社団法人　情報処理学会　研究報告
IPSJ SIG Technical Report

2006－MBL－36（46）
2006－UBI－10（46）
2006／2／17

端末の高速移動に対応するOLSRの機能拡張に関する検討

近藤　　毅†　　加藤　聰彦†　　伊藤　秀一†

† 電気通信大学　大学院情報システム学研究科東京都調布市調布ヶ丘１－５－１
E-mail: †{kondoh,kato,itoh}@net.is.uec.ac.jp

あらまし　近年無線アドホックネットワークが広く検討されており，プロアクティブ型のルーチングプロトコルではOLSR (Optimized Link State Routing) が注目されている．OLSR は隣接ノードの発見やトポロジ情報の流布を定期送信に依存している．このためノードが高速に移動するようなトポロジが頻繁に変化する状況下では，迅速なリンク変化の検出およびトポロジ情報の入手ができない．これに対し本稿では，OLSR に対して，端末が高速に移動する場合の性能を改善する手法を提案する．具体的には，移動方向の前方では，これまでに隣接関係を確立していないノードからのHello に対して，すぐに Hello を返信するすることにより早期に隣接関係を確立する．これまでにネットワークに存在しないノードを発見した場合は，そのノードに対して MPR (Multipoint Relay) がトポロジ情報をユニキャストで注入することにより経路を確立させる．また，移動方向の後方のノードに対しては，移動の強度に従い，隣接関係有効時間をフィードバック制御するという方法を提案する．本稿では，これらの方法に対するシミュレーションによる評価を行った結果も示し，その有効性を明らかにする．
キーワード　アドホックネットワーク，ルーチングプロトコル，OLSR，高速移動

A Study on Enhancement of OLSR for High Speed Mobility

Tsuyoshi KONDOH†, Toshihiko KATO†, and Shuichi ITOH†

† Graduate School of Information Systems, University of Electro-Communications
1-5-1 Chofugaoka, Chofu-shi, Tokyo, 182-8585 Japan
E-mail: †{kondoh,kato,itoh}@net.is.uec.ac.jp

Abstract　Recently, Mobile Ad hoc Networking (MANET) is being widely studied, and OLSR (Optimized Link State Routing) is paid much attention as a proactive routing protocol. OLSR depends on periodic transmission of control messages for the neighbor discovery and the flooding of topology information. So, it is considered that OLSR has some problems on the link establish and the maintenance of topology infomation for the case of frequent topology change resulting from high speed node mobility. This paper proposes some extensions of OLSR for improving performance for this case. First, for a moving node and nodes located in the direction of the movement, it proposes a method that nodes which receive a Hello message from a non neighbor node will send a Hello message instantly. It also proposes a method that, if some nodes find a node that is not a member of the network, the multipoint relay selected by the moving node unicasts Topology Control messages. For nodes located in the reverse direction of the movement, it proposes a method that nodes feedbacks the neighborhood maintaining timer according to the relative speed of the node movement. This paper also describes the simulation results for our proposals and shows their effectiveness.
Key words　Ad hoc Network, Routing Protocol, OLSR, High Speed Mobility

1. はじめに

近年，新たな無線ネットワークの形態として，アドホックネットワーク [1] が活発に検討されている．ここでは，無線インタフェースを持つノードが，端末としてだけでなくルータとしても動作し，IP ルーチングによるマルチホップネットワークを自律的に構築する．IETF で検討されたOLSR (Optimized Link State Routing) [2] は，プロアクティブ型のルーチングプロトコルの代表例であるが，ノードが別のネットワークへ移動し通信を再開するまでに遅延が生じるという問題点がある [3].

