無線アドホックネットワークにおける
パケット転送制御方式の研究

Study on Packet Transmission Control Scheme for Wireless Ad Hoc Networks

2013年7月

July 2013

早稲田大学大学院国際情報通信研究科
国際情報通信学専攻 情報通信ネットワーク研究II

山本 嶺

目 次

第1章　序論

1.1　研究の背景と目的

FTTH（Fiber to the Home）やADSL（Asymmetric Digital Subscriber Line）などにより，一般家庭でもインターネットに常時接続可能な環境が普及してきている．また，第3世代移動通信や更に高速なデータ通信が可能なLTE（Long Term Evolution）などの携帯電話網の発達により，屋外や移動中でもインターネットを利用可能な環境が普及してきている．そのため,「いつでも，どこでも」ネットワークに接続可能なユビキタス社会に対するユーザの欲求が増大している．従来，屋外や移動中にネットワーク接続を実現するためには携帯通信網が主に利用されていたが，スマートフォンやマルチメディアコンテンツの普及によるトラヒックやユーザ数の増加によって通信帯域が圧迫され，通信品質の低下が問題となっている．そのため，携帯事業者は無線LANアクセスポイント（以下，無線APと称す）にトラヒックをオフロードすることで，通信品質の向上を図っている．このように，これまではあらかじめ設置しておいた無線基地局や無線APなどのインフラストラクチャ（以下，インフラと称す）を利用してネットワーク接続が行われていた．一方，特定のインフラを利用することなく無線端末のみでネットワークを構築し，通信を行うことが可能な技術として無線アドホックネットワークが注目されている．無線アドホックネットワークでは，ネットワークに参加する無線端末にルータとしての役割を与えることにより，端末同士が互いの通信を中継することが可能となり，電波が直接到達しない端末へマルチホップ通信を用いたデータ転送が可能となる．これにより，災害時や山間部などインフラが存在しない場合においても，自律分散的なネットワーク構築が可能となり，局所的な情報配信や外部ネットワークに接続したゲートウェイを用いたネットワーク接続等を実現することができる．

無線アドホックネットワークでは，前述したように端末同士で自由度の高い自律分散的なネットワーク構築が可能である．一方，ネットワークの接続状態や通信状態を管理するための機構が備わっていないことやパケット転送に無線通信を用いること，端末の移動性が高いといった特徴から，有線通信と比較して安定した通信を行うことが困難である．また，無線アドホックネットワークでは，一般的にはMACプロトコルとしてIEEE 802.11を用いることを想定しているため，チャネルや通信帯域に制約があることや通信端末としてスマートフォンやノートPCなどを想定しているため，十分な無線資源，端末資源を確保することは困難であると考えられる．そのため，無線アドホックネットワークでは，主に以下のような問題点が生じる．

1. 端末移動や障害物によるリンク切断

2. ネットワークトポロジーの変化による経路再構築

3. 不安定な通信環境に起因する高いデータ損失率

4. データ損失に伴う TCP 性能の低下

5. 輻輳の発生

6. 安定した通信経路への負荷集中及び公平性の低下

無線アドホックネットワークでは，移動可能な携帯端末などを利用したネットワーク構築が想定されているため，端末の移動によって 1. に示すようにリンク切断が発生する．また，端末間に障害物が存在する場合には，受信信号強度が低下することとなり，リンクで正常な通信を行うことが困難となる．これらのリンク切断は，リンク上での再送の頻発や経路切断の要因となることから，信頼性や通信効率の低下，再送の頻発による通信負荷の増加を招くこととなる．

2. は 1. に関連し，通信環境の変化によって発生したリンク切断により，構築した通信経路が利用できなくなることで発生し，経路再探索のための経路探索パケット送信による通信負荷の増加や一時的にパケット転送が行えなくなることで遅延が増加し，通信効率の低下を引き起こす．また，端末の移動性が高い場合には，通信途中にも随時ネットワークトポロジーが変化するため，端末間の通信距離の変化などによって通信開始時に選択された通信経路が一定時間後には適切な経路とならない場合がある．そのため，端末の移動性が高い環境で一定の通信品質を確保した通信を行うためには，定期的に経路再構築を行う必要がある．

3 は，1. や 2. に関連し，信号が空間を伝搬する無線通信では，有線通信と比較して，安定した搬送路を確保することが困難であることや外的要因によって通信環境が変化しやすいという特徴に起因している．無線 AP などを用いて有線通信の一部を無線化するような使用方法では，端末の移動性が低いことや無線 AP と端末間の受信信号強度の変化幅が小さいことなどから，比較的安定した通信が可能である．一方，無線アドホックネットワークでは，前述したように端末移動などによって通信環境が随時変化するため，安定した通信環境を確保することが困難であり，有線通信や無線 AP を利用した無線通信などと比較してデータ損失の発生確率が高くなる．データ損失が高くなることにより，再送の頻発やリンク切断が発生することとなり，通信効率の低下を引き起こすこととなる．

4. は，伝送途中で損失が発生しやすいという無線アドホックネットワークの特徴に起因し，損失を輻輳の発生と誤認することで発生する．TCP[1] では，確認応答（ACK：Acknowledgement）を用いて送信データの到着確認を行っているが，あて先端末から重複 ACK を受信した場合には，伝送途中で損失が発生したと認識

し，再送を行う．また，TCP では損失を輻輳の発生と認識し，送信側ウィンドウサイズを縮小する制御が行われ，通信効率が低下する．通常，有線通信では輻輳以外の要因で損失が発生する可能性が低いため，輻輳回避に有効な手法としてウィンドウサイズの縮小が用いられているが，輻輳以外の要因によって損失が頻発する無線アドホックネットワークでは，不必要なウィンドウサイズの縮小が行われることとなり，通信効率が著しく低下する．

無線アドホックネットワークでは，端末間のリンク構築に無線接続を用いているため，有線通信と比較して狭帯域となる．そのため，複数のフローが同一リンクへ流入した場合，リンク容量を容易に超過することとなり，5. に示したように，輻輳が発生することとなる．更に，無線アドホックネットワークでの経路構築では，通信環境が優れていて安定した経路を優先的に利用するように経路構築が行われるため，輻輳の他に 6. で述べたような特定の端末への負荷集中が問題となる．

このように，無線アドホックネットワークでの通信は，有線接続やインフラを用いた通信方式と比較して，通信環境の変動が大きいため，インターネットなどで用いられている通信方式をそのまま適応することは困難である．そのため，無線アドホックネットワークで高効率，高信頼な通信を実現するためには，以下に示す 1.〜4. 課題を解決した通信方式が必要となる．

1. リンク及びエンドエンド間での信頼性確保

2. 生存時間が長く，安定した経路の構築

3. 限られた無線資源を有効活用するための送信速度制御

4. 広範囲の通信状態に基づく自律分散的な通信制御

本論文では，これらの課題を解決し，高効率，高信頼な無線アドホックネットワーク向けの通信方式を実現することを目的として議論する．このため，本論文では，データリンク層，トランスポート層，及びクロスレイヤ制御を用いた通信方式に着目し，検討を行う．

1.2 研究の概要

本論文では，無線アドホックネットワークにおける通信効率の向上，及び信頼性確保を行うことを目的とし，無線アドホックネットワークの通信特性に基づいた通信方式の検討を行っている．本論文では，データリンク層での制御，トランスポート層での制御，及びクロスレイヤでの制御についてそれぞれ検討を行い，前述した目的を達成する．

以下，各章ごとに概要を述べる．

第 1 章は序論であり，研究の背景，目的，及び概要について述べている．

第 2 章は，無線アドホックネットワークの概要，通信特性，及びそれらに付随する問題について述べている．無線アドホックネットワークでは，自律分散的に通信制御が行われるため，インフラを用いた通信と比較して，通信効率や信頼性が低下する．そこで無線アドホックネットワークで高効率，高信頼な通信を実現するため，本論文で取り扱う課題について述べている．

第 3 章では，リンクでの信頼性及びエンドエンド間での通信効率向上に関し，データリンク層において経路近傍の端末を用いて自律分散的に再送を行う手法について検討を行っている．通常，無線アドホックネットワークのデータリンク層では，リンク間でのフレーム損失時に自動再送要求（ARQ: Automatic Repeat Request）によって損失フレームの回復が行われる．しかし，再送を行うためには，タイムアウト時間まで待機する必要があることや，利用不能なリンクに対しても繰り返し再送が行われることから，遅延の増加やトラヒックの増加が問題となる．本章では，この問題に対し，無線通信の特徴である通信の同報性を利用し，経路近傍の端末がフレーム送信を漏れ聞くことによって，リンク利用の可否，及びフレーム損失の有無を判断し，自律的に再送制御を行う手法について検討を行っている．これにより，リンク間でのフレーム損失回復時間を短縮することで，エンドエンド間の遅延低減を達成し，通信効率の向上を実現している．更に，リンク間の信頼性を向上させることによって，通信経路の安定性を向上させ，経路再構築に伴う遅延やオーバヘッドの増加を低減している．

第 4 章では，エンドエンド間の信頼性確保，及び無線資源活用のため，トランスポート層において再送及び送信速度制御を行う手法について検討を行っている．無線アドホックネットワークでは，端末移動や障害物などの影響によって，一般的な無線通信と比較して，伝送途中に損失が発生する確率が高くなる．そのため，セグメント損失を輻輳の発生としてウィンドウ制御を行っている TCP Reno などの通信方式では，不要なウィンドウサイズの縮小が行われ，エンドエンド間のスループット低下の要因となる．また，選択的確認応答（SACK: Selective Acknowledgement）[2] を用いていない TCP 通信の場合，ACK を受信したセグメントから損失セグメントまで再送が行われるため，冗長な再送制御となり，ネットワーク負荷が増加する要因となる．この問題に対し，本章では経路上の通信状態を往復遅延時間（RTT: Round Trip Time）によって評価し，ウィンドウ制御を行う手法について検討を行っている．これにより，セグメント損失の有無に関わらず，通信状態に応じた送信速度制御が可能となり，限られた無線資源を有効に利用することを実現している．また，セグメント損失時には，必要なセグメントのみ再送を行うよう再送制御を改良したことにより，不要なセグメント送信の抑制を実現している．更に，同一リンク上に複数の TCP フローが存在する場合に，端末間，及びプロトコル間の公平性を確保した通信が実現可能であることを示している．

第 5 章では，自律分散的なネットワーク構築から生じる通信負荷集中を分散す

るための手法について検討を行っている．前述したように，無線アドホックネットワークでは，端末同士が互いに連携してネットワーク構築を行うことから，通信環境が優れているリンクや端末を優先的に利用することにより，安定した経路構築を行っている．そのため，そのような箇所や端末に流れるトラヒック量が相対的に増加することになり，ネットワーク内の公平性が低下することや通信環境の悪化が問題となる．従来，無線アドホックネットワークでは，主に空間的に負荷分散を実現する手法が提案されていた．しかし，これらの手法では，短い周期で変動する通信環境に対応することが困難であることや近傍の端末密度によって負荷分散性能が劣化するという問題があった．また，これらの手法では，経路上のみの通信状態を評価し，負荷分散制御を行っているため，経路近傍の端末に経路上の通信が与える影響が考慮されていなかった．そこで，本章では，経路近傍の通信状態を，漏れ聞きを利用して評価し，経路上の負荷，及び経路近傍の端末へ与える影響に基づいた負荷分散手法について検討を行っている．更に，本章では，従来の空間的負荷分散ではなく，送信速度制御を利用して時間的に負荷分散を実現する手法について検討を行い，通信環境が随時変動する無線アドホックネットワークで効率的な負荷分散を実現可能であることを示している．

第 6 章は結論であり，本論文で得られた成果をまとめ，考察を行っている．

最後に，本論文における約束事に関して述べる．

- 章，節は次のような順序で大項目から小項目へと移る．
  例： 第 3 章，**3.2**，**3.2.1**
- 式，図，及び表は，章単位で通し番号をつける．
  例：図 2.1
- 参考文献は，文章中及び文章の末尾に文献番号で示す．
- 本論文では各層での通信を区別するため，一般的に用いられている通信単位である「パケット」ではなく，トランスポート層では「セグメント」，ネットワーク層では「パケット」データリンク層では「フレーム」をそれぞれ用いる．

第2章　無線アドホックネットワーク

2.1　まえがき

ノートPCやスマートフォンなどの無線通信端末の普及に伴って，端末のみで自律分散的にネットワークを構築し，通信が可能な無線アドホックネットワークが注目されている．無線アドホックネットワークでは，携帯通信網のように固定されて基地局などのインフラ設備を用いることなく，端末同士が互いに中継制御を行うことによって電波範囲外の端末との通信を実現している．しかし，高い端末の移動性や無線通信を用いるという特徴から，障害物や電波干渉，トポロジー変化に伴う経路切断などにより，伝送途中での損失が頻発し，通信効率が低下するという問題がある．また，伝送失敗による損失によって，TCPのウィンドウ縮小が行われることで，エンドエンド間の通信効率低下が発生する．更に，無線アドホックネットワークでは，自律分散管理を基本に通信が実現されているため，通信環境の優れいている経路や端末を通過するトラヒック量が相対的に増加することとなり，ネットワーク全体の公平性を損なうといった問題がある．

本章では，本論文の主題となる，無線アドホックネットワークの概要，特徴について述べ，無線アドホックネットワークで高効率，高信頼な通信を実現する上での技術課題について述べる．

2.2　無線アドホックネットワークの概要

一般に，ネットワークを介した通信を実現するためには，端末，交換機やルータ，伝送経路などの設備が必要となる．電話網では，古くから回線交換方式が利用されてきたが，近年急速に普及したインターネットでは，パケット交換方式を用いた通信が行われている．インターネットでは，ルータがパケット交換の役割を担い，光ファイバなどの伝送経路とともに，インフラとしてあらかじめ敷設されている．近年では，伝送経路を無線化した携帯電話や無線LANなどのネットワークが普及しているが，これらのネットワークにおいても，基地局やアクセスポイントなどの設備をあらかじめ設置する必要がある．

これに対し，無線アドホックネットワークでは，図2.1に示すように，既存のインフラを用いることなく，端末同士で自律分散的にネットワーク構築を行い，直接無線通信を行う．あて先端末が送信元端末の電波範囲内に存在しない場合においても，図2.2に示すようなネットワーク内の端末がデータの中継を行うマルチホップ通信を用いることで，あて先端末への通信を可能にしている．

このように，無線アドホックネットワークでは，既存のインフラに依存せず通信を行うことから，様々な場面での利用が期待されている．例として，持ち寄った

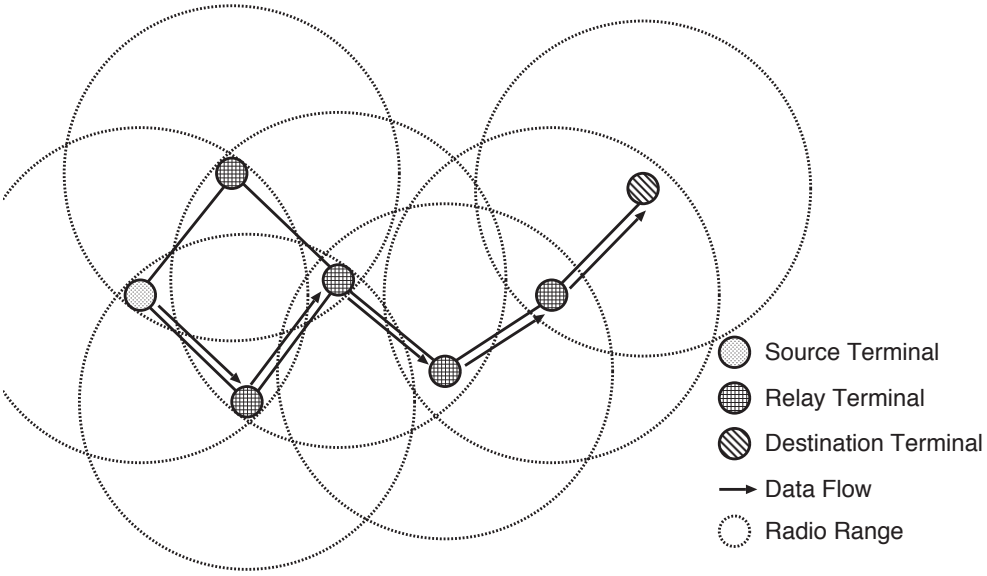

図 2.1 アドホックネットワークの構築例

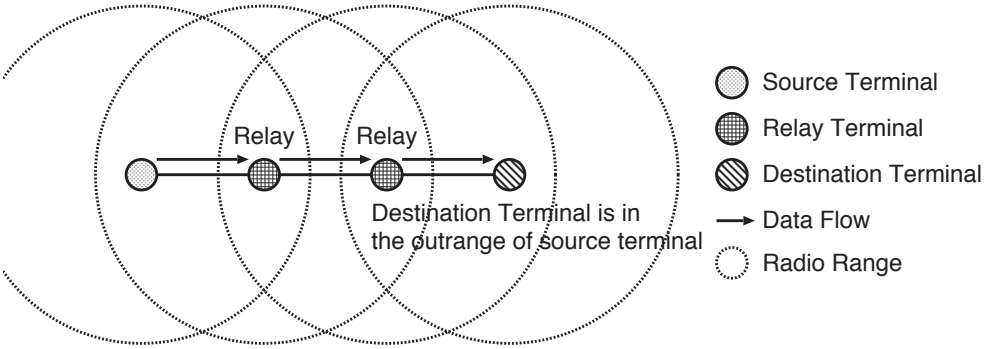

図 2.2 マルチホップ通信

ノート PC などの端末間でファイルを共有することや，既存のネットワークから切り離した別のネットワークを一時的に構築ことが容易に可能となる．また，マルチホップ通信を用いてデータ転送を行うことで，無線 LAN などの電波が到達しない箇所でも，端末を経由したネットワークアクセスを可能にすることができる．近年では，携帯ゲーム端末にアドホックモードが搭載され，持ち寄った端末間で相互接続を行うことで，多人数でのプレーを容易に楽しむことも可能である．また，高度道路交通システム（ITS）の一部である車車間通信についても，無線アドホックネットワークの一例として，活発な研究が行われている．このように，様々な場面で無線アドホックネットワークの応用が期待されているが，現在，最も注目されているのは災害時での応用である．災害時には，既存のインフラが損害を受けることがあり，インターネットや携帯電話網が寸断される可能性がある．そのような場合に，一時的なインフラとして，無線アドホックネットワークを利用することが期待されている．

