

アドホックネットワークにおける TCP コネクション確立時間を短縮するための低遅延ルーティング手法の提案

山本 貴之[†] 菅野 正嗣^{††} 村田 正幸[†]

[†] 大阪大学 大学院基礎工学研究科 〒 560-8531 豊中市待兼山町 1-3
^{††} 大阪府立看護大学 医療技術短期大学部 〒 583-8555 大阪府羽曳野市はびきの 3-7-30
E-mail: †{tak-ygmt,murata}@ics.es.osaka-u.ac.jp, ††sugano@osaka-hsu.ac.jp

あらまし アドホックネットワーク上のトランスポート層プロトコルとして TCP を用いた場合の性能に関する研究が多くなされている。しかしそのほとんどは、TCP コネクションは永続的なものであると仮定し、定常状態の性能を重視していた。ところが実際には明らかに、多くの TCP コネクションの生存時間は短いものであり、このようなコネクションにおいては、経路制御の遅延時間が無視できないものになると考えられる。本稿では、経路探索時の遅延時間を短縮することにより、このようなコネクションの性能を向上させる新しいルーティングプロトコルを提案する。またシミュレーションを通してその性能を調べ、他のプロトコルと比較することにより、その優位性を示す。

キーワード アドホックネットワーク、トランスポート層プロトコル、ルーティングプロトコル、短生存時間 TCP コネクション、経路探索遅延

A Proposal of Low-latency Routing Protocol for Short-lived TCP Connections in Ad Hoc Networks

Takayuki YAMAMOTO[†], Masashi SUGANO^{††}, and Masayuki MURATA[†]

[†] Graduate School of Engineering Science, Osaka University
1-3 Machikaneyama, Toyonaka, Osaka, 560-8531, Japan
^{††} Osaka Prefecture College of Health Sciences
3-7-30, Habikino, Habikino, Osaka, 583-8555, Japan

E-mail: †{tak-ygmt,murata}@ics.es.osaka-u.ac.jp, ††sugano@osaka-hsu.ac.jp

Abstract Wireless ad hoc networks are expected to be integrated with wired networks and many applications will operate over TCP/IP (Transmission Control Protocol/Internet Protocol), so many studies have been dedicated to improving the performance of TCP over ad hoc networks. However, most of these studies assumed a persistent TCP connection. This is clearly inadequate because many TCP connections are actually short-lived. For such short-lived connections, the routing latency in ad hoc networks is considerable. We describe a new routing protocol that is suitable for a network in which many TCP connections are short-lived. Through simulations, we investigate its performance and show its advantage on some other existing routing protocols.

Key words Ad hoc networks, Transport protocol, Routing protocol, Short-lived TCP connections, Routing latency

1. はじめに

アドホックネットワークは、複数の無線端末が相互に通信を行い、ネットワークの構成情報を交換することで、自律的に構築されるネットワークである。これらの無線端末は他の端末へデータパケットを中継する機能を備えており、広範囲のマルチホップ無線ネットワークを形成することが可能である。アドホックネットワークでは有線のバックボーンネットワークや基地局などを必要としないため、ネットワークの構築や拡張、撤去を簡単に、素早く行うことができるという特徴がある。このような無線通信インフラは、例えば、分散コンピューティング

や災害時ネットワークなどへ適用可能である。また、多くの分散配置されたセンサから情報収集を行うセンサネットワークは、アドホックネットワークの有用な適用例の一つである。

有線ネットワーク上で運用されている多くのアプリケーションは TCP (Transmission Control Protocol) を用いている。そのためアドホックネットワークが有線ネットワークと統合運用される場合、TCP/IP が用いられるのは確実であり、アドホックネットワーク上での TCP の性能に関する研究が多くなされてきた [1-6]。しかしこれらの研究の大部分は、TCP コネクションが永続的なものであると仮定し、定常状態のスループッ

トを性能指標としていた．ところが例えば，一般的な Web サーバにおける個々の Web ドキュメントの平均サイズは 10 Kbytes であると [7] で報告されているように，実際には多くの TCP コネクションの生存時間は短いものである．さらにセンサネットワークでは，各コネクションで配送されるデータは少量であるため，ほとんどのコネクションの生存時間は短いものであるといえる．そのため，前述のような性能指標には問題があると考えられる．そのため本稿では，アドホックネットワークにおける，生存時間の短い TCP コネクションに適した，新しいルーティング手法を提案する．TCP は無線や有線の統合ネットワークにおけるエンド間通信プロトコルであるため，TCP に変更を加えることによる性能改善は不適切であるといえる．LHR は既存の TCP の実装には変更を加えず，ルーティングの改善による性能向上を目標としたプロトコルである．