ノードは通信開始までに，隣接ノードを発見し，さらにトポロジ情報を受信し経路計算を行う．前者は Hello message の送信間隔である hello_interval によって，後者は TC message (Topology Control message) の送信間隔である TC_interval に依存して遅延を生じさせる．RFC での標準値を用いた場合には，前者によって生じる遅延が2〜6 秒，後者によって生じる遅延が約 5 秒である．このため，新規に通信可能になったノードが実際に通信を開始するまでには最悪で 11 秒の遅延が生じる．これに対し本稿

では，Hello のトリガ送信及び近接ノードによるトポロジ情報の注入を用いてこの遅延を低減する方法を提案する．

さらに移動後方に目を向けると，それまで通信可能であったノードが，移動に伴い通信範囲から離れることになる．離れるノードを事前に知ることは不可能であり，タイムアウト (aging) 処理により隣接ノード情報から削除される．この時間は neighbor_holdtime(=6sec) として定められているが，移動速度や通信範囲に依らない一定値であるため，高速移動環境下では通信不可能なノードが増え，パケット到達率を下げることになると考えられる．このようなノードを減らすため，移動の激しさを隣接ノード情報の変化から測定し，これに従って隣接関係の有効時間をフィードバック制御するという方法を提案する．

さらに本稿では，提案する方法をネットワークシミュレータを用いて評価した結果についても報告する．

2. OLSR の動作とその問題点

2.1 OLSR の動作

OLSR は無線メディアの特性に合わせて最適化されたリンクステート型のルーチングプロトコルである．OLSR では経路の決定に隣接ノード情報とトポロジ情報の 2 つの情報を用いる．この 2 つの情報の生成と入手方法に着目しながら，OLSR の動作概要を説明する．

2.1.1 隣接ノードの発見

ネットワークに参加している全ノードは通常 2 秒に設定される hello_interval ごとに Hello message を送出する．Hello を受信することで通信可能範囲に存在するノードを知る．Hello には自身の隣接ノード (Neighbor) が列挙されており，この中に受信ノード自身があれば，Hello を送出したノードとの間で双方向通信が可能と判断できる．さらに OLSR では Hello を用いて，自身から 2 ホップで到達可能なノード (2hop neighbor) の情報を入手する．受信した Hello 内の Neighbor の論理和から直接通信可能なノードを引くことで 2hop neighbor を求める．この情報を用いてどのノードを MPR (multipoint relay) に選定するべきかを判断する．

MPR とはトポロジ情報流布のための TC message の生成や転送を行うノードである．MPR は 2hop neighbor のカバー率の大きいノードから順に自身の全 2hop neighbor をカバーするように選ばれる．MPR の選定通知は Hello で行われる．MPR を選んだノードは MPRSelector と呼ばれ，また MPR も他のノードを MPR と選ぶため，MPRSelector となる．

2.1.2 トポロジ情報と経路の決定

MPR は，自身の MPRSelector の情報を TC に載せて広告する．MPR 同士の間で TC をブロードキャストで再転送することで，結果として全てのノードが TC を受け取る．この最適化により，従来のリンクステート型のプロトコルに比べルーチングトラヒックを減らすことができる．

最後に各ノードにおいて受信した TC の集合であるトポロジ情報から SPF (Shortest Path First) Tree を作成し，それ基づきルーチングテーブルを決定する．

2.2 OLSR の問題点

OLSR はダイナミックにトポロジが変化する環境を想定したプロトコルである．OLSR におけるノードの移動に対する取り扱いと，その問題点を検討する．

2.2.1 OLSR における移動の取り扱い

ノードが移動するとノード間の隣接関係が変化する．移動方向前方では，新たなノードと隣接関係を確立する必要があり，移動方向の後方では，通信できなくなったノードとの隣接関係を無効にする必要がある．Hello を用いた隣接関係の確立には複数回の Hello の送受信が必要である．しかし，hello_interval は移動速度に依らない一定値であり，ノードの移動速度が増加すると隣接関係の確立までの時間が相対的に大きくなる．

一方，一度確立した隣接関係の有効時間も一定値である．この時間は neighbor_holdtime と呼ばれ，hello_interval の 3 倍の 6 秒が規定されている．この有効時間が経過した後に，リンクが切れたと判断され，その後新たな経路が探されるため，長いダウンタイムが生じる．このため，高速移動環境下では無効になったリンクの発見が遅れ，パケットの到達率が下がる．さらに，ノードが新規にネットワークに参加する場合，隣接関係確立後，ネットワーク中の TC を受信するまで経路を確立することができず，通信開始までに長い遅延が生じる．