次にアドホックネットワークの特徴について述べる．アドホックネットワークでは有線通信と比較して以下のような特徴を有している．

（1）無線通信

特定のインフラを用いずに自律的にネットワークを構築することが可能な無線アドホックネットワークでは，無線マルチホップ通信を用いてあて先端末への通信を行う．無線を用いた通信の特徴の一つとして，電波干渉や障害物などにより電波強度が大きく変化することが挙げられる．このことから，有線通信と比較して通信の品質が低下し，データ損失が生じやすいといった特徴がある．しかし，無線の電波範囲内であれば端末の位置に依存することなく通信が可能となるため，有線通信と比較して自由度の高い利用が可能となる．一方，一般に無線リンクの帯域は有線と比較して狭帯域となるため，通信負荷集中による帯域の圧迫によって輻輳が頻発することとなる．

（2）ネットワークトポロジーの変化

無線アドホックネットワークでは各端末が高い移動性をもつため，ネットワークトポロジーが端末の移動に伴って随時変化する．端末の移動により，各端末間の距離や相対位置は常に変化し，通信経路の切断や新たな経路の発生が生じる．また，あて先端末までの物理距離が電波範囲内であっても，障害物や電波干渉などによってリンクの切断が発生し，ネットワークトポロジーは変化する．そのため，送信元端末からあて先端末までの通信経路を構築するためには，有線通信におけるルーチングプロトコルでは対応が困難であるため，無線アドホックネットワークに対応したルーチングプロトコルが必要である．

（3）ルーチング

無線アドホックネットワークでは，ネットワークの状態を統合的に管理するためのサーバなどを用いず，端末のみで自律分散的なネットワーク構築が行われる．そのため，各端末はあて先端末への通信経路を確立する際に，ネットワーク内の他の端末の存在をあらかじめ知る必要がある．無線アドホックネットワークでのルーチングプロトコルは，主にプロアクティブ形（テーブル駆動形）とリアクティブ形（オンデマンド形）に大別される．プロアクティブ形プロトコルは，各端末が保持する情報を定期的に交換することで，経路情報を常に最新の状態に保つ方式である．代表的なプロトコルとして DSDV（Destination Sequenced Distance Vector）[3]，CSGR（Cluster Switch Gateway Routing）[4] などがある．リアクティブ形プロトコルは，データを送信する必要がある場合に，随時経路探索を行い，経路情報を取得する方式である．代表的なプロトコルとして AODV（Ad hoc On-demand Distance Vector）[5]，DSR（Dynamic Source Routing）[6] などがある．その他，テーブル駆動形，オンデマンド形双方の特徴をもち合わせたハイブリッド形ルーチングプロトコルも提案されている．

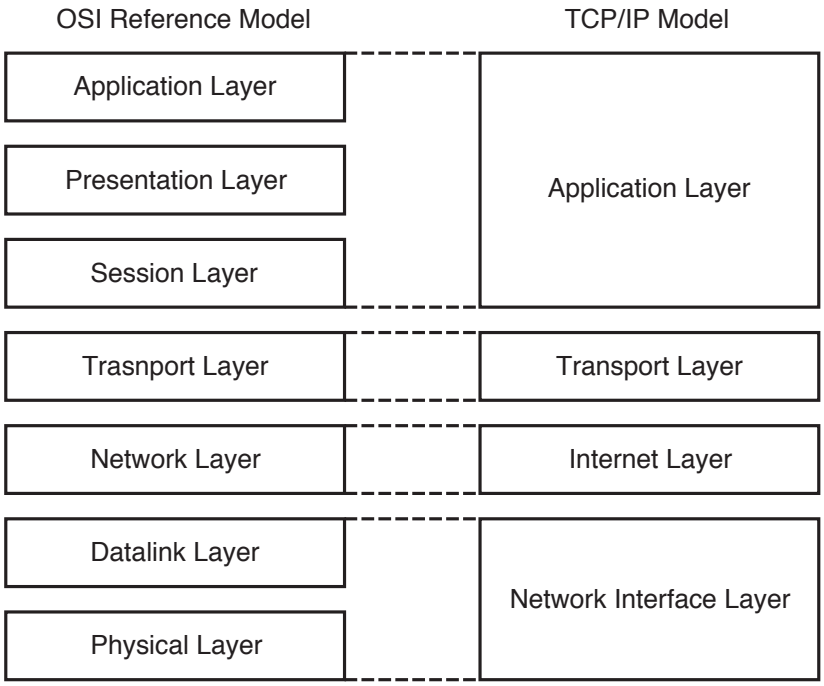

図 2.3 OSI 参照モデル

2.3 無線アドホックネットワークでの通信とその問題

無線アドホックネットワークでは，一般にインターネットと同様の OSI 参照モデルに基づいたプロトコル構成を用いて通信を実現している．図 2.3 に OSI 参照モデルの概略図を示す．その中から本節では，データリンク層，ネットワーク層，トランスポート層について述べる．

一般に，データリンク層では，通信先への物理的な通信経路を確立し，経路に流れるデータの誤り検出などを行う．無線アドホックネットワークでは，無線 LAN などで広く用いられている IEEE 802.11 を利用することが主に想定されている．ネットワーク層では，あて先端末まで送信すべきデータを届けるための通信経路構築や IP アドレスに代表されるような通信経路内のアドレス管理等を行う．無線アドホックネットワークでは，IP を用いた通信が想定され，経路構築には無線アドホックネットワークのための AODV や DSR などのルーチングプロトコルが用いられる．トランスポート層では，エンドエンド間での通信を管理し，送信速度制御や再送制御，誤り訂正などを行う．また，TCP/IP を用いた場合には仮想的な通信路であるセッションの制御も行う．無線アドホックネットワークでは，トランスポートプロトコルとして，主に UDP（User Datagram Protocol）[7] や TCP が用いられる．

次に各層での制御について述べる．

（1）データリンク層

ここでは，無線アドホックネットワークで用いられる代表的なデータリンクプ

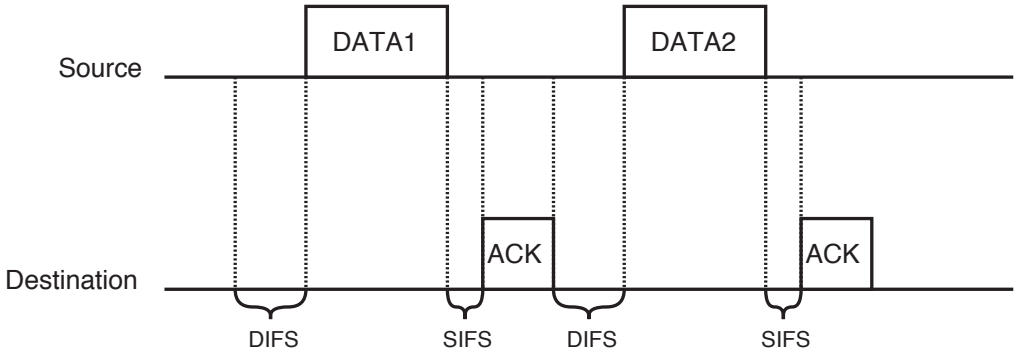

図 2.4 CSMA/CA でのデータ送信例

ロトコルである IEEE 802.11 について述べる．IEEE 802.11 では，データ転送のために無線通信を用いることから，有線通信と比較してビット誤りが発生しやすいという特徴があるため，フレーム損失が発生した場合には，自律的に再送を行う仕組みを備えている．有線通信で用いられるデータリンクプロトコルでは，フレーム送信時に ACK を用いた通信の可否の確認を行っていないが，IEEE 802.11 では各データフレームに対応した ACK フレームを受信確認に用いる．送信元端末が，送信データフレームに対する ACK フレームを受信すると次のデータフレームの送信を行うが，フレーム損失が発生し，ACK フレームを受信できない場合には再送を行い，再送フレームに対する ACK フレームが返答されるまで次のフレーム送信を行わない．また，各フレームはシーケンス番号を格納しており，あて先端末はデータフレームのシーケンス番号が正しい場合のみ，送信元端末へ ACK フレームを返答し，受信データを上位層へ転送する．

IEEE 802.11 では CSMA/CA（Carrier Sence Multiple Access with Collision Avoidance）を利用することでフレームの衝突を回避している．CSMA/CA における通信例を図 2.4 に示す．

CSMA/CA を用いた通信を行う場合，送信元端末は以下の手順でフレーム送信を行う．送信元端末は，データフレーム送信前にキャリアセンスを行うため，DIFS（Distributed Inter-Frame Space）で定義されている時間待機する．この DIFS 間に，他の端末が通信していることを検知した場合には，衝突回避を行うためフレーム送信を中止し，ランダム時間待機した後，再度キャリアセンスを行う．DIFS 間に，他の端末からのデータ送信が検出されなかった場合には，データフレームの送信を行う．データフレームを受信したあて先端末は，受信後に SIFS（Short Inter-Frame Space）で定義されている時間待機し，ACK フレームを返送する．

このように，CSMA/CA を用いることによって，電波を受信することが可能な隣接端末とのフレーム衝突を低減することが可能となる．しかし，2 ホップ先の端末と同時に通信を行った場合，送信元端末は電波送信を検知することができないため，受信端末側でデータ衝突が発生する問題がある．これを隠れ端末問題とい

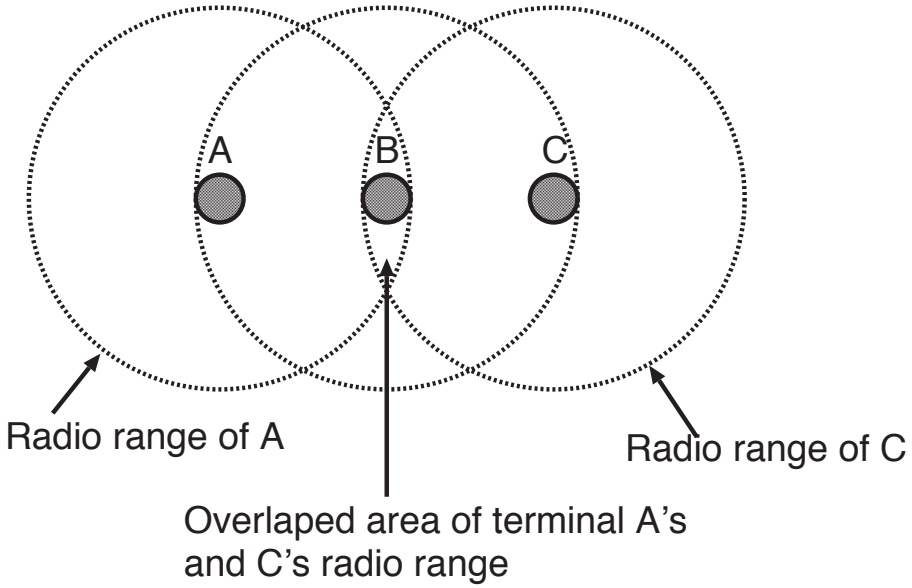

図 2.5 隠れ端末問題

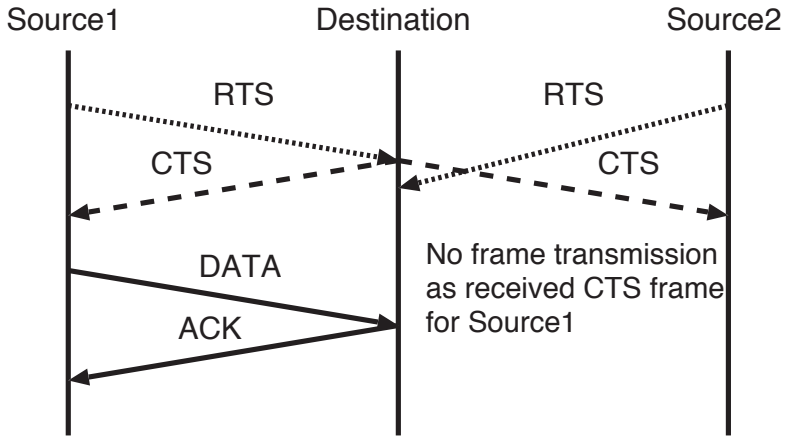

図 2.6 動作手順

い，送信データ量に比例して発生しやすいという特徴がある．図 2.5 に隠れ端末問題の概略を示す．図の例では送信元端末 A，送信元端末 C があて先端末 B へフレーム送信を行うが，端末 A と端末 C はお互いの通信状態を把握することができないため，それぞれ自身のタイミングでフレーム送信を行う．しかし，あて先端末 B では端末 A と端末 C からのフレームを同時に受信することができないため，フレーム衝突が発生することとなる．また，隠れ端末問題は，マルチホップ環境でないシングルホップの場合でも発生する可能性がある．

この問題に対して RTS（Request To Send），CTS（Clear To Send）メッセージを用いてフレーム衝突を回避する手法が用いられている．端末が RTS しきい値以上のデータサイズをもつフレームを送信する場合には，RTS/CTS を用いてあらかじめ送信権を取得する必要がある．送信元端末からの RTS をあて先端末が受信

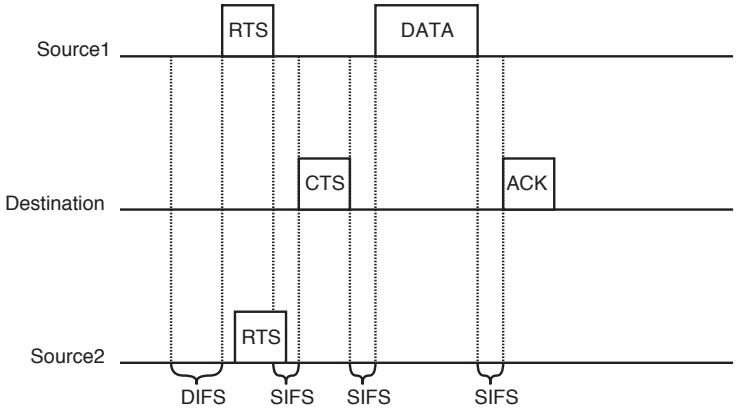

図 2.7 データ送信例

すると送信端末へ向けて CTS フレームを送信する．この時，複数の端末から RTS フレームが送信された場合でも，ただ一つの送信元端末へ CTS フレームを返答する．また，あて先端末以外の端末が RTS フレームを受信した場合や CTS フレームが返答されない場合には，各端末は一定時間通信を行わない状態へと遷移する．図 2.6，図 2.7 に RTS/CTS の動作手順とデータ送信例を示す．

このように，RTS/CTS を用いて，通信開始前に受信端末主導で通信の優先権を設定することにより，複数端末によるフレーム送信を抑制することで，衝突発生を低減することが可能となる．しかし，送信データ量の増加に従って，RTS/CTS によるフレーム送信がチャネルを消費することとなり，オーバヘッドの増加や通信効率の低下が発生することとなる．そのため，RTS/CTS の利用と衝突発生はトレードオフの関係にあり，適切なしきい値設定が必要となる．

（2）ネットワーク層

ここではアドホックネットワークにおけるルーチング手法について述べる．アドホックネットワークにおけるルーチング手法として代表的なものを以下に示す．

（a）プロアクティブ形ルーチングプロトコル

プロアクティブ形ルーチングプロトコルの代表例として DSDV について以下に述べる．DSDV においてネットワーク内の各端末は，同一ネットワーク内の到達可能なすべてのあて先端末とそのあて先端末までのホップ数を記録したルーチングテーブルを管理している．従って，送信元端末が経路を必要とする，しないに関わらず，常に利用可能な経路情報が準備されている．また，シーケンス番号を用いて経路が新しいか古いかの判断を行っている．ルーチングテーブルの更新情報は，定期的にネットワーク全体に向けて送信されるため，ネットワークの負荷を上昇させる要因となる．一方，DSDV ではフルダンプ，ネットワークプロトコルデータユニットと呼ばれる 2 種類のルーチン情報更新パケットを用いてこの問題を軽減している．新しいルーチング情報を送信する時には，あて先端末のアドレス，あて先端末までのホップ数，シーケンス番号を格納する．

