

アドホックネットワークにおける適応的クラスタリング手法

大田 知行[†] 井上 伸二[†] 角田 良明[†]

† 広島市立大学 情報科学部 情報工学科
〒 731-3194 広島市安佐南区大塚東 3 丁目 4 番 1 号
E-mail: †{ohta,inoue,kakuda}@ce.hiroshima-cu.ac.jp

あらまし アドホックネットワークにおいて、クラスタリングは多数の端末を階層的に効率良く管理、また、周波数等の資源を効率良く割り当てるために用いられる。アドホックネットワークでは、ノードの移動により、トポロジ変化が頻繁に起こる。その結果、クラスタの再構成が頻繁に起こってしまう。ノードの移動速度が速くなるにつれ、クラスタの再構成はさらに頻繁に起こる可能性がある。この頻繁に起こるトポロジ変化に対して、ネットワーク全体ではなく、各ノードが自律的、かつ、局所的に適応してクラスタリングを行なう必要がある。また、ネットワーク中の各クラスタのサイズをできるだけ均一にすることにより、ノードの密度に依存しないクラスタリングを行なうことができる。このことにより、クラスタヘッドの負荷分散もまた実現可能となる。本論文では、以前我々が提案したクラスタリングスキームがノード密度やノードの移動速度に関係なく、均一なクラスタ構成を行なうことができることや、クラスタの再構成を比較的短い時間で行なうことができることをシミュレーション実験により示している。

キーワード アドホックネットワーク、クラスタリング

An Adaptive Clustering Scheme
for Highly Mobile Ad Hoc NetworksTomoyuki OHTA[†], Shinji INOUE[†], and Yoshiaki KAKUDA[†]

† Dept. of Computer Engineering, Faculty of Information Sciences,
Hiroshima City University

Ozuka-Higashi 3-4-1, Asa-Minami-ku, Hiroshima, 731-3194 Japan

E-mail: †{ohta,inoue,kakuda}@ce.hiroshima-cu.ac.jp

Abstract A clustering scheme for ad hoc networks is aimed at managing a number of mobile devices by utilizing hierarchical structure of the networks. In order to construct and maintain an effective hierarchical structure in ad hoc networks where mobile devices may move at high mobility, the following requirements must be satisfied. (1) The role of each mobile device for the hierarchical structure is adaptive to dynamic change of the topology of the ad hoc networks. The role of each mobile device should thus change autonomously based on the local information. (2) The overhead for management of the hierarchical structure is small. The number of mobile devices in each cluster should thus be almost equivalent. This paper proposes an adaptive multihop clustering scheme for highly mobile ad hoc networks. The results obtained by extensive simulation experiments show that the proposed scheme does not depend on mobility and node degree of mobile devices in the network, which satisfy the above requirements.

Key words Ad Hoc Networks, Clustering

1. ま え が き

アドホックネットワークは、ルータの機能を備えているモバイル端末の集合によって構成される。各端末が他の端末と通信するとき、通信範囲内に他の端末を中継することにより、目的の端末へデータパケットを送ることができる。つまり、各端末

が前もって設置された基地局に依存することなく目的のノードと通信が可能となる。ネットワーク中のすべての端末が自由に動き回っても、その集合から離れない限り端末間での通信が可能である。本論文では、移動端末からなる分散システムであるアドホックネットワークに関して、ネットワーク上の端末をクラスタに分割するクラスタリングについて考える。クラスタリ

ングは、多数の端末を階層的に効率良く管理、また、周波数等の資源を効率良く割り当てるために用いられる。現在、クラスタリングに関して多くの研究がなされている [4]~[6]。

また、クラスタリングを利用したルーティングアルゴリズムも数多く提案されている [1]~[3], [7]。現在提案されている代表的なルーティングアルゴリズムは、flooding に依存している。flooding により、端末の移動速度に関わらず接続性を維持することが可能である。しかし、接続性を維持するための制御パケットの量がネットワーク規模の増大にともない、指数関数的に増加する可能性がある。そこで、クラスタリングを利用することにより、できるだけ少ない制御パケット数でルーティングを行なうことが可能となる。