生存時間の短いコネクションにおいては，隣接端末との経路表交換の際の大きなオーバーヘッドや，初期経路探索のための大きな遅延時間，リンク切断にともなう経路探索のための大きな遅延時間などが性能に影響を及ぼすと考えられる．もし TCP コネクションが永続的なものであると仮定するならば，これらの問題は，たとえモビリティや負荷の高いネットワークにおいても，それほど大きな影響を及ぼさないであろう．しかし短い TCP コネクションに対しては，これらの問題は性能に大きな影響を及ぼすことになる．

本稿では，我々は低遅延複合ルーティング手法 (LHR: Low-latency Hybrid Routing protocol) を提案する．これは上記のような問題を解決するために，TCP コネクションに関わる遅延時間を減らすことを目標としたプロトコルである．特に，TCP コネクション確立時に経路探索を行う場合と，リンク切断に対応するための経路再探索における遅延時間を減らすことを主眼に置いている．

本稿は以下のように構成されている．2. 章では，我々の提案するルーティング手法 LHR の詳細を説明し，どのようにして経路探索の遅延時間を削減するかについて述べる．3. 章では LHR を実装したシミュレーションを行い，他のいくつかのルーティング手法と性能を比較することにより，LHR の優位性を確かめる．最後に 4. 章においてまとめと今後の課題について述べる．

2. 短生存時間 TCP コネクションのための低遅延ルーティング手法

アドホックネットワークにおける経路探索は，未知の経路を探索するための初期経路探索と，端末の移動などにともなうリンク切断に対処するための経路再探索にわけることができる．本稿で提案するルーティング手法 LHR は，これらの経路探索それぞれについて，遅延時間を短く抑えることができる．以下，それらの探索手法の詳細を述べる．

2.1 新規経路探索の遅延削減

アドホックネットワークにおけるルーティング手法は，大別してオンデマンド型とテーブル駆動型の二つの種類がある．DSR [8] のようなオンデマンド型のルーティング手法では，パケット転送要求に応じてブロードキャストによる経路探索を行うため，転送開始までの遅延時間が長くなる傾向がある．一方 DSDV [9, 10] に代表されるテーブル駆動型の手法は，周期的に経路表を隣接端末と交換し，あらかじめ経路を探索しておく手

法である．そのため，パケット転送要求が発生した際に経路が既に探索されている場合には，端末は素早く転送を開始できる．しかしこの手法は，ネットワーク全体の経路情報を収集するのに長時間を要し，経路が未知の送信先に対しては転送を開始できないという問題がある．LHR では，各端末はデータ受信端末 (DR: Data Receiver) に対する経路のみを管理しており，それらを周期的に隣接端末と交換することで更新する．DR に対する有効な経路を保存していない場合は，端末はオンデマンド型の経路探索を行う．このため，既知の経路は経路表から素早く経路を選択することができ，未知の経路については能動的に経路探索を行うことができる．

データ送信元端末は，送信先への有効な経路が未知の場合は送信先端末を宛先とする Route Request (RREQ) パケットをブロードキャストし，オンデマンドに経路探索を行う．RREQ パケットを受け取った端末は，RREQ 送信元端末へ戻る経路を保存し，その端末自身が RREQ パケットの宛先ではない場合は，RREQ パケットを再度ブロードキャストして中継を行う．RREQ パケットを受信した端末が，RREQ パケットの宛先として指定された端末であれば，経路を応答するための Route Reply (RREP) パケットをブロードキャストする．RREQ パケットの中継と同様に，RREP パケットを受け取った端末は，RREP 送信元端末，つまり DR までの経路を保存し，RREP パケットをブロードキャストで中継する．これらの二つのパケットには送信元によってシーケンス番号がつけられており，経路更新の際に，その経路の新しさを示すようになっている．これについては後述する．このようにして，これらの二つのパケットを受け取った全ての端末は，DR までの経路を記録することができる．この経路探索で得られた経路は，周期的に隣接端末と交換し更新を行うテーブル駆動型の経路探索によって更新される．一般にテーブル駆動型経路探索を行う端末は，リンク切断が起こった際，それを回復するためには隣接端末から得られる経路更新を待たなければならず，遅延の増加により性能に大きな影響を受けることになる．この遅延時間を減少させるために，LHR は次節で述べるような経路再探索手法を適用している．

