>10)</sup> に対するノードの重み割り当て手法を提案する。RCSA は頂点重み付きグラフに対し、ノードの重みに従ったクラスタ構造を出力するアルゴリズムである。MANET ではノードの移動によりトポロジが時々刻々

\*1 代表的な通信モデルとして他にメッセージの送受信によってノードが通信を行う**メッセージパッシングモデル**がある。局所的な共有メモリモデルに対して設計されたアルゴリズムをより一般の無線通信に近いメッセージパッシングモデル上で動作するよう変換する研究がさまざまところで行われている<sup>6)</sup>。本研究では提案手法のアイデアの有効性に焦点を当てるために、局所的な共有メモリモデルを用いることにする。

と変化する。これはクラスタリング問題への入力の変化と見なせる。Johnenらは入力の変化に対する頑健性を提案した<sup>9)</sup>。RSCA<sup>10)</sup>はトポロジ変化が起こった際にも通常ノードが常にクラスタヘッドに所属するという頑健性を保証する。

ノードへの重み割り当て問題とは、MANET環境を抽象化したグラフ  $G = (V, E)$  を入力とし、各ノード  $p_i \in V$  の重みを出力とする問題である。本稿では特に、各ノードが局所的に自身の重みを計算する重み割り当て関数を持つとする。ノード  $p_i$  は入力変数として自身の  $x$  座標  $x_i$ ,  $y$  座標  $y_i$ , 移動ベクトル  $\vec{v}_i$  を持つ。状況  $c_k$  において、ノード  $p_i$  の重み割り当て関数は、 $p_i$  と各  $p_j \in N_i$  の位置、速度ベクトル、状態を入力とし、ノード  $p_i$  の重み  $w(p_i)$  を出力する。

#### 4. 提案手法

本章では、RSCA に対してノードの移動特性を考慮し、安定したクラスタ構造を実現する重み割り当て手法  $GPGND$  を提案する。

##### 4.1 RSCA

RSCA によって形成されるクラスタの例を図 1 に示す。条件 3.1 より、RSCA ではクラスタヘッドと通常ノードは常に隣接する。提案手法はノードが移動を行うたびに重み変更を行う。RSCA では重みを変更しても通常ノードは常にいずれかのクラスタヘッドに常に所属し続けることが保証される。

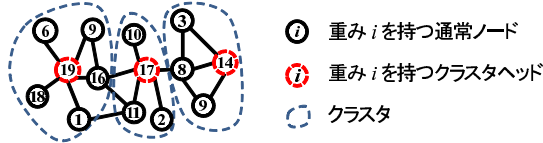


図 1 RSCA によるクラスタ構造の例

##### 4.2 ノードの重み割り当て手法

提案手法  $GPGND$  では、イベント会場や駅周辺などでしばしば見られる、歩行者がある程度のモビリティグループを形成して移動する状況を想定する。このような状況ではクラスタ変更のオーバーヘッドやクラスタの安定性のため、以下の 3 点を考慮する必要がある。

- (i) **モビリティグループの検出**: 複数のノードが 1 つのモビリティグループを形成して移動している間は、モビリティグループに所属するノード同士でクラスタを形成し、維持することが望ましい。
- (ii) **モビリティグループの合流への対処**: 異なるモビリティグループ (クラスタ) が合流する場合、クラスタを合併させる方が良く、サイズの小さいクラスタがサイズの大きいクラスタに加わることで、クラスタ変更にかかるオーバーヘッドを抑制することが望ましい。
- (iii) **モビリティグループの交差への対処**: 異なるモビリティグループが一時的に接近して離れる (交差する) 場合、クラスタを合併させず、モビリティグループごとのクラスタを維持することが望ましい。

以上の 3 点を踏まえ、提案手法では、(i) モビリティグループに所属するノードの検出、(ii) モビリティグループの移動軌跡、(iii) モビリティグループのサイズ、(iv) モビリティグループの移動性について考慮し、ノードの重みを決定する重み割り当て関数を与える。