現在提案されているクラスタリング手法では、各クラスタは、クラスタを管理する端末 (以下、クラスタヘッド (clusterhead)) とその隣接端末 (つまり、クラスタヘッドから 1 ホップしか離れていない端末) によって構成される。以下、クラスタに所属するノードのことをクラスタメンバと呼ぶ。また、二つ以上のクラスタヘッドと隣接する端末は、所属する二つのクラスタ間ゲートウェイ (gateway) となる。以下、ゲートウェイと呼ぶ。

このクラスタ構成には以下の問題点があると考えている。第一に、クラスタヘッドや各端末の所属するクラスタが頻繁に変わってしまう可能性がある。これは、クラスタヘッドの通信範囲から外れた端末は、その端末自身がクラスタヘッドとなるか、新しく通信範囲に入ったクラスタヘッドのクラスタメンバになるからである。つまり、クラスタヘッドと少しでも異なる方向に移動しただけで別なクラスタの端末となってしまう可能性がある。また、端末の移動速度が速い場合においても各端末の所属するクラスタが頻繁に変わってしまう可能性がある。第二に、ネットワーク中の端末の密度にクラスタヘッドの負荷が依存してしまうことである。これは、端末の密度が高い場合にはクラスタメンバの管理等でクラスタヘッドに非常に大きな負荷がかかってしまう。また、端末の密度が小さい場合には各クラスタが小さくなってしまい、クラスタリングの利点が損なわれる可能性がある。

そこで、クラスタヘッドからクラスタメンバのホップ数を 1 ホップ以上のホップ数によって構成するマルチホップなクラスタを提案した。ここで、クラスタヘッドからクラスタメンバまでのホップ数が 1 ホップ以上のホップ数で構成されるクラスタをマルチホップクラスタと呼ぶ。マルチホップクラスタでのクラスタの条件は以下の通りである。

(1) クラスタヘッドから 2 ホップ以上離れたクラスタメンバは、同じクラスタヘッドに所属するクラスタメンバによって連結されてなければならない。

(2) クラスタヘッドが管理するクラスタメンバの数を指定した範囲内とする。

これらの条件により、クラスタの地理的な大きさが大きくなることにより、端末移動速度が高い場合においても効率の良いクラスタリングができると考えている。また、クラスタをクラスタメンバ数によって定義しているため、端末の密度に依存しないクラスタリングを行なうことができる。また、すべてのクラ

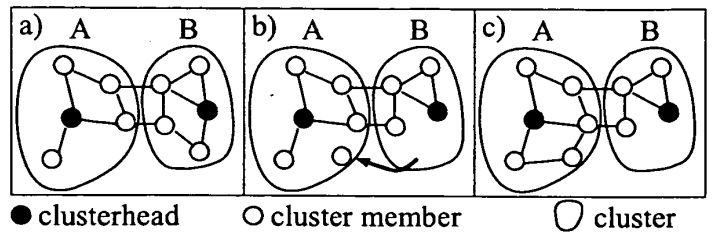


図 1 自律ノードの例

スタサイズを均等にすることによりクラスタヘッドの負荷を均一にすることが可能となると考えられる。

本論文では、文献 [8] で提案したマルチホップクラスタリングについて実験的評価に基づいて考察を行なっている。以下、端末はノードと呼ぶ。

本稿の構成は、以下の通りである。2. 節で、本手法の特徴について述べる。3. 節で、マルチホップクラスタリングスキームについて述べる。4. 節で、本手法のシミュレーション実験による評価を述べる。最後に、5. 節で、まとめとともに今後の課題を述べる。