（b）リアクティブ形ルーチングプロトコル

リアクティブ形ルーチングプロトコルの代表例としてAODVについて以下に述べる．AODVはDSDVプロトコルを基に構築されており，AODVではデータ送信を行う際に経路を作成することによって，必要なブロードキャスト数を少なくするように改良を加えている．送信元端末はあて先端末へデータを送信する際に，有効な経路が自身の経路テーブルに存在しない場合は経路探索を行う．送信元端末は，隣接端末に向けて経路要求（RREQ:Route Request）パケットをブロードキャストする．このとき，RREQパケットを受信した隣接端末はさらに自身の隣接端末にRREQパケットを転送する．あて先端末，またはあて先端末への有効な経路を保持している端末に到達するまで同様の手順で転送を繰り返す．また，AODVではシーケンス番号を用いて経路がループしないことと最新の経路情報を保持することを保証している．RREQパケットを転送する際に，端末はRREQパケットを受信時にRREQを送信した隣接端末をルーチングテーブルに記録することで，逆方向の経路を確立する．RREQパケットがあて先端末，またはあて先端末への有効な経路を保持している端末に到達すると，その端末は経路要求応答（RREP：Route Reply）パケットを送信元端末へ向けて送信する．経路探索時のRREQパケット，RREPパケットの送信例を図2.8に示す．ここで，RREPパケットは，RREQパケットを転送する際に記録しておいた逆方向の経路を利用して送信元端末へ送信される．また，RREPパケットが送信される際，経路上の端末はそのRREPパケットの送信元端末への経路を自身のルーチングテーブルに記録する．また，送信元端末が移動するとあて先端末への新しい経路を見つけるために経路探索を再び行う．経路上の端末が移動するとその上流端末が移動を検知して上流端末に対してリンク切断通知メッセージを送信する．これにより，経路のその部分が削除される．同様の処理が上流に向かって送信元端末に到達するまで行われる．更に，AODVではHelloメッセージを用いることで，自身の存在を隣接端末に知らせることができる．Helloメッセージは定期的にブロードキャストされ，リンク接続状況などに関する情報を送信することができる．

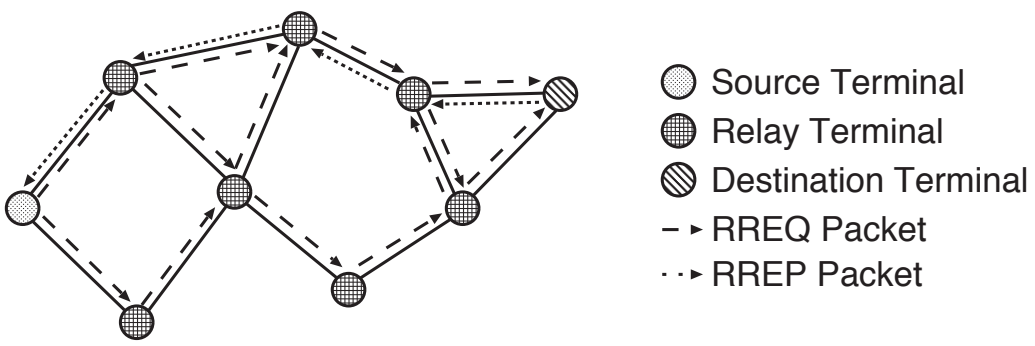

図 2.8 RREQ，RREPパケット送信例

（c）オポチュニスティックルーチングプロトコル

オポチュニスティックルーチング（OR：Opportunistic Routing）では，無線アドホックネットワークの特徴を利用し，特定の経路に依存しないパケット転送を実現している．無線アドホックネットワークにおける OR では，無線通信の同報性を利用し，漏れ聞きやマルチパスルーチングを応用し，あて先端末までの経路に冗長性をもたせることによって，通信成功率及び通信確率の向上を実現している．OR において，あて先端末までパケットを転送するためには，動的な中継端末選択が必要となる．そのため，通信成功率，ホップ数，位置情報などの指標を利用して中継端末の優先度設定が行われる．通信成功率を中継端末選択の指標として用いる手法 [8]–[11] では，通信開始前に取得していたあて先端末までの通信成功率や各リンクの通信成功率を基に，あて先端末までの推定転送回数が少なくなるよう中継端末選択が行われる．ホップ数を指標として用いる手法 [12]–[15] では，通常のルーチングと同様に，あて先端末までのホップ数がより短い端末により高い優先度を割り当て，パケット転送が行われる．位置情報を指標として用いる手法 [16]–[19] では，通信開始前に取得していたネットワーク内の全端末の位置情報を基に，通信範囲内に存在している端末の中からあて先端末へ近い順に優先度割り当てが行われる．これにより，ホップ数を指標とした手法と同様に，あて先端末へ近づくようパケット転送が行われる．また，消費電力を中継端末選択の指標として用いる手法 [20] やユーザ効用を基に中継端末選択を行う手法 [21] なども提案されている．

このように，無線アドホックネットワークでのルーチングプロトコルでは，端末移動によるネットワークトポロジーの変化に対応するため，有線通信の場合と比較して，柔軟な経路構築を行っている．しかし，プロアクティブ形，リアクティブ形のルーチングプロトコルでは，通信開始時に選択された通信経路を継続的に利用するため，端末の移動性が高い環境ではトポロジー変化によって経路の生存時間が短くなることや頻繁な経路再構築によるオーバヘッドの増加などが問題となる．また，OR では，一般的にパケット転送をブロードキャストによって実現しているため，ネットワーク内の総転送量が増加することとなり，ネットワーク資源を消費することとなる．

（3）トランスポート層

現在インターネットなどの有線通信で用いられている代表的なトランスポートプロトコルとして UDP と TCP がある．また，TCP の改良プロトコルとして BIC [22]，CUBIC [23]，H-TCP [24]，TCP Hybla [25]，STCP [26]，YeAH [27]，TCP Illinois [28]，Compound TCP [29]，TCP Veno [30] などが提案され，近年では様々な OS に実装されている．ここでは，トランスポートプロトコルの役割と UDP，TCP の基本的な動作について述べる．

アドホックネットワークでの基本的なトランスポートプロトコルの役割は，イ

ンターネットの場合と同様に，送信元端末からあて先端末までデータを届けることである．そのため，セッション管理，フロー制御，誤り訂正などの機能を備えたプロトコルが用いられる．また，アプリケーションから受け取ったデータを適切な単位に区切り，伝送する役割も担っている．トランスポートプロトコルでは基本的に，送信元端末とあて先端末のエンドエンド間の制御のみを行うため，中継ルータや中継端末などではトランスポートプロトコルを意識せずに通信を行っている．そのため，トランスポートプロトコルはネットワークの状態を正確に把握することが困難であり，経路切断や輻輳などを把握するためには下位層からの情報が必要となる．以下に代表的なトランスポートプロトコルである UDP と TCP について述べる．

（a）UDP

UDP はコネクションレス形のトランスポートプロトコルであり，軽量で高速なデータ転送が可能であるという特徴を有している．そのため，実時間性の高いアプリケーション（ネットワークゲーム，ストリーミング配信など）で用いられることが多い．コネクションレス形プロトコルとは，通信を行う前に仮想的な通信路（コネクション）を構築せず，通信要求を上位層から受信するとすぐにデータ転送を開始する．また，UDP では上位層からのデータをデータグラム（Datagram）という単位に分割し，下位層へ転送するだけのプロトコルであるため，送信速度を制御するといったフロー制御機能を備えていない．また，送信元端末からあて先端末へ一方的にデータを送信するプロトコルであるため，通信途中でセグメント損失が発生した場合でも，損失したデータを再送することなく通信を継続する．そのため，信頼性よりも即応性や通信速度を重視する前述したようなアプリケーションで用いられることが多い．これは，伝送途中に 1 つのセグメントが損失しても，ネットワークゲームでは一時的に動作がとまるだけであり，ストリーミング配信でも後続のデータにより問題なく再生が可能であることが多いためである．

（b）TCP

TCP はコネクション形のトランスポートプロトコルであり，UDP と比べて高い信頼性を達成することができる．また，ACK を待たずに一度に送出可能なデータ量であるウィンドウを用いた送信速度制御を行うことや，ACK を用いた信頼性確保を行っている．そのため，通信速度よりも信頼性が必要となるアプリケーションで用いられることが多い．コネクション形プロトコルとは，コネクションレス形プロトコルとは逆に，通信を行う前に送信元端末とあて先端末間でコネクションを確立し，専用の仮想的な通信路を用いて通信を行う．アプリケーション側からは，このコネクションを用いることによって特別に意識することなく，通信の信頼性を保証できる．また，TCP では，ウィンドウサイズを調整することによって送出可能なデータ量を制御することが可能であり，ネットワークの状態に合わせた送信速度を調整する．また，シーケンス番号を用いて到着順序を管理しているため，

セグメント損失や到着順序逆転が発生した場合には自動的にそれらを訂正する機能を備えている．このような TCP の特徴から，TCP はインターネットで多く用いられ，Web サイトの閲覧などに用いる HTTP（Hyper Text Transfer Protocol）や電子メールの送受信に用いる IMAP（Internet Message Access Protocol），POP3（Post Office Protocol 3），SMTP（Simple Mail Transfer Protocol），ファイル転送に用いる FTP（File Transfer Protocol），遠隔操作に用いる TELNET，SSH（Secure SHell）などで用いられている．

このように，UDP と TCP はそれぞれ高速，高信頼な通信が可能という特徴を有しているが，無線アドホックネットワークでは，次のような問題が発生する．UDP では，信頼性確保の仕組みが存在しないため，損失が発生しやすい無線アドホックネットワークでは，受信側で正常にデータを受信することが困難である．また，UDP を用いた通信では，通信路の通信状態を考慮することなくデータ送信が行われるため，多数の端末が同時にデータを送信するような場合には，トラヒックがリンク容量を超過する可能性があるため，輻輳発生の要因となる．TCP では，ACK を用いて通信の可否を判断しているため，無線アドホックネットワークで頻発するセグメント損失を回復することが可能であるが，セグメント損失が頻発する環境では別の問題が生じる．一般に，TCP では重複して受信した ACK を輻輳検知に用いているため，セグメント損失が頻発する無線アドホックネットワーク

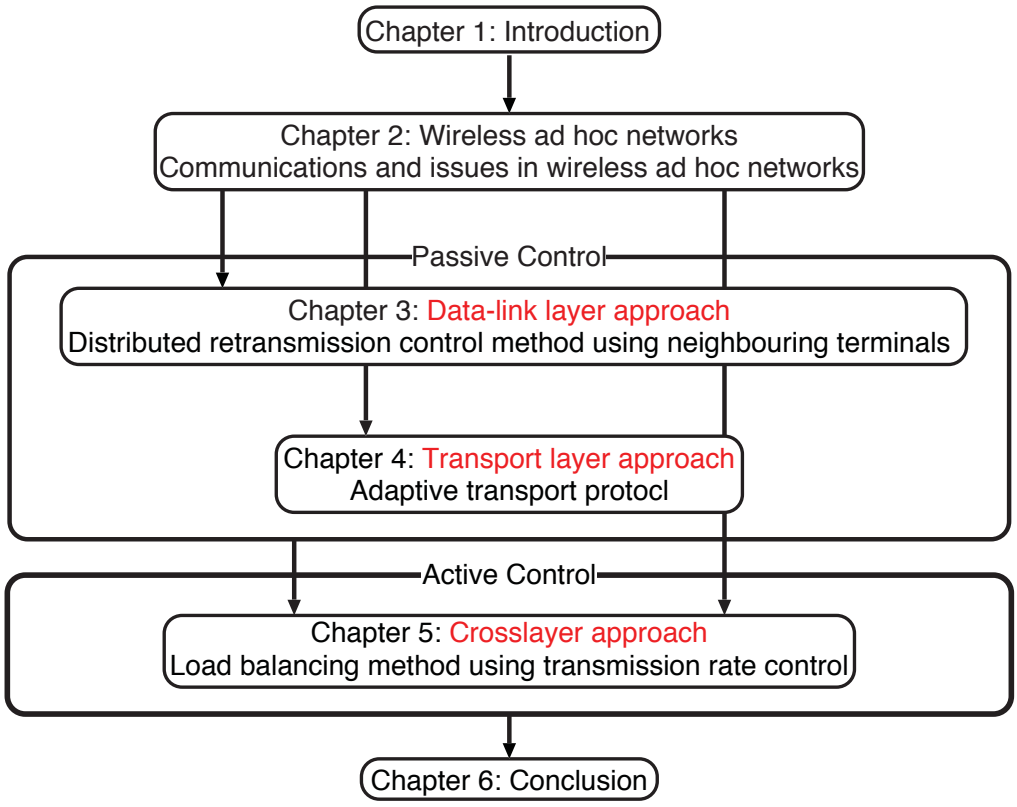

図 2.9　本論文の構成

では，データ損失を輻輳の発生と誤検知する可能性がある．輻輳発生を検知した場合，TCP では送信側のウィンドウサイズを減少させることによって，単位時間あたりに送出するデータ量を制限し，輻輳回避を行う．そのため，輻輳以外の要因によってセグメント損失が頻発する無線アドホックネットワークでは，不必要に伝送速度が低下することとなる．従って，無線アドホックネットワークで TCP を用いた効率的な通信を実現するためには，セグメント損失とウィンドウ制御を切り離した送信速度制御により通信効率向上を図る必要がある．

本論文では，前述した無線アドホックネットワークでの問題点を克服するため，図 2.9 に示すように，受動的制御として第 3 章ではデータリンク層での再送制御方式について論じ，第 4 章では無線アドホックネットワークの通信特性に基づくトランスポートプロトコルについて論ずる．また，第 5 章では，能動的制御として，クロスレイヤ制御を用いた負荷分散手法について論ずる．

2.4 むすび

本章では，無線アドホックネットワークの概要と無線アドホックネットワークでの通信について述べた．2.2 では，無線アドホックネットワークの概要を示し，有線通信と異なる特徴を有することを述べた．また，有線通信と異なる環境で通信を行うため，既存のプロトコルをそのまま用いることが困難であることを示した．2.3 では，無線アドホックネットワークで用いられている代表的なデータリンク層，ネットワーク層，トランスポート層のプロトコルを挙げ，動作の概要について述べた．

第3章　近傍端末を用いた自律分散再送制御手法

3.1　まえがき

端末の移動性が高く，ネットワークトポロジーの変化が大きいアドホックネットワークでは，DSDV，OLSR（Optimized Link State Routing）[31] などのプロアクティブ形や DSR，AODV などのリアクティブ形のルーチングプロトコルが数多く提案されている．しかし，これらのプロトコルは主にホップ数を基準に経路選択を行うため，経路の安定性や物理的な伝送距離が考慮されておらず，リンクでの伝送誤りやリンク切断などが発生する可能性がある．

これらの問題に対し，複数の通信経路をあらかじめ構築し，リンク切断時に代替経路へ切り換えることで通信効率の低下を抑制するマルチパスルーチング手法が提案されている [32]．しかし，複数経路を構築することからネットワーク資源を通常のルーチングプロトコルと比較して多く消費することや，端末の移動性が高い場合には適切な代替経路を構築することが困難となる．一方，利用できないリンクや不安定なリンクを一時的に迂回し，フレームの再送や中継を行う再送制御手法 [33]，[34]，[36] や一時的経路変更手法 [35] が提案されている．これらの手法では，リンクごとの制御に焦点を当て，近傍端末によるフレーム漏れ聞きを利用した再送制御，または中継制御を行うことで通信効率の低下を抑制している．しかし，再送制御手法では，フレームごとに通信の可否を判断しているため，端末負荷や再送処理時の遅延の増加などが問題となる．また，一時的経路変更手法では，適切な経路変更期間を設定することが困難であることから，不必要に迂回経路を利用することで通信効率の低下が発生する．

本章では，アドホックネットワークにおけるこれらの問題に対応した再送制御手法を提案する．提案手法では，従来の再送制御手法と同様に，径路近傍の端末による漏れ聞きを利用した再送制御及びフレームヘッダの拡張を行うことによって，効率的，自律分散的に再送制御を行う．また，コンピュータシミュレーションを用いて，従来手法との性能比較を行うことで，提案手法の有効性を明らかにする．

3.2　無線アドホックネットワークにおける再送制御

3.2.1　一般的な再送方式

TCP では，信頼性を確保するためにデータを受け取ったことを送信元端末に明示的に通知する必要がある．無線アドホックネットワークでは，端末が移動することによってネットワークトポロジーが常に変化し，利用可能な経路も常に変化

する．そのため，経路探索時に利用可能であった経路が通信開始時，または通信途中に利用できなくなる可能性がある．また，無線を利用して通信を行っているため，電波干渉や障害物の影響などによりリンクが一時的に利用できなくなる可能性があり，フレーム損失が発生する原因となる．フレーム損失が発生した場合，信頼性を確保するために損失したフレームを再送する必要があり，いくつかの再送方式が存在する．無線アドホックネットワークにおける再送として，各リンクごとに再送を行う方式と送信元端末からあて先端末へのエンドエンド間での再送方式が考えられる．前者の方式は，MAC プロトコルで行われている再送方式であり，後者の方式は TCP で用いられている再送方式である．また，UDP ではデータを一方的にあて先端末へと送信するため，再送制御は用いられていない．