前の 2 つは隣接ノード情報のメンテナンスに関わる問題であり，最後の 1 つはトポロジ情報の入手に関わる問題である．ノードの移動速度が大きく，ネットワークのトポロジが頻繁に変化する環境では性能が低下すると考える．そこで本稿では，このようなノードの高速移動に伴う頻繁なトポロジの変化に対して，早期に対応できるような OLSR の機能拡張について検討した結果を述べる．

3. 高速動作に対応する OLSR の改良提案

3.1 設 計 方 針

繰り返しになるが OLSR は経路決定のために，“ 隣接ノード情報 ”と“ トポロジ情報 ”の 2 つの情報を用いる．本稿ではこの 2 つの情報を新しく保つことで，ルーチング精度を高めパケットの到達率を上げるための提案を述べる．具体的には下のような提案を行う．

- 隣接関係を結んでいないノードからの Hello を受信した場合，すぐに Hello で返事をすることで高速に隣接関係を確立する
- ノードの移動度に従ってダイナミックに隣接関係の有効時間を制御することで，通信不可能になったノードを早期にタイムアウトさせる
- 新規参加ノードへトポロジ情報を注入することで，新規参加ノードが早期に通信を開始できるようにする

隣接ノード情報はネットワーク中の各ノードが保持している情報で，直接通信可能であることを確認した（隣接関係を確立した）ノードの集合である．ある移動中ノードの前後における隣接ノード情報のメンテナンスをもとに検討を行う．

3.2 移動方向前方における提案

移動方向前方では，移動に伴い新たに通信できる範囲に入ったノードと複数回の Hello のやりとりにより隣接関係を確立する．この手順を図 1 に示す．

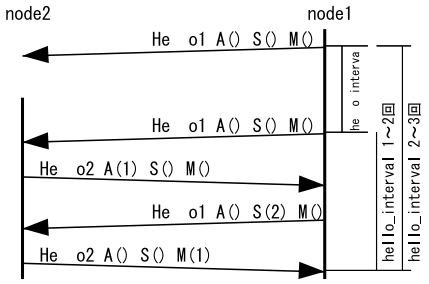

図 1 既存の隣接関係確立手順

両方のノードが隣接関係を確立するまでに 3 回，MPR の選定通知も入れると 4 回の Hello 送信が必要である．これにかかる時間は hello_interval が 2 秒とすると，図 1 のように 2〜6 秒となる．この時間はノードの移動速度に関わらず一定である（図中 A,S,M はそれぞれ ASYM, SYM, MPR を示す）．

3.2.1 Hello のトリガ送信

この時間を短縮するために以下の提案を行う．すなわち，隣接関係の確立を高速化するために，現状 Neighbor で無いノードからの Hello を受信したら，即座に Hello で返信することとする（図 2 参照）．

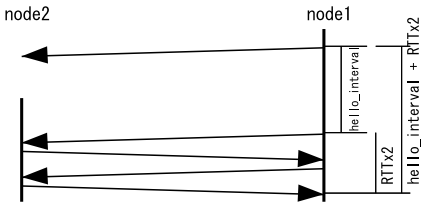

図 2 隣接関係の確立の高速化

それまで Neighbor でないノードからの Hello がトリガになり，Hello の送信を呼ぶため，最初の Hello 以降は実質通信に必要な時間だけで隣接関係を確立できるようになり，Neighbor になるまでの時間が大幅に縮小すると考えられる．

3.3 移動方向後方における提案

前述のように，移動後方においては，それまで通信可能であったノードが，移動に伴い通信不能となる．これらのノードを検出するために，neighbor_holdtime(=6sec) を用いたタイムアウト処理を行う．この時間は移動の激しさに依らず定められているため，この時間を移動速度に応じてフィードバック制御する手法を提案する．