2. マルチホップクラスタリングスキームの特徴

最初に、本手法の特徴を述べる。

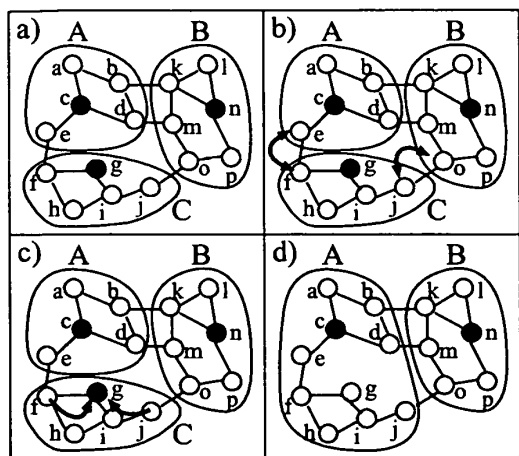
- 各ノードは自律的にクラスタを構成
- 各クラスタは、自律的にクラスタを維持

アドホックネットワークでは、移動ノードによってネットワークが構成される。あるノードから別なノードへパケットを送るために経路を生成しても、次の瞬間にはノードの移動によりその生成経路が使用不可能な状態になっていることが頻繁におこる。

クラスタリングにおいても同様なことが頻繁に起こり得る。ある生成されたクラスタが次の瞬間には、クラスタ内のノードの移動によりそのクラスタがクラスタとして機能していないような状態が頻繁に起こり得る。しかしながら、クラスタリングを行なうためには、どのようにノードが移動したとしても、ネットワーク中のクラスタは維持されなければならない。

一つ目の特徴について、図 1 で例を用いて説明する。図 1a) で示す二つのクラスタから成るネットワークを想定する。図の左側のクラスタをクラスタ A、右側のクラスタをクラスタ B とする。今、図 1b) で示すように、クラスタ B のクラスタメンバが移動によって、クラスタ A へ移動した場合を考える。その結果として、図 1c) のような状態となる。このとき、そのクラスタメンバがクラスタ B から出ていったにも関わらず、クラスタ B には何の影響を与えることなく、今まで通りにクラスタ B はクラスタとしての機能を果たす。また、クラスタ A に移動したクラスタメンバは、新しくクラスタ A のクラスタメンバとして機能を果たす。このとき、このクラスタメンバ加入によりクラスタ A に対して何らかの影響を与えることはない。

次に、二つ目の特徴について説明する。本手法では、クラスタヘッドが管理するクラスタメンバの数を指定した範囲内となるようにしている。もし、クラスタメンバの数が範囲より小さ



● clusterhead ○ cluster member □ cluster

図2 自律クラスタの例

い場合、このクラスタを隣接する他のクラスタと合体させる。クラスタの合体を行なうためには、クラスタヘッドは隣接クラスタのクラスタメンバ数を知る必要がある。これは、クラスタが合体した結果、クラスタメンバ数が指定された範囲を越えてしまうことを防ぐためである。また、クラスタメンバの数が範囲より大きい場合は、このクラスタを二分割する。次に、クラスタメンバの動作と持つ情報を説明する。

動作1: クラスタメンバは、定期的に隣接ノードと情報交換を行なう。

動作2: クラスタメンバは、定期的に自分の持つ情報をクラスタ内にブロードキャストする

ここで、動作1により、クラスタメンバは隣接ノードの情報を持つことができる。また、動作2により、クラスタメンバは同じクラスタに所属するすべてのクラスタメンバを知ることができる。

クラスタ同士の合体について、図2で例を用いて説明する。図2a)で示すようなクラスタA, B, Cの三つのクラスタから成るネットワークを想定する。各クラスタのゲートウェイは、クラスタAでは、b, d, e, クラスタBでは、k, m, o, クラスタCでは、f, jとなっている。今、クラスタCのクラスタメンバ数が指定された範囲より小さいと仮定する。このとき、クラスタCは隣接クラスタAかBのどちらかと合体を行なう。クラスタCのクラスタヘッドgが隣接クラスタA, Bのクラスタメンバ数を知る方法について説明する。

すべてのクラスタメンバは、動作2より同じクラスタに所属するクラスタメンバを知っている。その結果、ゲートウェイeはクラスタAのクラスタメンバ数を持っている。ここで、2b)に示すように、ゲートウェイfとクラスタAのゲートウェイeの間で情報交換を行なうことにより、ゲートウェイfは、クラスタAのクラスタメンバ数を得ることができる。そして、2c)に示すように、fがfの持つ情報をクラスタ内にブロードキャストすることにより、ゲートウェイfの持つクラスタAのクラスタメンバ数がクラスタヘッドgに届けられる。

