

確率的ルーティングアルゴリズム ARH の MANET への適応手法

岩田 元[†] 松尾 啓志[†]

[†] 名古屋工業大学大学院工学研究科 〒 466-8555 名古屋市昭和区御器所町

E-mail: [†]gensan@matlab.nitech.ac.jp, ^{††}matsuo@nitech.ac.jp

あらまし 我々は、確率的ルーティングアルゴリズムの一つである、ARH (Ant Routing with routing history) に対する、MANET の特徴を考慮した改良手法を提案してきた。本稿では、我々が提案してきた改良手法に対して、シミュレーション評価を行い、提案手法と ARH, AODV (Ad hoc On-demand Vector Routing) を比較する。また、IEEE 802.11b 無線ネットワークを用いた実証実験を行い、提案手法と ARH, AODV を比較する。

The method adapting probabilistic routing algorithm ARH to MANET

Hajime IWATA[†] and Hiroshi MATSUO[†]

[†] Graduate School of Engineering, Nagoya Institute of Technology

Gokiso-cho, Syouwa-ku, Nagoya-shi, Aichi, 466-8555 Japan

E-mail: [†]gensan@matlab.nitech.ac.jp, ^{††}matsuo@nitech.ac.jp

Abstract We have proposed a technique that reform probabilistic routing algorithm ARH (Ant Routing with routing history) using the characteristic of MANET. In the proposed technique, a node sends hello packet with a data packet, doesn't use unreliable links, learns route using intercepted (not received) packet, and improve retransmitting method. In this paper, we compare with AODV (Ad hoc On-Demand Vector Routing), ARH, and proposed technique by conducting experiments in simulation.

1. はじめに

近年、無線端末の急速な普及により、MANET (モバイルアドホックネットワーク) [1] に期待が寄せられている。MANET では、無線基地局を介して通信を行う従来の無線ネットワークとは異なり、端末同士が直接通信を行うことにより、通信設備を設置することなく無線ネットワークを構築可能である。そのため、災害地、僻地での利用やセンサーネットワーク、車々間通信などの大規模ネットワークから、屋内での利用や携帯型ゲーム機における通信対戦などの小規模ネットワークまで、様々な用途への適用が考えられており、MANET ルーティングプロトコルが多数提案されている [2] ~ [5]。

また、Q-Routing [6], Ants Routing [7] など、強化学習を応用した確率的手法を、MANET のような頻繁にトポロジが変化するネットワークでのルーティングに用いることが考えられている。中でも、ARH (Ant Routing with routing history) [8] は、経路の履歴を用い、使用し

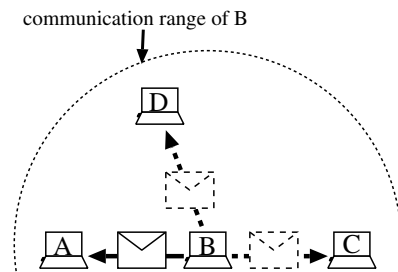


図 1 パケットの送信
Fig. 1 Sending packet.

た経路と逆向きの学習を行うことで効率的な学習を実現している。

ARH は、動的なネットワークへの適応を目的としたルーティングアルゴリズムである。しかし、ARH は、移動端末間の不安定な無線リンクによって構成される MANET 特有の環境を考慮していない。例えば、MANET の中でも、複数の端末が 1 個のチャンネルを共有し、無指向性

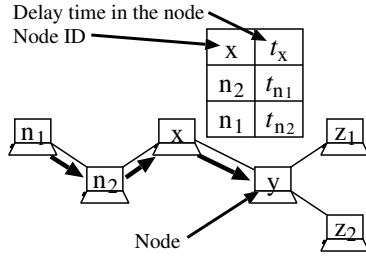


図 2 経路履歴情報

Fig. 2 Information of route history.

の電波を用いて通信を行うネットワークでは、各端末は近隣端末が送信したパケットを傍受することが可能である(図 1)。また、MANET 環境では、パケットの受信が非常に不安定な領域(グレーゾーン)[9]が存在する。

我々は、上記のような MANET 特有の環境を考慮した、ARH に対する改良手法を提案してきた[10]。しかし、文献[10]では小規模な MANET を用いた実証実験しか行っており、詳細な検討がなされていない。

本稿では、我々が提案してきた手法を、シミュレーションを用いて評価する。また、Linux カーネル 2.6 上に実装した提案手法を用いて、実環境を用いた実験を行い、AODV (Ad hoc On-Demand Distance Vector Routing)[3]、および改良手法を適用しない ARH との比較結果を示す。

2. ARH

ARH は、強化学習を応用したルーティングアルゴリズムであり、中継端末を確率的に選択し、固定的な経路を確立しない。各端末は、パケットを受信した際に受信したパケットと逆向きの経路が使用されるように、中継端末の選択確率を学習していく。

