

アドホックネットワークにおける複数経路の利用による TCP/UDP 通信の性能向上の検討

森 拓海[†], 高橋 修[‡]

[†] 公立はこだて未来大学大学院システム情報科学研究科

[‡] 公立はこだて未来大学

概要 モバイルアドホックネットワーク(MANET)はモバイル機器をノードとする P2P ネットワークとして研究されている。MANET では IETF より多くのルーティングプロトコルが議論され、AODV, OLSR, TBRPF などが RFC で仕様化されている。それらルーティングプロトコルのなかで、複数の経路を構築し通信を行うもの(マルチパスルーティング)が存在する。マルチパスルーティングのおもな目的は、通信データを分散させること(負荷分散)や経路切断時に瞬時に代替経路に切り替えることによるデータ通信遅延の短縮なのである。上位の通信プロトコルとしては、おもに UDP や TCP といったプロトコルが使用されることを想定しているが、特に TCP に関してはアドホックネットワーク上でスループットの低下が顕著に現れることが知られている。現在までに研究されてきた手法の多くは TCP もしくはルーティングプロトコルとの両方の実装が必要であり、ルーティングプロトコルになんらかの制約が発生する。本稿ではマルチパスルーティングの改良のみで行う TCP パフォーマンスの向上手法を検討・評価した。マルチパスルーティングには複数経路を構築する場合の経路特性やスケジューリング、上位層の通信プロトコルなど様々な要素が存在する。これらを計算機シミュレーションにより評価を行い、TCP パフォーマンスを向上可能なマルチパスルーティング手法が存在するかを検討する。

The Study of the Performance Enhancement of the TCP/UDP Communication by Multipath Routing in Ad-hoc Networks

Takumi Mori[†], Osamu Takahashi[‡]

[†] The graduate school of Future University-Hakodate, Systems Information Science

[‡] Future University-Hakodate

Abstract MANET (Mobile Ad-hoc NETWORKS) is studied as the P2P network which assumes mobile device as the nodes. Many routing protocols are standardized with RFC in MANET by IETF such as AODV, OLSR and TBRPF. There are multipath routing protocols which establish multiple routes in those routing protocols. Purposes of multipath routing are what distribute the communication data, and shorten data communication delay by changing with alternative route when main route was disconnected. UDP and TCP as upper layer communication protocol is known but TCP performance is dropped down significantly in ad-hoc network. Many of method that has been studied by the present need the implementation in TCP or in the both with the routing protocols, and some kind of limitation occurs in the routing protocols of the network layer. In this paper, we implemented and evaluated the improvement TCP performance methods in only improvement of routing protocols. There are various elements in multipath routing such as route characteristic, scheduling and the communication protocol of the upper layer. We evaluated by computer simulation whether there is the multipath routing method that can improve TCP performance.

1. はじめに

モバイルアドホックネットワーク(MANET)上で、TCP 通信を行うことの必要性は現在の通信用途から考えれば妥当なものである。またアドホックネットワーク上のセキュア通信を考える上で、データ損失に最も憂慮しなくてはならないデータ転送方法でもある。これはデータパケットや ACK の損失が通信レートに深く影響するためである。しかし TCP は高エラー、高移動性環境の MANET を想定して設計されておらず、MANET 上で TCP 通信を行う場合は十分なスループットを確保することは難しい。

IETF^[1]で仕様化されている MANET のルーティングプロトコルとしては、AODV (RFC3561)^{[2][3]}, DSR (RFC4728)^[4], OLSR (RFC3626)^[5], TBRPF (RFC3684)^[6]などが存在する。高移動性環境では OLSR のようにあらかじめ通信経路を形成するプロアクティブ型ルーティングでは、頻繁な通信トポロジ変化に対応することが困難であるため不向きである。通信要求時に経路を形成する AODV, DSR のようなリアクティブ型ルーティングプロトコルは移動性に強いが、経路形成が完了するまでの時間が通信遅延となる。また通信経路が切断されると再び経路検索のための Route Request のフラッディングを行うため、ネットワークへの負荷が大きい。この問題の解決法として複数の経路を構築し通信を行う

マルチパスルーティングが存在する。マルチパスルーティングのおもな目的は、通信データを分散させること(負荷分散)や経路切断時に瞬時に代替経路に切り替えることによるデータ通信遅延の短縮なのである^{[7][8]}。またセキュリティ面の応用として、攻撃者を含む経路へのデータ送信リスクの低減^{[9][10]}や、攻撃者またはそれを含む経路を回避するための経路としての利用^{[11][12]}が研究されている。これらのルーティングプロトコルでは上位層の通信プロトコル性能への影響を評価していない。本稿では複数経路を用いた TCP^[13](Transmission Control Protocol) および UDP (User Datagram Protocol)^[14] データ通信の性能改善が可能かを検討する。

2. 関連研究[17]

アドホックネットワーク上で TCP 通信を効率化する研究は多数行われている。本章ではアドホックネットワークにおける TCP パフォーマンス向上の研究例を紹介する。

2.1. 既存の無線通信における TCP 通信の改善例

衛星ネットワークにおける TCP パフォーマンス改善方法のうちアドホックネットワークへ適用可能なものは以下 3 つがある。

(1) TCP ヘッダ圧縮

逆経路で送られる ACK のサイズを減らす技術である。中間ノードが実装する。

(2) ACK フィルタリング

ACK の転送数を減らす技術である。TCP シンクで実装されるため TCP の技術である。

(3) ACK 輻輳制御

受信者に逆経路の輻輳を制御させる技術である。

これらの技術は送信側でバーストラフィックを引き起こし、輻輳ウィンドウサイズの成長を遅くしてしまう。したがって送信側で輻輳を制御することが必要となる。

2.2. アドホックネットワークにおけるTCPパフォーマンス向上手法

(a) ELFN (Explicit Link Failure Notification)

アドホックネットワークとして以下の環境における TCP パフォーマンスの研究がされている。

表 1 TCP 通信の評価に用いた MANET 環境

ノード移動モデル	ランダムウェイポイントモデル
ポーズタイム	0秒
移動速度	[0.9v - 1.1 v] の一様分布 v:移動速度
ルーティングプロトコル	DSR