同様の動作により、クラスタヘッドgは、ゲートウェイjか

らクラスタBのクラスタメンバ数を得ることができる。このクラスタAとBのクラスタメンバ数を元に、クラスタCは自律的に合体相手を選択し、合体を行なう。結果として、図2d)で示すようにクラスタ構成となる。この図では、クラスタAと合体を行ない、クラスタAのクラスタメンバ数が増え、クラスタCは消滅している。

3. マルチホップクラスタリング

3.1 アドホックネットワーク

アドホックネットワーク(以下、ネットワーク)は、有線リンクやそれを経由する基地局を持たず、移動ノードだけで構成される。各ノードは無線が有効な通信範囲内のノードとだけ通信可能である。ネットワークを無向グラフ $G = (V, E)$ で表す。各接点 $v_i \in V$ はノードID i のノード(以下、ノード i と呼ぶ)に、各無向枝 $(v_i, v_j) \in E$ はノード i, j 間の無線リンクに対応する。このときノード v_i と v_j は隣接しているという。

3.2 クラスタの構成

クラスタは、ノードの部分集合で、一つのクラスタヘッドと複数のノードからなる。クラスタヘッドとなるノードの識別子がこのクラスタのクラスタIDとなる。このとき、クラスタヘッドとクラスタヘッド以外のノードは、同じクラスタIDを持つノードで連結していなければならない。また、クラスタの大きさを制限するために、あらかじめ最小構成数と最大構成数を決める。

[定義1] クラスタヘッドが v_i であるクラスタ C をクラスタ C_i と呼ぶ。

[定義2] クラスタ C_i のクラスタメンバとなるノードの個数 $|C_i|$ は、 $L \leq |C_i| \leq U$ でなければならない。

3.3 ノード

ノードは、以下の二つの基本的な動作を行なう。

- (1) 隣接ノードとの情報交換による情報収集。
- (2) クラスタ内ブロードキャストによるメンバ情報の収集。ノードの持つ情報。ノードID, クラスタID, 状態, クラスタメンバリスト。ブロードキャストによりクラスタメンバリストが更新される。

3.4 クラスタヘッド

クラスタを管理するノードをクラスタヘッドと呼ぶ。クラスタメンバリストにより、クラスタの大きさを管理する。大きさが L よりも少ない場合、隣接クラスタのいずれかと合体を行なう。隣接クラスタの大きさについては、ゲートウェイから情報を収集する。合体は、他のクラスタとの関係が必要となるため、合体するためにはネットワークに多少の影響を与える。

また、大きさが U よりも大きい場合、クラスタを二分割する。しかしながら、分割はそのクラスタ単独で行なうため分割中に他のクラスタに影響を与えることはない。どちらの場合においても、合体の場合はクラスタの消滅、分割の場合は新しいクラスタの生成により、その周辺のクラスタでは、隣接クラスタ情報の書き換えが必要となるためネットワークに多少の影響を与える。しかしながら、どちらの場合においても、隣接クラスタのみにしか影響を与えないため局所的にクラスタ構成の更

新を行なうことができる。分割、結合がクラスタ構成の更新に与える影響については、文献[9]にて考察を行なっている。

3.5 ゲートウェイ

ノード v_i の隣接ノードの中に、 v_i と異なるクラスタ ID を持つノード $v_j (j \neq i)$ が存在する時、ノード v_i をゲートウェイと呼ぶ。ゲートウェイ v_i は、 v_j とノード動作 1 により、 v_j の持つクラスタメンバリストにより、隣接クラスタのサイズを知ることができる。その情報をノード動作 2 により、 v_i のクラスタヘッドに伝える。

