

アドホックネットワークにおける隣接端末発見精度向上方式の提案

杉浦 健太郎[†] 岩田 彰[†] 若山 公威^{††}

無線端末が自律的にネットワークを構築するアドホックネットワークではネットワーク維持のためにメッセージ交換を行う必要があり、端末数が増すにつれてこのメッセージが通信帯域を圧迫するという問題がある。そこで我々はアドホックネットワークにおいて動的なルーティングパラメータの変更による通信メッセージ削減方式を提案し、従来方式と同程度の端末発見精度を保ちつつメッセージ数を削減できることを確認した。しかし、この方式には静的なトポロジーに属する端末が他の静的なトポロジーに移動する場合、隣接発見精度が低下する課題がある。今回さらに移動する端末の速度ベクトル情報を用いることで課題を解決する隣接端末発見精度向上方式を提案し、メッセージ数と隣接端末発見遅延時間について評価を行う。

Improving Discovering Neighbor Node in Ad-Hoc Networks

KENTARO SUGIURA[†], AKIRA IWATA[†]
and KIMITAKE WAKAYAMA^{††}

Each wireless node has to send messages to create and maintain an Ad-Hoc networks. However when the number of nodes participating in the network increase, the number of packets flowing in the system increases. This in return will cause packet collisions. Ito et al suggested a method which can reduce the number of packets in the system by changing the routing parameters dynamically. However if the node moves from one static topology group of nodes in the network to another static topology group, this system will require a longer time for that node to be realized by its new neighbors. In this paper we propose a method which reduces the delay time by using velocity vector of the nodes. Also we evaluate the proposed method by the no of messages flowing in the network and the node addition time.

1. はじめに

近年、無線 LAN の利用の拡大に伴いアドホックネットワークへの注目が高まっている。

アドホックネットワークは無線 LAN によって各端末が自律的にネットワークを構成し、通信を行うために基地局などのアクセスポイントを必要としないネットワークのことである。このネットワークを利用する場面としては、災害時などのインフラストラクチャの利用ができない状況下での通信や、道路上での車車間通信、またセンサネットワーク等の様々な場面での利用が期待され、それぞれの分野での応用として研究が進められている。

このアドホックネットワークの問題点の一つとしてネットワーク中の端末数の増加に伴い、ネットワークを維持するためのメッセージ数が増加することが上げ

られる。特に全てのノードが移動している場合、常にトポロジーが動的に変化しているので、ネットワーク維持のために各端末が頻繁にメッセージ交換をしなければならずメッセージ数も多大である。

伊藤らはこの問題を改善する方法として、各端末が隣接端末数の状況にあわせて動的にルーティングパラメータを変更することにより、ネットワーク維持のための通信メッセージ数を削減する手法を提案した¹⁾。通常アドホックネットワークのルーティングプロトコルでは、ネットワーク維持のためのルーティングパラメータは静的である。伊藤らの方式ではこのルーティングパラメータを各端末が動的に変化させることにより、従来の方式と同程度の遅延時間を保ちつつメッセージ数を削減することができた。

しかし、伊藤らの方式では静的なトポロジーに属するノードが他の静的なトポロジーに移動する場合、発見精度が低下する課題がある。

本研究ではこの課題を解決するために、移動するノードの速度ベクトル情報と位置情報を用いた隣接端末発見精度向上方式の提案を行う。

[†] 名古屋工業大学

Nagoya Institute of Technology

^{††} 名古屋外国語大学

Nagoya University of Foreign Studies

2. 伊藤らの提案方式

一般的なルーティングプロトコルでは、ネットワーク上の全ての端末が分散協調的に動作する必要性から、原則として全ての端末で同じパラメータを設定する必要がある。また、一度設定したパラメータは、ネットワーク全体をリセットするまで変更されることはない。

