

# 移動端末のためのトポロジ変化に強いクラスタリング手法 A Robust Clustering Technique for Moving Objects Changing Their Topology

羅 勇<sup>†</sup>    天笠 俊之<sup>†</sup>    波多野 賢治<sup>†</sup>    宮崎 純<sup>†</sup>    植村 俊亮<sup>†</sup>  
Yong Luo    Toshiyuki Amagasa    Kenji Hatano    Jun Miyazaki    Shunsuke Uemura

## 1. まえがき

IT技術の著しい発展に伴い、ノートパソコン、PDAや携帯電話などに代表される移動端末が爆発的な普及を見せている。これに伴い、移動端末によるアドホックネットワークに関する研究も盛んになっている。アドホックネットワークは複数の端末により一時的に構成されるネットワークであり、軍事、災害救助、センサネットワークなどにおいて幅広い応用が期待される。その一方では現状において帯域幅、端末の性能制約及びルーティング法などといった検討すべき所が多い。

アドホックネットワークにおける以上の問題点を効率良く対処できる解決手法の一つにクラスタリングがあげられる。例えばトポロジの変化があるネットワークから、ある基準に基づきクラスタ代表 (clusterhead) を選び、そしてクラスタ代表を中心にネットワークを複数のクラスタに分けるという考え方である。しかし、既存の手法 [1, 2, 3] では、端末の離脱などによるトポロジ変化のネットワークに有効であるが、端末の移動によるトポロジ変化のネットワークを考える場合、クラスタ代表の選択、および端末があるクラスタを離脱し、他の加入先のクラスタを選択する際において、各端末の移動を考慮しないので、得られるクラスタが短い期間で崩れる可能性が高い。つまりクラスタが移動によるトポロジ変化には安定性に欠けている。

そうした既存手法の問題点を対処するため、本稿では、移動端末のトポロジ変化が激しいアドホックネットワークを想定して、トポロジ変化に強いクラスタリング手法を提案する。すなわち端末の移動を意識して、端末間の通信リンクが長く維持できるようなクラスタリング手法を提案する。提案手法は、端末の移動ベクトルを元に端末間の通信可能な時間を概算し、長時間通信可能な移動端末同士をクラスタ化する。この手法により高い安定性を持つクラスタの形成が可能となる。

## 2. 端末間の通信可能時間のモデル化

各ノードが位置情報  $p = (x, y)$ 、移動ベクトル  $\vec{s} = (s_x, s_y)$  を自身で知ることができるものとする。二つのノード  $u, v$  が互いに通信範囲内にあれば、リンクがあるとす。リンクが切れるまでに維持できる概算時間を  $t_{(u,v)}$  (以降単にリンク維持時間と呼ぶ) で表す。時刻  $t_0$  の時、二つのノードが位置などの情報交換により、自身および相手の位置変化を時間  $\Delta t$  に関する線形関数

$$f(\Delta t) = \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x_{init} & s_x \\ y_{init} & s_y \end{pmatrix} \begin{pmatrix} 1 \\ \Delta t \end{pmatrix}$$

で表すことができる。

よって二つのノード間の距離も時間に関する関数となる。

$$D_{(u,v)}(\Delta t) = \sqrt{(x_u - x_v)^2 + (y_u - y_v)^2}$$

$$D_{(u,v)}(\Delta t) \geq p \times \min\{TRange_u, TRange_v\} \quad (1)$$

式(1)となる時、互いの通信リンクが切れることになる。 $0 \leq p \leq 1$  は、建物などの影響を考慮したパラメータである。よって、 $t_0$  から、リンクが切れるまでの時間  $\Delta t$  は、式(1)を解くことで得られる。すなわち、

$$t_{(u,v)} = \Delta t$$

一般的にノードの移動は道路などの制約で一定のスピードである方向に向かって一定の距離を進むことの繰り返しであることが多いと期待される。更新されるまでリンク維持時間は一定の期間内に有効と考えられる。よって、二つのノード間のリンク維持時間を時間で評価することが可能である。