3.6 状態とノードの動作

この階層構造構成法では五つの状態を定義し、その状態により、果たす役割を変化させる。各状態にあるノードの役割について説明する。

Normal State Node(NSN):ノードはクラスタヘッドや、ゲートウェイの機能を持たないノード。

Control Node(CN):ノードはクラスタヘッドの役割を果たす。

Border Node(BN):ノードはゲートウェイとなる。

Border Control Node(BCN):ノードはクラスタヘッドとゲートウェイの役割を担うことになる。

Orphan Node(ON):ノードの状態が ON のとき、ノードはクラスタ ID を持たない。新しくネットワークに加えられたノードは、最初、この状態である。

アドホックネットワークでは、常にノードが移動するので、ノードの移動に対して適応力があることは重要な要素の一つである。本階層構造構成法は、ノードの移動に対して高い適応性を示している [9]。

3.7 ノードの状態遷移

アドホックネットワークでは、クラスタを構成するための中心的存在となるクラスタヘッドや、クラスタとクラスタを結ぶゲートウェイの移動に伴い、別のノードがそのクラスタでの新しいクラスタヘッドやゲートウェイとして機能する必要がある。このような理由からノードは自律的に状態遷移をすることにより、クラスタの維持を行う必要がある。各ノードの状態の変化は、それらのノードの隣接ノードの状態によって決まる。図 3 にその状態遷移を示す。A から E までの 5 つの状態遷移条件を元に、クラスタ維持を各ノードの状態遷移により行う。

A: ノードは現在の役割に、クラスタヘッドの役割を追加する。
B: ノードの状態は NSN に遷移し、通常のノードになる。
C: ノードは現在の役割に、ゲートウェイの役割を追加する。
D: ノードは現在の役割から、ゲートウェイの役割を削除する。
E: ノードが保持していたクラスタ ID を削除し、ノードは状態 ON に遷移し孤立したノードになる。

4. シミュレーション実験

4.1 実験環境

今回のシミュレーション実験での実験環境について説明する。ネットワークの大きさを $750m \times 750m$ とし、その中に存在するノードを 200 ノードとする。ノードの初期位置はランダムに配置する。ノードの移動方法は、randomway point model を採用している。この方法は、目的座標まで与えられた平均速度で

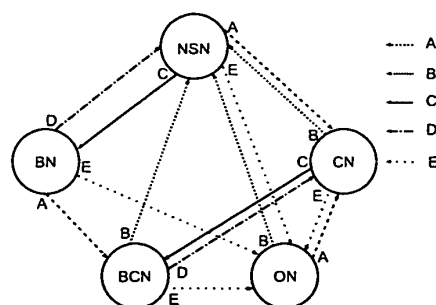


図 3 ノード v_i の状態遷移図

移動し、目的地に到着すると、ある一定時間その目的地で停止する。これを停止時間と呼ぶ。今回の実験では、この停止時間 (pause time) を 0 とした。そして、新しい目的座標を決め、再び新しい目的座標まで移動を開始する。以上の動作を繰り返す。

また、各ノードの通信可能範囲は半径 $100m$ とし、任意のノード間の遅延時間は、パケットを送受信の処理時間も含めて $1msec$ とする。

次に、各ノードが隣接ノードの状態やクラスタ ID を定期的にチェックするための Hello の間隔を $200ms$ とし、すべてのノードでの定期通知の間隔もそれぞれ $200ms$ とする。シミュレーション全体の時間を $80sec$ に設定し、最初の $5sec$ のデータはすべて破棄し、 $5sec$ 以降から、 $60sec$ 間ノードを動かす。

最後に、ノードの平均移動速度を $5m/s$, $10m/s$, $20m/s$ とする。また、各クラスタのクラスタメンバ数の範囲 (L, U) を、それぞれ、 $(10, 25)$, $(20, 50)$, $(30, 75)$ とした。

4.2 クラスタに関する実験

ある一つのクラスタヘッドに多くのクラスタメンバが集まることなく、すべてのクラスタメンバがクラスタヘッドに対して均一であることは、クラスタヘッドの負荷分散にとって非常に望ましいことである。そこで、本実験では、どのようにノードの移動速度を変化させたとしても、ネットワーク中のすべてのクラスタのサイズが均一となっていることをシミュレーション実験により示した。