ノード同士が同一のモビリティグループに所属するか否かの基準として、ノードの位置と移動ベクトルを用いる。ノード  $p_i$  は定数  $\mu$  に対して  $a_1 |x_i - x_j| + a_2 |y_i - y_j| + a_3 |\vec{v}_i - \vec{v}_j| < \mu$  を各隣接ノード  $p_j$  と評価し、この式が成り立つとき、 $p_i$  は  $p_j \in N_i$  が自身と同じモビリティグループに所属すると判定する。上記の式により、各ノードは自身と距離が近く、移動ベクトルが近いノードを同一のモビリティグループとして判定する。 $p_i$  が所属するグループを以降では  $g_i (\subseteq N_i \cup \{p_i\})$  とする。

各ノード  $p_i$  は  $g_i$  を識別した後、 $g_i$  内のノードのモビリティをもとに自身の重みを計算する。重み割り当て関数は  $GPG$ ,  $GPN$ ,  $GPD$  の 3 つの項の和である。 $p_i$  はこれら 3 つの項を  $g_i \subseteq N_i \cup \{p_i\}$  に含まれるノード間でのみ計算する。

$GPG$  はモビリティグループが交差する状況において、クラスタを維持させるための項である。ノード  $p_i$  の重みに、 $p_i$  の移動軌跡と  $g_i$  の重心の移動軌跡の差を反映する。モビリティグループ  $g_i$  の重心座標は、 $(x_{g_i}, y_{g_i}) = \frac{1}{|g_i|} (\sum_{p_j \in g_i} x_j, \sum_{p_j \in g_i} y_j)$  と定義される。過去  $T'$  秒間における重心の軌跡を計算し、 $p_i$  の移動軌跡が  $g_i$  の重心の移動軌跡に近いノードほど重みが大きくなるように重みを割り当てる。

$g_i$  に所属するノードの移動により、 $(x_{g_i}, y_{g_i})$  は頻繁に位置が変化する可能性があり、それによりクラスタヘッドの交代が発生する可能性がある。そこで、クラスタヘッドを維持させるために、閾値  $\varphi$  に対し  $g_i$  の重心の多少の移動は無視する。すなわち、ノード  $p_i$  がクラスタヘッドで  $dist_{g_i}(t) = |x_i^k - x_{g_i}^k| + |y_i^k - y_{g_i}^k|$  が閾値  $\varphi$  以下である場合、 $p_i$  と  $(x_{g_i}, y_{g_i})$  間の距離は 0 であるとみなす。 $x_i^k(y_i^k)$  は時刻  $k$  におけるノード  $p_i$  の  $(x(y))$  座標である。ここで、以下の  $h_{g_i}(t)$  を導入する。式 (1) で  $CH_{p_i}$  は  $p_i$  がクラスタヘッドであるときかつその時のみ、 $true$  となる。

$$h_{g_i}(t) = \begin{cases} dist_{g_i}(t) & (dist_{g_i}(t) > \varphi \vee CH_{p_i} \neq true) \\ 0 & (dist_{g_i}(t) \leq \varphi \wedge CH_{p_i} = true) \end{cases} \quad (1)$$

時刻  $t$  において、ノード  $p_i$  に対する  $GPG_{g_i}(t)$  は  $GPG_{g_i}(t) = \frac{1}{\sum_{k=t-T'}^t (h_{g_i}(k)+1)}$  とする。 $p_i$  の移動軌跡が重心の移動軌跡に近いノードに大きな重みが割り当てられる。

$GPN$  はノード  $p_i$  の重みに、 $g_i$  のサイズを反映させるための項である。 $G$  におけるノード数の上限を  $N$  とすると、 $p_i$  に対する  $GPN_{g_i}(t)$  は  $GPN_{g_i}(t) = \frac{|g_i^t|}{N}$  とする。この式において  $|g_i^t|$  は現時刻  $t$  における、 $g_i$  のノード数である。 $g_i$  のノード数が多いほど  $p_i$  に大きな重みが割り当てられる。

$GPD$  はノード  $p_i$  の重みに、 $g_i$  の移動ベクトルを反映させるための項である。各  $p_j \in g_i$  の移動ベクトルの平均値を求め、ノード  $p_i$  の移動ベクトルとの差に応じた重みを割り当てる。現時刻  $t$  におけるモビリティグループ  $g_i$  の移動ベクトルの平均値  $E_{g_i}(t)$  を

$E_{g_i}(t) = \frac{1}{|g_i|} \sum_{p_j \in g_i} \vec{v}_j$  とする.  $p_i$  の移動ベクトルと  $E_{g_i}(t)$  との差は  $D_{\vec{v}_i}(t) = |E_{g_i}(t) - \vec{v}_i|$  である. これらの式から, ノード  $p_i$  に対して  $GPD_i(t)$  は  $GPD_i(t) = \frac{1}{D_{\vec{v}_i}(t)+1}$  とする.  $p_i$  の移動ベクトルが  $g_i$  の移動ベクトルに近いほど大きな重みが割り当てられる.