表 1 の環境で分析した結果ノード移動速度 2m/s~10m/s でスループットが激減し、10m/s~30m/s でわずかにスループットが減少することが示されている。またスループットの特徴として次の 2 点があげられる。

① 特定のパターンで 0 近くのスループット

キャッシュに切断されやすい経路が存在するため頻繁に Route Failure が発生し TCP 転送タイムアウトにより通信できない。

② その他のパターンで高いスループット

送信者と受信者の距離が近く、DSR が有効な経路を維持できる。しかし送信者と受信者の距離が離れると Route Failure が発生し、TCP 転送タイムアウトが発生しそれ以後通信できない。

TCP 輻輳制御の誤作動を防ぐために移動性によって引き起こされるパケット損失の場合は ELFN (Explicit Link Failure Notification) と呼ばれる技術がある。これは TCP とルーティングプロトコルの両方を改良する技術である。ネットワーク層で Link Failure を検出すると、その情報を TCP 層に通知することにより、輻輳以外の理由でパケットを損失したことを検知することが可能となり、輻輳制御の誤作動を防ぐことができる。

(b) RTO 固定法

MAC で Link Failure が発見されるまでの遅延は、TCP がシングルコネクションの場合は独立である。マルチコネクションの場合はノード速度の関数になる。経路計算待ち時間は Link Failure 後の再計算時間である。加えて、送信元まで Link Failure が伝達するまでの時間を考慮する必要があり、ノード移動によりかなりの遅延が発生すると考えられる。そのため RTO(再送タイムアウト)が長くなり、TCP のアイドル時間が増加する。この問題の解決法として AODV,DSR において TCP の RTO を固定することによるスループット改善手法が存在する。しかしこの手法は本来の輻輳制御に悪影響を及ぼす恐れがある。

(c) CWL : Congestion Window Limit

TCP 輻輳ウィンドウサイズは有線ネットワーク上では定数として与えられる。有線ネットワークではネットワークの帯域幅の急激な変化が発生しないためである。アドホックネットワークで

はネットワーク帯域幅は経路を構成するノードにより変化する。そのため TCP 輻輳ウィンドウサイズを状況に応じて制限するための手法として、輻輳ウィンドウサイズ制限(CWL :Congestion Window Limit)が存在する。最適な CWL をマルチホップの BDP (Bandwidth Delay Product)により決定する。TCP 輻輳ウィンドウは BDP より小さくしなければならない。また TCP で用いられている RTT (Round Trip Time)に代わる概念として RTHC (Round Trip Hop Count)を定義している。MAC 層の仕様に関わらず、マルチホップ経路のバイトあたりの BDP が RTHC ×データパケットサイズより大きくならないことが、順経路と逆経路の帯域幅ボトルネックを想定すると証明可能である。この方法を用いて評価した結果、マルチホップ経路が 5 ホップ以上の場合にスループットが激減することがわかっている。

現在研究されているアドホックネットワーク上での TCP パフォーマンス改善手法の多くは TCP ないしはルーティングプロトコルの両方で改良が必要な手法である。また TCP より下位のルーティングプロトコルとの連携が必要な場合はルーティングプロトコルになんらかの制約を設ける必要がある。さらに、これらの手法はアドホックネットワーク上でのネットワーク帯域幅や移動性に着目されたものであり、現実世界で問題となる高エラーによるパケット損失を考慮していない。本稿では高エラー環境の MANET を想定し、ネットワーク層の複数経路を用いたルーティングプロトコルのみの改良により TCP スループットの向上が可能であるかを検討した。

3. 高エラー環境が及ぼす影響

本章では、高エラー環境がアドホックルーティングプロトコルや TCP にどのような影響を及ぼすかを解説する。

3.1. ルーティングプロトコルの影響

多くのルーティングプロトコルは制御パケットやデータパケットがエラーにより頻繁に損失することが考慮されていない。たとえば AODV のような距離ベクトルをメトリックとしたルーティングプロトコルの場合、必ずしも最短経路から形成できるとは限らないため、通信経路が安定しないなどの影響が考えられる。DSR のようなゾースルーティングの場合は、RREQ(Route Request)などの制御パケットにゾースルートと呼ばれる通信経路ノードリストが付属するため、制御パケットが肥大化しやすい。特に RREQ が伝搬する際にホップしたノードをスタックするため、通信ノード同士の距離が長いほど RREQ のサイズも大きくなる。一般的に Bit Error と呼ばれる、ノイズやコリジョンに起因する自然のデータ損失はパケットのサイズが大きいほどその影響を受ける傾向がある。そのため、通信距離が長い場合は AODV よりも経路形成が困難になる。上記以外の理由(制御パケットへの電子署名など)により制御パケットサイズが増加した場合にも経路形成に影響がある。次の実験はその例である。

1000m 四方に 50m 間隔にノードを 100 配置し、リンク間 Byte Error Rate を 0.00025 に設定する。図 1 は AOMDV(Node Disjoint Mode)の RREQ サイズを増加させた場合の通信経路形成実験の結果である。RREQ サイズ増加による経路形成数の変化は少ないが、経路形成時間が確実に増加する。これは、Expanding Ring Search^aによる経路形成時に 1 度の RREQ では RREQ を損失

a Expanding Ring Search
アドホックルーティングプロトコルで経路検索を行う際に、RREQ パケットの TTL を徐々に増加させる方式

し経路が発見できず、数度のExpanding Ring Searchをおこなうためである。

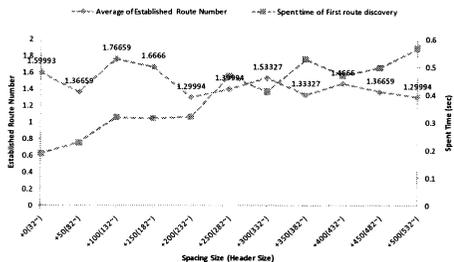


図 1 AOMDV: RREQ サイズ増加による経路形成への影響

次にDSRのマルチパス拡張であるDSR-NDMR (提案方式)を見てみると、同様に経路数の変化はないが経路形成時間が増加していることがわかる (図 2)。またAOMDVと異なり多くの時間が必要となることからわかる。この理由はAOMDVの場合の理由に加え、Expanding Ring Searchを繰り返すことによるソースルートの肥大化により、AOMDVよりRREQの到達率が悪くなるためと考えられる。