（1）リンクでの再送

無線アドホックネットワークにおける通信では，主に各端末が中継を行うマルチホップ通信が利用されている．各リンクでは，ACK を用いることでフレーム送信の可否を確認し，信頼性を確保している．中継端末はフレーム送信後，一定時間経過後に ACK フレームを確認できない場合には，フレーム損失が発生した，またはACK フレームの損失が発生したと判断し，リンク上で再送を行う．図 3.1 にリンクでの再送の概略を示す．再送までの間隔はプロトコルの実装に依存し，再送を行う回数も同様である．一定回数再送を行っても ACK フレームを受信できない場合には，そのリンクが利用不可と認識し，経路の再構築が行われる．また，リンクでの再送は，IEEE 802.11 では ARQ（Automatic Repeat reQuest）と呼ばれ，通信の信頼性を確保するため広く用いられている再送方式である．

（2）エンドエンド間での再送

エンドエンド間での再送は，TCP によって行われ，送信元端末とあて先端末間で行われる再送である．一般に，無線アドホックネットワークでは，送信元端末が経路を構築し，あて先端末へデータを送信するが，中継端末の移動や障害物の影響などによって経路が利用できなくなる可能性がある．このような場合，リンク上でのフレーム再送が最初に行われ，フレーム再送が一定回数失敗した場合には

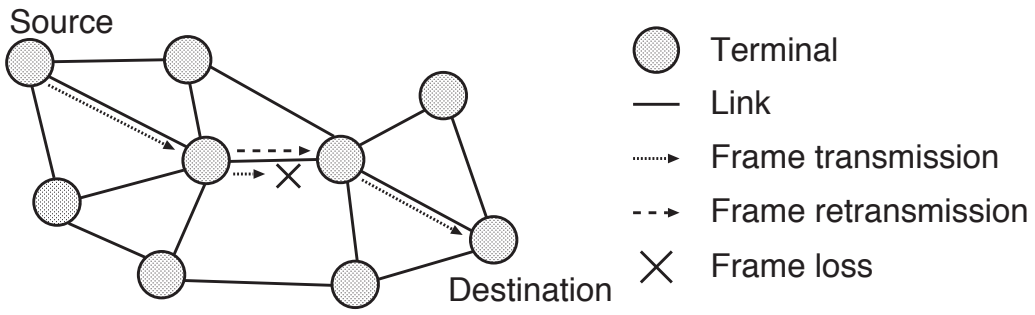

図 3.1　リンクごとの再送

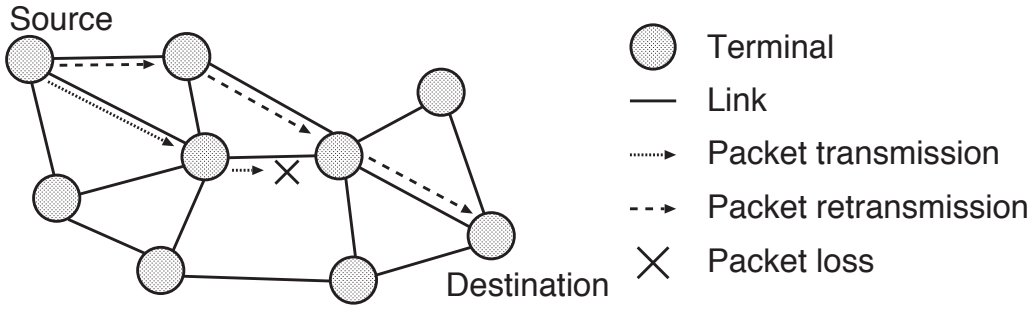

図 3.2 エンドエンド間での再送

送信元端末が経路を再構築し，あて先端末へのパケット送信を再び行う．図 3.2 にエンドエンド間での再送の概略を示す．図 3.2 では，リンク上でフレーム損失が発生した場合を想定している．この場合，まずはリンク上での再送制御が行われるが，一定回数試行した後，正常にフレーム再送が完了しなかった場合には，エンドエンド間での再送が実行される．またこのとき，ルーチングプロトコルによっては，経路再構築が実行され，経路再構築後に新たな経路でエンドエンド間の再送が行われる．

3.2.2 DRNT

DRNT（Distributed Retransmission Method using Neighbour Terminal）[33], [34] は，フレームの漏れ聞きを利用して経路近傍の端末が一時的にフレームを保持し，フレーム損失発生時に近傍端末から自律的に損失フレームを再送する手法である．IEEE 802.11 では，通常は自端末あてでないフレームを受信すると破棄するが，DRNT ではあて先端末が自端末の近傍端末テーブルに存在する場合には一定期間保持する．一定時間経過した後，近傍端末があて先端末からの ACK を確認できない場合には，保持していたフレームを用いて，送信元端末に代わって再送を行う．

DRNT における再送手順を図 3.3 に示す．送信元端末 A があて先端末 B へフレーム送信を行った際に．フレーム損失が発生した場合を想定する．この際，無線通信の特徴から，端末 A の近傍端末である端末 C も端末 B へ送信したパケットを受信することができる．端末 A からのフレームを漏れ聞いた端末 C は，フレームのあて先である端末 B が自身の近傍端末テーブルに存在するか確認し，存在する場合には一定時間保持する．A–B 間でフレーム損失が発生した場合には，端末 B からの ACK が返答されないため，端末 C はフレーム損失が発生したと認識し，保持していたフレームの再送を行う．この際，再送フレームヘッダの送信元端末は本来の送信元端末である端末 A へ書き換え，再送を行う．端末 C からの再送フレームを受信した端末 B は，通常と同様に ACK を返答するが，A–B 間のリンク

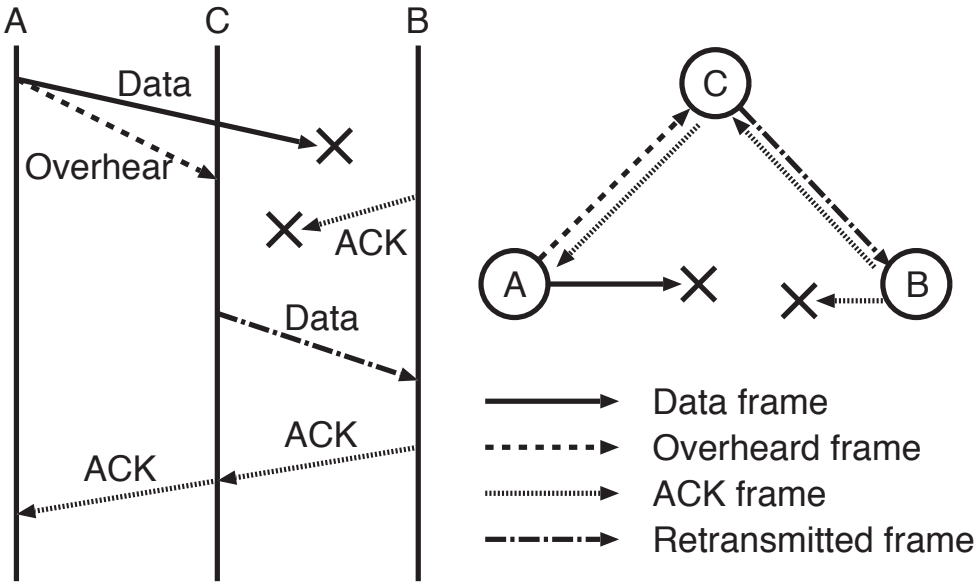

図 3.3 DRNT の動作例

が利用できない可能性が高いため，再送を行った端末 C は端末 B から返答された ACK を端末 A へ中継する．また，近傍端末が複数存在する場合には，再送フレームの衝突が発生する可能性があるため，再送を行うまでのタイムアウト時間をランダムタイマを用いて調整する．更に，近傍端末が他の端末からの再送フレーム，または ACK を受信した場合には，自身は再送を行う必要がないと判断し，保持していたフレームを破棄する．

このように，DRNT では経路の近傍端末を利用して自律的に損失したフレームの再送を行うことで，信頼性，接続確率，及び通信効率の向上を実現している．しかし，フレームごと処理を行うため，連続したフレーム損失が発生した場合には，ACK 確認のために再送まで一定時間待機する必要があり，遅延が増加する問題がある．

3.2.3 TRM

TRM（Temporary Route Modification）[35] では，連続したフレーム損失時に DRNT の性能が低下する問題に対し，一定回数連続した中継制御を行うことで，ACK 確認の待ち時間を低減し，通信効率の向上を図っている．TRM におけるフレームの漏れ聞き及び ACK 確認の手順は DRNT と同様に行われるが，連続した中継制御のため，新たに制御メッセージ，承認メッセージを用いる．制御メッセージには，損失フレームの送信元端末，及びあて先端末アドレス，制御メッセージの送信元アドレス，及び中継回数が含まれる．ここで，中継回数は，近傍端末を利用した中継を行う回数を示し，初期値を 2 とし，予め設定した間隔内に再度中

継制御を行うごとに中継回数を 2 倍に増加させる．

TRM における中継制御手順を図 3.4 に示す．送信元端末 A があて先端末 B へフレーム送信を行った際に，フレーム損失が発生した場合を想定する．この際，リンク A–B の近傍端末である端末 C は，漏れ聞きを用いてリンクの通信の可否を監視する．端末 B から端末 A あての ACK が返送されず，リンク A–B 間でフレーム損失が発生したと端末 C が判断した場合には，端末 C は本来のあて先である端末 B に対し，制御メッセージを送信する．ここで，制御メッセージには，損失フレームの送信元端末及びあて先端末情報として，端末 A 及び端末 B のアドレスを格納し，制御メッセージの送信元として端末 C のアドレスを格納する．また，制御メッセージ内の中継回数には，累積中継制御回数に応じて算出した中継回数を格納する．制御メッセージを受信した端末 B は，自身に対するフレーム送信が失敗したことを認識し，近傍端末による中継制御を許可するため，承認メッセージを端末 C あてに送信する．このとき，端末 B が複数の端末から制御メッセージを受信した場合，先着した制御メッセージのみに承認メッセージを返送する．あて先端末 B からの承認メッセージを受信した端末 C は，自身が中継端末として選択されたと認識し，中継制御を行うために，送信元端末 A にあて先端末 B へ送信したものと同じ制御メッセージを送信する．これにより，送信元端末 A 及びあて先端末 B は，端末 C を経由したフレーム送信が行われることを認識し，一時的な経

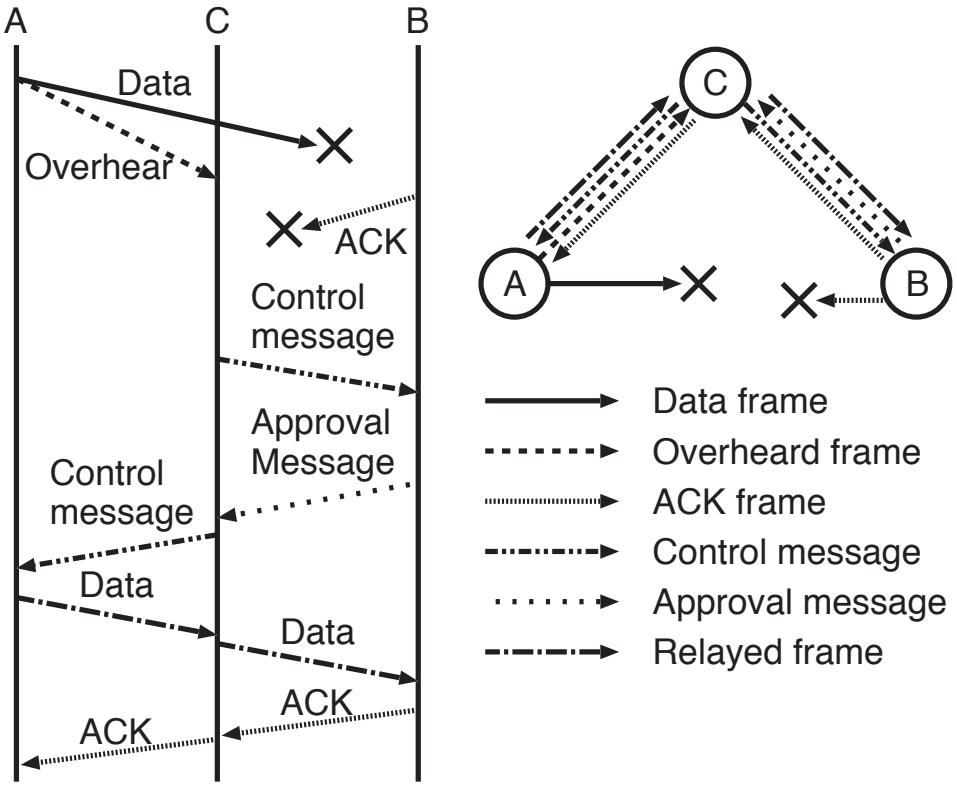

図 3.4 TRM の動作例

路変更が行われる．あらかじめ設定した回数中継制御が行われた後，近傍端末によるフレーム中継を停止し，リンク A–B を用いた通常の通信へと復帰する．

このように，TRM では経路の近傍端末を利用した一時的経路変更による中継制御を行うことで，連続したフレーム損失時の通信効率低下に対応している．しかし，中継制御によってホップ数が増加することから，ネットワーク負荷やエンドエンド間の遅延が増加する問題がある．また，アドホックネットワークでは通信環境が随時変化するため，リンク利用の可否を正確に把握することが困難であり，適切な中継回数設定を行うことが困難である．

3.2.4 CTB

CTB（Cognitive Temporary Bypassing）[36] では，DRNT と同様に，漏れ聞きを利用して一時的にフレームを保持し，近傍端末を利用して再送を行う手法が提案されている．CTB では，BCM（Bypass Candidate Message），BRM（Bypass Request Message）という 2 種類のメッセージを用い，中継端末とあて先端末間で再送制御を行っている．BCM は，フレーム損失を検知した経路の近傍端末から送信され，送信元端末及びフレームのシーケンスナンバが含まれている．BCM を受信したあて先端末は，BCM に含まれるシーケンスナンバに対応したフレームを受信しているか確認し，受信が確認できない場合には，BRM を経路の近傍端末へ返送する．BRM を受信した経路の近傍端末は，自身が保持している漏れ聞きフレー

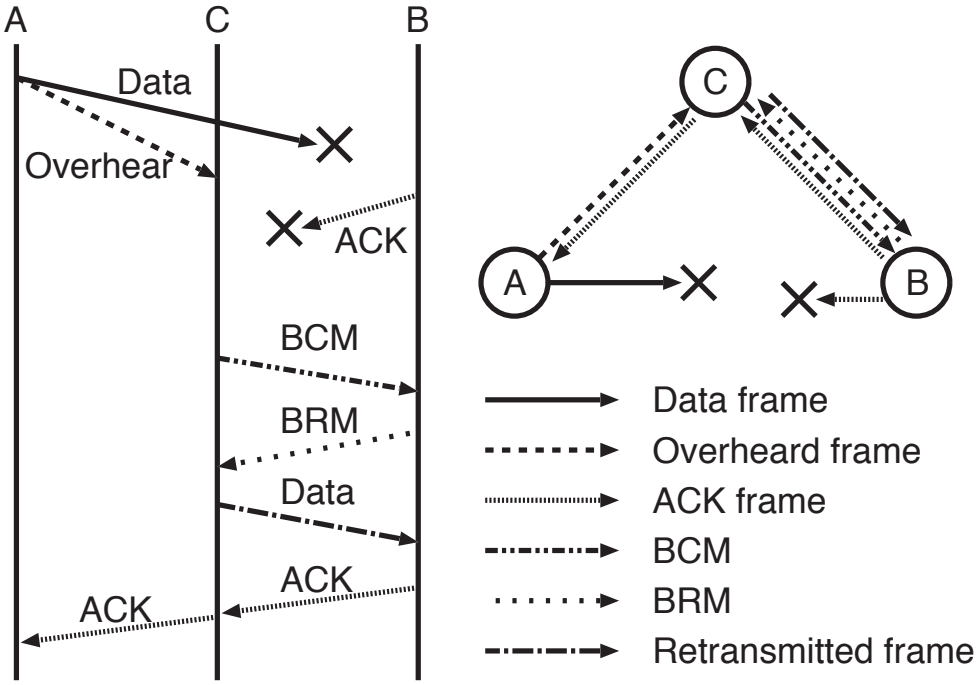

図 3.5 CTB の動作例

ムをあて先端末へ送信する．再送フレームを受信したあて先端末は，ACK を返送し，経路の近傍端末はこの ACK フレームを送信元端末へ転送する．これにより，送信元端末からのフレーム送信がタイムアウトすることなく成功し，不必要なフレーム送信を低減することができる．

CTB における再送手順を図 3.5 に示す．送信元端末 A があて先端末 B へフレーム送信を行った際に，フレーム損失が発生した場合を想定する．この際，DRNT と同様に，送信元端末 A から送信されたフレームを端末 C は漏れ聞き，自身があて先端末 B へ再送可能な場合には，一時的に保持する．あて先端末 B からの ACK 返信を端末 C が検知できない場合には，リンク A–B 間でフレーム損失が発生したと認識し，BCM を端末 B へ送信する．BCM を受信したあて先端末 B は，自身に対するフレーム送信が失敗したことを認識し，近傍端末による再送を要求するため，BRM を端末 C へ送信する．BRM を受信した端末 C は，保持していたフレームをあて先端末 B へ送信し，返送された ACK を送信元端末 A へ中継する．