ノード  $p_i$  の重みを  $w(p_i, t) = b_1 \cdot GPG_i(t) + b_2 \cdot GPN_i(t) + b_3 \cdot GPD_i(t)$  で定義する.

RSCA は各ノードが互いに異なる重みを持つネットワークに対して設計されたクラスタリングアルゴリズムであるが, 提案した重み割り当て関数  $f(p_i, t)$  では, いくつかのノードが同じ重みを持つ場合がある. その場合, ノードの重みの比較に**クラスタヘッド時間**を用いる.  $p_i$  のクラスタヘッド時間  $t_{ch_i}$  はクラスタヘッド  $p_i$  が過去  $T'$  単位時間のうち, 連続でクラスタヘッドであり続けた時間である (すなわち, 通常ノード  $p_j$  に対して,  $t_{ch_j} = 0$ ). ノード  $p_i, p_j$  に対して  $w(p_i) = w(p_j)$  であり,  $t_{ch_i} < t_{ch_j}$  であるとき, ノード  $p_j$  の重みは  $p_i$  の重みよりも大きいと判断する. さらに  $t_{ch_i} = t_{ch_j}$  の場合はノードの ID によって比較を行い,  $id_i < id_j$  であるとき, ノード  $p_j$  の重みは  $p_i$  の重みよりも大きいと判断する.

### 5. 理論モデルにおける解析

提案手法の有効性を示すために, 従来手法<sup>11)</sup> と提案手法に関して, 単純な移動モデルのもとでクラスタの安定性, クラスタリングの質, 収束時間の理論解析を行った. 本章では, はじめに従来手法の重み割り当て方法について述べ, その後従来手法と提案手法の理論解析結果を示す.

#### 5.1 従来手法<sup>11)</sup>

文献 11) では RSCA に対する 3 つのノード重み割り当て手法  $Nei$ ,  $NumN$ ,  $DifN$  が提案されている. 3 つの重み割り当て手法はノードの相対位置 (隣接ノードの集合) の変化に注目している.  $Nei$  は時刻  $t$  での  $p_i$  の重みとして  $|N_i(t)|$  を割り当てる. ここで,  $N_i(t)$  は時刻  $t$  における  $p_i$  の隣接ノードの集合である.  $NumN$  は時刻  $t$  での  $p_i$  の重みとして  $1/dfn_i(t)$  を割り当てる.  $dfn_i(t)$  は  $0 \leq k \leq T'$  に対して  $|N_i(t-k)| - |N_i(t-k-1)|$  の和であり, 隣接ノード数の変化が小さいノードほど大きな重みが割り当てられる.  $DifN$  は時刻  $t$  での  $p_i$  の重みとして  $1/dfs_i(t)$  を割り当てる.  $dfs_i(t)$  は  $0 \leq k \leq T'$  に対して  $|N_i(t-k) \setminus N_i(t-k+1)|$  の和であり, 隣接ノードの集合差のサイズが小さいノードほど大きな重みが割り当てられる.  $NumN$  と  $DifN$  は相対位置の変化が頻繁に起こるノードの重みを小さくする.  $p_i$  と  $p_j$  の重みが等しい場合は ID で比較を行い,  $id_j < id_i$  ならば  $p_j$  の重みを  $p_i$  よりも大きいとみなす.

#### 5.2 解析に対するシステムモデル

任意のフィールド, 任意の移動状況では各手法の性能を理論的に解析することは困難なので, ノードがライン上を移動する状況での従来手法, 提案手法の性能を比較する.

解析における実行モデルは**同期ラウンドモデル**である. 各ラウンドのはじめで全ノードが隣接ノードの局所変数の値を読み, その後各ノードはアルゴリズムに基づいて局所変数を更新する. 単純化のために, 正当な状況へ収束した後は次にノードの移動が起こるまでラウンドは進まないとする.