2.1 経路選択

ARH では、端末 y がパケットを送信する場合、中継端末を、自身が持つルーティングテーブルに記録された確率値 $P_y(d, z)$ (y が、宛先が d であるパケットの中継端末として近隣端末 z を選択する確率)にしたがって、近隣端末から確率的に選択する。

2.2 経路履歴情報

ARH では、端末が送信するパケットに、送信した端末の ID とその端末内部での遅延時間からなる、経路履歴情報を付加する(図 2)。経路履歴情報は、中継端末がパケットを中継する際に更新される。

2.3 経路の学習

各端末は、パケットを受信(中継)した際に、付加された経路履歴情報を用いて経路の学習を行う。学習は、経路履歴情報に記録された端末(図 2 の x, n_2, n_1)を宛先としたルーティングテーブルのエントリ全てに対して、式(1)、式(2)を用いて行う。

$$\Delta p = \gamma^h \cdot \frac{k}{f(t_{d'})} \quad (0 < \gamma < 1, k > 0) \quad (1)$$

$$P_y(d', z) = \begin{cases} \frac{P_y(d', z) + \Delta p}{1 + \Delta p} & (z = x) \\ \frac{P_y(d', z)}{1 + \Delta p} & (z \in \text{neighbor of } y, z \neq x) \end{cases} \quad (2)$$

ここで、 k は学習率、 γ は割引率、 $f(t)$ は単調非減少関数^(注1)、 d' は受信したパケットの経路履歴情報に記録された端末、 x は受信したパケットの直前の中継端末、 $t_{d'}$ は d' から x までの所要時間、 h は d' から x までのホップ数である。

式(1)は確率の変化量を表し、送信元端末からの所要時間 $t_{d'}$ 、ホップ数 h が小さいほど大きくなる。式(2)で d' 宛のパケットの中継端末として x を選択する確率を Δp だけ増加させた後、確率を正規化することにより、受信したパケットと逆向きの経路を学習する。

3. MANET に適応させるための改良方式

本章では、我々が提案してきた改良方式について述べる。

3.1 Hello パケットの相乗り

一般的に、MANET におけるルーティングアルゴリズムでは、近隣端末の管理を行うために、各端末は Hello パケットを定期的にブロードキャストする[3]、[4]。しかし、ある端末が送信したデータパケットは、近隣端末が傍受でき、Hello パケットと同様に近隣端末の管理に利用できるため、Hello パケットの送信は冗長である。

改良方式では、データパケットに Hello パケットを相乗りにさせる手法を用いる。この手法では、各端末は、近隣端末が送信するデータパケットを傍受し、Hello パケットと同等に処理することで、一定期間パケットを送信していない場合にのみ Hello パケットを送信すれば良いため、制御メッセージの削減が期待できる。

3.2 リンクの安定性確認

改良方式では、3.1 節で説明した Hello パケットの相乗りに加えて、Hello パケットに対して到達通知パケットを返し、Hello パケットが届いた端末を管理する手法を用いる。この手法では、各近隣端末から返送されてくる到達通知パケットから、一定割合以上の Hello パケットが届いているかどうかを調べることで、各近隣端末が自身の安定した無線通信範囲内に存在しているかどうかを判定することができる。したがって、Hello パケットの到達率を調べることで、グレーゾーンに存在する端末にパケットを中継しないことを実現としている。

しかし、Hello パケットを受信するたびに到達通知パケットを返すと、制御メッセージ数が爆発的に増加する。そのため、Hello パケット同様、到達通知パケットをデー

(注1): 本稿では、 $f(t_{d'}) = t_{d'}$ とし、 Δp は $t_{d'}$ に反比例する。

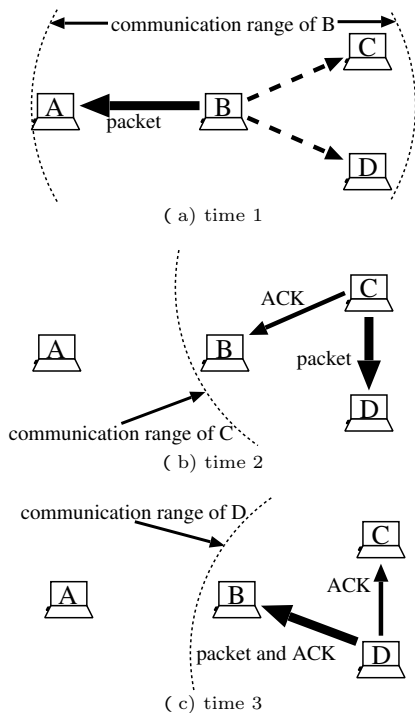


図 3 改良方式でのパケットの交換
Fig. 3 Exchanging packet of proposal.