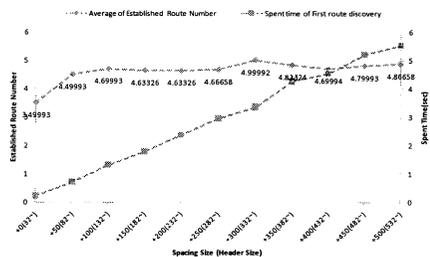


図 2 DSR-NDMR: RREQ サイズ増加による経路形成への影響

以上の理由から高エラー環境のMANETでは、制御パケットのサイズをなるべく抑える必要があるといえる。

3.2. TCPへの影響

実環境で無線通信を行う場合、必ずデータの損失が発生する。有線のネットワークの場合、データの損失の主な原因は輻輳によるものである。TCPは有線のネットワークを対象に設計されているため、パケットの損失は輻輳と解釈される。TCPは輻輳を起こしたと判断した場合、通信レートを下げることによって輻輳の解消を試みる。この機構は輻輳ウィンドウ制御により実現され、輻輳時の対応方法はTCPの方式により異なる。TCPの機能であるスロースタート、輻輳ウィンドウサイズ制御、Fast Retransmit, Fast Recoveryの基本についてはRFC2001^[16]に記述されている。

3.2.1. TCP輻輳ウィンドウサイズ制御

TCPにはネットワークの許容量以上のデータを転送すると輻輳が発生するという問題がある。この問題は1980年初頭に指摘され、年々問題が深刻化する傾向にある。

TCPには輻輳制御機構として、終端ノードによる輻輳ウィンドウ制御がある。TCPの各方式によりその制御方法は異なる。1988年に登場したTCP Tahoeでは、スロースタートと輻輳回避アルゴリズムを採用している。ACKを受け取る毎に1ずつウィンドウサイズを増加させ、スロースタート時には指数的、輻輳回避機構では線型にウィンドウサイズを増加させる特徴がある。また、Fast

Retransmitアルゴリズムの採用し、パケット損失時に再送タイムアウトを待たずに再送パケットを転送することができる。1990年に登場したTCP RenoではFast Recoveryアルゴリズムの採用している。これはTahoeで採用されているFast Retransmitアルゴリズムの欠点である、「Fast Retransmit後のウィンドウサイズの極端な減少(ウィンドウサイズが1になる)」問題を改善している。Fast Retransmitが成功した場合に輻輳の度合いが少ないと判断し、転送速度をパケット喪失検出前の50%に抑えることで実現している。1996年に登場したTCP NewReno^[16]では、さらにFast Recoveryアルゴリズムの修正が行われている。パケットの喪失率がやや大きい場合に対するアルゴリズムの不具合の修正として、パケット再送後さらに重複再送パケットが到着すると、輻輳ウィンドウサイズを一時的に増加させ新しいセグメントを送出する。また新しいACKが到着すると、輻輳ウィンドウサイズをパケット喪失前の1/2にした後、スロースタートには入らず、輻輳回避段階へ移行する。

●ホップ数による輻輳ウィンドウサイズの変化

アドホックネットワーク上では通信の際に中継ノードをホップすることにより輻輳ウィンドウに影響を及ぼす。図 6に注目すると、ホップ数が増加することにスロースタート終了時のウィンドウサイズ(はじめの閾値)が増加する。またスロースタートの終了時間も増加し、その後のウィンドウサイズの減少速度が低下する。ホップ数増加によるRTT(Round Trip Time)増加により、ネットワーク上に漂流するパケットが増加することでウィンドウサイズがホップ数が短い場合よりも高くなる。しかし、RTTの増加によりFast Retransmit時のタイムアウトがバックオフ値により指数増加する。TCP NewRenoのFast Retransmit^[16]の再送タイムアウト(RTO)は以下の式で示される。

$$RTO = \text{平均 RTT} + 4 \times \text{平均偏差}$$

Fast Retransmit時は以下のようにFast Retransmit フェーズ移行直前のRTOにバックオフをかけた値となる。

$$RTO = \text{直前の RTO} \times \text{指数バックオフ}$$

$$\text{指数バックオフ} = 2^n (n=1,2,3,4,...)$$

したがってFast Retransmit時は、0ホップ時のRTTの2倍、4倍、8倍のタイムアウトの増加が発生し、輻輳によるFast Retransmit失敗が判断されるまでの時間も長くなる。

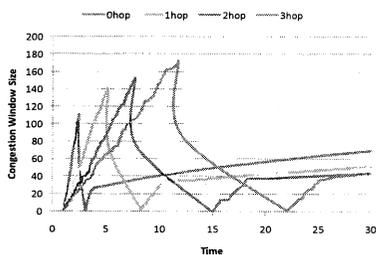


図 3 ホップ数による輻輳ウィンドウサイズの変化

実世界の通信ではこの現象のほか、リンク間にエラーが存在するため、さらに輻輳ウィンドウの制御に影響が出る。図 4の通信トポロジにおいて、ノード0-1, 0-4間にそれぞれ0.00025, 0.0005のByte Error Rateを設定しDSRプロトコルによりTCP(NewReno)通信を行った場合の輻輳ウィンドウサイズの変化は図 5のようになる。エラーにより頻りにデータパケットを損失することで、TCPでは輻輳が頻発しているとみなし、輻輳ウィンドウサイズが上下していることがわかる。このため、通信時の

スループットは安定せず通信レートも増加しない。

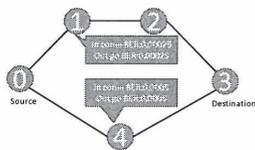


図 4 通信トポロジ

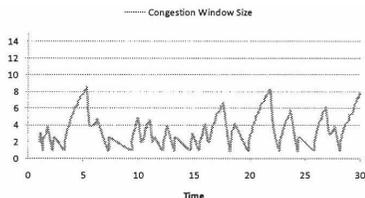


図 5 高エラー環境における輻輳ウィンドウサイズの変化

4. 複数経路通信を用いたTCP性能改善

本章ではノードディスジョイントな経路を用いて効率的なデータ転送を行う際の経路の使用方式を検討する。データ転送プロトコルとして用いられている TCP および UDP を対象とし、ノードディスジョイントな経路を用いた場合の通信方法を考える。