ノードの移動モデルとして, 図 2 のようなラインポロジを想定し, ノードの移動方向は一方方向であるとする. 初期配置として各ノードは単位距離ごとに  $p_1, p_2, \dots, p_N$  の  $N$  個が置かれており, 各ノードは隣接ノードとのみ通信が可能である. フィールドの右に存在するノードほど ID が大きく, 左の隣接ノードよりも ID は 1 大きい (したがって, フィールド左端のノードは最も小さい ID を持つ). 1 回の移動で各ノードは単位距離分左へ移動する. フィールド左端のノードが移動を行うとフィールドから離脱し, また移動ごとにフィールド右端に 1 つノードが新たに現れる. 新たに出現するノードはフィールドに存在するノードよりもさらに値が 1 大きい ID を持つ. 図 2 において,  $p_1 < p_2 < \dots < p_N < p_{N+1}$  である. さらに新規出現のノードは局所変数の値が初期化され, 過去の履歴を何も保持していない状態である. ノードの移動は  $t$  ラウンドごとに行われ,  $t$  はクラスタリングの計算の完了に十分な時間であると想定する. クラスタヘッド同士は隣接しないとする. 上記のような規則的なモデルを考慮することで, 提案手法の基本的な性能を解析する. このようなモデルでノード数  $N$ , 経過時間  $Tt$  (初期状態から  $T$  回移動が行われた状態) に対して, 以下の 3 つの指標に関して解析を行う.

**クラスタヘッドの変化回数 ( $c_{ch}$ )** はクラスタの安定性を示す. 各ノードに関して, 状態が変化 (クラスタヘッドから通常ノード, もしくはその逆の変化) した回数を計算する. クラスタヘッドが変化すれば, クラスタの再構築が行われ, システムのオーバーヘッドが高くなる. そのため, クラスタヘッドの変化回数が少ない方が性能が良い. **平均クラスタ数 ( $n_{ch}$ )** はクラスタリングの質を示す. ここで, 平均クラスタ数とは単位時間あたりでのクラスタの数であり, クラスタヘッドの数と等しい. クラスタが極端に多ければクラスタヘッドの変化回数は小さくなる可能性はあるが, 各クラスタのサイズが小さくなり, クラスタリングの効果が無い. そのため, 平均クラスタ数が少ないほど, よいクラスタリングと言える. **収束時間 ( $r_{ch}$ )** は計算時間を示す. 同期実行モデルでは, 計算に対するラウンド数が増加するにつれ, クラスタ形成に必要なメッセージの総量も増加する. そのため, 収束するためのラウンド数が少ないほど性能が良いと言える.

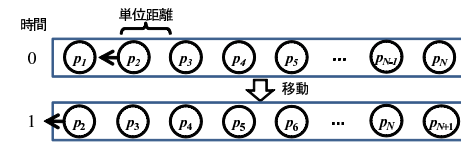


図 2 解析に用いるモビリティモデル

#### 5.3 解析結果

まずノードの重みをそのノードの ID とする ID 基準の手法の結果を述べる. 従来手法<sup>11)</sup> はノードの重みが等しい場合 ID で比較を行うため, ID 基準の手法は最悪時の性能が従来手法と等しくなることを示す. その後提案手法の結果を述べ, 従来手法との優位点を示す.

**ID 基準の重み割り当て:** 最初の状況で, フィールド右端のノードが最も大きい重みを持つ. その後, 右端のノードがクラスタヘッドとなり, その左に位置するノードが通常ノード

となる。以後各ノードは交互にクラスタヘッドと通常ノードになる。よって平均クラスタ数は  $\lceil N/2 \rceil$  となる。その後移動が起こるたびに重み最大のノードが出現し、クラスタヘッドとなる。移動前フィールド右端に位置していたノードは通常ノードとして新規出現ノードに所属し、さらにその左のノードは通常ノードからクラスタヘッドとなる。結果として全ノードが移動のたびに状態を変更する。また各ノードの状態変更は一度に行われるものではなく、(右端の) 重みが大きいノードから低いノードへ (左へ) 順に遷移する形となる。以上のことから、1 回の移動に対する収束時間は  $O(N)$  ラウンド、 $Tt$  ( $T$  回移動後) 期間では  $O(NT)$  ラウンドとなる。さらに移動 1 回におけるクラスタヘッド変化回数は  $O(N)$  となり、 $Tt$  期間では  $O(NT)$  となる。