タパケットに相乗りさせている。このとき、各端末は近隣端末が送信するパケットを傍受することができるので、データパケットの中継先端末とは異なる端末に対する到達通知パケットも相乗りさせることが可能である。このように、到達通知をパケットに相乗りさせることで、制御メッセージ数の増加を抑制している。

図 3 にリンクの安定性確認手法を用いたパケット交換方式を示す。この例では、time 1 で B が A にデータパケットを送信し、time 2 で C が D に、time 3 で D が B にデータパケットを送信している。この 3 個のデータパケットには、それぞれ Hello パケットが相乗りしている。また、time 2 で送信されるデータパケットには time 1 で送信されたパケットに対する到達通知パケット (ACK) が、time 3 で送信されるデータパケットには time 1 および time 2 で送信されたパケットに対する到達通知パケットが相乗りしている。

3.3 経路の学習

従来の ARH では、パケットを受信した時点で経路の学習を行っていた。例えば、端末 A が図 4 (a) に示す経路で転送されたパケットを受信した場合、端末 A は図 4 (b) の破線の経路を学習する。したがって、パケットの転送経路に含まれない端末 (図 4 の端末 D) は、他の端末までの経路を学習することができなかった。しかし、パケットを送信・中継する端末の近隣端末は、送信されたパケットを傍受することができる。

改良方式では、傍受したパケットも受信したパケット

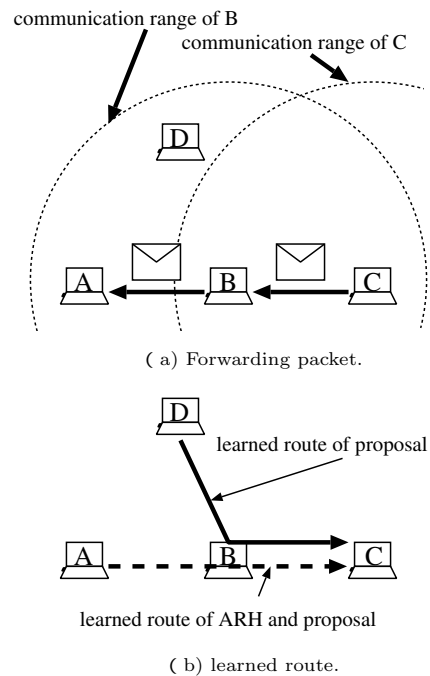


図 4 経路の学習
Fig. 4 Route learning.

と同様に経路の学習に用いる (端末 D が図 4 (b) の実線の経路を学習する) ことで、パケットを中継・受信した端末だけでなく、パケットを送信・中継した端末の近隣端末にも他の端末までの経路を学習させ、学習効率を高めている。

3.4 再 送

改良方式では、3.2 節で説明した到達通知を ACK として用い、パケットの再送を行っている。さらに、再送するときに、以下の処理を行っている。

- (1) 送信に失敗したリンクに対して、負の学習を行う。
- (2) 再送パケットの中継端末を再選択する。

これらの処理は、パケットが正しく送信できず、再送を行う場合、再送前に用いたリンクは信頼できない可能性があるため、再送前に用いたリンクの使用を抑制するために行う。

一般に、MAC 層も再送機構を有しており、到達通知を用いた再送は冗長となる。一方で、MAC 層の再送機構は、衝突や外乱による送信失敗を想定しており、物理リンクの切断は想定していない。しかし、MANET では、頻繁に物理リンクの切断が発生するため、同一端末に対して再送を何度行っても、宛先端末にパケットが届かない場合がある。

そこで、本稿では、クロスレイヤアプローチを用い、MAC 層の再送機構に改良を施すことで、本節の手法を実装した。

表 1 実験環境

Table 1 Experimental environment.

シミュレーションエリア	1000m × 1000m
ノード数	100 台
モビリティ	Random Way Point
最低移動速度	15m/s
最大移動速度	15m/s
停止時間	なし
MAC プロトコル	IEEE 802.11
伝送レート	11Mbps
受信 SNR 閾値	10dB
送信電力	5dBm

表 2 実験で用いたパラメータ

Table 2 Parameters of Experiment.

k	2.0
γ	0.5
選択確率上限値	0.9
最大経路履歴付加数	10 件
Hello パケットの最低送信間隔	0.5s
Hello パケットの有効期間	1.5s
パケット傍受後、到達通知を送信する期限	0.5s
パケット送信後、到達通知を監視する期間	1.5s
1 パケットに相乗りさせる最大到達通知数	5 個
近隣端末を有効とみなす Hello パケット到達率	60%
負の学習を行うときの Δp	0.1