このように，TRM では DRNT と同様に経路の近傍端末を利用した再送制御を実現し，信頼性，接続確立，及び通信効率の向上を実現している．しかし，DRNT と同様にフレームごとの制御のため，ACK 確認の待ち時間による遅延増加や制御メッセージの送受信によるネットワーク負荷の増加が問題となる．

3.3 近傍端末を用いた再送制御

前節で述べたように，従来手法では経路の近傍端末を利用した再送制御及び中継制御によって利用不能なリンクを回避したフレーム送信を行うことで，接続確率の向上を達成している．しかし，DRNT と CTB ではフレームごとの制御が行われているため，オーバヘッドが増加することや，TRM での適切な中継回数設定が困難であるといった問題がある．そこで，MAC ヘッダの拡張を用いて経路の近傍端末が自律的にリンクの接続性を確認し，再送が必要な場合のみ制御を行うことで，オーバヘッドや不必要な制御を抑制した再送制御方式を提案する．

3.3.1 近傍端末協力形再送制御

提案手法では，DRNT や TRM と同様に，近傍端末の把握のために Hello メッセージを用いる．各端末は，一定時間ごとに Hello メッセージをブロードキャスト

Frame control	Duration /ID	Address1	Address2	Address3	Sequence control	Address4
2byte	2byte	6byte	6byte	6byte	2byte	6byte

図 3.6 IEEE802.11MAC データフレームヘッダ

し，自身の存在を近傍端末へ通知する．また，提案手法での再送制御には，制御メッセージとその応答メッセージを用いる．制御メッセージと応答メッセージには，それぞれ保持しているデータフレームのシーケンスナンバー，送信元端末アドレス，あて先アドレス，再送を行うノードのアドレスを格納する．近傍端末が再送を行う場合には，図 3.6 に示すように，再送するデータフレームのアドレス 3 領域に自身のアドレスを格納することで，再送フレームの送信元端末が近傍端末によるものかを判断することが可能である．更に，提案手法では，再送制御に加えて，優先的に再送を行う近傍端末を自律的に決定することで，通信遅延の増加や中継制御時の冗長の再送データフレームの発生を抑制する．また，優先的に再送を行う端末を決定することにより，複数の近傍端末が存在する場合でも，再送制御フレームの衝突を抑制することができる．更に，複数の近傍端末が再送端末の候補となることで，接続率の高い端末が再送を行う確率が高くなり，再送制御による通信環境の劣化を低減することが可能となる．

3.3.2 動作手順

図 3.7 に提案手法の動作フローチャートを示す．提案手法における再送制御は，次の手順で行われる．

1. 近傍端末が送信したデータフレームを漏れ聞き，アドレス 3 領域を確認する．アドレス 3 領域が空の場合，通常のフレーム送信と判断し，再送制御のために漏れ聞いたフレームを保持する．アドレス 3 領域が空でない場合，近傍端末が送信した再送データフレームであるため，漏れ聞いたフレームを破棄する．

2. 漏れ聞いたフレームが制御メッセージか応答メッセージの場合，自端末では再送制御処理を行わないため，破棄する．それ以外の場合，一定時間漏れ聞いたフレームを保持する．フレームの保持時間は，各端末においてランダムに決定される．

3. 自端末が，漏れ聞いたデータフレームの送信元端末とあて先端末の間のリンクに対して再送を行う中継端末として設定されている場合，漏れ聞いたデータフレームを利用して優先的に再送を行う．それ以外の場合，漏れ聞いたフレームのあて先端末から ACK が返送されるのを待機する．

4. 一定時間内に，漏れ聞いたフレームのあて先端末から ACK が返送された場合，正常にフレーム送信が完了したと判断し，保持していたフレームを破棄する．一定時間経過後に，あて先端末からの ACK を確認できない場合には，フレーム損失が発生したと判断し，制御メッセージをあて先端末へ送信する．

ここで，制御メッセージには，漏れ聞いたフレームのシーケンスナンバー，送信元端末アドレス，あて先端末アドレス，自端末のアドレスを格納する．

5. 近傍端末から送信された制御メッセージを受信したあて先端末は，メッセージ内のシーケンスナンバーに該当するフレームをすでに受信しているか確認する．該当フレームが受信済みの場合，受信した制御メッセージを破棄する．また，近傍端末側では，応答メッセージが返送されないことから，再送の必要は無いと判断し，タイマーのタイムアウトを待って保持していたフレームを破棄する．該当フレームが未受信の場合，近傍端末へ再送を要求するため，応答メッセージを送信する．ここで，複数の端末から制御メッセージを受信した場合，最初に到着した制御メッセージのみに応答することとする．

6. あて先端末からの応答メッセージを受信した近傍端末は，自端末による再送が要求されたことを確認し，保持しているデータフレームのアドレス 3 領域に自身のアドレスを格納後，送信元端末に代わり，あて先端末へフレーム再送を行う．

7. 保持していたフレームの送信元端末は，近傍端末による再送フレームやあて先端末からの応答メッセージを漏れ聞いた場合，再送制御の権限が自端末から近傍端末へ委譲されたと判断し，自身による再送制御を停止する．

8. 再送を行った近傍端末は，再送フレームに対する ACK を待機し，これを確認した場合，自身の再送が成功したと判断し，以降の再送制御に備え，自端末を当該リンクに対する中継端末と設定する．

9. 中継端末として設定された近傍端末は，再送制御を行っているリンクに対し，他の近傍端末による再送フレームや再送制御を検知した場合，再送の優先権が他端末に移ったと判断し，自身の優先的再送制御を停止する．

3.3.3 動作例

提案手法による動作例を図 3.8 に示す．送信元端末 A があて先端末 B へフレーム送信を行った際に，フレーム損失が発生した場合を想定する．この際，他の手法と同様に，送信元端末 A から送信されたフレームを端末 C は漏れ聞き，自身があて先端末 B へ再送可能な場合には，一時的に保持する．あて先端末 B からの ACK 返信を端末 C が検知できない場合には，リンク A–B 間でフレーム損失が発生したと認識し，制御メッセージを端末 B へ送信する．制御メッセージを受信したあて先端末 B は，自身に対するフレーム送信が失敗したことを認識し，近傍端末による再送を要求するため，応答メッセージを端末 C へ送信する．応答メッセージを受信した端末 C は，保持していたフレームをあて先端末 B へ送信し，端末 B か

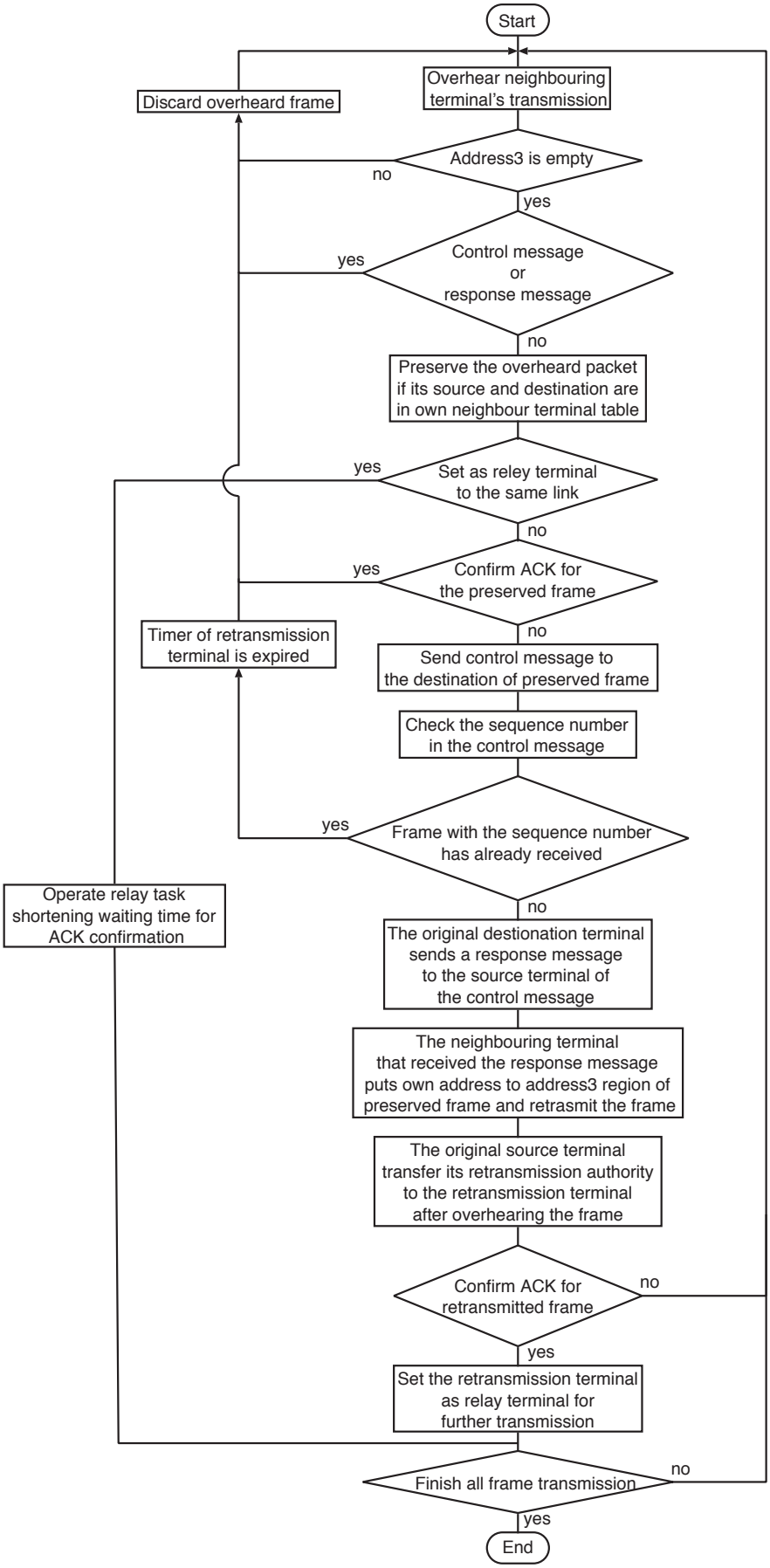
Start
Overhear neighbouring terminal's transmission
Discard overheard frame
Address3 is empty
no
yes
Control message or response message
yes
no
Preserve the overheard packet if its source and destination are in own neighbour terminal table
Set as reley terminal to the same link
yes
no
Confirm ACK for the preserved frame
yes
no
Timer of retransmission terminal is expired
Send control message to the destination of preserved frame
Check the sequence number in the control message
Frame with the sequence number has already received
yes
no
Operate relay task shortening waiting time for ACK confirmation
The original destionation terminal sends a response message to the source terminal of the control message
The neighbouring terminal that received the response message puts own address to address3 region of preserved frame and retrasmit the frame
The original source terminal transfer its retransmission authority to the retransmission terminal after overhearing the frame
Confirm ACK for retransmitted frame
no
yes
Set the retransmission terminal as relay terminal for further transmission
Finish all frame transmission
no
yes
End

図 3.7 フローチャート

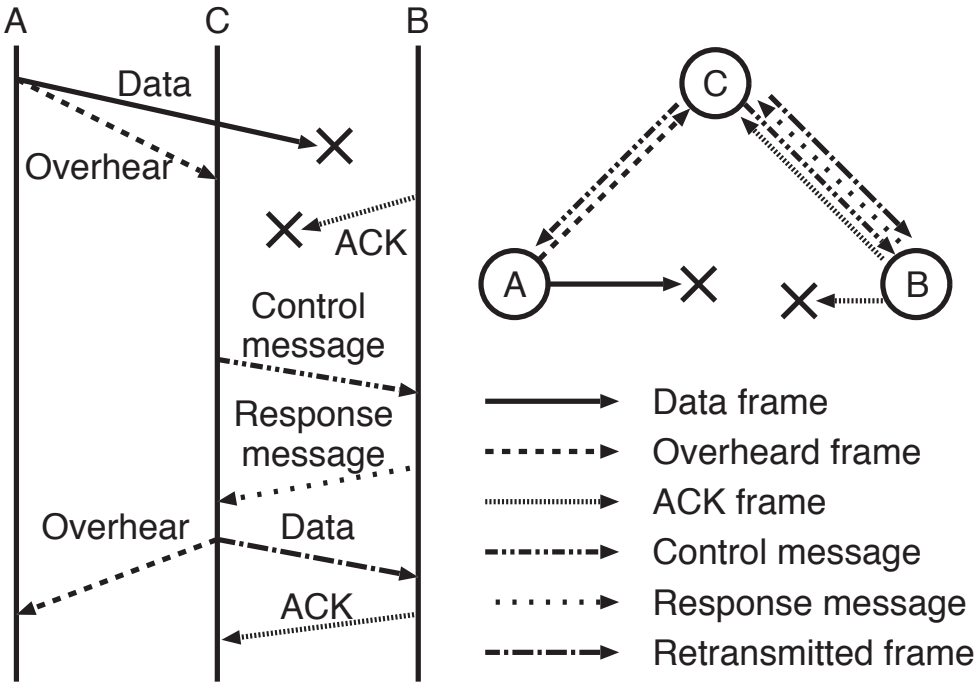

図 3.8 提案手法の動作例

ら ACK が返送されることを確認する．このとき，端末 C による再送フレームは端末 A も同時に漏れ聞くことが可能であり，端末 A はこのフレームによって再送の権限が端末 C へ委譲されたと認識する．

提案手法では，前述のような制御を行い，ACK 確認待ち時間を減少させることで，通信遅延の増加を低減した再送制御を行い，通信効率の向上を実現している．更に，近傍端末の中から優先的に再送制御を行う中継端末を選択することによって，複数の端末が再送可能な状況下での制御メッセージの衝突を抑制することが可能となる．また，中継端末として設定された近傍端末が自律的にフレーム損失の連続性を判断することで，不必要な再送フレームの生成を抑制し，フレーム損失の連続性に適応した再送制御を実現している．

3.4 シミュレーション評価

3.4.1 シミュレーション環境

本節では，コンピュータシミュレーションを用いて，通常の IEEE 802.11，従来手法，提案手法について性能評価を行う．シミュレーション条件は以下のとおりである．QualNet[37] を用いて，1,000m 四方の領域に 100 端末をランダムに配置する．各端末の移動速度は 0m/s〜10m/s の間でランダムに設定され，ウェイティングタイムを 20 秒とする．無線通信方式には，IEEE 802.11b を用いるが，RTS/CTS は

利用しない．有効電波半径は 100m とする．ルーチングプロトコルには AODV を利用し，Hello メッセージの送信間隔は 3 秒に設定する．各端末は，あて先端末をランダムに決定し，500Byte のデータを送信する．この際，通信セッションの平均生起間隔を 5 秒から 30 秒まで 5 秒間隔で変化させる．トランスポートプロトコルには，TCP と UDP を用い，それぞれのプロトコルでの特性を評価する．シミュレーション時間は 20 分とする．

評価指標には，IEEE 802.11 における ACK フレームのタイムアウト回数，リンク破損通知回数，TCP による再送回数，パケット送信成功率，エンドエンド間のスループット及び通信遅延を用いる．

ACK フレームのタイムアウト回数は，IEEE 802.11 において，送信データフレームに対する ACK フレームを確認することができず，タイムアウトとなった回数を示している．ACK タイムアウトが発生した場合，送信元端末はあて先端末に対し，フレーム再送を行う．そのため，タイムアウト回数を用いることで，リンクでの再送回数を評価することができる．また，一定回数データフレーム再送が連続して行われ，再送上限回数（SRL：Short Retry Limit）に達した場合，リンクが破損したと認識し，上位層に通知される．リンク破損通知回数は，前述したデータリンク層での再送回数が，IEEE 802.11 で規定されている再送上限に達した回数を表している．そのため，リンク破損通知回数を用いることで，データリンク層での通信の安定性を評価することができる．TCP による再送回数は，リンクは損が発生し，経路再構築後に TCP によって行われた再送回数を示している．この値を用いることで，経路の安定性，再送制御の効果を評価する．また，提案手法では，再送制御を変更することで，再送効率が変化するため，通信効率に影響を及ぼす．そのため，更なる評価指標として，セッション当たりのエンドエンド間におけるパケット送信成功率，スループット，通信遅延も評価指標として用いる．

3.4.2 UDP での評価結果

図 3.9〜図 3.13 に，トランスポートプロトコルとして UDP を利用した場合の評価結果を示す．

図 3.9 より，提案手法では，通常の IEEE 802.11 や従来手法と比較して，ACK フレームのタイムアウト回数が減少していることが分かる．これは，提案手法では特定の中継端末が優先的に再送制御を行うため，ランダムタイマを用いて再送端末を毎回決定する従来手法よりも再送効率が向上しているためである．また，同様の理由により，通信負荷によらず，常に一定の効果を得ることができている．図 3.10 より，提案手法，従来手法ともに一般的な IEEE 802.11 と比較して，リンク破損通知回数の低減を実現している．しかし，提案手法では，従来手法と比較してより顕著な改善効果が見られる．これは，提案手法の中継ノードを用いた優先