**従来手法<sup>11)</sup>**: 従来手法はいずれの手法も隣接ノードによって重みを決定する。しかし図 2 のモデルではノードの移動による隣接関係の変化が小さい。よって各ノードの重みの差が小さく、例えば  $Nei$  ではフィールド両端の 2 つのノードの重みが 1、それ以外のノードの重みが 2 と等しくなる。従来手法では重みが等しい場合 ID で比較を行うため、結果的に ID 基準のものと同様の結果となる (表 1)。

**提案手法**: 提案手法で考慮する 3 つの項のうち、 $GPN$  は  $Nei$ 、 $GPG$  は  $NumN$  と同じ重み分布となる。また  $GPD$  は全ノードの重みが等しくなる。しかし重みが等しい場合提案手法ではクラスタヘッド時間で比較する。したがって、一度クラスタヘッドとなったノードは通常ノードとなることはなく、ノードの状態変化は新規参加のノードのみである。よって、1 回の移動に対するクラスタヘッドの変化回数は  $O(1)$ 、 $Tt$  期間では  $O(T)$  となる。さらに移動 1 回における収束時間は  $O(1)$  ラウンド、 $Tt$  期間では  $O(T)$  ラウンドとなる。

各手法における解析結果を表 1 に示す。提案手法は ID 基準、従来手法と比較して平均クラスタ数を維持しつつ、収束時間、クラスタヘッド変化回数を削減できている。本移動モデルのような単純な環境では、提案手法は非常に有効的な手法であると言える。

## 6. シミュレーション実験

提案した重み割り当て手法の有効性を確認するために、Java でシミュレータを作成し、従来手法<sup>11)</sup> との比較実験を行った。

### 6.1 シミュレーション環境

評価実験では、RPGM と RWP の 2 つのモビリティモデルを用いて、従来手法と比較実験を行った。シミュレーション領域は  $200[m] \times 200[m]$  の正方形領域と  $20[m] \times 200[m]$  の長方形領域の 2 種類とした (詳細な設定は表 2 に示す)。各ノードはこの領域内をモビリティモデルに従い同期して移動する。移動のたびに各ノードは重み計算と RSCA を実行する。人が徒歩で移動する道路領域を想定した長方形領域ではノードの離脱・参加を想定し、長方形の上下端に近づいたノードはフィールドから離脱すると同時に長方形領域の逆側に新しいノード (履歴を持たない状態) として現れる。RPGM に関して、モビリティグループ内の最大のノード数を 10 としている。

ノードの移動に関するパラメータを表 2 に示す。同じモビリティグループに所属するノ

表 1 解析結果のまとめ

手法	$c_{ch}$	$n_{ch}$	$r_{ch}$
ID	$O(NT)$	$\lceil N/2 \rceil$	$O(NT)$
$Nei$	$O(NT)$	$\lceil N/2 \rceil$	$O(NT)$
$NumN$	$O(NT)$	$\lceil N/2 \rceil$	$O(NT)$
$DifN$	$O(NT)$	$\lceil N/2 \rceil$	$O(NT)$
$GPG$	$O(T)$	$\lceil N/2 \rceil$	$O(T)$
$GPN$	$O(T)$	$\lceil N/2 \rceil$	$O(T)$
$GPD$	$O(T)$	$\lceil N/2 \rceil$	$O(T)$

表 2 モビリティパラメータ表 3 重み割り当てパラメータ

ノード数 (正方形領域)	1000	提案手法	$a_1, a_2, a_3$	1	
ノード数 (長方形領域)	400		$\mu$	10	
シミュレーション時間 [s]	1800		$T$	15	
最大速度 [m/s]	1		$N$	1000	
最大停止時間 [s]	60		$b_1$	25	
通信範囲 [m]	10		$b_2$	10	
			$b_3$	1	
			$\varphi$	2 [m]	
			従来手法	$M$	30
				$h(x)$	1