従来の Proactive 型ルーティング方式における `advertise_rate` と `neighbor_deletion` はネットワーク全体で常に同じ値である必要があったパラメータであるが、伊藤らの提案方式ではこれらをネットワークの状況に応じて動的に変化させてメッセージ数の削減を行う。

ここで、`advertise_rate` や `neighbor_deletion` のようなパラメータに関する説明をする。これらのパラメータはアドホックルーティングミドルウェアである株式会社スカイリー・ネットワークス社の製品、DECENTRA²⁾ における名前である。

DECENTRA は Hybrid 型ルーティングプロトコルに分類され、スコープという概念を元に、スコープ内は Proactive 型ルーティング、スコープ外は Reactive 型ルーティングによってルーティングを行う。

`advertise_rate` は、周囲に自らの存在を広告するための hello パケットを送信する時間間隔を設定するものである。`neighbor_deletion` は、隣接端末から一定時間 `advertise` を受け取らない場合に、その端末を隣接でないとして判断するまでの時間間隔である。従って、`neighbor_deletion` は、必ず `advertise_rate` より大きな値を設定する必要がある、`neighbor_deletion` を小さくするためには、必然的に `advertise_rate` も小さくする必要があるといったように、これらのパラメータ間には密接な関係がある。

`advertise_rate` と `neighbor_deletion` の値は、ネットワークでどの程度の発見、および消滅遅延時間を許容するかを考慮して決定する。発見遅延時間とは、2つのノード同士が互いの電波到達範囲に入ったとき、それをノードが実際に検知するまでの時間差である。消滅遅延時間とは、あるノードがあるノードの電波到達範囲から離脱したことをノードが実際に検知するまでの時間差である。

発見遅延時間を小さくするためには、`advertise_rate` を小さくする必要がある。消滅遅延時間を小さくするためには、`neighbor_deletion` を小さくする必要がある、同時に `advertise_rate` も小さくする必要がある。一方、`advertise_rate` を小さくすると、メッセージ送信間隔が小さくなるために、送信されるメッセージ数が大きくなる。従って、遅延時間とメッセージ数はト

レードオフの関係にある。

動きの激しい、より動的なネットワークにおいては、ノードの発見、消滅がより頻繁に起こるため、発見、消滅遅延時間を小さくするために `advertise` メッセージ数を大きくする必要があるが、動きの少ない、相対的に静的なネットワークにおいては、相対的に動的であるネットワークと同等の遅延時間とするために動的なネットワークと同等の `advertise` メッセージ数は必要とされないと思われる。伊藤らの方式は、ネットワークで相対的に静的な部分においてメッセージ数を削減することで、ネットワーク全体のメッセージ数を削減しつつ、ネットワーク全体として、発見、消滅遅延時間が大きく増加しないことが期待できる。

2.1 伊藤らの方式におけるパラメータの決定方法

伊藤らの方式においてルーティングパラメータを動的に決定する方法を説明する。以下で新たに導入される用語、パラメータについて説明する。

`node_increase_rate(1/s)`

状況に応じて決定される、単位時間あたりの新規ノード増加率。

`advertise_rate(s)`

`node_increase_rate` にしたがって動的に変化する hello パケット送信間隔。

`neighbor_deletion(s)`

ある端末から見て、他の特定の端末をルーティングテーブルから削除するまでの時間。他の端末の `advertise_rate` にしたがって変化。

`max_advertise_rate(s)`

最大 `advertise_rate` を決定するパラメータ。

`min_advertise_rate(s)`

最小 `advertise_rate` を決定するパラメータ。

`max_advertise_rate_increase(s)`

単位時間あたりに `advertise_rate` が増加する上限の保証値を決定するパラメータ。

2.2 `advertise_rate` の決定

`advertise_rate` は、単位時間あたりの隣接端末増加率 `node_increase_rate` を引数とする次の式 (1) によって決定される。

$$\begin{aligned} x &= \text{node_increase_rate} \\ \min &= \text{min_advertise_rate} \\ \max &= \text{max_advertise_rate} \\ \text{advertise_rate}(x) &= \frac{\min * \max}{x/\alpha * (\max - \min) + \min} \end{aligned} \quad (1)$$