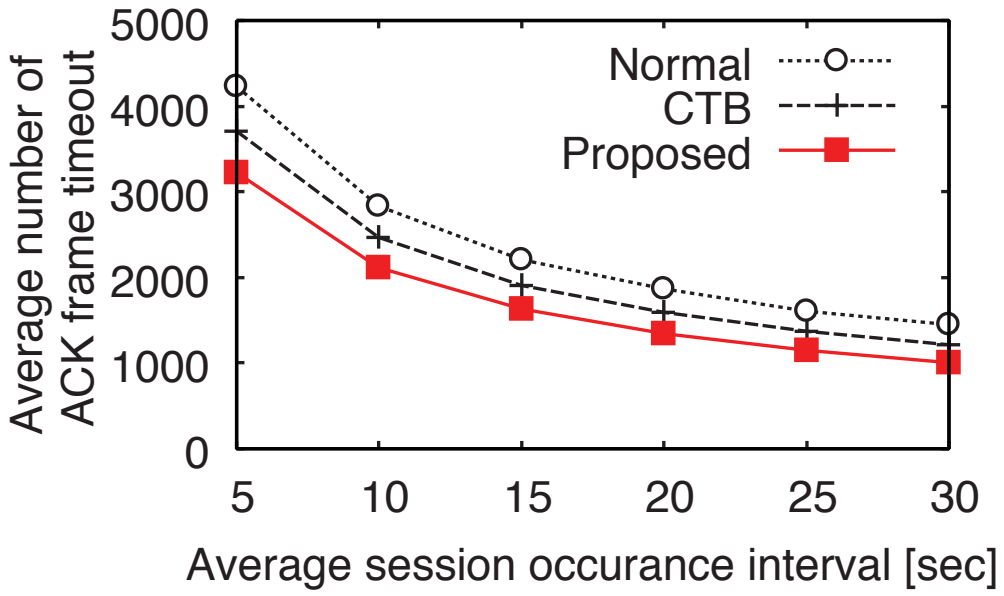

図 3.9 UPD 通信下での ACK フレームタイムアウト回数 〈発表文献 3.2〉

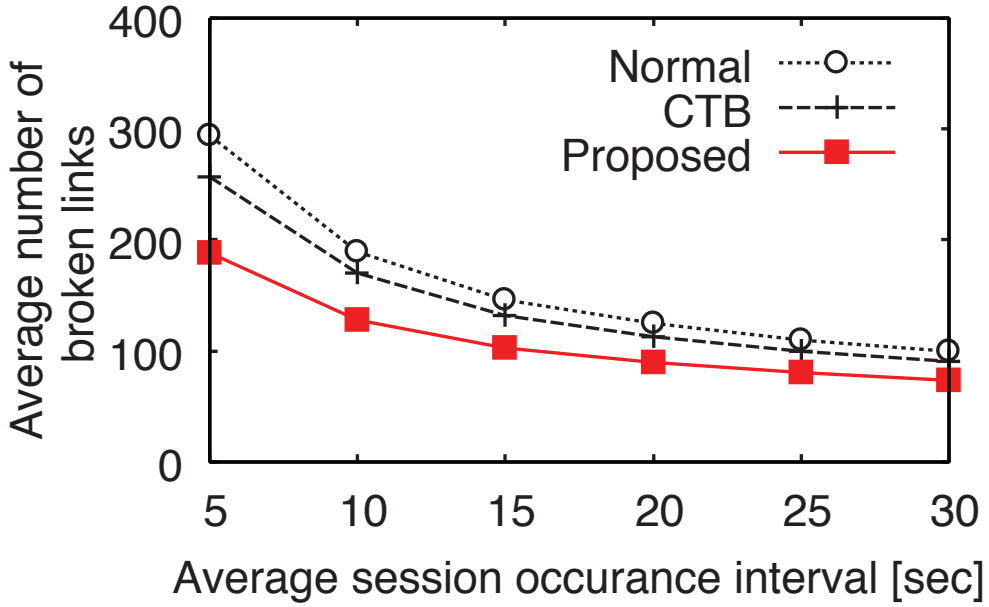

図 3.10 UDP 通信化でのリンク破損通知回数 〈発表文献 3.2〉

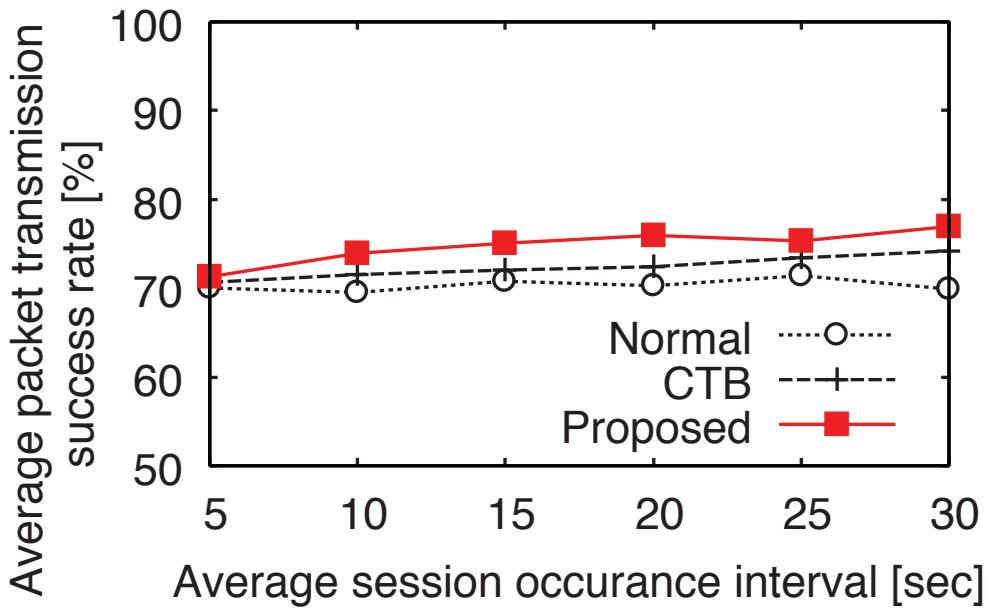

図 3.11 UPD 通信下でのパケット送信成功率 〈発表文献 3.2〉

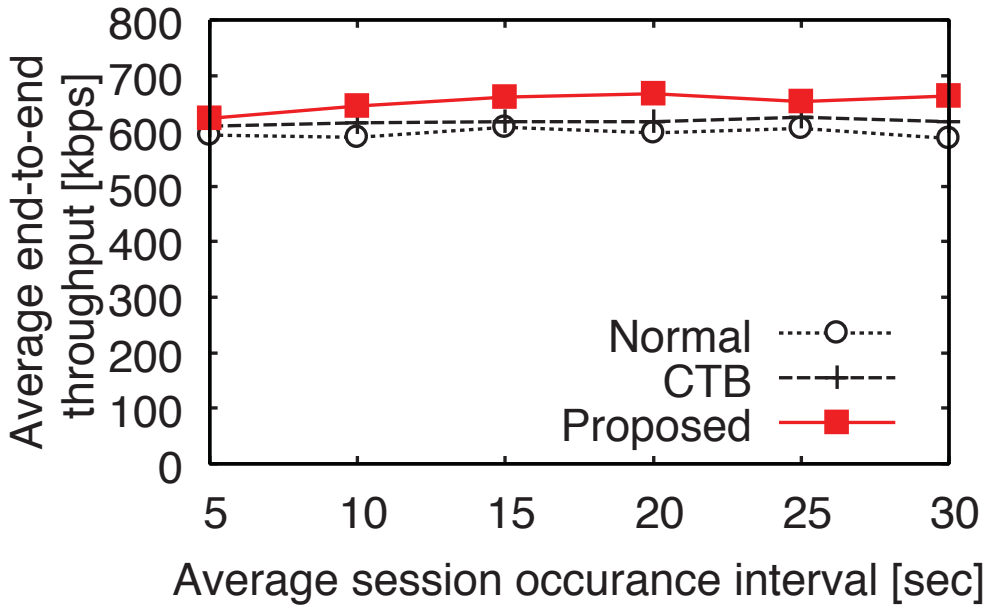

図 3.12 UDP 通信化でのスループット 〈発表文献 3.2〉

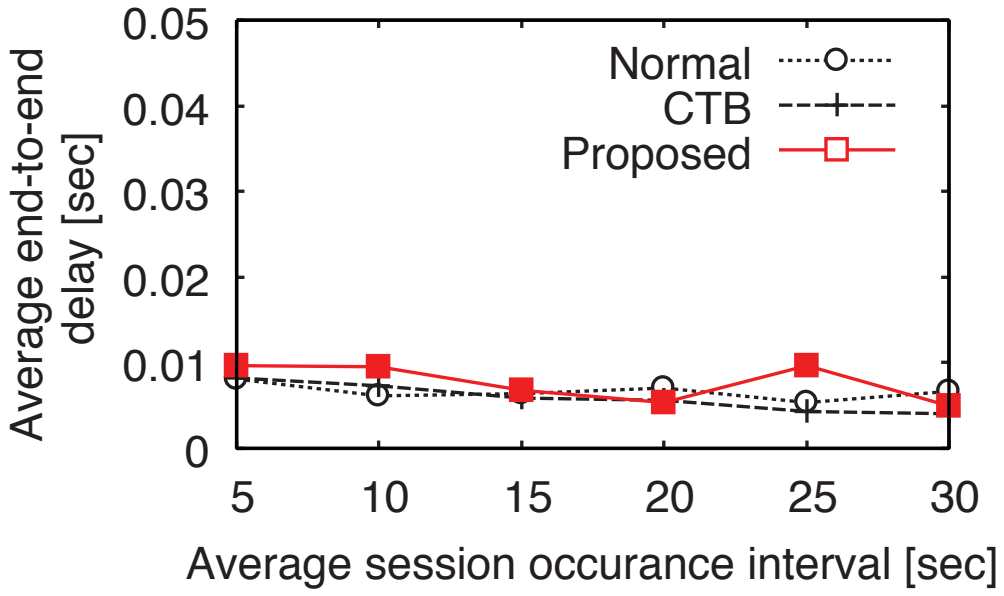

図 3.13 UDP 通信化での通信遅延 〈発表文献 3.2〉

的な連続再送によって，連続して損失が発生するような状況下でより効率的な再送制御が可能になったためである．以上のことから，UDP 通信下では，提案手法による再送効率向上が実現されたと結論できる．

再送制御導入による通信効率への影響を評価するため，図 3.11 〜図 3.13 に，エンドエンド間におけるパケット送信成功率，スループット，遅延の評価結果を示す．図 3.11 より，提案手法によるパケット送信成功率が向上していることが分かる．これは，提案手法による再送効率向上によって，伝送途中で発生した損失を回復することが可能になり，パケット到達率が向上したためである．また，パケット送信成功率の向上によって，エンドエンド間のスループットが向上していることが，図 3.12 より分かる．このように，提案手法による再送効率の改善が全体的な通信効率の向上に寄与していると考えられる．一方，図 3.13 より，提案手法や従来手法では，エンドエンド間の遅延が増加していることが分かる．これは，再送制御によって通常は 1 ホップで伝送可能なリンクの途中に中継端末が介在することで，フレーム損失時にはホップ数が増加することとなり，経路長が長くなるためであると考えられる．しかし，通常の IEEE 802.11 と比較して，遅延の増加がわずかであるため，通信性能に与える影響は小さいと考えられる．

3.4.3 TCP での評価結果

図 3.14〜図 3.19 に，トランスポートプロトコルとして TCP を利用した場合の評価結果を示す．

図 3.14，図 3.15 より，トランスポートプロトコルに TCP を用いた場合でも，

ACK フレームのタイムアウト回数，リンク破損通知回数ともに，低減することができている．従来手法においても，一定の効果が得られているが，提案手法ではより顕著な効果を見ることができる．このような特徴は，UDP 通信下での評価結果と同様であるが，提案手法がデータリンク層で再送制御を行っているため，トランスポートプロトコルによらず，同様の効果が得られるためである．図 3.16 は，TCP によるエンドエンド間の再送が行われた回数を示している．提案手法，従来手法ともに通常の IEEE 802.11 と比較して，再送回数を大きく低減することができている．これは，データリンク層での再送制御によって，損失が発生した場合でもリンクの信頼性を向上させることが可能なため，結果的にエンドエンド間の信頼性向上に繋がり，再送回数の低減を実現しているためである．

再送制御導入による通信効率への影響を評価するため，図 3.17 〜図 3.19 に，エンドエンド間におけるパケット送信成功率，スループット，遅延の評価結果を示す．図 3.17 より，従来手法と比較して，より大きなパケット送信成功率の改善が見られる．これは，図 3.15，図 3.16 で示したように，データリンク層での再送効率を向上させたことにより，トランスポート層で必要な再送回数が低減され，パケットの到達率が改善されたためである．また，図 3.18，図 3.19 から，従来手法と比較し，提案手法ではエンドエンド間のスループット及び遅延の改善効果が高いことが分かる．これは，データリンク層での再送効率の改善により，伝送途中で損失するパケットが減少し，TCP の通信制御効率が向上したためである．また，提案手法では，UDP 通信下ではわずかに遅延が増加していたが，TCP 通信下では遅延の低減を実現している．これは，TCP では，損失が発生した場合にエンドエ