3.3.1 neighbor_holdtime のフィードバック制御

まず，隣接関係の有効時間を適切にフィードバックするために，考慮すべきパラメータを検討する．

リンクの変化頻度

リンクの変化頻度は無線到達距離と移動速度に依存する．これは移動の激しさと言い換えることができる．

Hello の受信成功確率

neighbor_holdtime が hello_interval の複数倍に設定されているのは，Hello の取りこぼし，つまりまだ有効なノードを誤って無効であると誤認することを防ぐためである．無線 LAN において，ブロードキャストフレームである Hello に対してはリンク層での確認応答や再送が行われない．このため，無線チャネルが混雑してくると，隠れ端末問題による取りこぼしが生じやすくなる．つまり，闇雲に neighbor_holdtime を短くすればよいわけではない．

本提案では，この 2 つの要因を考慮し，隣接関係有効時間をフィードバック制御する．以下，それぞれのパラメータを定量的に扱うための指標を示す．

3.3.2 リンクの変化頻度・移動の激しさ

リンクの変化頻度・移動の激しさを表す指標を検討するために，ノードの通信範囲と移動に伴うその変化を見る．図 3a の 2 つの円は $t1, t2$ 各時点におけるノードの無線伝搬範囲を示している．r(m) は無線伝搬半径，x(m/s) はノードの移動速度，$t1, t2$ は時刻で，$t2 = t1 + \text{nb\_holdtime}$ とする．

図中 $t1$ の場所に居たノードが nb_holdtime 秒の時間をかけて $t2$ の場所へ速度 x(m/s) で移動する．nb_holdtime は既に述べたとおり隣接関係の有効時間で，通常 6 秒に設定されている．$t2$ の時点で実際に通信可能であるノードの範囲は図 3a の網掛け部で，面積は πr^2 である．これを $S_{\text{real\_neighbor}}$ と呼ぶ．$t2$ の時点で隣接ノードとして扱われる範囲は図 3b の網掛け部で，面積は $\pi r^2 + \text{nb\_holdtime} \times 2rx$ となる．この範囲を S_{neighbor} と呼ぶ．移動方向後方には隣接ノードとして扱われているが，実際には通信することができないノードが発生している．また，$t1$〜$t2$ の間に新たに隣接ノードとなった範囲は図 3c の網掛け部で，面積は $\text{nb\_holdtime} \times 2rx$ となる．これを $S_{\text{new\_neighbor}}$ と呼ぶ

移動の激しさは無線伝搬半径と移動速度との比である

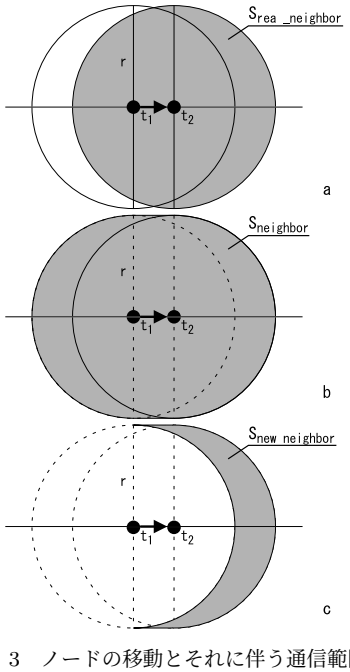

図 3 ノードの移動とそれに伴う通信範囲

と言える．これを表現するために本稿では以下のように new_nb_rate を定義する．

$$\text{new\_nb\_rate} = \frac{S_{\text{new\_neighbor}}}{S_{\text{neighbor}} - S_{\text{new\_neighbor}}}$$

各範囲の面積はすでに述べたとおりであるため，$\text{new\_nb\_rate} = \frac{\text{nb\_holdtime} \times 2x}{\pi r}$ となる．式中 r, x 以外は定数であり，new_nb_rate は無線伝搬半径と移動速度の比である．

これを実際のシミュレーションに適応するため，面積を該当するノードの数に置き換える．

$$\text{new\_nb\_rate} = \frac{\#\text{new\_neighbors}}{\#\text{neighbors} - \#\text{new\_neighbors}}$$

これは隣接ノード情報に多少の拡張をすることで容易に観測可能な値であり，電界強度や位置情報などの付加情報を一切用いる必要がない．new_nb_rate を用いることで，ある node から見た周囲の移動速度を表すことができる．これは”相対移動速度”とも言え，移動の”激しさ”を表している．