## 3. 提案手法

本稿では、リンク維持時間を考慮したクラスタリング手法 CLT (Clustering based on Link Time) を提案する。図1で提案手法の考え方を説明する。

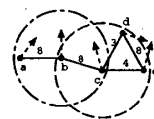


図1: リンク維持時間を考慮したクラスタリング

図1では、数字は計算されたリンク維持時間を表す。もしノード  $b$  をクラスタ代表としてクラスタ  $C(b) = \{a, b, c\}$  を形成すれば、 $C(b)$  は、少なくとも8単位時間で維持できると分かる。一方  $c$  をクラスタ代表とする  $C(c) = \{b, c, d, e\}$  を形成するとすると、4単位時間が経つと、ノード  $d, e$  が離脱し、他のクラスタに加入するための再計算をしなければならない。明らかにクラスタ  $C(b)$  のほうのクラスタ内のトポロジ関係が長く保ち、安定度が高い。よって図1を  $C(b) = \{a, b, c\}$ 、 $C(c) = \{c, d, e\}$  の二つのクラスタに分けたほうがいい。

以上のことから、安定度の高いクラスタを得るためには、できるだけリンク維持時間の長いノードをクラスタに分けることが重要である。

### 3.1 クラスタ代表選択

安定の高いクラスタを得るためには、二つの側面が考えられる。(1) 適当なクラスタ代表の選択。クラスタ代表が変われば、そのクラスタを一旦解散して、クラスタ代表を再選し、クラスタを再構成しなければならない。よってクラスタ代表を長く務めるノードを選ぶことによりクラスタの再構成を低減することでクラスタを更新するコストを抑えることができる。(2) メンバノードの離

<sup>†</sup>奈良先端科学技術大学院大学 情報科学研究科, Graduate School of Information Science, Nara Institute of Science and Technology

脱が少ない。クラスタから離脱するメンバノードの個数が少なければ、再加入するクラスタを選択するためのコストを省くことができる。

結論として、クラスタの安定を考慮して、適当なクラスタ代表を選ぶことはクラスタの品質を大きく改善することが期待できる。

ここで二つのクラスタ代表選択法を提案する。

[方法1] ノード  $v$  のそれぞれの近接ノード  $u \in N(v)$  が持つリンク維持時間の中において、 $t_{(v,u)}$  が高い順位にあれば、ノード  $v$  は、長いリンク維持時間の近接ノードを多く持つと判断できる。よって  $v$  の近接ノードからの評価により、 $v$  に重みをつけることによって、 $v$  がクラスタ代表に適合できるかどうかを判断できる。

$$w_{(v,u)} = \sin\left(\frac{t_{(v,u)}}{2\sum T(u)}\pi\right) \quad (2)$$

$T(u) = \sum_{z \in N(u)} t_{(u,z)}$ . 式(2)を使うことで、 $t_{(v,u)}$  がそれぞれの近接ノードにおける重みが  $(0, 1]$  に分布する。次に

$$w(v) = \sum w_{(v,u)}$$

を求め、ノード  $v$  の重みとする。もし  $v$  の重みが近接ノードのどれよりも大きければ、 $v$  をクラスタ代表と選択する。

[方法2] ノード  $v$  の最大リンク維持時間を意識して、もし他の近接ノードとのリンク維持時間が最大リンク維持時間との差が小さければ、その  $v$  は、多くの長いリンク維持時間の近接ノードを持つと分かる。 $v$  のそれぞれのリンク維持時間と最大リンク維持時間との差の平均  $avg(v)$  を求め、そして最大リンク維持時間から  $avg(v)$  を引いて、得た結果を重みとする。つまり