複数経路を用いた通信方法は、(1)経路選択メトリック、(2)経路特性、(3)経路使用法の3つを考慮する必要がある。

4.1. 経路選択メトリック

複数経路ルーティングを用いる場合、保持している経路からどの経路を選択するかは重要な問題である。一般的には以下のような方式が存在する。

- ① 最短ホップ：ホップカウントの短い経路を選択
- ② 最短遅延：最も早く RREP が返信された経路を選択
- ③ 経路品質：パケットの到達/損失率などから最も適切と思われる経路を選択
- ④ 確率：あるメトリックにより確率的に経路を選択
- ⑤ ランダム：保持している経路からランダムに選択

①の方式については、AODVやDSRに用いられている距離ベクトルによる経路選択である。経路を複数構築する場合、最も短い経路が切断されると次に短い経路を選択する。②は最短遅延経路を選択するもので、経路切断後は次に遅延の短い経路を選択する。③はデータの通信品質以外にセキュリティなどのメトリックを考慮する場合もあり、数多くの方式が研究されている。④は③の経路品質などから確率を決定し、ルーティングを行う際には確率により経路を決定する。これは、必ずしも採用したメトリックが最適とは限らないためである。しかし、そのメトリックが通信品質に良い影響を与える傾向がある場合に選択確率を高く設定することで、非最適な経路を多用することを抑止している。⑤の方式は保持している経路からランダムに選択する方式である。この方式は保持している経路を均等に使用することができる。また、使用経路の予測が困難なため、セキュリティにも応用できる。しかし、送信されるパケットの到着順序に不整合が起きやすいため、選択対象の経路の特性をなるべく同一のものにする必要がある。

本稿ではMANETを対象としているため③、④のような計算負荷が大きく、経路維持のための記憶資源を用いる方式は対象外と

する。⑤のランダムな選択方式は、状況による性能変化が大きく計算機シミュレーションによる定量的な評価が難しいため本稿での評価は見送る。また、計算機シミュレータによる評価を実施するため、RREQにブロードキャストジッターが発生する。そのため②の方式は正確なRTT(Round Trip Time)計測が困難であることから採用しない。

また TCP 通信特有の現象として次のようなものがある。

●ホップ数による TCP スループットへの影響

アドホックネットワークにおけるTCPスループット特性は図6のようになる。ホップ数が増加するごとにTCPスループットは $1/N$ (N :ホップ数)で減少していくことがわかる。

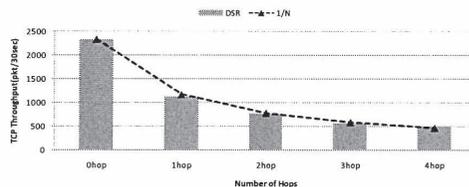


図 6 ホップ数による TCP スループットの変化

ホップ数増加はTCP通信を行う際には大きな差となって表れることが予想される。したがって本稿では①最短ホップ選択方式を採用する。

4.2. 経路特性

複数経路ルーティングでは、保持する経路の特性としてディスジョイント^[10]と呼ばれるものがある。ディスジョイントにはさらにノードディスジョイントとリンクディスジョイントの2つに分けられる(図7)。ノードディスジョイントとは送信元から宛先ノードまでの経路同士が中継ノードを共有していないことである。中継ノードは同一の宛先に対し、1つの経路のみを保持する。リンクディスジョイントはある2つのノード間がノードディスジョイントなことである。

保持経路同士の中継ノードを共有した場合、あるリンク間で切断が発生すると連鎖的に他の経路が切断される。リンクディスジョイントな経路の場合、中継ノードで切断を検知するとその先からノードディスジョイントな経路を用いるが、送信元からみる宛先ノードまでの経路がノードディスジョイントではない可能性がある。これは、送信元が経路を把握する必要があるようなセキュアルーティングの場合、送信元の意図に関わらず攻撃者を含む経路で通信する可能性がありセキュリティ上の脆弱性となる。一方ノードディスジョイントな経路は効率的な経路分散が行え、送信元による経路制御が可能である。本稿では送信元による経路使用方式を用いるため、ノードディスジョイントを対象とする。

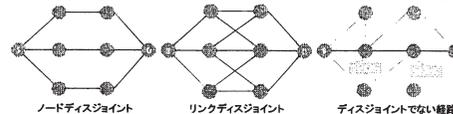


図 7 ノードディスジョイントな経路特性

4.3. 経路使用法

本節では、データ転送プロトコルとしてUDPとTCPを対象とした複数経路の利用方式を考察する。UDPはデータ到達確認を行わず、高い転送速度で通信を行うことができる。一方TCPはACKによりデータ到達を確認する信頼性通信を行うほか、ACKによりデータ転送速度を制御する。低信頼性のUDPデータ

は損失を許可するため、セキュア通信としては想定されない。

TCPは輻輳制御をデータ損失により検出するという特徴から、輻輳以外のデータの損失を避ける必要がある。アドホックネットワークではエラーや移動によりデータの損失が発生するため、これらによる損失を低減させる必要がある。

4.3.1. UDP通信

UDP通信ではデータの到達率が重要となる。複数経路を利用してデータ到達率を向上させるため、次の2点に着目する。

① 経路切断時の瞬時の代替経路への切り替え

② Packet Salvage

①の方式は図8-Clone0のように経路0-1-2-3と経路0-4-3の2経路を保持している場合、最短経路である経路0-4-3で通信を行い、同経路が切断されると、瞬時に経路0-1-2-3に切り替える方式を指す。経路切断時に経路が再検索される間にバッファオーバーフローによるパケットの損失を防ぐことを目的として行う。②の方式は経路0-4-3が切断された場合、MAC Callback^bにより損失したデータパケットを代替経路0-1-2-3に再送することで、経路が切断された際のトリガパケットを損失することを防ぐ。

このほかの方式として、パケットを保持経路に重複して送信する方法が考えられるが、複数経路に重複してパケットを送信することによる帯域圧迫による転送速度低下を考慮し、対象外とした。