4. シミュレーション評価

提案手法を glomosim [11] に実装し、シミュレーション評価を行い、提案手法と、改良前の ARH、ARH に 3.1 節の手法のみを追加した手法（手法 1）、3.1 節と 3.2 節の手法を追加した手法（手法 2）、3.1 節～3.3 節の手法を追加した手法（手法 3）、AODV を比較した。実験環境を表 1^(注2)に、ARH、手法 1～3、提案手法で用いたパラメータを表 2 に示す。端末の内部遅延時間は、シミュレーションでは計算困難であるため、送信待ち行列長で代用した。また、グレーゾーンの存在をシミュレートする目的で、SNR の閾値を用いてパケットが受信可能かどうかを判定する部分を、式 (3) を用いて確率的に決定することとした。

$$p(snr, snr_{thr}) = \frac{1}{1 + \exp\{\alpha(snr_{thr} - snr)\}} \quad (3)$$

p は受信成功率、 snr は受信信号の SNR、 snr_{thr} は SNR の受信閾値である。また、 α は不安定度を表すパラメータで、 $\alpha = \infty$ では閾値判定となり、 α が小さくなるほどグレーゾーンが大きくなる^(注3)。

(注2): この時、端末の通信半径は約 200m となった。

(注3): 本稿では、 $\alpha = 10$ として実験を行った。

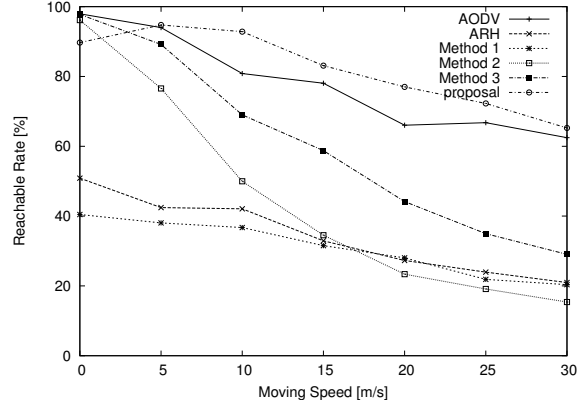


図 5 実験 1 の結果

Fig. 5 Result of experiment 1.

シミュレーション時間は 70 秒間とし、シミュレーション開始 5 秒後から 60 秒間、1 個の端末が別の端末に対して UDP パケットを用いて RTT の測定を行う。測定は 546 バイトのパケットを用い、秒間 30 回行った（したがって、通信帯域は往復で 256kbps となる）。

4.1 実験 1

Random Way Point における移動速度を変化させ、AODV、ARH、手法 1～3、提案手法における測定パケットの到達率を測定した。

試行を 100 回行い、中央値を求めた結果を図 5 に示す。図 5 から、手法 1 は ARH と比較して到達率が低下していることがわかる。これは、Hello パケットの相乗りにより、Hello パケット受信間隔が短くなり、リンクの切断検知が遅れるためである。

手法 2 では、移動速度が小さな場合は到達率が向上した。このことから、移動速度が小さな場合には到達通知が有効であるといえる。一方で、移動速度が大きな場合には、物理リンクが切断されても、古い到達通知が残ってしまうために、リンク切断検知に遅れが生じてしまい、到達率が低下する。

さらに、IEEE802.11 で用いられる RTS/CTS 機構により、実際には送信されないパケットが生じ、相乗りしている Hello パケットや到達通知が失われる場合があり、リンク確認精度悪化の原因となった。これは、RTS を一定回数再送しても中継端末から CTS を受け取れない場合、パケットの送信が中止されるために起こる。

手法 3 を用いることで、到達率は向上した。このことから、3.3 節の改良を行うことで、宛先までの経路を効率的に発見できるようになり、ネットワークの動的な変化に対する適応力が向上したといえる。

提案手法では、更なる到達率の改善が見られた。したがって、提案手法では、信頼できないリンクの使用を抑制し、リンクの切断が原因で起こる再送においてもパケット

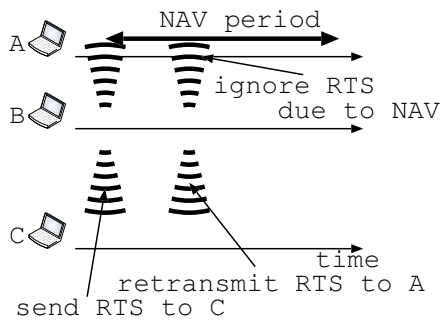


図 6 再送の問題点

Fig. 6 Problem of retransmission.

を正しく転送できるといえる。しかし、端末が停止している状態（移動速度が 0m/s ）では、到達率は低下した。これは、端末が停止している状態では、リンクの切断はほとんど発生せず、再送の原因は衝突が大半を占めるため、負の学習を行うことでルーティングテーブルの破壊を招いてしまうためである。

また、再送時に中継端末を変更することで、図 6 に示す問題を引き起こす場合があった。図 6 では、端末 B が端末 C に RTS を送信している。端末 A はこの RTS を受け取り、NAV 期間を設定する。その後、C に RTS が届かないため、RTS は A 宛に再送される。しかし、RTS を受け取った A は NAV 期間中であり、RTS が無視され、CTS が送信されないため、正常にパケットの交換ができず、問題である。