2.2 経路再探索の遅延削減

LHR の特徴は，端末の移動や無線環境の変化に起因するリンク切断を回復するための手法にある．このように切断されたリンクを回復し，経路を再探索するための手法は，以下に示すようなものがある．

(1) 経路表を周期的に隣接端末と交換することによって，隣接端末の持つ新しい経路情報を得る．問題は，新しい経路が必要になった際に，その要求に応じて経路探索を行えないことである．この方法は，テーブル駆動型のルーティング手法が用いているものである [9, 10] ．

(2) 経路が失われたことを示すメッセージが，TCP 送信元端末まで送られる．このパケットを受け取った送信元端末は，再度経路探索を開始する．オンデマンド型手法である DSR はこの方法を用いている．この方法ではエンドホスト間で経路が再探索されるため，新しく得られる経路が短く，特に長時間続くコネクションに対して効果的である．しかし短時間コネクションに対しては，リンクの回復まで時間がかかってしまうことが問題になる [8] ．

(3) リンク切断を検出した端末から，経路探索パケットを

表 1 図 1 の端末 0 から端末 6 への経路表の例

Table 1 Example of the Route Table of Node 0 in the Figure 1

	隣接端末	ホップ数	シーケンス番号
route 1	4	2	1
route 2	3	3	1
route 3	2	3	1

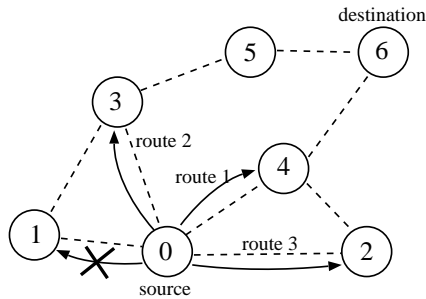


図 1 複数経路エントリ例

Fig. 1 Multiple Routes' Entries

再度ブロードキャストする．この手法では，新しく得られる経路が手法 2 よりも長くなってしまふことがあるが，接続時間が短い場合は，大きな問題とはならないと考えられる．

(4) あらかじめ，宛先端末それぞれに対する複数の経路を管理しておくことで，ある経路が使用不可能になった場合，すばやく別経路を試せるようにする．ただし，代理経路が常に存在するとは限らない．

LHR では我々は，手法 3 と 4 を統合して使用した．リンク切断を検出した端末は，まず手法 4 による再送を試みる．もし有効な別経路が無い場合は，端末は手法 3 を用いて，できるだけ短時間で経路を再構成する．以下ではこの手法について詳しく説明を行う．

2.1 節で述べたように，RREQ パケットを受信した端末は RREP パケットをブロードキャストする．ある端末が同一の RREP パケットを複数の隣接端末から受け取る場合があるが，それは DR までの経路が複数あることを示している．端末はリンク切断に備え，これらの経路を保存する．ある端末が，その隣接端末から RREP パケットを通して経路情報を受け取ったとき，その端末は RREP パケットの送信元に対して管理している経路の更新作業を行う．この時，管理している経路と RREP のシーケンス番号を比較する．もし RREP のシーケンス番号のほうが大きければ，その端末は古い経路を全て削除し，新しい経路を保存する．もしそれらが同じ値なら，端末は RREP の示す経路を予備経路として使用できる．RREP のシーケンス番号のほうが小さければ，端末はそれを無視する．

このように各 DR に対して複数の経路を管理すると，経路エントリの数が多くなりすぎることが考えられる．この問題を回避するため，LHR では各 DR に対して，最短ホップ数の経路と，最短+1 ホップの経路のみを保存するという制限を設けている．例えば図 1 では，端末 0 から 6 への最短経路は 2 ホップである．この場合，端末 0 は端末 6 に対しては 2 ホップと 3 ホップの経路のみを管理する．表 1 は端末 0 における経路表の例である．このように端末が管理する経路のホップ数をどのように制限するのが適切なのかを見積もることは難しいが [11] についての結果から，最短+1 ホップまでに制限するのが適切