表 1 は、それぞれノードの移動速度、定義したクラスタサイズにおけるネットワーク中に存在したクラスタ数を示している。括弧の中は、各クラスタの平均クラスタメンバ数を示している。各実験は、5 秒から 60 秒までの間、5 秒毎のクラスタ数を計測し、その平均値を計算している。クラスタメンバ数についても同様である。

表 1 より、各クラスタサイズの範囲の定義において、ノードの移動速度に関わらずネットワーク中のクラスタ数がほぼ同じである。また、クラスタメンバ数も同様に、ノードの移動速度に関係なく同じ値となっている。つまり、本手法でのクラスタリングはノードの移動速度に全く影響されないと言えることができる。

表 2 は、それぞれノードの移動速度、定義したクラスタサイズにおける各クラスタにおけるクラスタヘッドからクラスタメンバまでの平均ホップ数を示している。括弧の中は、各クラスタで最もホップ数の多かったノードのホップ数を示している。各

表 1 クラスタ数

Mobility (m/s)	Cluster Range		
	10-25	20-50	30-75
5	10.41 (18.31)	5.35 (36.74)	3.89 (51.26)
10	10.09 (17.81)	5.55 (34.65)	3.72 (54.12)
20	10.50 (18.38)	5.68 (35.13)	3.75 (53.60)

表 2 クラスタメンバの平均ホップ数と最大ホップ数

Mobility (m/s)	Cluster Range		
	10-25	20-50	30-75
5	1.97 (5.63)	2.74 (7.52)	3.12 (7.92)
10	2.00 (5.85)	2.76 (7.46)	3.29 (8.16)
20	1.95 (5.72)	2.62 (6.94)	3.13 (8.25)

表 3 クラスタ数とクラスタメンバ数

Mobility (m/s)	Transmission Range (m)		
	100	200	300
5	5.35 (36.74)	5.09 (38.86)	4.37 (42.14)
10	5.55 (34.65)	4.95 (40.13)	4.12 (41.41)
20	5.68 (35.13)	4.89 (39.02)	4.57 (41.17)

表 4 クラスタメンバの平均ホップ数

Mobility (m/s)	Transmission Range (m)		
	100	200	300
5	2.74	1.64	1.25
10	2.76	1.90	1.35
20	2.62	1.97	1.54

表 5 クラスタリングの適応性評価

Mobility (m/s)	Cluster Range		
	10-25	20-50	30-75
5	101.00	80.40	0.40
10	360.70	301.50	100.70
20	583.30	481.00	161.00

unit: ms

実験は、5秒から60秒までの間、5秒毎のホップ数を計測し、その平均値を計算している。最もホップ数が多かったノードのホップ数についても同様である。

表2は、クラスタメンバの平均ホップ数がノードの移動速度を高くしてもほぼ同一のホップ数となっている。クラスタメンバの平均ホップ数と平均最大ホップ数の差が極端に大きくないため、各クラスタは、クラスタヘッドを中心に比較的均一に分布していると考えられる。もし、クラスタが極端に横や縦のサイズが大きい楕円形になっていれば、平均最大ホップ数は大きくなるだろう。

次に、ノードの通信可能範囲を、100m, 200m, 300mと変化させることにより各ノードの隣接ノード密度を高めることにより、本手法がネットワーク中のノード密度に依存するかどうかを調べた。本実験では、クラスタサイズの範囲20-50についてのみ実験を行なった。

表3は、それぞれノードの移動速度、ノードの通信可能範囲におけるネットワーク中に存在したクラスタ数を示している。括弧の中は、各クラスタの平均クラスタメンバ数を示している。各実験は、5秒から60秒までの間、5秒毎のクラスタ数を計測し、その平均値を計算している。クラスタメンバ数についても同様である。