4.3.2. TCP通信

TCP通信で扱われるDATAパケットとACKパケットの転送方法として5つの転送方式(Clone Mode)を定義した。

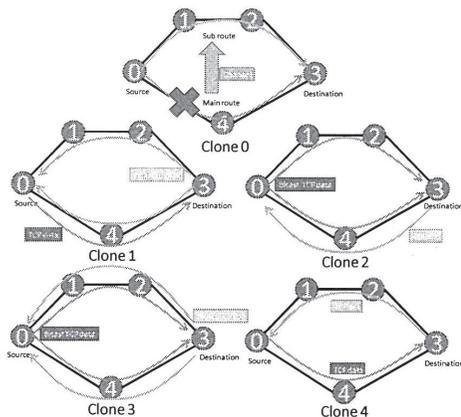


図8 経路使用方式

- **Clone 0** TCP DATA, ACKともに1本の経路で通信を行う。経路切断が発生すると代替経路に切り替える方式である。これはUDP通信で用いる方式と同一のものである(図8-Clone0)。
- **Clone 1** 信頼性通信を目的とするTCP通信では、データパケットの到達率を向上させると共に、到達/損失したパケットを確認するためのACKの到達率を向上させる必要がある。ACKパケットはデータパケットと比較しサイズが小さいためエラーによる影響は少ないがACKの損失はデータの再送を引き起こすため無駄なデータ通信が増加する。TCP DATAは1本の経路

で通信し、ACKを2つの経路によりバイキャストする(図8-Clone1)。

- **Clone 2** TCP DATAを2本の経路によりバイキャストし、ACKを1経路で通信する(図8-Clone2)。エラーの影響を受けやすいTCP DATAの到達率を上げるための方式である。
- **Clone 3** TCP DATA, ACKを2経路にバイキャストする方式である(図8-Clone3)。両パケットの到達率を上げる方式である。
- **Clone 4** TCP DATAとACKを別経路にて通信する方式である(図8-Clone4)。TCP DATAとACKの同経路使用による輻輳を防ぐための方式である。

次章では、これらの通信方式がTCP通信の性能改善につながるかを計算機シミュレーションにより評価する。

5. シミュレーション評価

前章で提案した通信方式の性能評価を計算機シミュレーションを行う。シミュレータにはNS-2^[20]を使用し、ルーティングプロトコルとして、NS-2付属のAODVとそのマルチパス拡張であるAOMDV^[21]、AODV-UU^[22]とそのマルチパス拡張である提案方式AODV-NDMR、そしてDSR-UU^[23]とそのマルチパス拡張である提案方式DSR-NDMRの6プロトコルを評価対象とした。前章で提案した複数経路利用方式を提案方式上にて実装し、AOMDVおよびシングルパスルーティングと比較評価する。

5.1. Byte Error Rateの設定

エラーはパケットのBitに対して発生する。Bit Error RateはMAC層でのパリティチェックがあるためパケットの損失が一定に発生しない。一定のパケット損失の評価を行うためNS-2上でByte Error Rateを設定した。Byte Error Rateによるパケットの損失率は以下で定義される。

$$\text{Packet Error Rate} = 1 - (1 - \text{Byte Error Rate})^n$$

n : Packet Size(Byte)

Byte Error Rate値は4ホップ程度の通信を想定した場合の制御パケットの到達率とデータパケットの損失率より決定した。各リンク間のデータ通信時の到達率はパケットの送信時および受信時にPacket Error Rateの影響を受ける。したがってKホップ先にデータが到達する確率は以下ようになる。

Packet Reachable Rate

$$= \{1 - (\text{Packet Error Rate} \times \text{Packet Error Rate})\}^{(K-1)}$$

K = 0, 1, 2, ...

本稿では4ホップ程度の通信を想定した場合にルーティング制御パケットを損失せず、データパケットを半分程度損失するような高エラー環境を想定した。多くのルーティング制御パケットは100bytesより小さい。表2に制御パケットの大きさを100bytesと想定した場合の4ホップ先までのパケット到達率の理論値を示す。Byte Error Rateを0.0001から0.0005程度まで想定しても制御パケットの到達率は98%以上であるため、経路構築にはほぼ影響のないレベルであることがわかる。

表2 Byte Error Rateとパケット到達率(制御パケット)

DATA SIZE		100 (System packet recommend)					
Byte Error Rate	Packet Error	0hop	1hop	2hops	3hops	4hops	
0.0001	1%	100%	100%	100%	100%	100%	100%
0.0002	2%	100%	100%	100%	100%	100%	98%
0.00025	2%	100%	100%	100%	100%	100%	99%
0.0003	3%	100%	100%	100%	99%	98%	98%
0.0004	4%	100%	100%	99%	98%	98%	98%
0.0005	5%	100%	100%	99%	98%	98%	96%

^b MAC Callback

送信したパケットが宛先(ネクストホップ)に到達しなかった場合に、ルーティングプロトコルに損失したパケットが通知される機能のことを指す。

表 3 Byte Error Rate とパケット到達率 (データパケット)

DATA SIZE	1000 (Data packet recommend)					
Byte Error Rate	Packet Error	0hop	1hop	2hops	3hops	4hops
0.0001	10%	99%	98%	96%	93%	86%
0.0002	18%	97%	94%	87%	77%	59%
0.00025	22%	95%	90%	82%	67%	45%
0.0003	26%	93%	87%	76%	57%	33%
0.0004	33%	89%	79%	63%	40%	16%
0.0005	39%	85%	71%	51%	26%	7%

次にデータパケットとしてサイズを1000bytes程度と想定した場合の4ホップ先までの到達率の理論値を表3に示す。Byte Error Rateを0.00025に設定すると約45%と到達率となり50%程度の損失率を出すことができる。ルーティング制御パケットはこの数値でも99%の到達率があるため、本稿ではByte Error Rateとして0.00025を採用した。

5.2. 通信トポロジ

経路の作成を評価対象のルーティングプロトコル間で公平に評価するために1000m四方に50m間隔で100ノードをグリッド状に配置した(図9)。各ノードはシミュレーション中は停止し、送信元ノードを32、宛先ノードを67として通信を行う。