ただし、 α は $\text{advertise_rate}(\alpha) = \min$ となる値である。

式 (1) は、`node.increase_rate` の逆数を、`advertise_rate(0) = max, advertise_rate(α) = min` となるように正規化したものである。

隣接端末の増加率が大きいということは、その端末の周囲のネットワークの変化が大きく、相対的に動的であることを意味する。反対に、隣接端末の増加率が小さい場合には、その端末の周囲のネットワークの変化は小さく、相対的に静的であることを意味する。式 (1) は、隣接端末の増加率、`node.increase_rate` が最大のときに `advertise_rate` が最小となり、`node.increase_rate` が最小のとき、`advertise_rate` を最大とすることを意味する。つまり、より動的なネットワークでは、頻繁な `advertise` を行い、より静的なネットワークでは、`advertise` 間隔が大きくなり、送信メッセージ数を減少させる。式 (1) 内の α の値は、`advertise_rate` を最小とするときの、`node.increase_rate` の最大値を表す。

`advertise_rate` は、式 (1) で決定されるが、前回の `advertise_rate` から、今回の `advertise_rate` までに増加する最大値 `max_advertise_rate_increase` を越えないように調整する。

2.3 neighbor_deletion の決定

`neighbor_deletion` は、自分から見た、他のノードそれぞれに対し、それぞれの `advertise_rate` から決定される値である。

A における、B の最後の `advertise_rate` を `last_advertise_rate(A,B)` と定義する。

`last_advertise_rate` により、A における B に対する `neighbor_deletion` は次のように定義される。

$$\begin{aligned} \text{neighbor_deletion}(A, B) \\ = (\text{last_advertise_rate}(A, B) \\ + \text{max_advertise_rate_increase}) * \beta \quad (2) \end{aligned}$$

ただし、 β はメッセージ衝突による `advertise` メッセージのロストを考慮し、`advertise` メッセージを何回受け取らなかった場合に隣接端末を削除するかを定めるパラメータである。`max_advertise_rate_increase` は、`advertise_rate` の変動における 1 回あたりの最大増加値であり、`advertise_rate` が変動するために必要となる。

2.4 伊藤らの方式の課題

ここまで述べてきた伊藤らの提案方式を用いることで従来の方式と同程度の発見遅延時間でメッセージ数を削減することができる。しかし、この方式の課題として、静的なネットワークにおいて各端末の、`advertise_rate` が `max_advertise_rate` に近い状況のとき、隣接でなくなった端末をルーティングテーブルから削除するまでの消滅遅延時間が従来の方式よりも大

きくなってしまふことが上げられる。

ここで、一般的な Proactive 型ルーティングの場合、ルーティングテーブル作成のためには端末同士が互いに自身の持つ隣接端末等の情報のやり取りが必要となる。このとき隣接端末の情報を際限なく通信パケットに入れて送ってしまうと通信帯域の圧迫となる恐れがある。特に端末数の大規模なアドホックネットワークの場合はそれが問題となり、本来の目的である通信の妨げとなる可能性が大きい。そのため、ルーティングテーブルに登録できる隣接端末の最大値を決めてルーティングテーブルを作成する。

このように最大値を決めてルーティングテーブルを作成するとき、ルーティングテーブルが隣接端末の登録で一杯になった場合、新しく隣接となった端末が現れてもその端末が登録されるためには既に登録されている端末のうち、隣接でなくなった端末がルーティングテーブルから削除されるまで待たなければならない。これが、伊藤らの方式での静的なネットワークであった場合だと、`max_advertise_rate` の定数倍の時間経過して、隣接でなくなった端末がルーティングテーブルから削除されない限り、ルーティングテーブルの書き換えが行われないため、通常よりも大きな発見・消滅遅延時間となってしまふ。