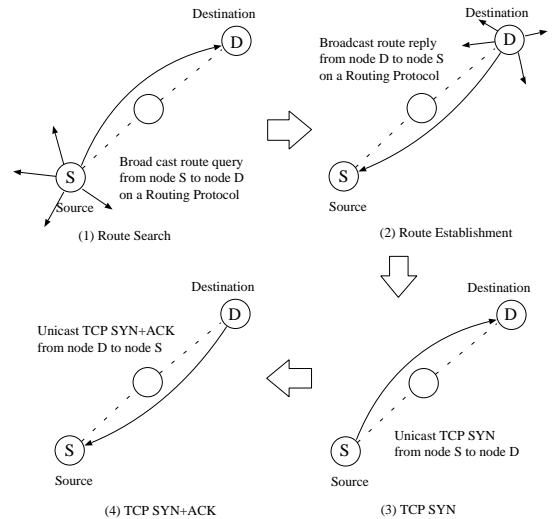


図 2 経路探索と TCP コネクション確立の逐次実行

Fig. 2 Sequential Operation of Route Search and TCP Connection Establishment

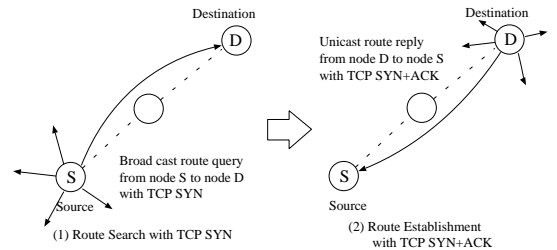


図 3 経路探索と TCP コネクション確立の並列実行

Fig. 3 Concurrent Operation of Route Search and TCP Connection Establishment

であると考えられる．図 1 の端末 0 が端末 6 を宛先とするパケットを持っている場合，端末 0 はまず最初に最短経路である経路 1 による送信を試す．もしそれに失敗した場合，端末 0 は経路 1 を無効にし，2 番目に短い経路を試す．このようにして全ての経路を試しても転送できなかった場合，端末 0 は RREQ パケットを生成し，経路を再探索する．

2.3 経路探索と TCP コネクション確立の統合

TCP コネクションは 3 ウェイハンドシェイクによって確立される [12]．まず TCP 送信端末と受信端末は SYN, SYNACK, ACK の 3 つのパケットの交換を行う．この過程は，実際のコネクション継続時間にかかわらず必要となるため，短時間 TCP コネクションにおいては，コネクション確立に必要な時間がコネクション全体の時間に対して占める割合が大きくなる．

LHR では TCP エンドホスト間の経路が未知である場合，RREQ と RREP の 2 つのパケットが交換される．従って新規経路を使って TCP コネクションを確立する時には，データパケットを送り始めるまでには，図 2 に示されるように，合計 4 つのパケットが交換されなければならない．これは生存時間の短いコネクションにとっては無視できない遅延の原因となる．LHR では図 3 に示されるように，TCP コネクション確立を経路探索と統合することにより，この遅延時間の削減を図っている．SYN パケットを生成したある端末が，その宛先に対する有効な経路を持っていない場合，その端末は SYN パケットを載せた RREQ パケットをブロードキャストする．同様に TCP

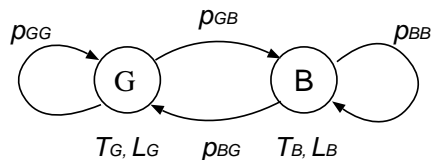


図 4 2 状態遷移エラーモデル

Fig. 4 Two-state Transition Error Model

受信端末から返される RREP パケットは、SYNACK パケットを同時に運ぶ。これによりネットワーク負荷が高まるのは避けられないが、この手法による遅延時間の削減効果による性能向上は大きいと考えられる。またこの方法は TCP には何らの変更を加えないため、既存の TCP の実装をそのまま適用可能である。

3. 性能評価

3.1 シミュレーションモデル

我々はシミュレーションのために、LHR を CMU Monarch Project [13] による無線環境向けの拡張を含んだネットワークシミュレータ ns-2 [14] に実装した。シミュレーションでは LHR との比較を行うルーティングプロトコルとして、ns-2 に実装された DSDV と DSR を利用し、データリンク層には IEEE 802.11 MAC プロトコルを利用した。各端末の無線伝搬距離は 250 メートルであり、パケットバッファサイズは 50 パケットである。