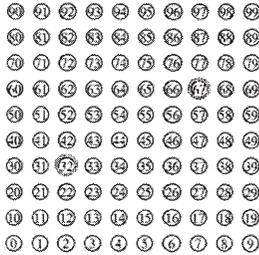


図 9 ノードの配置

5.3. シミュレーションパラメータ

シミュレーションパラメータの詳細を表4に示す。UDP、TCP通信ともにデータパケットのサイズを1000bytesとした。TCP通信については輻輳ウィンドウ制御にFast RetransmitとFast Recoveryを用いるTCP NewRenoとした。

表 4 シミュレーションパラメータの詳細

環境	UDP実験	TCP実験
通信エリアの幅	1000m x 1000m	1000m x 1000m
電波伝送距離	半定約200m	半定約200m
ノード数	100ノード	100ノード
シミュレート時間	31秒	31秒
実行回数	30	30
通信プロトコル	2ノード間の単方向通信。 プロトコル: UDP パケット生成間隔: 0.1秒 データパケットサイズ: 1000Byte パケットの開始時刻から1秒後に通信開始 (30秒間の通信を行う)	2ノード間の単方向通信。 プロトコル: TCP NewReno 輻輳ウィンドウサイズの最大値: 200 データパケットサイズ: 1000Byte パケットの開始時刻から1秒後に通信開始 (30秒間の通信を行う)
ルーティングプロトコル	NOF, DSR, AODV-NDMR, DSR-NDMR, AODV (Node Disjoint node)	NOF, DSR, AODV-NDMR, DSR-NDMR, AODV (Node Disjoint node)
マルチパス通信方法 (Clone Mode)	Clone: 経路切替時、代替経路がある場合には切り替える Packet Salvageあり	Clone 0: 経路切替時、代替経路がある場合には切り替える Clone 1: 経路があればTCP ACKパケットを1経路にバーストする Clone 2: 経路があればTCP DATA ACKともに2経路にバーストする Clone 3: 経路があればTCP DATA ACKともに2経路にバーストする Clone 4: 経路があればTCP DATAとTCP ACKを別経路で通信する Packet Salvageあり
通信経路制御	複数経路ルーティ: プラセは以下の2方式について行う WAIT MODE: 最終2本の経路がなくなるまで通信を中断しない。経路が見つからない場合は再検索する NO WAIT MODE: 最終1本でも見つければ通信開始。経路が2本以上あるときはclone mode有効	複数経路ルーティング場合は以下の2方式について行う WAIT MODE: 最終2本の経路がなくなるまで通信を中断しない。経路が見つからない場合は再検索する NO WAIT MODE: 最終1本でも見つければ通信開始。経路が2本以上あるときはclone mode有効
最大マルチパス本数	10本	10本
送信速度(マックス)	25Mbps	25Mbps
リンク遅延	0.00025	0.00025
ノードの配置	50m間隔のグリッド上	50m間隔のグリッド上
ノードの動き	なし	なし

5.4. 結果と考察

本節では計算機シミュレーションの結果を示す。図10から図14に、複数経路形成において1本目の経路を発見した時点から通信を開始する「NO WAIT MODE」の結果を示す。図15、図16は通信経路を2本確保したのちに通信を開始する「WAIT MODE」の結果を示す。さらにノードディスジョイントではない経路との比較を図17から図19に示す。

5.4.1. UDP通信

はじめにUDP通信の結果を図10に示す。シングルパスルーティングの中でも、Hop By Hop方式のAODV(NS-2標準)が最も到達率が高い。同じAODV系の実装であるAODV-UUは70%程度の到達率であった。ソースルーティングであるDSR-UUは到達率が半分程度と非常に低い。これはソースルートがデータパケットに記載されている分Byte Error Rateによる影響が強く受けたためである。次に複数経路のAODV-NDMR, DSR-NDMRの両方については90%程度の到達率といずれのベースとなったシングルパスルーティングのプロトコルの性能を上回った。これは、WAIT MODE, NO WAIT MODEに関わらず性能向上が見られた。また経路の構築時間についても概ね0.5秒以内に納まっている。しかしAOMDVについては、NO WAIT MODEの場合は提案方式と同等の性能を示すのに対し、WAIT MODEでは経路形成時間が10倍以上である。そのためデータパケットの到達率が40%と非常に悪い。これは、AOMDVのRREQ伝搬制限機能の一つであるAdvertised hop countに起因するものと考えられる。この機能は送信元のネクストホップ(First hop)からのホップカウントのうち最小のものをRREQフォワード時に周囲のノードに通知するもので、無用に長い経路の構築を抑止するために行われる。この機能のため高エラー環境では経路を構築し難くなりデータパケットの到達率が低下したと考えられる。

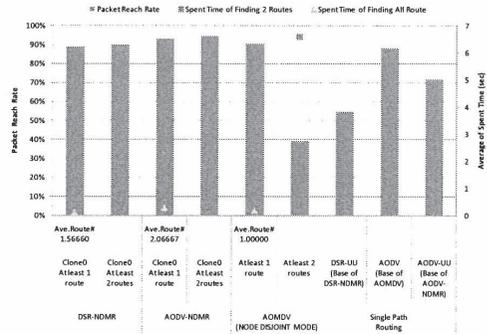


図 10 UDP 実験結果

5.4.2. TCP通信

次にTCP通信の結果をNO WAIT MODEとWAIT MODEに分けて以下に示す。

● NO WAIT MODE

はじめにDSR-NDMRの結果について着目する。図11のスループットの結果を見ると、シングルパスのDSRと比較していずれの通信方式でも性能が改善されている。また図12を見るとAOMDVと比較して経路形成時間、初期構築経路数ともにDSR-NDMRが優れていることがわかる。しかしながら、Clone Modeによる性能差はあまり見られない。これは経路の初期構築数が1.65から1.7程度であることから、2本の経路が確保されず、Clone Modeが機能していないためと考えられる。