この情報を元に neighbor_holdtime をフィードバックし，ノードの移動方向後方に発生する隣接ノードとして扱われるが実際には通信することができないノードを減らす．

フィードバックを行うと各面積が変化してしまうが，これは容易に補正可能である．フィードバックにより nb_holdtime が current_nb_holdtime に変化し，これに伴い $S_{new\_neighbor} = current\_nbholdtime \times 2rx$ となる．これを補正するために $\frac{\text{default\_nbholdtime}}{\text{current\_nbholdtime}}$ との積を求める．

$$\text{new\_nb\_rate} = \frac{S_{\text{new\_neighbor}}}{S_{\text{neighbor}} - S_{\text{new\_neighbor}}} \times \frac{\text{default\_nbholdtime}}{\text{current\_nbholdtime}}$$

同様に各範囲のノード数により求めた値も補正できる．

$$\text{new\_nb\_rate} = \frac{\#\text{new\_neighbors}}{\#\text{neighbors} - \#\text{new\_neighbors}} \times \frac{\text{default\_nbholdtime}}{\text{current\_nbholdtime}}$$

3.3.3 Hello の受信成功確率とリンク失効の誤認

Hello の取りこぼしの主要因は，受信端末における隠れ端末問題によるコリジョンである．802.11 無線 LAN のデータリンク層においてブロードキャスされる Hello は確認応答・再送がされず，RTS/CTS も用いないため受信ノードが CTS を返すこともない [4]．このため，受信側におけるフレーム衝突が生じやすい．この確率は，無線チャネルの混雑率と，フレームの長さに依存する．フレームの長さは，Hello のみであれば，隣接ノード数で決まる．しかし，Hello に TC が piggy back されている場合，フレームの長さはより長くなる．

piggy back によりフレーム長がダイナミックに変化するため，本提案では無線チャネルの busy 率を用いて，Hello の受信成功確率を表すのが妥当であると考える．

3.3.4 フィードバック係数の検討

2 つのフィードバック係数を元に neighbo_holdtime を決定すると，neighbo_holdtime は 3 次元マップで示される．しかし，今回無線チャネルの busy 率を測定することができなかったため，フィードバックは new_nb_rate の増加に従い，2〜6 秒の範囲で neighbor_holdtime を必要な分だけ短縮することで行う．

実際に用いるフィードバック係数は評価と共にシミュレーション環境で new_nb_rate を測定した上で決定する．

3.4 新規参加ノードへの提案

経路を決定するために必要なもう 1 つの情報はトポロジ情報である．トポロジ情報は TC の集合で，TC は MPR のみが送信するメッセージである．

MPR にとっての MPRSelector に変化が生じた場合，標準で TC が出る．それ以外の場合，一定間隔毎（5sec）に TC が送信される．

ネットワークに新たなノードが参加した場合を想定する．新規参加ノードは Hello により隣接ノードを発見し，MPR を選ぶ．MPR に選ばれたノードは新たにやってきたノードの情報を TC によってネットワーク中に知らせる．これにより，以前からネットワークに参加していたノードは新規参加ノードを知る．

一方，新規参加ノードが，既存のノードへの経路を確立するためには，ネットワークを流れる TC を一通り受信する必要がある．これには TC の送信間隔である 5 秒が必要である．隣接関係の確立は Hello のトリガ返信による高速化を用いることができるが，TC の送信間隔に依存するこの時間は変化しない．

そこで，新規参加ノードから MPR に指定されたノードが自身の持つトポロジ情報を新規参加ノードへ注入すると言う手法を提案する．これにより，新規参加ノードが迅速に経路を確立することができるようになる．

以下に具体的な手法を記述する．まず，新規参加ノードであるか否かの判断は，ルーチングテーブルを見ることで行う．プロアクティブ型ルーチングプロトコルはネットワークに参加している全ノードへの経路を持っているため，ルーチングテーブルにエントリがなければ新規参加ノードである．情報の注入を行うノードは，新規参加ノードから MPR に選ばれたノードとする．注入する情報は，トポロジ情報として全ノードが保管している情報である．これを再度 TC の形に構成し，ユニキャストで該当ノードへ送信する物とする．これにより，小変更で既存ノードとの矛盾も無く機能を実現することができる．