シミュレーションに用いたネットワークモデルは、50 台の端末が 500 × 2000 m の範囲にランダムに配置されている。端末の移動はランダムウェイポイントモデル [15] に基づいている。これは、各端末がランダムに目的地と速度を決定して移動し、目的地に到着後、再度ランダムに次の目的地と速度を決定して移動するというモデルである。端末移動の影響を調べるため、我々は以下に示すような 3 つの移動パターンを使用した。これらは、端末が動かないネットワーク、徒歩や自転車などによる移動が伴うネットワーク、自動車などの高速移動が伴うネットワークを想定している。

- 最高速度: 0 (m/sec)
- 最高速度: 5 (m/sec), 平均速度: 2.5 (m/sec), 停止時間: 0 (sec)
- 最高速度: 20 (m/sec), 平均速度: 10 (m/sec), 停止時間: 0 (sec)

無線回線で発生するエラーは、図 4 のような、ギルバートモデル [16] として知られている 2 状態遷移エラーモデルを用いてモデル化した。この図で状態 G はパケットが失われぬ状態、状態 B は必ずパケットが失われる状態である。平均のパケットエラー率 (PER) は [17] の計算式に基づき、0% と 1% の 2 種類を考えた。この時、状態 G と状態 B の滞在時間は、TCP データパケット 1 つを転送するのに必要な時間 (約 5.84 msec) と定義し、これを 1 スロットとした。PER を 1% とするとき、エラーが発生しない状態 G に滞在する平均時間を 20,000 スロット、エラー状態 B の平均滞在時間は 200 スロットとした。

シミュレーションで作成したネットワークモデルは、多くの生存時間の短い TCP コネクションが発生する、データ収集のためのネットワークである。各コネクションで転送するデータ量は一律に最大 10 パケットまでとし、各 TCP パケットのサイ

ズは 1460 bytes とした。シミュレーション時間は 210 秒であり、最初の 100 秒間は、DSDV が経路表を作成するのに使用されるものとし、TCP コネクション確立は 100 秒から 200 秒までの間に開始されるものとした。コネクション確立の開始要求はポアソン到着過程に従うものとし、ネットワーク全体における平均到着率は毎秒 5 本とした。センサネットワークのようなデータ収集ネットワークをシミュレートするため、全てのコネクションは、あるデータ収集端末 1 台を宛先とし、それ以外の端末から張られるものとした。

我々は性能指標として TCP コネクション確立遅延時間を調べた。これは、TCP SYN パケットの発生から、TCP 送信端末における SYNACK パケットの受信までの時間である。また TCP データ転送時間も同時に調べた。これは SYN パケット発生から、最後の ACK パケット受信までの時間、つまり、データ送信の開始から終了までの時間である。

3.2 シミュレーション結果

図 5 から 10 は、横軸に示された時間内に確立できたコネクションの割合の累積度数分布を示している。図 5 が示すように、エラーも端末移動もないネットワークにおいては、全てのプロトコル (LHR, DSR, DSDV) がほとんどの TCP コネクションを素早く確立できている。特に LHR は、全ての端末が DR に対する経路を管理することと、TCP コネクションを経路探索プロセスと統合できることから、3 つのなかで最もよい結果を示している。このネットワークモデルではリンク切断が起こらないため、DSDV のようなテーブル駆動型プロトコルも、素早いコネクション確立が可能である。しかし図 6 や 7 が示すように、リンク切断が多くなってくると、DSDV では素早いコネクション確立ができなくなってくる。これは、テーブル駆動型プロトコルは、端末が移動するネットワークにおける頻繁なトポロジの変更に対応できないことが原因である。DSR のようにオンデマンド経路探索が可能なプロトコルは、DSDV よりも短時間で多くのコネクションを確立できるが、LHR はそれを上回っていることがわかる。

図 8 から 10 のように PER の値を 1% に設定した場合、オンデマンド経路探索機能を持ったプロトコルと、持っていないプロトコルの間の性能差はさらに大きくなっている。3.1 節で述べたように、エラー発生モデルのエラー状態平均滞在時間は 200 スロットに設定されており、バーストエラーをシミュレートしている。DSDV では、端末はこのような短いバーストエラーに対応して、別経路を探索することができないため、リンクが回復するまでパケットをバッファに保存することになる。これに対し、オンデマンド手法を備えたプロトコルは、このようなエラーに対しても別経路を探索し、転送を行うことが可能である。特に LHR では、端末は複数の経路を管理できるため、DSR よりも素早く別経路を選択できる。