表3により、各クラスタはノードの密度がどうであれ、ノードの移動速度を高くしてもネットワーク中のクラスタ数や各クラスタのクラスタメンバ数は影響を受けず同一な大きさのクラスタとなっていることを示すことができた。このときの各クラスタでのクラスタヘッドからクラスタメンバまでの平均ホップ数を表4に示す。予想される通り密度が高くなったことにより、クラスタヘッドからのホップ数は短くなっている。

以上の実験により、ノードの密度、ノードの移動速度に関係なくネットワーク中に同一なサイズのクラスタの構成を行なうことができることを示した。

4.3 収束時間に関する実験

収束時間は、シミュレーション時間65secでノードの動きを止めて、そこから安定状態になったときの時間を計測する。ここで、安定状態のために必要な時間を5secとする。つまり、5sec続いて状態遷移やクラスタIDの変更が行われなければ、その状態は安定しているとみなし、安定状態となった時間から5sec差し引いた時間を安定状態になるまでにかかった時間とする。

この5secは、予備実験により経験的に得た値である。予備実験として、200ノードをランダムに配置して、ノードは動かない状態で、本手法によるクラスタ構成を行い、クラスタが構成されるまでの時間を計測する。まず、30sec間の実験を行う。30sec後の状態を安定状態として、いつ、その最後の30secの状態になったのか計測する。その結果として、最悪でも2sec程度だったので、今回の実験において、多く見積もって5secとした。

表5には、それぞれノードの移動速度、定義したクラスタサイズの範囲における収束時間を示している。

収束時間は、クラスタヘッドの変更数(表6)とクラスタの合体分割数(表7)は非常に関連している。ネットワーク中のクラスタ構成が変わってしまうために、クラスタヘッドの変更が起こる。クラスタヘッドが変更してしまう主な原因は、クラスタの合体と分割によるものである。合体は、隣接する二つのクラスタで合体を行なうため、どちらか一方のクラスタは消滅してしまう。また、クラスタの分割においても、一つの大きなクラスタを分割することにより新しいクラスタが生成される。クラスタの消滅とクラスタの生成により、ネットワーク中のクラスタ構成が変更される。収束時間を計測するためにノードの動きを止めて、そこからクラスタリングを行なう。ノードを動かしていた60secの間、クラスタ構成がノードの動きに対して適応していなければ、ノードの動きを止めたときに、クラスタの構成が非常に壊れ状態となっている可能性がある。ここで壊れた状態とは、クラスタのサイズが定義したクラスタ範囲の条件を満たしていない場合や、あるクラスタの中に別なクラスタのクラスタメンバが存在した場合のことを言う。そして、その壊れた状態から正常な状態に戻すために、各クラスタ、つまりクラスタヘッドが自律的にクラスタの合体と分割を行なう。このクラスタの合体、あるいは、分割のために行なうクラスタリン

表 6 クラスタヘッドの変更数

Mobility (m/s)	Cluster Range		
	10-25	20-50	30-75
5	47.80	12.80	7.90
10	84.00	36.40	22.60
20	169.40	62.70	41.70

表 7 合体分割数

Mobility (m/s)	Cluster Range		
	10-25	20-50	30-75
5	43.70 (39.40)	11.60 (6.60)	7.40 (3.40)
10	73.50 (70.60)	33.40 (21.90)	20.80 (8.70)
20	151.40 (137.40)	58.50 (34.50)	38.50 (14.60)

表 8 ノードのクラスタ ID 変化数と状態遷移数

Mobility (m/s)	Cluster Range		
	10-25	20-50	30-75
5	1132.00 (861.80)	658.10 (814.40)	572.60 (752.10)
10	2323.30 (1458.70)	1489.90 (1620.80)	1161.70 (1355.10)
20	4705.50 (3849.20)	2854.90 (3151.60)	2076.00 (2621.60)

グの手続きもまた収束時間に含まれる。もし、クラスタ構成がノードの動きに対して適応していれば、この収束時間はほぼ 0 に近い値となる。

ノードの移動速度を高くするとクラスタ構成の壊れた状態が起こりやすくなるため、結果として収束時間が長くなっている。しかしながら、最も長い場合でさえ、たかだか隣接ノードをチェックする間隔に換算して 3 回分 (600ms) である。クラスタサイズの範囲を多くすると、クラスタ構成のための条件が緩むことになるため、結果としてより短い収束時間となっている。以上により、本手法がノードの移動に対して適応的にクラスタリングを行なうことができていると言える。