そもそも Proactive 型ルーティングは Reactive 型ルーティングに比べ、高密度のアドホックネットワーク向きのプロトコルであるので、こういったルーティングテーブルの課題は検討すべきである。

そこで、伊藤らの提案方式の課題、および、ルーティングテーブルの書き換えの課題を元に、隣接端末発見精度向上方式を提案する。

3. 提案方式の概要

第 2 章の終わりで述べたような状況の時、どのようにすれば新規端末が隣接端末としてルーティングテーブルに登録されるかを考える。ここでアドホックネットワーク中で移動する端末の移動について考える。端末が移動する時、どのように動くのかを考えると、何か目的があつて移動を行っているはずである。何か目的があつて移動を行っていると仮定すると端末は目的地を目指してある一定の予測可能な移動をしているはずである。このことから端末の移動をそのときの移動速度などから予測してはどうかと考える。提案方式ではその予測をルーティングテーブルの登録時に用いる。

ルーティングテーブルに登録されるためには、既に登録されている端末がテーブルから削除されなければならない。では、どの端末ならば削除しても問題ない

かと考えた場合、以下の2種類の端末ならば削除しても問題がないと考えることができる。

- 既に通信可能範囲外に移動していて neighbor_deletion 経過するのを待つだけの端末
- 数秒後には通信可能範囲外に移動する端末

これらの端末をルーティングテーブルから削除し、新しく通信可能範囲内に入った端末を登録することで新規端末の発見精度が向上すると考える。

しかし、新しく通信可能範囲内に入った端末が上記の2番目の端末よりも長く隣接で居つづけるとは限らない。誤って隣接でいる時間の短い端末で、より長い時間隣接でいる端末を上書きしてしまうと、その後再び削除された端末が隣接であると判断され、他の端末を上書きするということが起こりうる。そのため、慎重に削除する端末を選択しなければならない。

そこで以下に示す方法により、長く隣接でいる端末を選んでルーティングテーブルの更新を行い伊藤らの方式、およびルーティングテーブルの課題の改善を図り、隣接端末発見精度向上を目指す。

3.1 速度と位置情報によるルーティングテーブル更新方式

登録数の限界に達したルーティングテーブルの書き換えを早めるために、速度と位置情報を用いて端末の予想離脱時間を算出し、その時間を各端末の優先度と見立てて、優先度の低い端末をルーティングテーブルから削除する方法を以下に提案する。

関連研究に、車両アドホックネットワークにおける Reactive 型プロトコルでの車両同士の相対速度と車間距離を用いた安定ルート構築のためのルーティングプロトコルを提案したもの³⁾がある。しかし、本提案方式は速度、位置の情報を利用するものの、第一に Proactive 型ルーティングであること、第二にルーティングにおける制御用メッセージ数の削減および、発見遅延時間の減少を目的としているという点で趣旨が違ふ。なお、本提案方式は Proactive 型ルーティングのルーティングテーブル更新に関する提案なので、DECENTRA に限らず他の Proactive 型ルーティングプロトコルに適用することが可能である。

速度と位置情報による優先度

まず、本論文で提案する方式では、advertise メッセージに送信元端末の速度と位置情報を加えて送信する。advertise メッセージを受け取った端末はルーティングテーブル登録時には速度と位置情報の登録も行う。

各端末は advertise メッセージを受け取りルーティングテーブルに登録する際、まずルーティングテーブルの登録状況を確認する。ルーティングテーブルに、

受け取った advertise の情報を登録する余裕がある場合はそのまま登録し、もしルーティングテーブルが一杯であった場合、自身と advertise メッセージに含まれている送信元端末、そしてルーティングテーブルに登録してあるすべての端末の速度と位置情報、そして各端末の advertise メッセージ受信（登録）時間を用いて受信端末自身に対する優先度を計算する。