表 2 はシミュレーション時間内に確立できたコネクションの割合を比較している。前出のグラフが示す最大値よりも、この表の値の方が大きくなっているが、これはこの表が、コネクション確立時にタイムアウトを起こし、SYN パケットの再送の結果コネクションが確立できたという場合も含んでいるからである。シミュレーションでは最初のタイムアウト時間は 6 秒であるため、グラフにはタイムアウト後の結果は含まれていない。DSDV は前述したようにリンク切断に際して、新しい経路を探索するのに時間がかかってしまうため、ノード移動や PER が

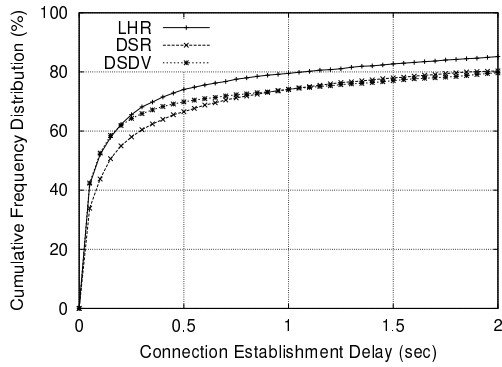


図 5 コネクション確立遅延時間 (最高速度 0 m/sec , PER 0%)
Fig.5 Connection Establishment Delay (Max speed 0 m/sec, PER 0%)

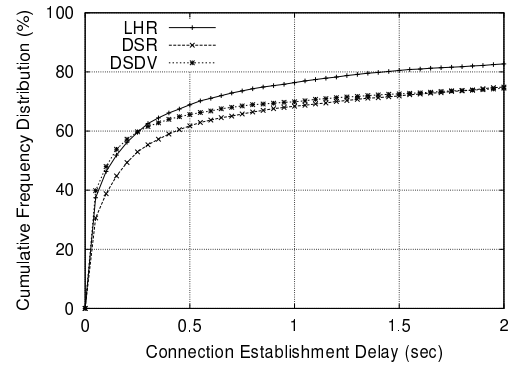


図 8 コネクション確立遅延時間 (最高速度 0 m/sec , PER 1%)
Fig.8 Connection Establishment Delay (Max speed 0 m/sec, PER 1%)

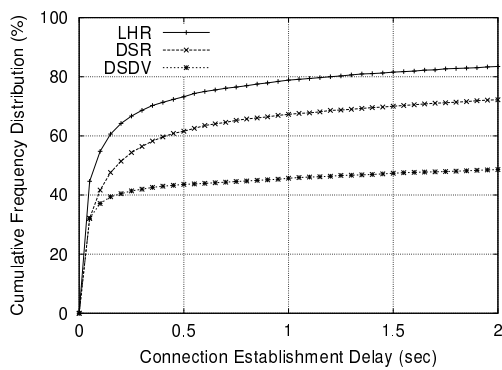


図 6 コネクション確立遅延時間 (最高速度 5 m/sec , PER 0%)
Fig.6 Connection Establishment Delay (Max speed 5 m/sec, PER 0%)

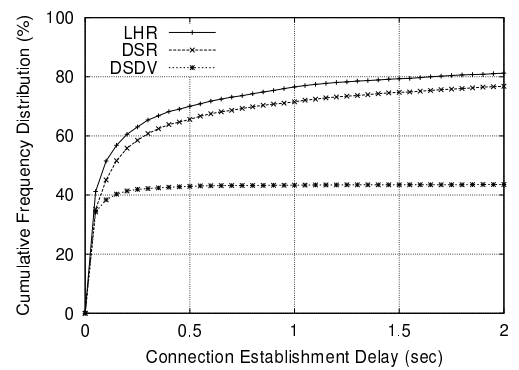


図 9 コネクション確立遅延時間 (最高速度 5 m/sec , PER 1%)
Fig.9 Connection Establishment Delay (Max speed 5 m/sec, PER 1%)

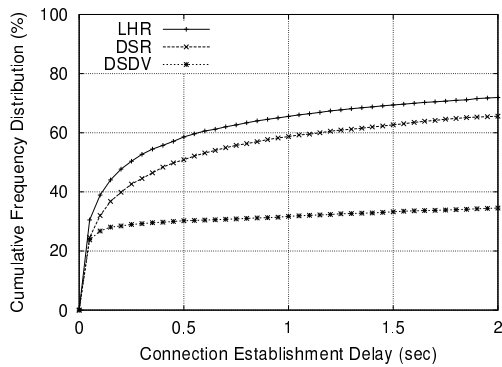


図 7 コネクション確立遅延時間 (最高速度 20 m/sec , PER 0%)
Fig.7 Connection Establishment Delay (Max speed 20 m/sec, PER 0%)