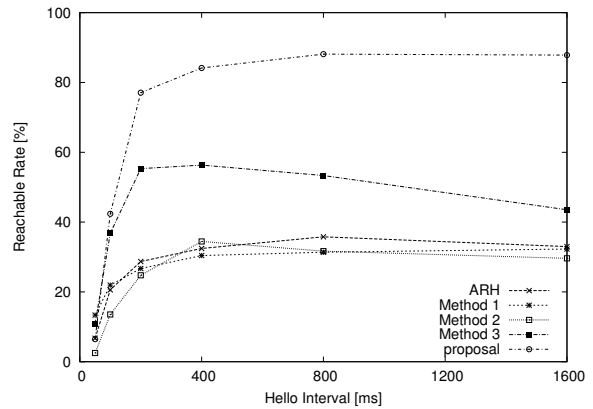
AODV では、固定的な経路を用い、MAC 層から再送失敗の通知を受け取って経路を無効化し、経路の修復を行う。したがって、経路が存在する間はパケットロスほとんど起こらず、パケットロスの多くが経路修復中に発生するため、確率的に経路を選択する ARH と比較して高い到達率となった。

しかし、提案手法を用いることで、端末が移動する場合には、AODV と比較しても高い到達率を達成できることがわかる。

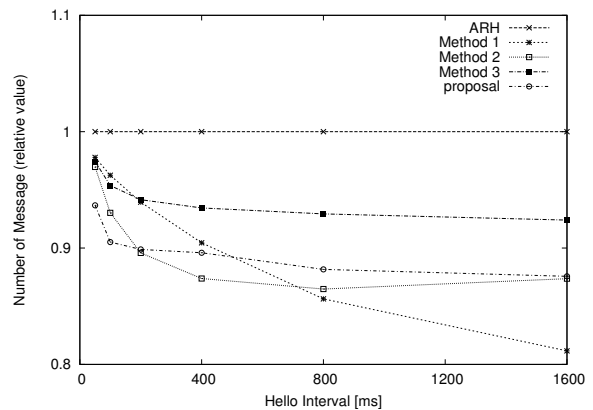
4.2 実験 2

Hello パケットの最低送信間隔を 50ms , 100ms , 200ms , 400ms , 800ms , 1600ms と変化させ、ARH、手法 1~3、提案手法における測定パケットの到達率および制御メッセージ数を測定した。試行は 100 回行い、到達率は中央値を、制御メッセージ数は平均値を求めた。また、Hello パケットの送信間隔の変化にあわせ、Hello パケットの有効期間、到達通知を監視する期間を Hello パケット送信間隔の 3 倍、到達通知の送信期限は Hello メッセージの送信間隔と同一とした。

実験結果を図 7 に示す。図 7 (a) から、Hello パケットの送信間隔を短くすると、到達率が低下することがわかる。これは、制御メッセージが増加し、帯域を消費する



(a) Reachable rate.



(b) Number of message.

図 7 実験 2 の結果

Fig. 7 Result of experiment 2.

ためである。

図 7 から、手法 1 は ARH と比べて制御メッセージ数が削減されたことがわかる。このことから、Hello パケットの相乗りににより、制御メッセージ数を削減できたといえる。削減効果は、Hello パケットの送信間隔を短くすると相乗りでできない Hello パケットが増加するため、Hello パケットの送信間隔が長いほど大きい。一方で、4.1 節と同様の理由により、到達率は低下してしまった。

手法 2 を用いると、Hello パケットの送信間隔が長い場合に、手法 1 と比較して制御メッセージ数は増加した。これは、Hello パケットの送信数が少ない場合には、他のパケットに相乗りでできない到達通知が多くなるためである。

手法 3 では、手法 2 と比較して到達率が向上した。一方で、学習の効率化により、送信されるパケット数が減少し、Hello パケットの相乗りが減少してしまうため、制御メッセージ数は増加した。しかし、Hello パケットと到達確認の相乗りににより、制御メッセージ数の増加を抑えることができ、ARH と比較すると制御メッセージ数は削減されている。

提案手法では、手法 3 と比べて、到達率、制御メッセー

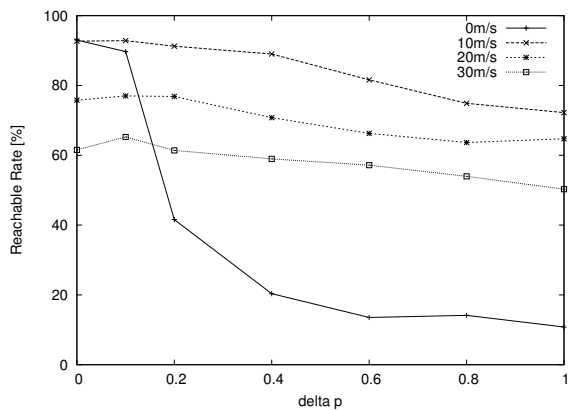


図 8 実験 3 の結果

Fig. 8 Result of experiment 3.