ド同士が通信できるようモビリティグループの最大の直径を 10 [m] とした。シミュレーション結果は 100 回試行を行った平均値である。比較手法として、毎秒各ノードにランダムに重みを割り当てる手法と、文献 11) で提案されている重み割り当て手法 ( $Nei$ ,  $NumN$ ,  $DifN$ ) を用いる。提案手法に関して、3 つの項  $GPG$ ,  $GPN$ ,  $GPD$  がどのようにクラスタリング結果に影響するか確認するために、それぞれの項を単独で重みに割り当てた場合の結果も示す ( $b_1 = 1, b_2 = 10, b_3 = 0$  等)。重み割り当てに関するパラメータを表 3 に示す。表 3 で  $b_1 = 25, b_2 = 10, b_3 = 1$  は、クラスタヘッド変化回数及び平均クラスタ数の値が小さくなるように調整した値である。この場合、ノードの重みに占める  $GPG$ ,  $GPN$ ,  $GPD$  の割合はおおよそ 6 : 1 : 3 となる。

本実験では理論解析で用いたクラスタヘッドの変化回数、平均クラスタ数に加えてメッセージ送信回数を評価指標とする。1 秒間あたりの各指標の平均値を結果として示す。本実験では MANET を想定し、メッセージパッシングモデルでの同期実行モデルを採用した。

### 6.2 実験結果

図 3 に RPGM モデルと RWP モデルにおけるクラスタヘッドの変化回数、図 4 に平均クラスタ数を示す。また、図 5~8 に各領域、移動モデルにおけるクラスタ数の時間遷移を示す。正方形領域では RPGM で平均隣接ノード数が 15、RWP で 11 となった。長方形領域では RPGM で平均隣接ノード数が 38、RWP で 34 となった。

図 3 で、移動モデルや領域の種類に関わらず、提案手法  $GPGND$  はいずれの手法と比較してもクラスタヘッド変化回数が小さく、最も安定性が高い。削減率が最も小さい長方形領域での RWP でも  $GPGND$  は従来手法よりもクラスタヘッド変化回数は 59% 値が小さくなっている。長方形領域で、 $GPGND$  は平均クラスタ数が最も小さい。正方形領域での RPGM モデルでも  $GPGND$  の平均クラスタ数は従来手法よりも小さい。しかし提案手法が対象としていない RWP モデルでは  $GPGND$  は平均クラスタ数が従来手法よりも増加する。提案手法は重みに周囲のノードの情報を用いるために、隣接ノード数が少ない正方形領域では効果が十分に発揮できなかったと考えられる。以上より、提案手法は RWP、特に正方形領域であまり効果的ではないが、RPGM ではクラスタヘッド変化回数と平均クラスタ数いずれも削減できている。図 5~8 を見ると、クラスタ数が時間遷移によって大きく変

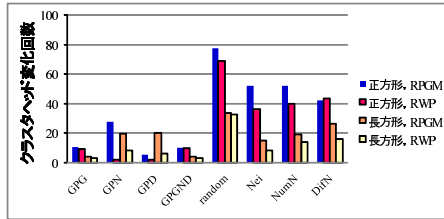


図 3 クラスタヘッド変化回数の平均値

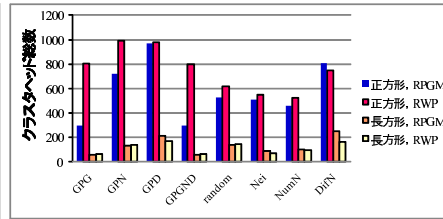


図 4 平均クラスタ数

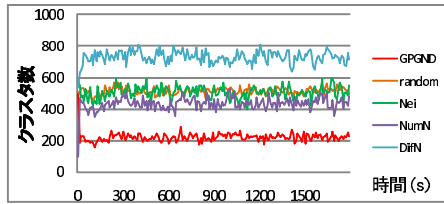


図 5 クラスタ数の時間遷移 (正方形, RPGM)

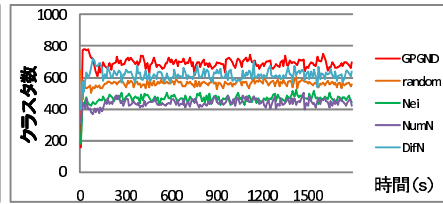


図 6 クラスタ数の時間遷移 (正方形, RWP)

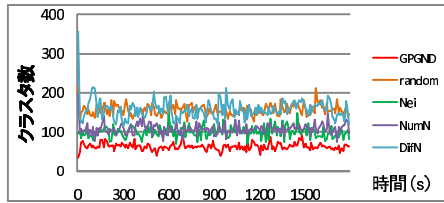


図 7 クラスタ数の時間遷移 (長方形, RPGM)