この優先度は、advertise の受信端末と送信端末があとどのくらいの時間で通信可能範囲よりも離れるかを表すので、自分の通信可能範囲外に最も早く離れていくであろう端末がどれであるか表しているといえる。そして既にルーティングテーブルに登録されている端末の中で、advertise メッセージを送信してきた端末よりも優先度の低い端末があれば、その端末をルーティングテーブルから削除して、新たに advertise を送ってきた端末を登録する。これにより、間もなく通信範囲外に移動する端末をルーティングテーブルから削除することができ、より端末の通信範囲内に居るであろう端末を登録することができると考える。

以下に端末 A と端末 B の優先度の決定方法を示す。ここで、端末 A が advertise 受信端末であり、ルーティングテーブルの更新を行う端末であり、端末 B が advertise を送信した端末、もしくはすでにルーティングテーブルに登録されている端末を示す。 \vec{V}_A 、 \vec{V}_B が端末 A、B それぞれの速度を表すベクトルで、 P_A が端末 A の現在の位置、 P_B が端末 B の advertise を送信した時の位置を表す。 P'_B が現在の端末 B の位置、 P'_A 、 P''_B が T 時間経過後のそれぞれの端末の位置を表す。 R_A が端末 A の電波到達範囲（通信可能範囲）であり、t が現在の時間、 T_B が端末 B が端末 A に対して advertise を送信した時間、T が端末 B が端末 A の通信可能範囲外まで移動するのにかかる時間を表し、この T が優先度となる。

優先度 T は、値が大きいくほど通信可能範囲をでるまでに時間がかかることを意味するので大きいほど優先度が高い。

$$\begin{aligned}
 \vec{V}_A &= (v_{Ax}, v_{Ay}) \quad , \quad P_A = (p_{Ax}, p_{Ay}) \\
 \vec{V}_B &= (v_{Bx}, v_{By}) \quad , \quad P_B = (p_{Bx}, p_{By}) \\
 P'_B &= (p'_{Bx}, p'_{By}) \\
 &= (p_{Bx} + v_{Bx} * (t - T_B), p_{By} + v_{By} * (t - T_B)) \\
 P'_A &= (p_{Ax}, p_{Ay}) \\
 &= (p_{Ax} + v_{Ax} * T, p_{Ay} + v_{Ay} * T) \\
 P''_B &= (p''_{Bx}, p''_{By}) \\
 &= (p'_{Bx} + v_{Bx} * T, p'_{By} + v_{By} * T) \\
 R_A &= \sqrt{(p'_{Ax} - p''_{Bx})^2 + (p'_{Ay} - p''_{By})^2} \quad (3)
 \end{aligned}$$

式 (3) より求められる T が、端末 B が端末 A の通信範囲外に移動するまでにかかる時間を表す。後は式 (3) を使い、ルーティングテーブルに登録されている端末全てと advertise を受け取った端末の優先度 T を求め、最も T の小さかった端末をルーティングテーブルから削除する。これにより端末のルーティングテーブルの更新精度が高まり、隣接端末発見精度が向上することが期待できる。

離脱予測時間による消滅遅延時間の改善

式 (3) によって端末の離脱予測時間を求めることができた。これを advertise メッセージ受信時のルーティングテーブル登録時以外にも利用できないか考える。

端末の離脱予測時間を求めることができたということは、各端末はルーティングテーブルに登録されている端末がいつ隣接でなくなるかをあらかじめ予測できていることになる。この予測を neighbor_deletion の判定に用いることで、より正確にルーティングテーブルから端末の削除を行えるはずである。

離脱予測時間と advertise_rate と neighbor_deletion の関係を以下のように用いることで端末削除時間をより正確に設定し、消滅遅延時間の減少を期待する。

- i. $T \leq \text{advertise_rate} + \text{max_advertise_rate_increase}$
 端末 A が端末 B の advertise_rate + max_advertise_rate_increase 時間内に次の advertise を受信しなければルーティングテーブルから削除
- ii. $\text{advertise_rate} + \text{max_advertise_rate_increase} < T \leq \text{neighbor_deletion}$
 時間 T 以内に次の advertise を受信しなければルーティングテーブルから削除
- iii. $\text{neighbor_deletion} < T$
 neighbor_deletion 時間内に次の advertise を受信しなければルーティングテーブルから削除