ジ数とともに改善された。これは、手法 3 ではリンクの切断によって再送に失敗するパケットが正しく転送されるようになったためと考えられる。

4.3 実験 3

提案手法を用いて、移動速度が 0m/s, 10m/s, 20m/s, 30m/s の場合に対して、3.4 節で説明した負の学習における Δp を 0, 0.1, 0.2, 0.4, 0.6, 0.8, 1.0 と変化させ、4.1 節と同様に測定パケットの到達率の中央値を測定し、負の学習の効果を検証した。この時、 $\Delta p = 0$ は負の学習を行わずに、再送時に中継先の再選択のみを行うことを意味し、 $\Delta p = 1$ は負の学習により、選択確率を 0 にすることを意味する。

実験結果を図 8 に示す。図 8 から、端末が移動しない状態では、 $\Delta p = 0$ の場合に到達率が最大となることがわかる。これは、4.1 節で説明したように、負の学習を行うことでルーティングテーブルの破壊が発生するためである。また、 $\Delta p = 0$ の場合においても、図 5 の手法 3 と比べて到達率が低下した。このことから、端末が移動しない状態では、中継端末を再選択することで、良い経路でパケットが転送される確率が減少してしまうと考えられる。

端末が移動する状態では、 $\Delta p = 0.1$ が最良となった。このことから、端末が移動する場合には、負の学習により、再送を効率良く行えるといえる。しかし、 Δp を増加させると、ルーティングテーブルの破壊が発生し、到達率は低下した。

5. 実証実験

Linux カーネル 2.6 上に実装した提案手法を用いて、実証実験を行った。本章では、提案手法の実装方法および実証実験について述べる。

5.1 提案手法の実装

ARH は確率的に中継端末を選択するため、決定的なルーティングを前提とした、Linux カーネルの持つルー

ティング機構を用いて ARH を実装することは困難である。そこで、MAC 層と IP 層の間に ARH で必要な処理を行う ARH 層を追加し、ルーティングを ARH 層と IP 層の 2 層にわけて行うことで ARH を実装した (図 9)。ARH 層では、以下の処理を行う。

- 有効な無線リンクの管理
- パケットに付加する経路履歴情報の更新
- 経路の学習
- 送信されるパケットの中継端末選択

ARH 層は、カーネル内で仮想的なネットワークデバイスとして実装した。また、ルーティング処理のうち、以下の処理は IP 層の機能を用いて行う。

- パケットの最終宛先が自身かどうか (ルーティングの必要性) の確認
- パケットを送出するデバイスの決定

このように実装し、各端末の仮想デバイスに同一サブネットの IP アドレスを割り当てることにより、提案手法を用いた通信を行うことができる。

5.1.1 送信処理

送信処理の際は、ARH 層は IP 層からパケットを受け取り、以下の処理を行う。

- (1) パケットの中継端末を選択する。
- (2) パケットに付加する経路履歴情報を更新する。
- (3) 到達通知をパケットに相乗りさせる。
- (4) パケットを MAC 層に渡す。

提案手法では、送信されるパケットを近隣端末が傍受する必要がある。これには、promiscuous モードを用いるべきであるが、本稿では、実装の簡単化のため、パケットを無線ネットワーク上にブロードキャストを用いて送出することで実現した^(注4)。

5.1.2 受信処理

受信処理の際は、ARH 層は MAC 層からパケットを受け取り、以下の処理を行う^(注5)。

- (1) 到達通知送信キューに受信したパケットの到達通知を挿入する。
- (2) 受信したパケットに相乗りしている到達通知を解析し、有効な物理リンクを更新する。
- (3) 受信したパケットに付加された経路履歴を用いて、学習を行う。
- (4) 受信したパケットの中継先が自身であった場合、パケットを IP 層に渡す。自身でなかった場合 (傍受したパケットの場合)、パケットを破棄する。

5.1.3 中継処理

ARH 層では、受信処理を行ったあとに、中継先が自身

(注4): データパケットの中継先は ARH ヘッダに挿入される。

(注5): 5.1.1 項で説明したように、送信するパケットはブロードキャストでネットワークに送出する。したがって、傍受したパケットも受信したパケットと同様に MAC 層から受け取る。

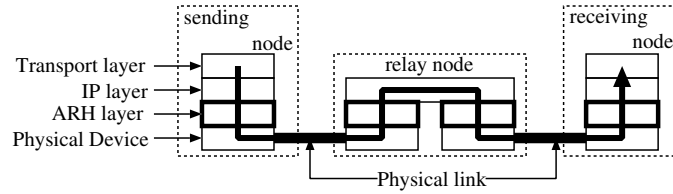


図 9 実装したプロトコルスタック
Fig.9 Implemented protocol stacks.