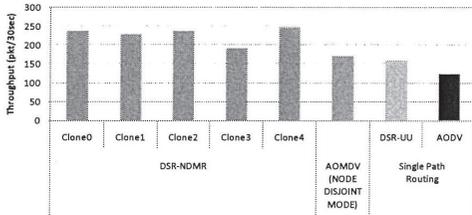


図 11 DSR-NDMR(NO WAIT MODE) : TCP スループット

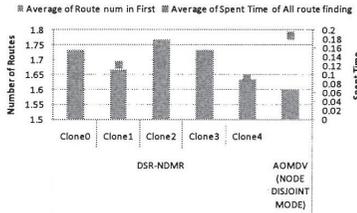


図 12 DSR-NDMR(NO WAIT MODE) : 経路形成数と形成時間
次にAODV-NDMRの結果を図 13, 図 14に示す。Clone0 と Clone1 に関してシングルパスのAODV-UUよりもすぐれたスループットが得られた。AOMDV(Node Disjoint Mode)はほぼAODV-NDMR(Clone0)と同じ動作をするため、同様の結果を得た。尚、AODV-NDMRではルーティングテーブルの特性上、Clone4 は実装できないため実験を実施していない。

Clone Mode による性能差は、Clone 0.1 ではベースとなったAODV-UUを上回るが、Clone 2,3 は逆に性能が低下している。Clone 0 はパケットの重複送信のための帯域圧迫が発生せず、経路切断時に再構築をすることなく素早く代替経路により通信が再開されるためと考えられる。Clone 1 では ACK の到達率が向上し、輻輳ウィンドウ制御の誤作動が抑制されたと考えられる。一方、Clone2,3 はサイズの大きいデータパケットの重複送信によりインターフェースキュー(IFQ)を圧迫し、IFQ オーバフローを誘発していることがスループット低下の一因であった。

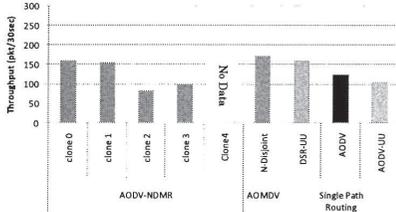


図 13 AODV-NDMR(NO WAIT MODE) : TCP スループット

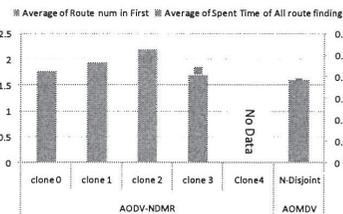


図 14 AODV-NDMR(NO WAIT MODE) : 経路形成数と形成時間

● WAIT MODE

Clone Mode を確実に使用するため、2本の経路を構築した後にはデータ通信を開始する場合の実験を行った。

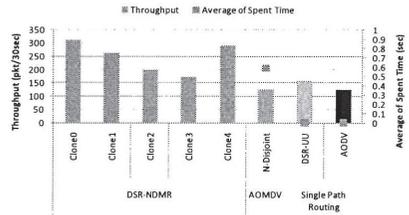


図 15 DSR-NDMR(WAIT MODE)の結果

まず図 15に示した結果のDSR-NDMRについては、1経路形成後に直ちに通信を開始するNO WAIT MODEと比較してスループットが上昇している。2経路形成時間は平均でも0.1秒程度なので、通信に多大な影響を及ぼすことはないと考えられる。一方AOMDVはNO WAIT MODEと比較しスループットが低下し、2経路形成時間も0.6秒程度を要する。UDP通信時と同様にAOMDVは2経路形成のため、再検索を繰り返すためである。Clone Modeに着目するとClone 0の性能が最も良く、ベースのDSR-UUの2倍のスループットを得た。またClone 1についてもNO WAIT MODEと比較して性能向上が見られた。Clone 2,3はNO WAIT MODEと同様にIFQオーバフローによる影響からほぼ変わらない結果となった。Clone 4はClone 1とほぼ同様の性能を有するが、若干のスループット低下がみられる。主経路で通信されるTCP DATAパケットに比べACKパケットはより長い経路で送信される可能性がある。そのためRTT増加によりRTOが増加し再送パケットの送信が遅れるためと考えられる。

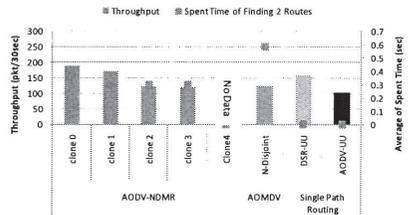


図 16 AODV-NDMR(WAIT MODE)の結果

最後にAODV-NDMRの結果を図 16に示す。こちらもNO WAIT MODEに比べスループットが増加する。2経路形成時間は若干DSR-NDMRより増加するがNO WAIT MODEよりも効果的にClone Modeを使用することが可能であるため、全体的にスループットが底上げされた。

5.4.3. ノードディスジョイントの有効性

本節ではノードディスジョイントではない経路を構築した場合のClone Modeの性能差を確認するための実験結果を示す。

● NO WAIT MODE

ノードディスジョイントの利点を検証するため、DSR-NDMRのNAF値制限が0の場合(ノードディスジョイント)と無制限(中間ノードの重複あり)の場合との性能比較を行った(図 17, 図 18)。NAF値は経路間の共有中間ノード数を表す。NAF値の詳細については文献8を参照してもらいたい。

NO WAIT MODE では経路形成数に大きな差が生じた。NAF=0の場合では初期経路は2本程度であるのに対し、

NAF=Unlimited ではほぼ最大の 10 本の経路を形成する（経路形成時間は増加）。しかし、スループットに注目すると性能は NAF=0 の方が優れているという結果となった。これは、パケット複製に使用する経路がノードディスジョイント内ではない場合、複製パケットが同一のリンク（ノード）を通過することで、リンク切断時に複製パケットも同時に損失してしまうために、パケット複製を行う有効性が低下してしまうためと考えられる。

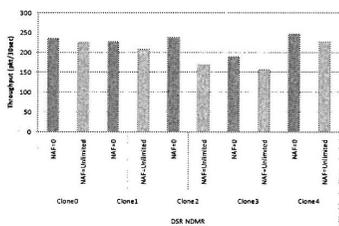


図 17 ノードディスジョイントでない通信との比較 (NO WAIT MODE, スループット)

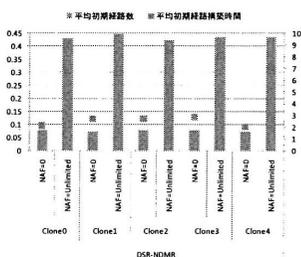


図 18 ノードディスジョイントでない通信との比較 (NO WAIT MODE, 経路数と経路構築時間)