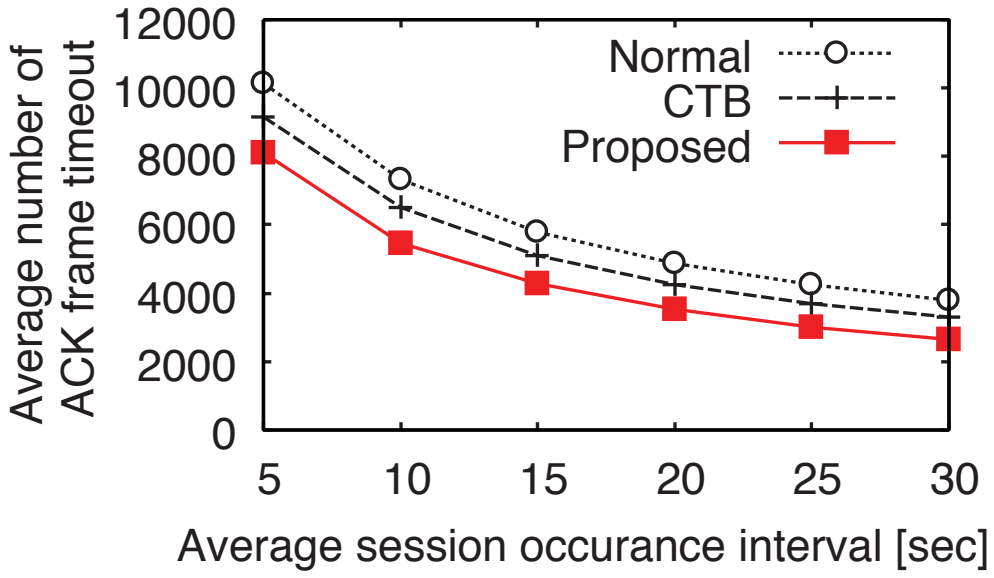

図 3.14 TCP 通信下での ACK フレームタイムアウト回数 〈発表文献 3.2〉

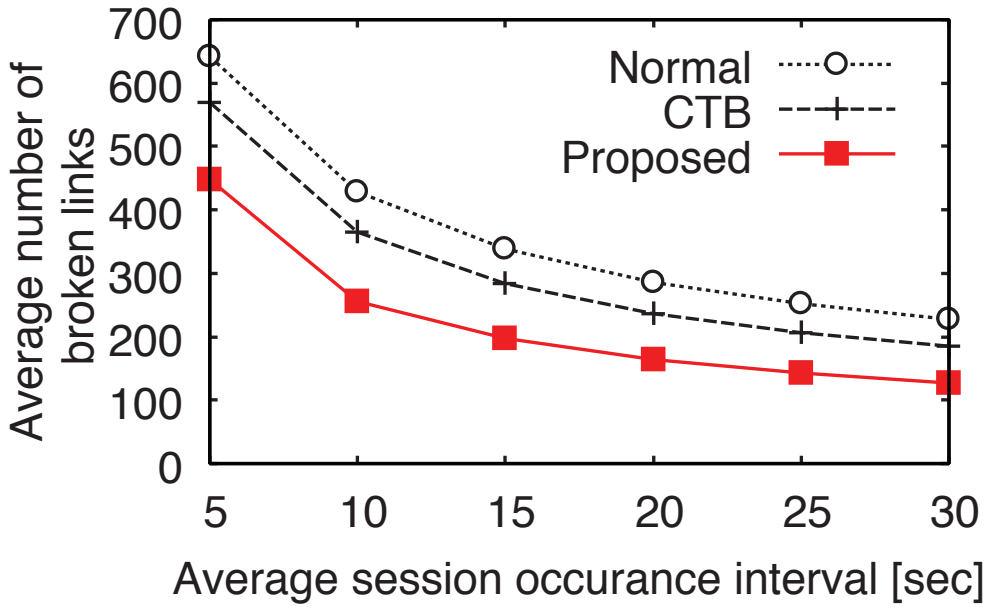

図 3.15 TCP 通信化でのリンク破損通知回数 〈発表文献 3.2〉

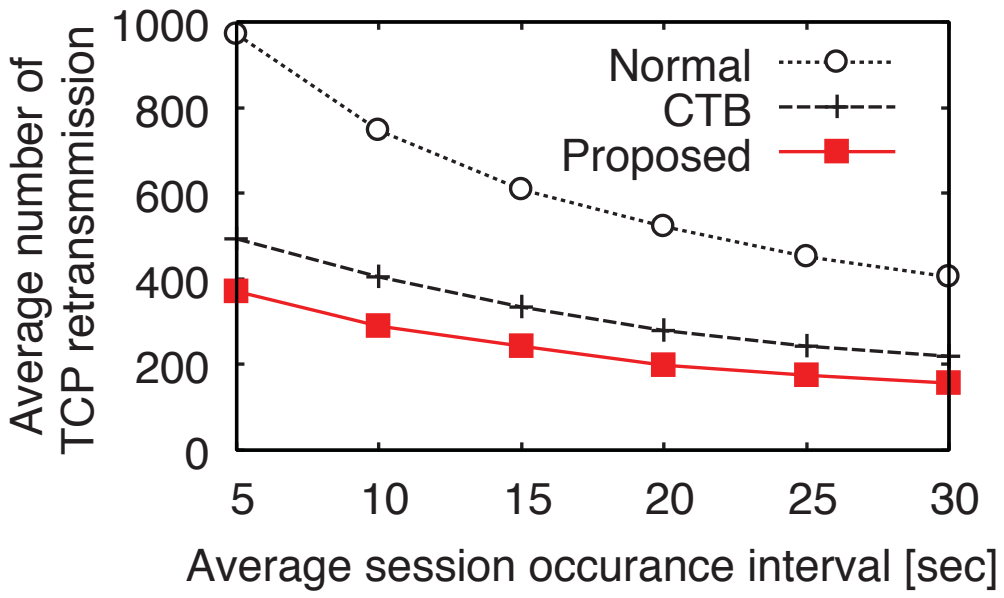

図 3.16 TCP 通信化での TCP による再送回数 〈発表文献 3.2〉

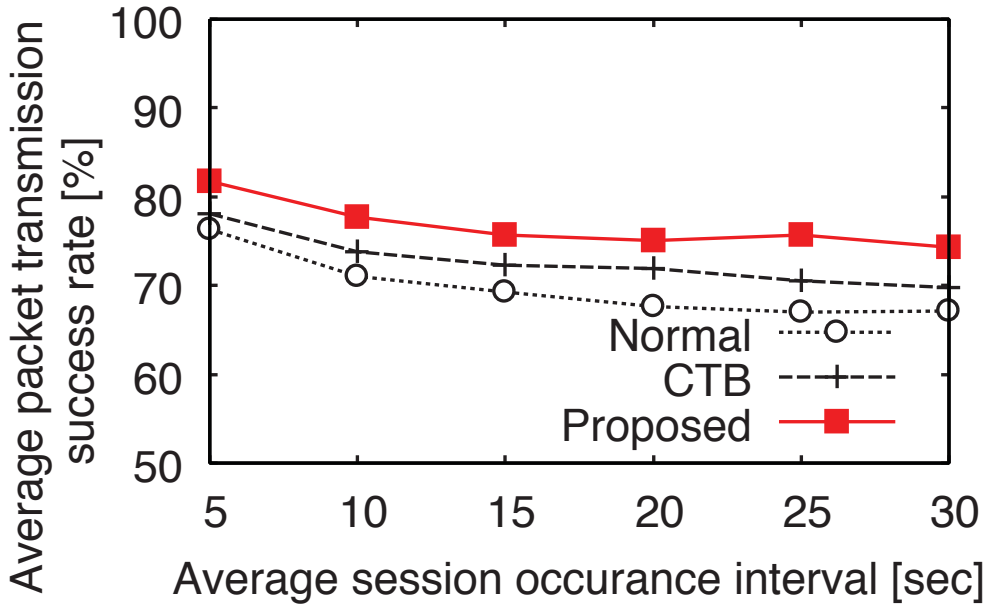

図 3.17 TCP 通信下でのパケット送信成功率 〈発表文献 3.2〉

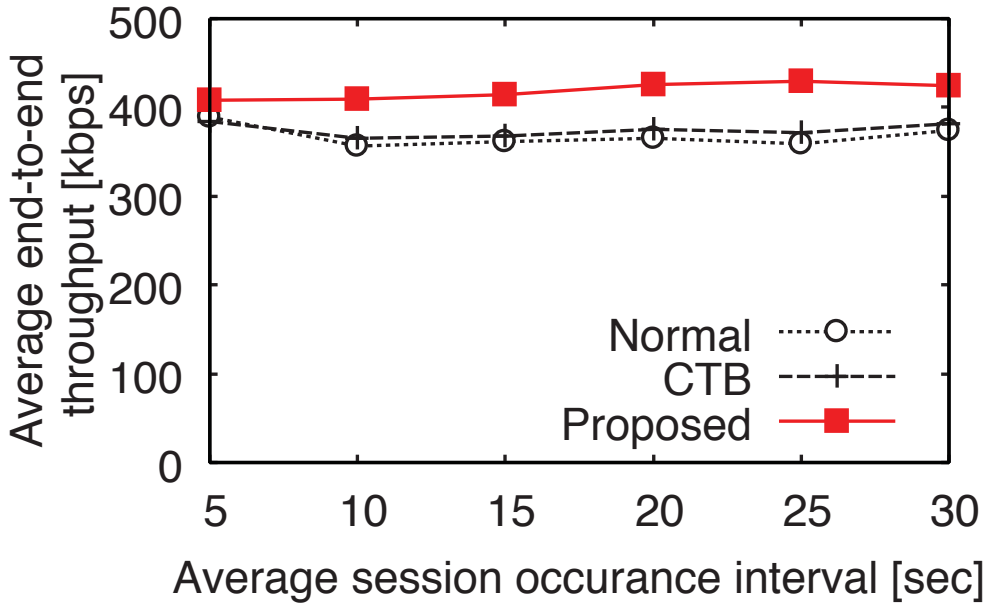

図 3.18 TCP 通信化でのスループット 〈発表文献 3.2〉

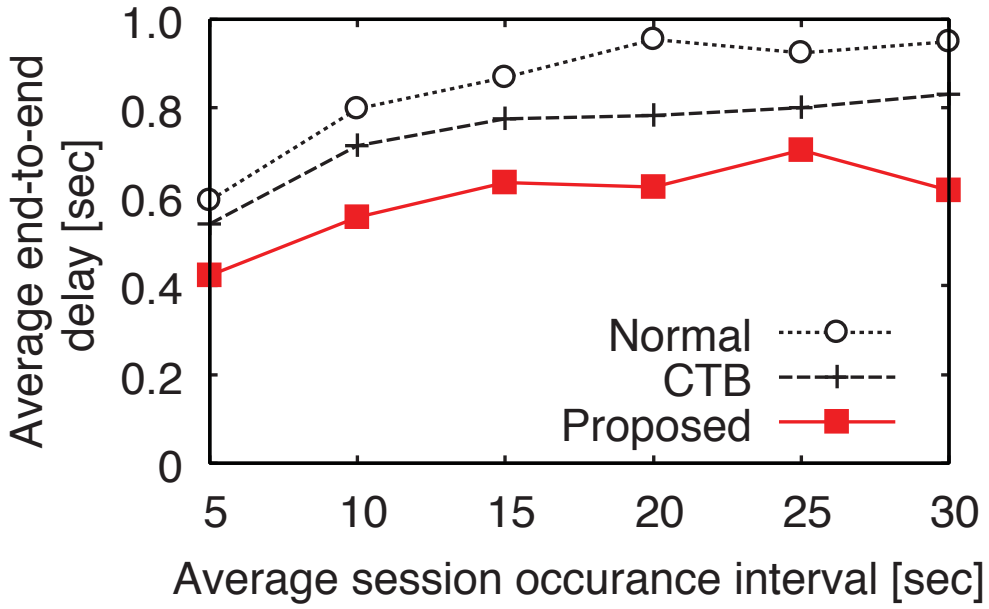

図 3.19 TCP 通信化での通信遅延 〈発表文献 3.2〉

ンド間での再送制御が行われ，遅延増加の要因となるが，提案手法による再送制御によって適切にリンク上での損失を回復を行うことで，エンドエンド間の損失を回避することができるためである．

3.5 むすび

本章では，無線アドホックネットワークにおける端末の高い移動性や無線通信の特徴に起因する高いデータ損失率を改善するため，データリンク層における再送制御手法を提案した．また，コンピュータシミュレーションによる評価によって，通常の通信及び従来手法と比較して，より効率的な再送制御を実現し，エンドエンド間での通信効率の改善を達成した．

提案手法では，無線通信の同報性を利用することによって，フレーム損失時にリンクの近傍端末が再送制御を行うことで損失を回復している．更に，特定の端末を中継端末として設定することで，従来手法で問題となっていた連続したフレーム損失時の再送性能低下を克服した．また，提案手法では，損失発生の初期の段階で回復動作を行うため，トランスポート層などの上位層の制御に影響を与えることなく，通信効率の改善を実現した．つまり，各リンクの信頼性を向上させることで，経路全体の信頼性，通信効率の向上を達成することができた．

第4章　適応形トランスポートプロトコル

4.1　まえがき

無線通信を行う無線アドホックネットワークでは，電波干渉や端末の高い移動性の影響によって，有線通信と比較してセグメント損失が頻発する．トランスポートプロトコルとしてTCP Renoを用いる場合，セグメント損失を輻輳発生のトリガとして利用しているため，輻輳回避によって送信速度が減少する．しかし，アドホックネットワークでは輻輳以外に起因するセグメント損失が頻発するため，不必要に送信速度が低下することとなり，余剰帯域があるにも関わらず通信効率が低下する問題がある．

一方，TCP Renoの改良形のプロトコルであるTCP Vegas [39] では，往復遅延時間（RTT：Round trip time）の変化によって残留セグメント数を予測し，通信状態を把握したウィンドウ制御を行う．しかし，通信状態が随時変動する無線アドホックネットワークでは，RTTが特定の値で安定せずに大きく変動することや，経路変更によってRTTが大きく変化することから，必ずしも正確な残留セグメント数予測とならない場合がある．そのため，通信環境に応じた適切な制御を行うことが困難となり，性能低下を引き起こす可能性がある．

これらに関連して，TCP Renoのウィンドウサイズの減少幅を帯域推定によって決定するTCP Westwood [38] やウィンドウサイズの上限値を帯域推定によって制限する手法 [40]，TCP Vegasのウィンドウ制御の閾値を動的に変化させるTCP Vegas-A [41] が提案されている．しかし，通信状態が随時変動する無線アドホックネットワークでは，正確な帯域推定は困難である．更に，TCP Vegas-Aでは大幅にRTTが変化した場合などに対応することが困難である．また，無線アドホックネットワークでは限られたネットワークや端末資源を有効に活用するため，端末間，及びプロトコル間での公平性を考慮した通信を行う必要がある．

本章では，無線アドホックネットワークにおけるこれらの問題に対応したトランスポートプロトコルを提案する．提案手法の特徴は，TCP Vegasを基本とし，RTTの相対変化量を利用して経路上の通信状態を把握し，ウィンドウ制御を行う点である．また，ネットワーク及び端末資源を有効に活用するため，セグメント送信を必要最小限に抑える再送制御，状態遷移を実現している．更に，コンピュータシミュレーションを用いて，提案手法の有効性評価を行う．

4.2　無線アドホックネットワークにおけるTCP通信

4.2.1　TCP Reno

TCP Reno は **2.3** で述べたように，インターネットなどで広く普及しているトランスポートプロトコルの一つである．TCP Reno は TCP 最初のバージョンである TCP Tahoe を改良したもので，以下のような特徴を有している．

1. コントロールセグメントを用いた状態，コネクション管理
2. 確認番号を用いて通信の可否を確認
3. シーケンス番号による到着順序保証
4. スライディングウィンドウを用いたデータ転送
5. ウィンドウサイズによって送信速度を制御

このような TCP Reno の特徴から多くの通信で TCP Reno が用いられ，デファクトスタンダードとしてインターネットでは用いられている．また，TCP Reno を改良した TCP New Reno も普及しているが，本節ではより基本的な TCP Reno について述べる．

（1）コネクション管理

TCP Reno では，アプリケーションが下位レイヤの状態を意識することなく，仮想的な通信路へデータを転送することができるようにセッション制御を行っている．そのため，TCP Reno ではコネクション管理のために，自身の状態をコントロールセグメントの送受信によって管理している．図 4.1 に TCP Reno における状態遷移図を示す．図中の長方形は各状態を示しており，状態遷移は矢印で表す．また，図中の TCB は Transmission Control Block と呼ばれる管理テーブルである．

状態遷移の例として，以下にコネクションの確立と切断の手順を示す．TCP におけるコネクション確立は次の手順を用いて行われる．

コネクション確立

コネクション確立時の動作を図 4.2 に示す．また，確立手順を以下に示す．コネクション確立時の動作は一般的に 3–way ハンドシェイクと呼ばれている．

1. 送信元端末は，コネクションを確立したい相手に向けて SYN フラグを立てたセグメントを送信し，自身の状態を SYN SENT へ遷移させる．
2. SYN フラグの立ったセグメントを受信した端末は，応答として SYN+ACK フラグを立てたセグメントを返答し，SYN RCVD へ遷移する．

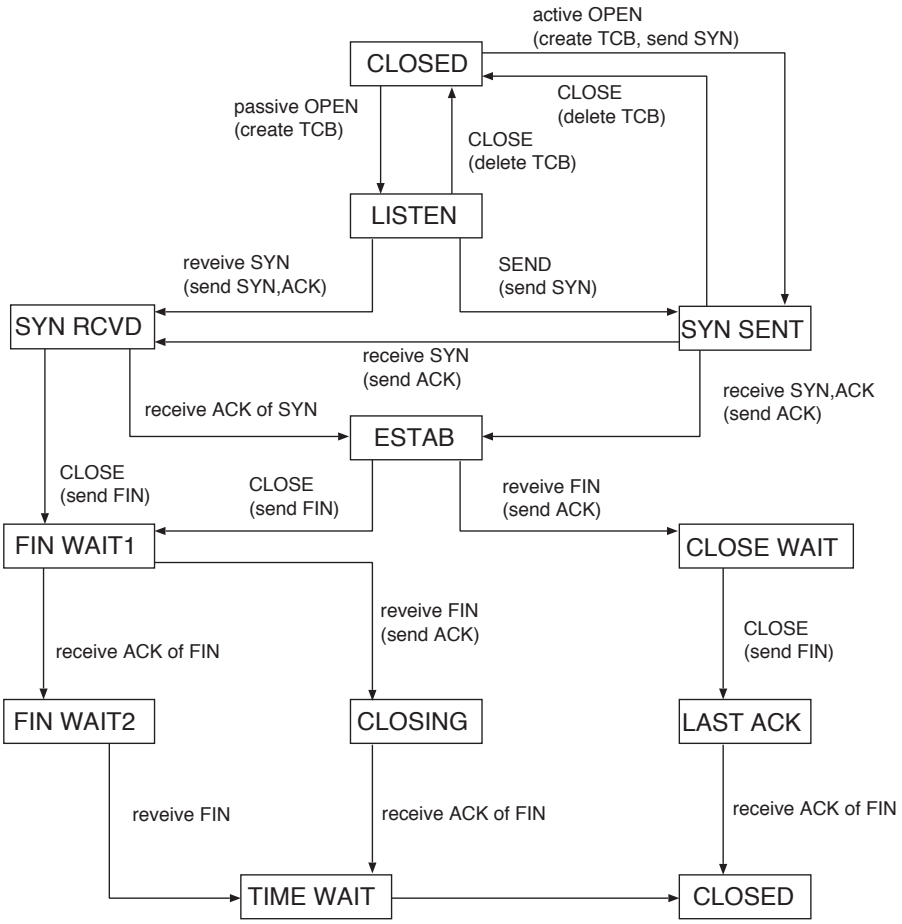

図 4.1 状態遷移図

3. 送信元端末は，自身が送信した SYN フラグの立ったセグメントに対する返答として SYN+ACK フラグの立ったセグメントを受信すると，ACK を返し，ESTAB へ遷移する．