表 6 は、ノードを動かしている間に起こったクラスタヘッドの変更数を示す。クラスタサイズの範囲を大きくするほど、ネットワーク中のクラスタ数が少なくなるため (表 1 参照)、クラスタヘッドの変更数が少なくなる。クラスタサイズの範囲が 10-25 の場合、ネットワーク中に存在するクラスタ数は 10 程度である (表 1 参照)。最もクラスタヘッドの変更数が多い場合においても、一クラスタ当たりのクラスタヘッド変更数は、16 程度である。ここで、ノードの移動速度の増加に対して、クラスタヘッドの変更数もまた同じ程度の割合で増加していることがわかる。つまり、ノードの移動速度に対して、クラスタヘッドの変更数が高々比例関係程度で収まっている。ノードの移動速度に対して、クラスタヘッドの変更数が爆発的に増大していない。さらにノードの移動速度を高くしたとしてもクラスタヘッドの数は爆発的に増大することはないと推測できる。各ノードのクラスタ ID 変化数と状態遷移数 (表 8 参照) に関しても同じことが言える。以上により、本手法ではノードの移動速度が高くなったとしてもネットワーク全体に爆発的な負荷を与えることなく十分に対応できると言える。

5. ま と め

本論文では、マルチホップクラスタリングの有効性について述べ、以前提案したマルチホップクラスタリングについて実験的評価を行ない考察を行なった。

今後の課題は、アドホックネットワークにおけるクラスタリングアルゴリズムに対して、ノードの移動に対してそのクラスタリングアルゴリズムの適応性が評価できるような関数を検討していく予定である。

謝 辞

本研究の一部は、広島市立大学特定研究費 (一般研究費) のもとに実施したものである。ここに記して謝意を表す。

文 献

- [1] S.Basagni, "Distributed Clustering for Ad Hoc Networks," Proc.99' Int'l Symp. on Parallel Architectures, Algorithms, and Networks (I-SPAN'99), 1999.
- [2] C.Chiang, H.Wu, W.Liu and M.Gerla, "Routing in Clustered Multihop, Mobile Wireless Networks with Fading Channel," Proc. of the IEEE Singapore Int'l Conf. on Networks (SICON), 1997.
- [3] C.R.Lin and M.Gerla, "Distributed Clustering for Ad Hoc Networks," IEEE Journal on Selected Areas in Communications, 1997.
- [4] S.Banerjee and S.Khuller, "A Clustering Scheme for Hierarchical Control in Multi-hop Wireless Networks," Proc. IEEE INFOCOM'2001, 2001.
- [5] A.D.Amis, R.Prakash, T.H.P.Vuong and D.T.Huynh, "Max-Min D-Cluster Formation in Wireless Ad Hoc Networks," Proc. IEEE INFOCOM'2000, 2000.
- [6] 谷口, 井上, 増澤, 藤原, "アドホックネットワークにおけるクラスタ構成法," 電子情報通信学会論文誌 D-I Vol.J84-D-I No.2, 2001.
- [7] C.E.Perkins, "AD HOC NETWORKING," Addison-Wesley, 2001.
- [8] T.Ohta, K.Ishida, Y.Kakuda, Shinji Inoue and Kaori Macda, "Maintenance Algorithm for Hierarchical Structure in Large Ad Hoc Networks," Proc. of Int'l Conf. on Fundamentals of Electronics, Communications and Computer Sciences (ICFCS2002), 2002.
- [9] T.Ohta, S.Inoue, Y.Kakuda, K.Ishida and K.Macda, "An Adaptive Maintenance of Hierarchical Structure in Ad Hoc Networks and Its Evaluation," Proc. of 22th IEEE Int'l Conf. on Distributed Computing Systems Workshops, 2002.