さらに，副作用として TC の定期送信間隔を延ばすことができ，定期送信に依るルーチングトラヒックを減らすことができる．もちろん変化があった部分はその都度 TC が送信されるため悪影響は小さい．

4. 提案手法の評価

4.1 移動方向前方に対する方式に対する評価

Hello のトリガ送信とトポロジ情報の注入の提案を 1 つの評価で確認する．評価には The Network Simulator ns2 を用いた．シミュレーション環境は以下の通りである．図 4 に示すように，無線伝搬半径は 150m とし，node1〜node7 を 150m 毎に直線的に配置する．シミュレーション開始時にはどのノードとも通信できない位置にいる node0 が 100 秒に node7 から 129m の位置に移動し，さらに 200 秒には左に向かって 5m/sec で移動を開始する．またシミュレー

ション開始時より node0 から node1 へ向けて CBR トラヒックを流す．

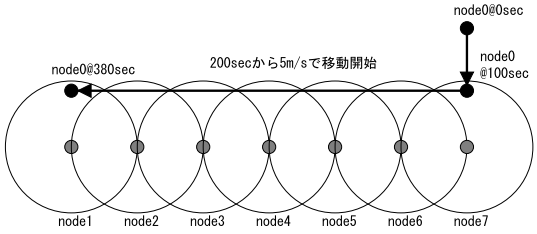

図 4　移動方向前方に対するシミュレーション

初期状態では通信不可能であるため node1 にはデータは到達しない．評価では，100 秒の時点の移動後に，7 ホップの経路を確立し node1 までトラヒックが到達するまでに必要な時間を観測する．さらに，200 秒の時点から移動を開始した後の通信状況の評価も行う．通常の OLSR を用いた結果を図 5 に，提案手法を取り入れた OLSR の結果を図 6 にそれぞれ示す．

通常の OLSR では，7 ホップの経路を確立するまでに 8〜10 秒かかった．今回の 2 つの提案を導入することで，この時間は 1〜1.5 秒へ短縮された．トリガとなる最初の Hello の受信までの時間，他のノードとの衝突を避けるためにジッタを挟んでいること考えると理想的な値である．

さらに，移動中のパケット到達率も改善している．これは Hello のトリガ返信により，移動中も新たな経路を早期に発見できるようになったためであると考えられる．

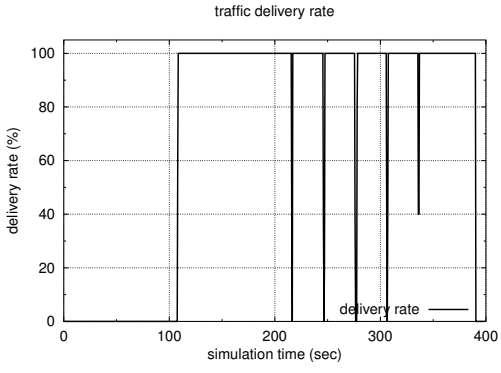

図 5　通常の OLSR による評価結果

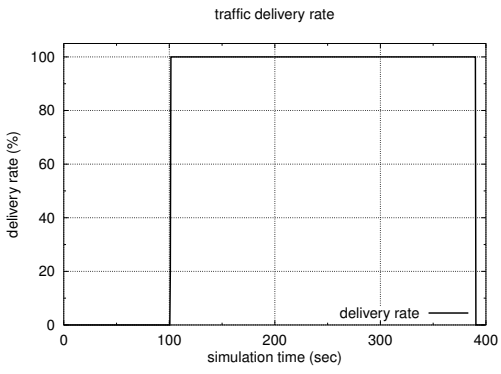

図 6　提案を取り入れた OLSR による評価結果

4.2　neighbo_holdtime のフィードバックを評価

次に neighbo_holdtime のフィードバック提案を検証する．今回の評価はダイナミックにトポロジが変化するような環境を想定する．また，実際の評価に先行して 3.3.2 節で検討した new_nb_rate を測定する．