$$w(v) = \text{Max}(t_{(v,u)}) - avg(v)$$

$avg(v) = \frac{1}{n} \sum_{n=|N(v)|} (\text{Max}(t_{(v,u)}) - t_{(v,u)})$  である。

### 3.2 初期クラスタ生成アルゴリズム

アルゴリズムは以下の通りである。すべてのノードが未定状態からスタートする。

(1) 各ノード  $v$  が近接ノード  $N(v)$  を求める。互いに位置や移動ベクトルなどの情報を交換することにより、リンク維持時間を計算する。

(2) 式(3.1)あるいは(3.1)を用いて、 $v$  が自分の重み  $w(v)$  を求める。

(3) 各ノードが計算した重み  $w(v)$  を含んだメッセージ CHC(ClusterHead Candidate) をブロードキャストする。

(4) 有限時間内に  $v$  が自分の  $w(v)$  より大きな重みの CHC を受信しなければ、自身をクラスタ代表として宣言し、さらに近接ノードにその宣言を通知する。一方近接ノードがその宣言を知ると、自身をクラスタメンバの状態にセットする。結果として、 $v$  をクラスタ代表として、その近接ノード  $N(v)$  をメンバとするクラスタが得られる。

(5) すべてのノードがクラスタ代表か、メンバノードかのどちらに属する状態に達した時点で、クラスタの生成が完了する。

生成されたクラスタには二つの性質がある。(1) 任意の二つのクラスタ代表は近接しない。(2) クラスタ間の

オーバーラップがあり、オーバーラップゾーン内にあるノードは、ゲートウェイノードとして振舞う。

### 3.3 クラスタの維持

クラスタの生成が完了するとクラスタの維持過程に入る。各ノードの位置及び移動ベクトルが随時変化する中で新しいノードが領域内に現れる、リンク切れにより消えるなど、全体としてのネットワーク構成は常に変化している。よってクラスタを効率よく維持することが重要である。また移動ベクトルの更新により、リンク維持時間を再計算する場合があり、その再計算によりクラスタを再構成する必要もある。一般的にクラスタの維持は大きく新しいノードが現れた時の処理と既存クラスタの維持に分けることができる。

新しいノードの加入:

- (1) 新しいノードが自分の近接ノードを求める。
- (2) 近接ノード中にクラスタ代表があるかどうかを確認する。あれば、そのクラスタ代表のメンバとなる。クラスタ代表が複数の場合、最もリンク維持時間の長いほうのクラスタ代表を選んで、そのクラスタに加入する。
- (3) 近接ノード中にクラスタ代表がなければ自身でクラスタ代表を宣言し、その近接ノードをメンバとする新しいクラスタを作る。

既存ノードの離脱:

- (1) クラスタ代表及びメンバは互いの状態を定期的に確認する。あるメンバがクラスタを離脱した場合、別の入るべきクラスタを新しいノードの加入と同じ手順で探す。
- (2) 最大リンク維持時間を持つノードが離脱した場合、或いはクラスタ代表同士が近接になった場合、クラスタを解散する。解散されたクラスタのノードを一旦未定状態に戻し、属すべきクラスタを探索するか、自分たちでもう一度クラスタ代表を選び、クラスタを形成する。

## 4. まとめ

本稿では、移動端末の移動ベクトルを考慮して、互いに通信範囲内にある移動端末間の通信可能な時間をリンク維持時間としてモデル化した。そしてリンク維持時間を元に、できるだけ長いリンク維持時間を多く持つクラスタ代表を選び、クラスタ生成アルゴリズムを提案した。

今後の課題としては、移動の方向とリンク維持時間を総合的に考慮するクラスタリング手法を考える。

## 謝辞

本研究の一部は、科学技術振興機構 (JST) の戦略的基礎研究推進事業 (CREST) プログラムの支援によるものである。ここに記して謝意を表す。

## 参考文献

- [1] Parekh, A.: Selecting Routers in Ad-Hoc Wireless Networks, In *Proceedings of the SBT/IEEE International Telecommunications Symposium* (1994).
- [2] Ephremides, A., Wieselthier, J. and Baker, D.: A design concept for reliable mobile radio networks with frequency hopping signaling, *Proceedings of the IEEE*, Vol. 75, No. 1, pp. 56 - 73 (1987).
- [3] Basagni, S.: Distributed Clustering for Ad Hoc Networks, *Proceeding of International Symposium on Parallel Architectures, Algorithms and Networks*, pp. 310 - 315 (1999).