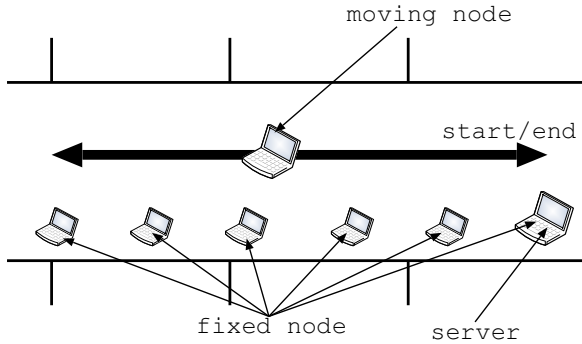


図 10 実験環境
Fig.10 Experimental environment.

であるパケットを全て IP 層に渡し、IP 層でルーティングの必要性を決定し、再び IP 層からパケットを受け取って送信処理を行うことでパケットの中継を実現する（図 9 の relay node）。

このようにすることで、Linux のルーティング機構を利用できるため、ファイアウォールを用いてパケットをフィルタリングするなど、柔軟な設定を行える。

5.1.4 再送処理

本実装では、5.1.1 項、5.1.2 項で述べたように、パケットを無線ネットワークにブロードキャストを用いて送出するため、MAC 層では再送機構が機能しない。また、3.4 節で説明した再送手法を実装するには、MAC 層に改良を施す必要があるが、物理デバイス固有の実装となり、実装は困難である。

そのため、3.4 節の手法は、クロスレイヤアプローチを用いて実装せず、3.2 節で説明した到達通知を用いて実装している。

5.2 実験

幅約 2m の廊下に、15m 間隔で 6 台の固定端末を配置し、その間を 1 台の移動端末が移動しながら IEEE 802.11b 無線ネットワークを用いて通信を行う実験を行った（図 10）。図 10 の環境では、もっとも離れた端末間でも 2~3 ホップで通信できる。実験で用いたパラメータのうち、4. 章とは異なるものを表 3 に示す。

実験では、TCP を用いて 30 フレーム/秒のデータ（128kbps）をサーバから移動端末に送信し、移動端末

表 3 実証実験で用いたパラメータ

Table 3 Parameters of experiment using real environment.

k	250 μ s
最大履歴付加数	5 件
負の学習を行うときの Δp	0.1
最大再送回数	4 回
到達通知のうち、再送に使われるものの送信期限	4ms
再送までの待ち時間	8ms

で各フレームの到着時刻を測定し、AODV（実装として、AODV-UU [12] をデフォルトのパラメータで使用した）、ARH、提案手法を比較した。測定は 4500 フレーム（150 秒間）を行い、測定中に移動端末を 1 往復させた。

実験結果を図 11 に示す。図 11 から、AODV、ARH では通信の断絶が見られることがわかる。AODV では、トポロジが変化し、経路を再構築する際にパケットロスが発生し、通信の断絶が発生したと考えられる。また、ARH では、トポロジが変化した場合に、すぐに新たな経路に切り替えることができず、通信の断絶が発生したと考えられる。一方、提案手法では、そのような通信の断絶は見られず、安定してフレームが届くことがわかる。このことから、提案手法では、AODV、ARH と比較して安定した通信を実現できるといえる。

6. 関連研究

提案手法では、最短経路ではなく、より安定した経路を用いることで安定な通信を行うことを実現している。同様に、安定な経路を用いる研究として、以下の研究がある。

Toh が提案した ABR（Associativity-Based Routing）[5] では、端末の移動を考慮し、端末間リンクの結合性と呼ばれる概念を用いる。ABR では、各端末は一定間隔でビーコンをブロードキャストする。近隣端末（1 ホップで通信可能な端末）の移動が低速、またはない場合や、近隣端末がビーコンを送信する端末と同一の方向に移動する場合、近隣端末が受信するビーコン数は多くなる。ABR は、受信したビーコン数に基づき、持続性が高いと期待できる経路を選択することで、ネットワークの動的な変化に対応する。

坂谷らが提案した OLSR-SS [13] では、OLSR に、Hello

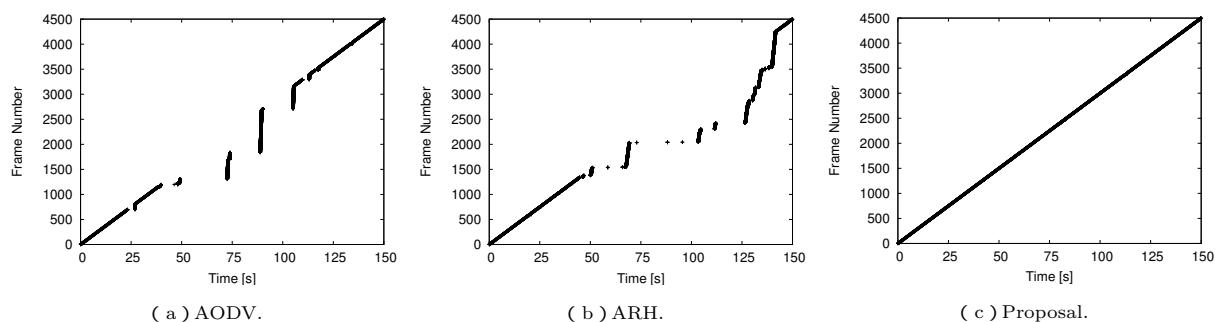


図 11 実証実験の結果

Fig. 11 Result of experiment using real environment.