4.2.1　new_nb_rate の実測

3.3.2 節では new_nb_rate を面積に基づく理想値として検討したが，実際には該当する範囲のノード数を用いて近似値を求める．実際にはノードの数は離散的で，ノードの配置もランダムであるため，誤差が予想される．特に隣接ノード数が少ない，ノードの配置が疎である場合，特に誤差が大きくなると予想される．

そこで実際の評価に先行して new_nb_rate をシミュレーション環境で実測する．環境は 1000×1000m の範囲に 100 ノードをランダムに配置し，そこで 1 ノードのみを環境の端から端まで直線的に一定速度で移動させる．100 ノードは静止しており，トラヒックは無い物とする．移動ノードの移動中の new_nb_rate の平均値を求めた．測定は 10 回行い平均値を図 7 に示す．さらに，同じシミュレーションを neighbo_holdtime を暫定的なフィードバック係数で制御した状態でも測定した．フィードバックに伴い変化する new_nb_rate は補正後の値を示している．

結果は理想値からある程度離れているが，移動の強度に伴いほぼ線形に new_nb_rate が上昇している．new_nb_rate によって移動の激しさを定量的に測ることができていると言える．

しかし，フィードバックを行った場合，移動速度が 10m/s を越えると new_nb_rate が著しく上昇し理想値から離れる．これはフィードバックに伴い Neighbor の数が少なくなり，さらに高速移動に伴い new_neighbor が増えることから誤差が大きくなったものと思われる．この問題を回避するために Neighbor の数が一定数を下回った場合フィードバックを停止するなどの対策が必要であると考える．これらを考慮して実際のフィードバックパラメータを決定する．

4.2.2　フィードバックパラメータの決定

実測した移動速度対 new_nb_rate の情報を元にフィードバックパラメータを検討する．変化させる範囲は標準となっている 6 秒から，hello_interval である 2 秒までとす

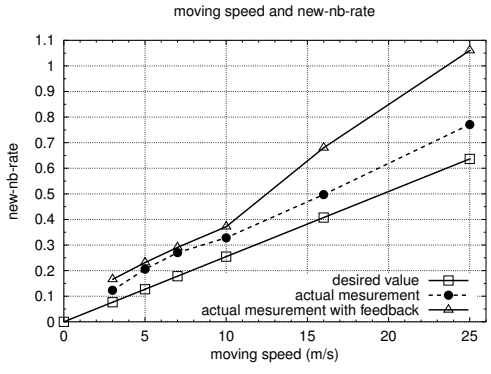

図 7　new_nb_rate の実測値

る．neighbor_holdtime を短縮するとパケット到達率が上昇するが，その一方で neighbo_holdtime を短縮しすぎると Hello の取りこぼしによりリンク切断を誤認してしまう．

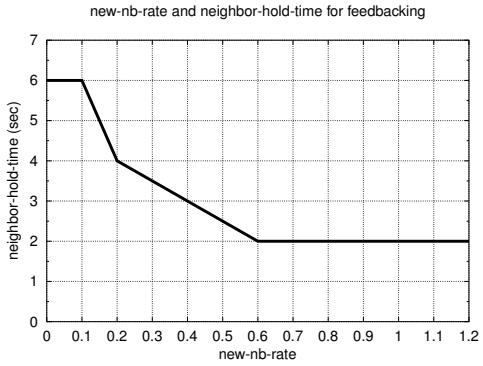

図 8　neighbor_holdtime のフィードバックに用いるパラメータ

今回は図 8 のようにフィードバック係数を定めた．さらに，neighbor の数が少ない（5 ノード未満）場合誤差が大きくなるためフィードバックを停止する．

4. 2. 3　neighbo_holdtime のフィードバックの評価

neighbo_holdtime を実際にフィードバック制御してその効果を検証する．シミュレーション環境は，670m × 670m の領域に 50 台のノードをランダムに配置し，全ノードが一定速度でランダムな方向へ移動し続けるものとする．無線伝搬半径は 150m である．トラヒックは送信・受信の 25 のノードペアの間に，各ペア間のスループットが 8Kbps の CBR トラヒックを発生させた．