● WAIT MODE

WAIT MODEの結果(図 19)を見ると、2 経路形成時間はNAF値無制限の場合が若干高速である。しかしスループットはNO WAIT MODEと同様に、NAF=0 の場合の方が優れている。

上記の結果から、Clone mode を有効に利用するためには、ノードディスジョイントな経路を用いることが良いといえる。

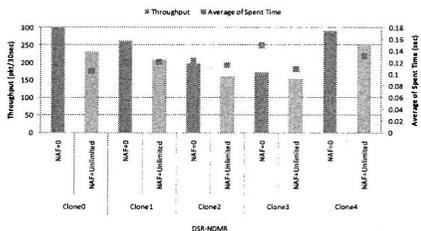


図 19 ノードディスジョイントでない通信との比較(WAIT MODE)

6. 結論と今後の課題

提案方式はベースのシングルパスルーティングプロトコルと比較してスループットの向上が認められた。TCPでは代替経路への切り替えを行うClone 0 が最も性能向上が見られ、WAIT MODEではさらに性能が向上することがわかった。ノードディスジョイントでない経路でClone Modeを用いた場合と比較し

た結果、ノードディスジョイントな経路で効果的なClone Modeを用いたTCPパフォーマンス改善が行えることを確認した。しかしClone Modeが確実に使用できるWAIT MODEにおいて2経路に重複してデータを転送するClone1,2,3 のTCPパフォーマンスがClone0よりも低下する問題がある。これは文献[17]の以下が関連していると思われる。

●マルチパスルーティングにおけるTCPパフォーマンス低下 SMR (Split Multipath Routing)の用いた実験の結果、次の2つの要因によりTCPパフォーマンスが低下する。

- ① RTT が早まることによる転送タイムアウト短縮
- ② 別経路データ転送によるTCPパケットの到着順序不整合
 - ①は平均RTT法が不正確なために発生する。複数経路の切り替えは経路の再検索よりも短いためRTTが短縮され、RTOが短縮される。そのためTCP DATAにわずかな遅延が発生しただけでも損失判定される。②は複数経路間の転送時間が異なる場合にTCP DATAの到達順序が逆転することである。これはTCPではパケット損失として検出されるため、TCP転換を誘発する。

本稿では複数経路形成が確実に進め、経路間のホップ数に差が発生しにくい限定された環境で性能が向上することを示した。実世界ではノードの配置およびノード移動を考慮する必要がある。今後はより実世界に近い環境において、提案方式が効果を発揮するかを確認する必要がある。また、パケットの到達率だけでなく、到着順序の整合性を確保することも課題となる。

7. 参考文献

- [1]. Internet Engineering Task Force (IETF), www.ietf.org/
- [2]. AODV: RFC3561 <http://www.ietf.org/rfc/rfc3561.txt>
- [3]. C. E. Perkins, E. Belding-Royer, and S. R. Das "Ad hoc On-Demand Distance Vector (AODV) Routing",
- [4]. DSR. <http://www.ietf.org/rfc/rfc4728.txt>
- [5]. T. Clausen, P. Jacquet "Optimized Link State Routing Protocol(OLSR)." TBTRPF: RFC3684 <http://www.ietf.org/rfc/rfc3684.txt>
- [6]. <http://www.ietf.org/rfc/rfc3626.txt>, 2003, RFC3626.
- [7]. 谷山健太, 櫻井裕介, 甲藤二郎 「アドホックネットワークにおけるディスジョイント・マルチパスルーティング」, 2004
- [8]. Ash Mohammad Abbas, "Disjoint Multipath Routing for Mobile Ad hoc Networks", 2007, April
- [9]. 森祐海, 森祐海, 高橋修 "アドホックネットワークにおける防衛法の分類と耐攻撃性アドホック・ルーティング・プロトコルアーキテクチャの提案" "MBL41,2007
- [10]. 森祐海, 森祐海, 高橋修 "AAA: Anti Attack Ad-hoc routing protocol の提案と実装・評価" "DICOM02007,2007
- [11]. 森祐海, 森祐海, 高橋修 "アドホックネットワークにおける攻撃法・防衛法の分類とAODVベースセキュアルーティングプロトコルの提案" "MBL41,2007
- [12]. 森祐海, 森祐海, 高橋修 "AODVベースセキュアルーティングプロトコルの提案とその実装・評価" "DICOM02007,2007
- [13]. TCP: Transmission Control Protocol, RFC793
- [14]. UDP: User Datagram Protocol, RFC 768
- [15]. RFC2001, <http://www.ietf.org/rfc/rfc2001.txt>
- [16]. RFC3782, <http://www.ietf.org/rfc/rfc3782.txt>
- [17]. Ahmad Al Hanbali, Eitan Altman, Philippe Nain "A Survey of TCP over Ad Hoc Networks", June 2005
- [18]. TCP NewReno, RFC2582, <http://www.ietf.org/rfc/rfc2582.txt>
- [19]. 上野裕介, 樺中達司, 小野良司, 渡辺尚 「ルートの独立性を考慮したマルチパスルーティングプロトコルの提案とその評価」, 情報処理学会論文誌 Vol45 No12, Dec. 2004
- [20]. NS-2: www.isi.edu/nsnam/ns/
- [21]. M. K. Marina, S. R. Das, "On-demand Multipath Distance Vector Routing in Ad Hoc Networks", IEEE ICNP 2001
- [22]. AODV-UU: <http://core.it.uu.se/core/index.php/AODV-UU>
- [23]. DSR-UU: <http://core.it.uu.se/core/index.php/DSR-UU>
- [24]. 田内雅之, 井手口哲夫, 奥田隆史, 田字軍 「経路の切断を回避するアドホックルーティングプロトコルの提案とその性能評価」, 情報処理学会, Oct 2006
- [25]. Matthew J. Miller, Jungmin So "Improving Fault Tolerance in AODV", Fall 2002
- [26]. Y.Nishibayashi, Y.Sakurai, J.Katto "Multipath Extension of AODV with Enhanced Route Establishment and Proactive Route Management", 2003
- [27]. 森祐海, 横山 信, 高木 剛, 山崎 健一, 高橋 修 「AODVにおけるGhost Attackとその防衛法」, MBL-39, 情報処理学会研究報告 pp.53-58