パケットの信号強度を監視し、信号強度の変動幅を考慮した中継端末候補選択手法を適用している。OLSR-SSでは、各端末が近隣端末からの Hello パケットの受信信号強度を監視し、各端末が自律的に受信信号強度閾値を求め、受信信号強度が弱い端末を中継端末として使用しない。

また、高橋らが提案した ETR (Estimated-TCP-Throughput Maximization based Routing) [14] では、各経路の往復遅延時間および往復パケット喪失率から TCP スループットを推定し、推定値が最良となる経路を用いる。

7. まとめと今後の課題

本稿では、我々が文献 [10] で提案した手法に対して、シミュレーションによる評価を行った。実験結果から、提案手法を用いることで、大規模な MANET においても、制御メッセージ数を削減させながら到達率を向上でき、AODV と比較しても良い到達率が得られることを示した。

さらに、Linux カーネル 2.6 上に実装した提案手法を用い、IEEE 802.11b 無線ネットワークを用いて実験を行い、AODV、改良前の ARH との比較結果を示した。実験の結果、AODV、ARH で見られた通信の断絶が見られなくなり、AODV、ARH と比較して安定した通信を実現できることが確認できた。

しかし、文献 [10] で提案したリンク確認手法では、古い到達通知の影響により、精度の高い確認ができていないことがわかった。今後、より精度の高いリンク確認手法を検討する必要がある。

また、相乗りさせた Hello パケットや到達通知が失われたり、正常なパケット交換を妨げる場合があるなど、MAC プロトコルとの連携が取れていないこともわかった。今後、MAC 層に対して改良を施し、ARH と MAC 層がより緊密な連携を行う必要がある。

さらに、大規模なネットワークでは、学習速度が経路長に依存する問題がある。今後、提案手法に対して、経路長に依存しない強化学習手法を適用することを検討している。

文 献

- [1] IETF MANET Working Group
<http://www.ietf.org/html.charters/manet-charter.html>.
- [2] Johnson, D. B., Maltz, D. A. and Hu, Y.-C.: *The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks*, IETF Internet-Draft (Jul. 2004).
- [3] E.Perkins, C., Belding-Royer, E. M. and R.Das, S.: *Ad hoc On-Demand Distance Vector Routing*, IETF RFC 3561 (Jul. 2003).
- [4] Clausen, T. H. and Jacquet, P.: *Optimized Link State Routing*, IETF RFC 3626 (Oct. 2003).
- [5] Toh, C.-K.: Associativity-Based Routing For Ad-Hoc Mobile Networks, *Wireless Personal Communications*, Vol. 4, No. 2, pp. 103–139 (Dec. 1997).
- [6] Littman, M. and Boyan, J.: A Distributed Reinforcement Learning Scheme for Network Routing, Technical Report CS-93-165, School of Computer Science, Computer Carnegie Mellon University (1993).
- [7] Subramanian, D., Druschel, P. and Chen, J.: Ants and Reinforcement Learning: A Case Study in Routing in Dynamic Networks, *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence*, Vol. 2, pp. 832–839 (1997).
- [8] 斎藤享, 松尾啓志: 動的な環境下における履歴情報を用いた確率的ルーティング, 電子情報通信学会技術研究報告・SST, スペクトル拡散, Vol. SST2001-168, pp. 289–296 (Mar. 2002).
- [9] Lundgren, H., Nordström, E. and Tschudin, C.: Coping with communication gray zones in IEEE 802.11b based ad hoc networks, *5th ACM international workshop on Wireless mobile multimedia (WoWMoM 2002)*, ACM Press, pp. 49–55 (2002).
- [10] 岩田元, 松尾啓志: 確率的ルーティングアルゴリズム ARH を用いた無線 LAN 環境におけるストリーミング配信実験, 情報処理学会研究報告, 2006-MBL-37, pp. 91–96 (2006).
- [11] UCLA Parallel Computing Laboratory: Glomosim.
<http://pcl.cs.ucla.edu/projects/glosim/>.
- [12] Uppsala University: AODV-UU.
<http://core.it.uu.se/AdHoc/>.
- [13] 坂谷聡子, 長谷川淳, 長谷川晃朗, Davis, P., 門脇直人, 小花貞夫: 不安定な無線通信環境における大規模アドホックネットワークの安定化, 情報処理学会論文誌, Vol. 46, No. 12, pp. 2048–2856 (Dec. 2005).
- [14] 高橋ひとみ, 斎藤匡人, 間博人, 戸辺義人, 徳田英幸: MANET における TCP スループット推定による経路選択機構の実環境評価, 情報処理学会論文誌, Vol. 46, No. 12, pp. 2048–2856 (Dec. 2005).