速度ベクトル変化と advertise_rate の再設定

伊藤らの方式の場合、静的なトポロジー中の端末の advertise_rate は max_advertise_rate に近い値となっている。この時ある端末が別の静的なトポロジーに移動した場合、移動先ではどの端末も advertise_rate が大きな値となっているため、advertise が行われるまで時間がかかり発見遅延時間が大きくなる。

そこで、以下に示すように端末に速度ベクトルの変化が起こった場合、つまり、進行方向の変化が起こった場合、advertise_rate を変化させる方式を提案する。

- (1) 速度ベクトルが変化する
- (2) advertise_rate の値を min_advertise_rate に再

設定

- (3) n 回 advertise を送信後、新規隣接端末を発見しなければ再設定前の advertise_rate に戻す
- (4) (3) で新規隣接端末を発見していたらそのままルーティングを続ける

これより、端末が静的なトポロジー間の移動を行ってもすぐに advertise を行うので、伊藤らの方式で発生していた発見遅延時間の減少が期待できる。

4. 評価

今回の評価ではルーティングテーブル更新に関して提案方式の有効性を検証するために、シミュレータを作成し、伊藤ら方式を従来方式とし、従来方式と提案方式それぞれについて、次のように条件を設定してシミュレーションを行った。

設定条件

- 電波到達距離： 100m
- フィールド： 250m × 250m , 500m × 500m
- 端末数： 100
- 端末の移動速度： 1m/s ~ 5m/s
- 移動端末のパターン：フィールド上でランダムな点を選択し、ランダムで決められた上記の範囲内の移動速度で目的地に直進、到着後再びランダムな点へと移動を繰り返す
- シミュレーション時間： 30min
- β (隣接端末をルーティングテーブルから削除するまでの advertise 回数)： 3
- ルーティングテーブルの最大登録数： 8
- advertise_rate : 2sec ~ 90sec
- max_advertise_rate : min_advertise_rate + 60sec
- max_advertise_rate_increase : 4sec

シミュレーションでは、従来方式、および提案方式のメッセージ数と遅延時間の関係を調べる。従来方式、提案方式ともに advertise_rate を 2 秒 ~ 62 秒の範囲から、90 秒 ~ 150 秒の範囲にそれぞれ変動する範囲を定めてシミュレーションを行い、それぞれのメッセージ数、および遅延時間を得る。以上の条件で行ったシミュレーションの結果を以下に示す。フィールドサイズ 250m × 250m の結果を図 1、フィールドサイズ 500m × 500m の結果を図 2 に示す。

また今回のシミュレーションでは、ルーティングテーブルの最大登録数が 8 と制限されているので、ある端末の通信可能範囲内で advertise メッセージを送信したが、ルーティングテーブルに空きがなく登録されない端末が発生する。フィールドサイズ 500m × 500m のシミュレーションにおいて、シミュレーション全体

で、登録数の制限でルーティングテーブルに登録されなかった端末の総数を測定したものを図3に示す。

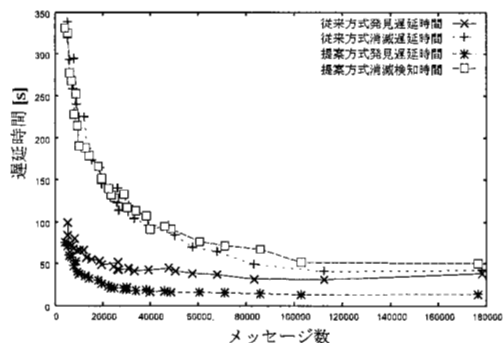


図1 シミュレーション結果1 [field size: 250 × 250]