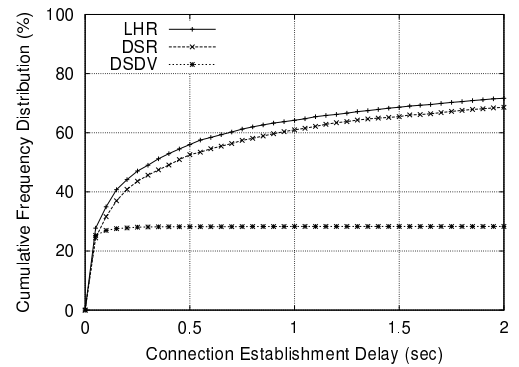


図 10 コネクション確立遅延時間 (最高速度 20 m/sec , PER 1%)
Fig.10 Connection Establishment Delay (Max speed 20 m/sec, PER 1%)

増加すると、コネクション確立成功率が下がることがわかる。
図 11 は横軸に示された時間内に全てのデータの転送が完了したコネクションの割合の累積度数分布のうち、端末最大速度が 20 m/sec で PER が 0% であるものである。図 10 に示されたコネクション確立遅延と同様の傾向を示していることがわかる。本稿では省略しているが、他のパラメータを設定した場合も、データ転送時間の分布はコネクション確立時間の分布と同様の結果を示している。また表 2 は、シミュレーション時間内にデータ転送を完了したコネクションの割合を示している。ここからも、オンデマンド型経路探索機能を持たないプロトコル

は、不安定な無線ネットワーク環境に対応しきれないことがわかる。また遅延時間を短くすることに主眼をおいた LHR は、同一時間内に DSR よりも多くのコネクションのデータ転送を完了できることがわかる。

4. まとめと今後の課題

本稿で我々は、センサネットワークのような、生存時間の短い TCP コネクションが多数発生するような環境に適したルーティングプロトコルを提案した。このようなネットワークでは、コネクション確立やデータ転送の遅延時間を抑えることが重要

表 2 コネクション確立成功率

Table 2 Connection Establishment Success Rate

パラメータ	LHR	DSR	DSDV
0 (m/sec) , PER 0%	98.7 %	97.8 %	98.2 %
5 (m/sec) , PER 0%	97.9 %	98.1 %	77.3 %
20 (m/sec) , PER 0%	99.0 %	97.6 %	63.9 %
0 (m/sec) , PER 1%	98.7 %	97.8 %	98.2 %
5 (m/sec) , PER 1%	97.8 %	98.0 %	69.3 %
20 (m/sec) , PER 1%	98.7 %	97.6 %	54.0 %

表 3 データ転送完了率

Table 3 Completed Data Transmission Rate

Parameters	LHR	DSR	DSDV
0 (m/sec) , PLR 0%	98.3 %	95.9 %	97.2 %
5 (m/sec) , PLR 0%	96.6 %	94.8 %	75.3 %
20 (m/sec) , PLR 0%	96.9 %	94.6 %	62.8 %
0 (m/sec) , PLR 1%	97.7 %	94.3 %	95.6 %
5 (m/sec) , PLR 1%	95.7 %	95.3 %	67.3 %
20 (m/sec) , PLR 1%	97.1 %	95.2 %	53.4 %

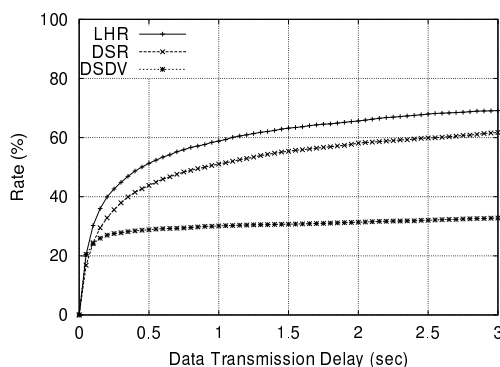


図 11 データ転送時間 (最高速度 20 m/sec, PER 0%)

Fig. 11 Data Transmission Time (Maximum speed 20 m/sec, PER 0%)