このような環境で，ノードの移動速度を変化させネットワーク全体のパケット到達率とルーチングトラヒック (Hello, TC 両方) の量を測定した．実験は標準状態の OLSR と今回の提案を取り込んだ OLSR の両方で行い，結果を図 9，図 10 に示す．

移動速度の全域において提案手法のパケット到達率が通常の OLSR のそれを上回っている．この傾向は移動速度が速くなるほど顕著であり 16m/s では 39.4%から 53.5%への改善となった．

移動が激しくなるにつれ，提案する手順に従い，new_nb_rate が上昇し，nb_holdtime が短縮される．これによって無効になったリンクを早期に発見し，代替えの経路を用いるようになる．パケット到達率の上昇は，これに伴うルーチング精度の上昇によるものであると考えられる．

ルーチングトラヒックを見ると，提案手法におけるルーチングトラヒックは移動が激しくなるに従いリニアに増大していることがわかる．通常の OLSR はルーチングトラヒックが頭打ちになる傾向にある．これは提案手法に取り入れた Hello のトリガ返信により，変化した隣接ノードを早期にに発見できるようになったためであると考える．通常の OLSR では Hello の送信間隔が移動速度に依らず一定であるため，高速移動環境下ではノードの発見が追いつかず，ルーチングトラヒックが頭打ちになったと思われる．提案手法は移動の強度に従いルーチングトラヒックが増大しているが，これはトポロジの変化応じた物であり正常な動作であると考える．

5. おわりに

本稿では，OLSR の移動に対する動作を検討し，頻度なトポロジ変化に対応するための拡張を提案した．Hello のトリガ返信及びトポロジ情報の注入により移動方向前方のノードや新規参加ノードの経路確立を高速化し，移動の激しさに従い隣接関係の有効時間をフィードバック制御することにより高速移動環境下において無効リンクを早期に発見できるようになった．以上 3 つの提案により迅速に適切な経路を確立することができるようになった．さらに，評価によりその有効性も確認した．

文　献

[1] S. Corson, J. Macker: *Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and E valuation Considerations.* Request for Comments 2501 (January 1999)

[2] T. Clausen, P. Jacquet: *Optimized Link State Routing Protocol (OLSR).* Request for Comments: 3626 (October 2003)

[3] 近藤 毅，加藤 聰彦，伊藤 秀一：端末の高速移動に対応する OLSR の機能拡張に関する検討．電子情報通信学会 第 2 回アドホックネットワーク・ワークショップ (May 2005)

[4] ANSI/IEEE 802.11 standard: *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.* (1999)

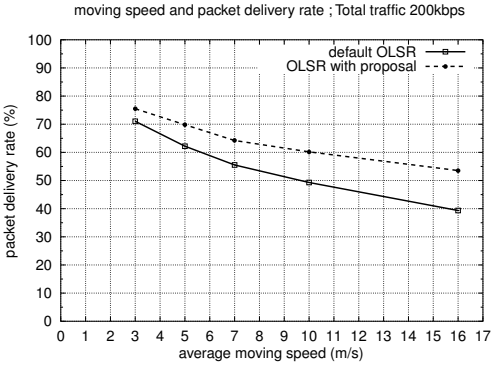

図 9　移動速度に対するパケットの到達率

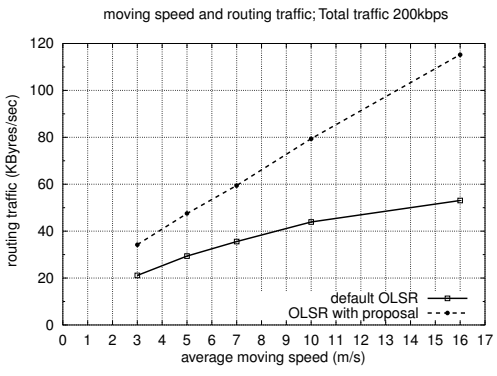

図 10　移動速度に対するルーチングトラヒック