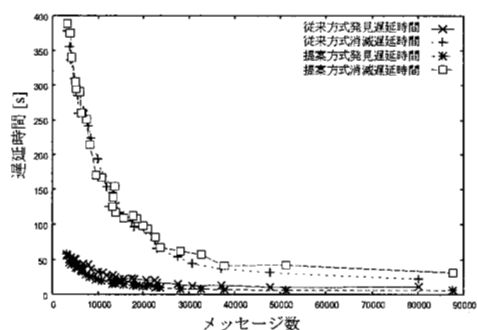


図2 シミュレーション結果2 [field size: 500 × 500]

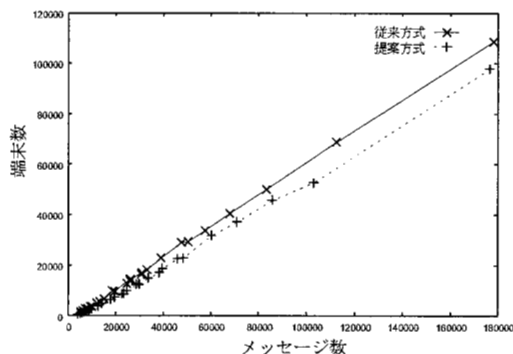


図3 シミュレーション結果3 [field size: 500 × 500]

図1、図2よりどちらもメッセージ数と消滅遅延時間の関係についてはメッセージ数の少ない部分ではほぼ同等であるといえるが、メッセージ数の多い部分では若干提案方式の方が劣った結果となっている。しかし発見遅延時間を見てみるとどちらも提案方式のほうが優れている。特に図1では従来方式の半分程度に抑えることができています。これは図1のフィールドの

端末密度が高いので、ルーティングテーブルが登録限界数まで埋まってしまうため、特に提案方式の効果が出ていていると言える。

また図3より従来方式と提案方式では、提案方式の方が、ある端末が他の端末の通信可能範囲内で advertise メッセージの送信をしたが、それを受信した端末のルーティングテーブルに登録されなかったという端末の数を少なくすることができていることがわかる。

以上から、提案方式では従来方式と同程度の消滅遅延時間を維持しつつ、発見遅延時間を減少させることが確認できた。

5. まとめ

本研究では、アドホックネットワークにおける動的なルーティングパラメータの変更による通信メッセージ削減方式の課題を解決するために、速度と位置情報を用いることで端末ごとに優先度を設定する方式を提案した。この優先度は端末毎の離脱予測時間であり、これによってルーティングテーブルの更新をやめ、発見遅延時間を減少させることができた。

また、離脱予測時間を advertise を受け取らなくなった端末の消滅判定に用いる手法と、速度ベクトルの変化で advertise_rate を変化させる手法の提案を行った。

今後は、この離脱予測時間を用いた消滅判定手法と速度ベクトル変化による advertise_rate 再設定手法に関するシミュレーションを行い、評価を行う予定である。

また、その評価の終了後は、これまではメッセージ数と隣接端末の発見・消滅に関する遅延時間の関係について検討を行ってきたが、今後の課題としては従来方式と伊藤らの方式および、提案方式を用いることにより作成されるルーティングテーブルから求めた経路についての、安定度やメッセージ到達率等に関する評価を行うことが上げられるので、こちらについても評価・考察を行っていく予定である。

参考文献

- 1) 伊藤健二, 若山公威, 岩田彰, 梅田英和: “愛・地球博における大規模モバイルアドホックネットワーク実証実験の検証” 情報処理学会研究報告 2006-DPS-126, 2006-CSEC-32, pp147-152, March 2006
- 2) “SKYLEY NETWORKS” <http://www.skyley.com/>
- 3) 吉田成志, 新井国充, 浅見重幸, 三木哲也: “車両アドホックネットワーク向けルーティング方式の提案”, 電子情報通信学会論文誌 B Vol.J88-B No.8 pp1434-1443, 2005