となる。我々の提案する LHR (Low-latency Hybrid Routing) プロトコルはオンデマンド型経路探索機能を持ち、経路表の更新はテーブル交換によって行われる。LHR では無線のエラーや端末移動によるリンク切断に対応するため、端末は各宛先端末に対する複数の経路を管理することができ、経路再探索の遅延時間を抑えている。また初期のコネクション確立時間を短縮するため、LHR の経路問い合わせと経路応答のパケットは、TCP コネクション確立のためのパケットを同時に転送している。シミュレーションを通して、LHR は、コネクション確立時間を短縮し、生存時間の短い TCP コネクションの性能を向上できることを確かめた。

今後の課題としては、他の TCP バージョンや SACK, delayed ACK といったオプション、TCP のウィンドウサイズや RTO の計算が、ネットワーク性能に与える影響を調べることが挙げられる。また LHR による経路探索のオーバーヘッドの大きさや、長いコネクションに対する性能、宛先端末が複数あるネットワークや、コネクション確立要求の到着率が大きい場合の性能評価を行うことも挙げられる。

謝 辞

本研究の一部は、文部科学省平成 13 年度科学技術振興調整費による「モバイル環境向 P2P 型情報共有基盤の確立」によって行われている。ここに記して謝意を表す。

文 献

- [1] D. Kim, C.-K. Toh, and Y. Choi, "TCP-BuS : Improving TCP Performance in Wireless Ad Hoc Networks," in *Proceedings of IEEE ICC 2000*, June 2000.
- [2] A. Ahuja, S. Agarwal, J. P. Singh, and R. Shorey, "Performance of TCP over Different Routing Protocols in Mobile Ad-Hoc Networks," in *Proceedings of IEEE VTC 2000*, May 2000.
- [3] K. Chandran, S. Raghunathan, S. Venkatesan, and R. Prakash, "A Feedback Based Scheme For Improving TCP Performance In Ad-Hoc Wireless Networks," in *Proceedings of ICDCS '98*, pp. 472-479, May 1998.
- [4] T. D. Dyer and R. V. Boppana, "A Comparison of TCP Performance over Three Routing Protocols for Mobile Ad Hoc Networks," in *Proceedings of MobiHoc 2001*, October 2001.
- [5] T. Goff, J. Moronski, and D. S. Phatak, "Freeze-TCP - A true end-to-end TCP enhancement mechanism for mobile environments," in *Proceedings of INFOCOM 2000*, March 2000.
- [6] G. Holland and N. H. Vaidya, "Analysis of TCP Performance over Mobile Ad Hoc Networks," in *Proceedings of ACM/IEEE MOBICOM '99*, pp. 219-230, August 1999.
- [7] M. Nabe, M. Murata, and H. Miyahara, "Analysis and Modeling of World Wide Web Traffic for Capacity Dimensioning of Internet Access Lines," *Performance Evaluation*, vol. 34, pp. 249-271, December 1999.
- [8] D. B. Johnson, D. A. Maltz, Y.-C. Hu, and J. G. Jetcheva, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR)," in *IETF Internet Draft*. <http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-06.txt>, November 2001.
- [9] C. E. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance Vector Routing (DSDV) for Mobile Computers," in *Proceedings of ACM SIGCOMM '94*, pp. 234-244, 1994.
- [10] C. E. Perkins, *AD HOC NETWORKING*. Addison-Wesley, 2001.
- [11] T. Yamamoto, M. Sugano, M. Murata, T. Hatauchi, and Y. Hosooka, "Performance Improvement in Ad hoc Wireless Networks with Consideration to Packet Duplication," in *Proceedings of APCC 2001*, pp. 348-351, September 2001.
- [12] M. Naugle, *Illustrated TCP/IP*. Wiley, 1998.
- [13] "The CMU Monarch Project." available at <http://www.monarch.cs.cmu.edu/>.
- [14] "The Network Simulator - ns-2." available at <http://www.isi.edu/nsnam/ns/>.
- [15] J. Broch, D. A. Maltz, D. Johnson, Y.-C. Hu, and J. Jetcheva, "A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols," in *Proceedings of MOBICOM '98*, pp. 85-97, October 1998.
- [16] E. N. Gilbert, "Capacity of a Burst-Noise Channel," *Bell System Technical Journal*, vol. 39, pp. 1253-1265, September 1960.
- [17] J. R. Yee and J. Edward J. Weldon, "Evaluation of the Performance of Error-Correcting Codes on a Gilbert Channel," *IEEE Transactions on Communications*, vol. 43, pp. 2316-2323, August 1995.