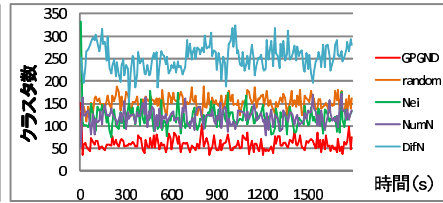


図 8 クラスタ数の時間遷移 (長方形, RWP)

化することは無く、やはり RPGM では平均クラスタ数の削減ができてることが分かる。メッセージ送信回数に関しては大きな差は見られず、GPGND と従来手法のメッセージ複雑度はシミュレーションではほぼ同じ結果となった。そのため、図の掲載は割愛する。さらに、重み割り当て関数の項である GPG, GPN, GPD それぞれを単独で重み割り当て関数とした結果を比較する。領域、移動モデルによらず平均クラスタ数は GPG の値が最も小さい。正方形領域では、RPGM では GPD, RWP では GPN がクラスタヘッド変化回数の値が最も小さくなった。しかし、GPN と GPD は平均クラスタ数の値が大きい。GPD と GPN は多くのノードがクラスタヘッドとなることを許すことによってクラスタヘッド変化回数を削減している。このことから、RPGM と RWP いずれのモデルにおいて

も、GPG がクラスタの安定性を改良するために最も重要な要因であると考えられる。

## 7. おわりに

本稿では RSCA<sup>10)</sup> によって得られるクラスタ構造の安定性を改良する重み割り当て手法を提案した。単純なモデルにおける理論解析より、提案手法は平均クラスタ数を維持しつつ、クラスタヘッド変化回数及び収束時間を改善した。シミュレーション実験により、提案手法はクラスタヘッドの変化回数に関して、従来手法に比べて 59% 改善したことを確認した。今後の課題としては、さまざまなシチュエーションを包括的に扱える重み割り当て手法の考案や、MANET での特定のアプリケーションの要求への対応が考えられる。

## 参考文献

- 1) Bai, F. and Helmy, A.: A SURVEY OF MOBILITY MODELS in Wireless AdHoc Networks, *Wireless Ad Hoc and Sensor Networks*, Kluwer Academic Publishers, chapter1 (2004).
- 2) Camp, T., Boleng, J. and Davies, V.: A Survey of Mobility Models for Ad Hoc Network Research, *WCMC: Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, Vol.2, pp.483-502 (2002).
- 3) Chatterjee, M., Das, S.K. and Turgut, D.: WCA: A Weighted Clustering Algorithm for Mobile Ad Hoc Networks, *J. of Cluster Computing, Special Issue on Mobile Ad hoc Networking*, Vol.5, No.2, pp.193-204 (2002).
- 4) Chinara, S. and Rath, S.K.: TACA: A Topology Adaptive Clustering Algorithm For Mobile Ad Hoc Network, *Proc. of WORLDCOMP'09* (2009).
- 5) Dijkstra, E. W.: Self-stabilizing system in spite of distributed control, *Comm. ACM*, Vol.17, No.11, pp.643-644 (1974).
- 6) Dolev, S., Israeli, A. and Moran, S.: Resource bounds for self stabilizing message driven protocols, *Proc. of PODC '91*, ACM, pp.281-293 (1991).
- 7) El-Bazzal, Z., Kadoch, M., Agba, B.L., Gagnon, F. and Bannani, M.: An efficient management algorithm for clustering in mobile ad hoc network, *Proc. of PM2HW2N '06*, ACM, pp.25-31 (2006).
- 8) Flauzac, O., Haggar, B.S. and Nolot, F.: Self-Stabilizing Clustering Algorithm for Ad Hoc Networks, *Proc of ICWMC'09*, IEEE Computer Society, pp.24-29 (2009).
- 9) Johnen, C. and Nguyen, L.H.: Route preserving stabilization, *Proc of SSS'03*, Springer, pp.184-198 (2003).
- 10) Johnen, C. and Nguyen, L.H.: Robust Self-Stabilizing Clustering Algorithm, *Proc of OPODIS'06*, Springer, pp.410-424 (2006).
- 11) Yamauchi, Y., Itou, T., Nishikawa, G., et al.: Clustering algorithms for mobile ad-hoc networks to improve the stability of clusters, *Proc. of IASTED'2008*, pp. 9-15 (2008).