4. あて先端末端末は，SYN+ACK フラグを立てたセグメントに対する ACK を受信すると，ESTAB へと状態を遷移し，コネクションを確立する

コネクション切断

1. データ送信が終了すると，送信元端末が FIN フラグを立てたセグメントをあて先端末へ送信し，FIN WAIT1 へ遷移する．

2. FIN フラグが立ったセグメントをあて先端末が受信すると，ACK を返答して CLOSE WAIT へ遷移する．

3. 返答された ACK を受信した送信元端末は，状態を TIME WAIT2 へ遷移させる．

4. これ以上通信する必要がない場合には，あて先端末から FIN フラグを立てたセグメントを送信元端末へ送信し，LAST ACK へ遷移する．

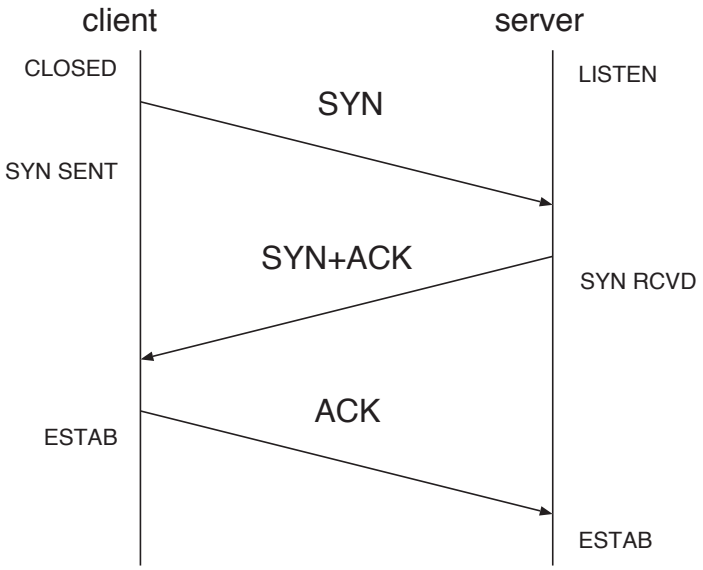

図 4.2 コネクション確立

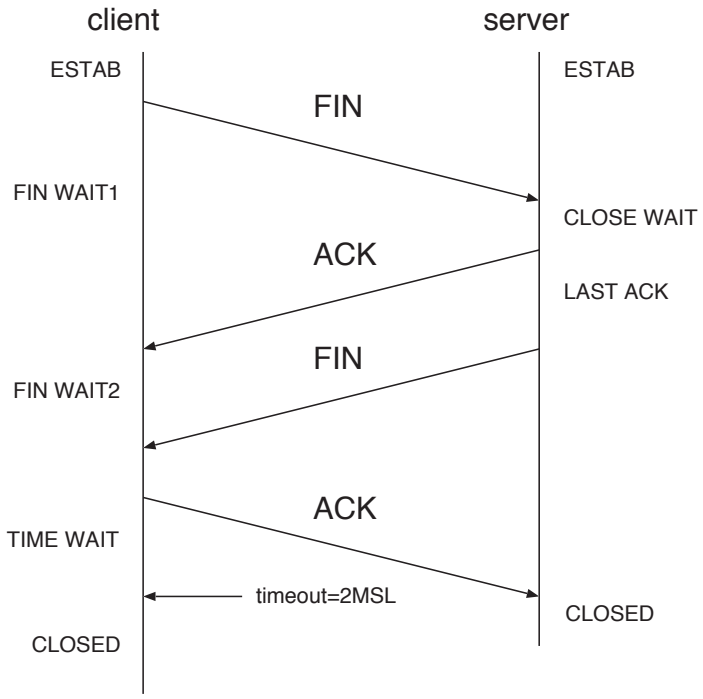

図 4.3 コネクション切断

5. 送信元端末は，あて先端末からFINフラグの立ったセグメントを受信すると，ACKを返してTIME WAITへ遷移し，それを受信したあて先端末はコネクションを切断する．

（2）ウィンドウを用いたセグメント送信

TCP Renoではウィンドウ制御を利用した送信速度制御が行われている．ウィンドウ制御を用いない通信では，セグメントを一つ送るごとにACKを待つ必要があるため，単位時間あたりに送出可能なセグメント数が一定であり，通信効率が低下する．そこで，ACKを待たずに次のセグメントを送るためにウィンドウを用いた通信が行われている．TCPのウィンドウ制御では，輻輳ウィンドウサイズ（cwnd：Congestion Window Size）に基づいた送信速度制御が行われる．ここで，輻輳ウィンドウサイズとはACKを待たずに送信できるデータの大きさを示している．輻輳ウィンドウ制御を用いた通信では，cwnd分のデータはACKを待たずに送信でき，ACKが返送されるたびに送信元端末はウィンドウを進める．つまり，送信できるデータの範囲をずらすことで次のセグメント送信を行う．この仕組みをスライディングウィンドウといい，動作例を図4.4に示す．

ウィンドウ制御を用いた通信を行う際に発生する問題点には，ACKの不着，送信元端末からのセグメントの不着などがある．そのため，ACKを送信元端末が受け取れなかった場合には，送信元端末はセグメント送信がタイムアウトになる前までに受信した最後のACKの情報を用いて次のセグメント送信を行う．動作例を図4.5に示す．また，送信元端末からのセグメントがあて先端末に届かなかった場合には，届かなかったセグメントを要求するACKを返信し，送信元が同じACKを3回連続して受信した場合，伝送途中でセグメント損失が発生したと認識し，セグメントの再送を行う．この際に，すでに受信していたセグメントを再送する必要はなく，損失したセグメントのみの再送を行う．動作例を図4.6に示す．更に，送信元端末からのセグメントをあて先端末で処理しきれない場合には，バッファ溢れなどによってセグメント損失発生の要因となる．そこで，TCP Renoでは，あて先端末の処理能力を送信元端末へ通知するため，受信側から送信元へ受信可能セグメントサイズを通知する仕組みが導入されている．これを広告ウィンドウサイズといい，TCPヘッダには広告ウィンドウサイズを格納するフィールドがあり，あて先端末のバッファが溢れそうになるとウィンドウサイズを減らすように送信元へ通知する．動作例を図4.7に示す．

TCP Renoにおけるウィンドウ制御はACKによって駆動され，ACKを受信するたびに輻輳ウィンドウサイズを更新することによって行われる．通信開始時には小さな輻輳ウィンドウサイズ（IW：Initial Window）に設定し，徐々に輻輳ウィンドウサイズを増加させるスロースタート（Slow Start）を用いることで，ネットワーク容量に合わせた送信速度を決定する．TCP Renoにおけるウィンドウ制御

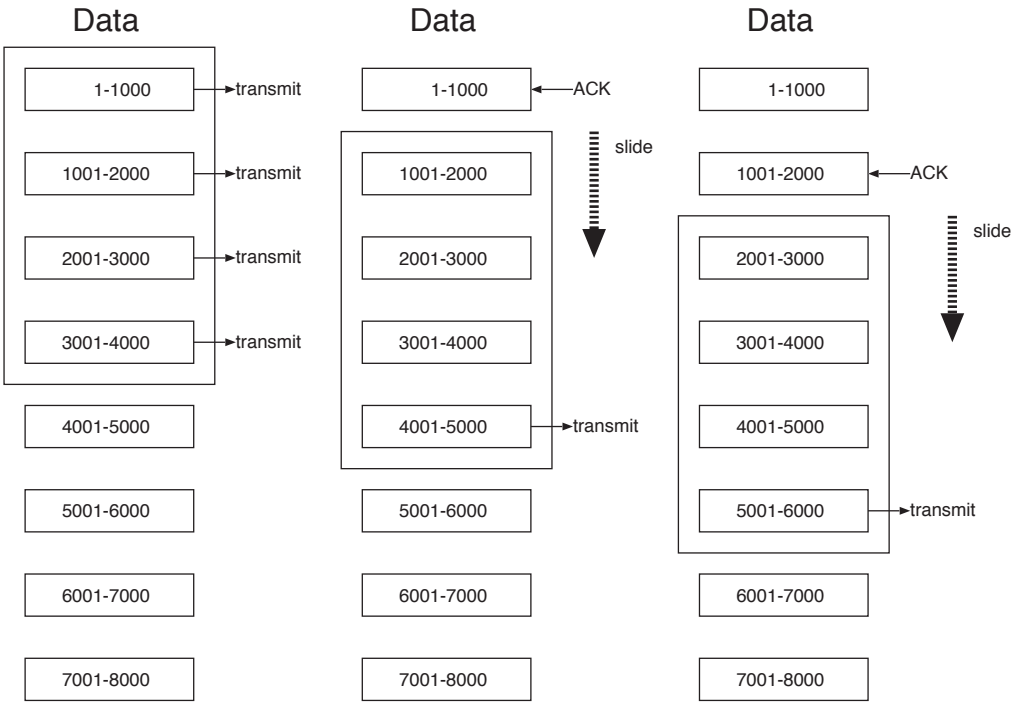

図 4.4 スライディングウィンドウ

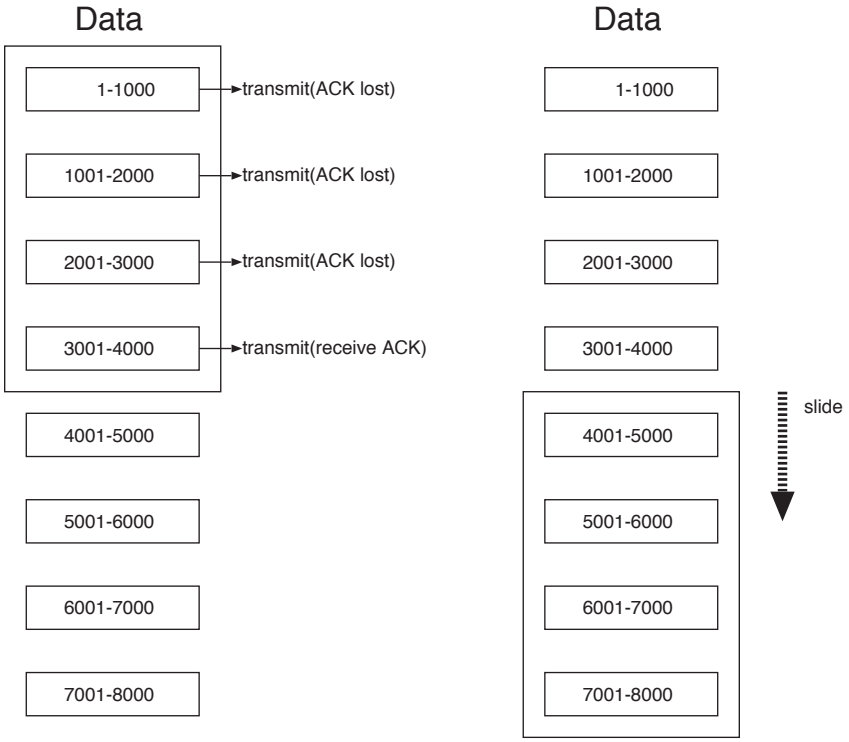

図 4.5 ACK 不着の場合

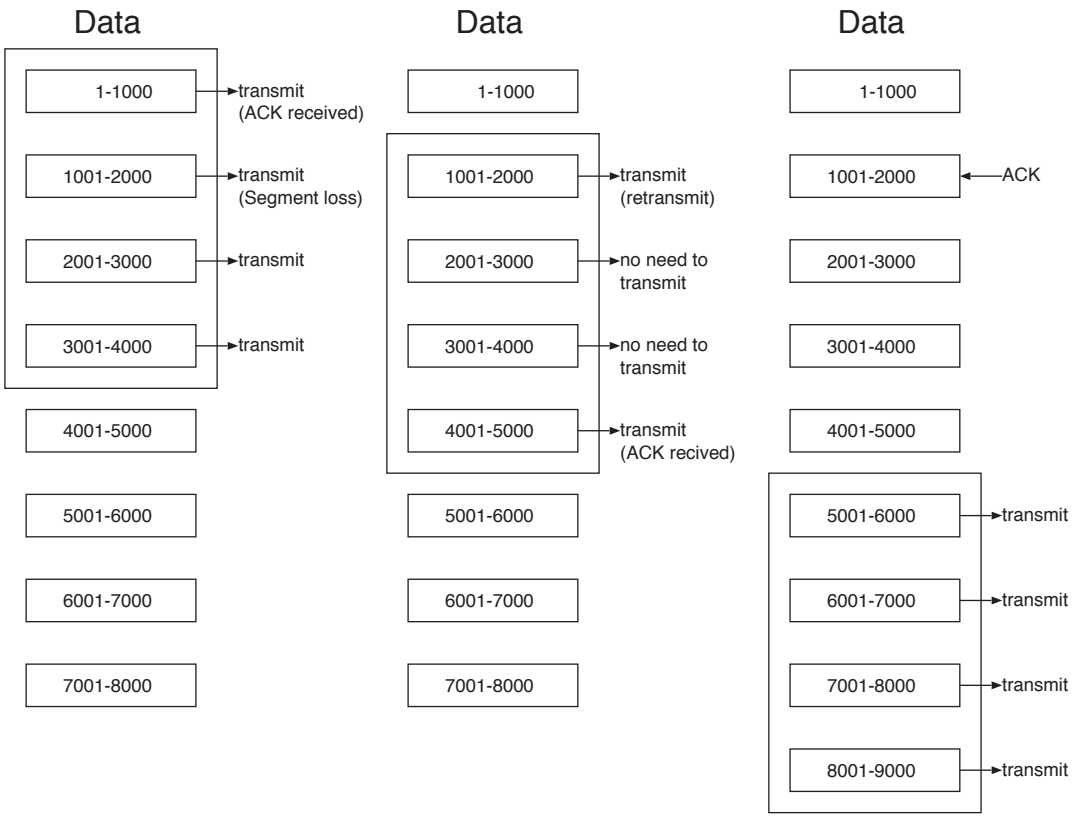

図 4.6 セグメント損失が発生した場合

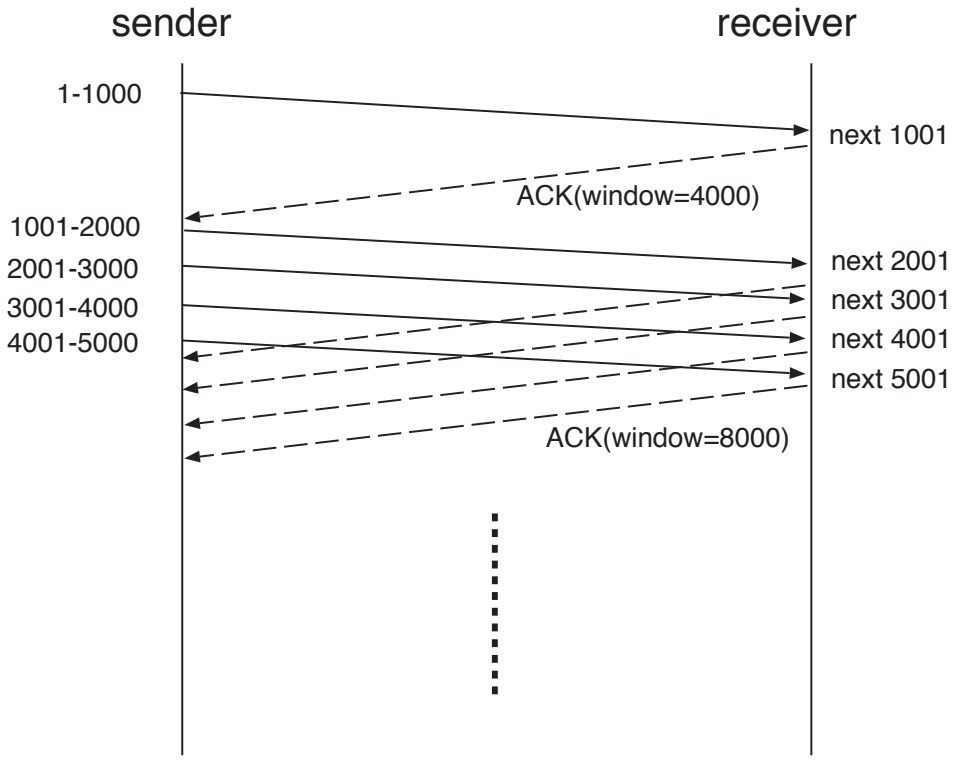

図 4.7 あて先端末からの広告ウィンドウサイズ通知

は ACK 受信ごとに以下の式 4.1 を用いて行われる．

$$c_{k+1} = \begin{cases} c_k + M_{\text{seg}} \ (c_k \leq s_{\text{th}}) \\ c_k + \dfrac{{M_{\text{seg}}}^2}{c_k} \ (c_k > s_{\text{th}}) \end{cases} \tag{4.1}$$

ここで，M_{seg} は送信可能最大セグメントサイズ，c は送信側ウィンドウサイズである．また，s_{th} は式 3.1 を用いたスロースタートから輻輳回避フェーズ（congestion avoidance phase）へと移行する際のしきい値で，セグメント損失が起こるたびに以下の式 4.2 を用いて更新される．

$$s_{\text{th}} = \max\left(\frac{\text{FS}}{2},\ 2M_{\text{seg}}\right) \tag{4.2}$$

ここで，FS（Flight Size）は送信されたがまだ ACK が返ってきていないデータ量を示す．TCP Reno ではあて先端末側から通知される広告ウィンドウサイズと送信側の輻輳ウィンドウサイズを比較し，より小さなウィンドウサイズを送信ウィンドウサイズとしてデータ転送を行う．

通信途中でセグメント損失を検出すると，TCP Reno では輻輳が発生したと認識し，輻輳回避を行う．セグメント損失の検出には連続した三つの重複 ACK を用いる．三つの重複 ACK を受信した場合にはセグメント損失が発生したと認識し，再送タイマを待たずに早期再送（Fast Retransmission）を行う．重複 ACK を受信できるということはあて先までセグメントが届いていることでもあるので，早期再送を行った後は早期回復（Fast Recovery）を用いてセグメントの送信を続ける．通常，早期再送と早期回復は組として以下のように行われる．

1. 三つの重複 ACK を受信するとセグメント損失が発生したと認識し，式 4.2 を用いて s_{th} を設定する．

2. 損失したセグメントを再送した後，ネットワーク内に残っていたセグメント分だけ c を増加させるため，$c = s_{\text{th}} + 3M_{\text{seg}}$ とする．

3. 更に追加の重複 ACK を受信すると，c を各 ACK を受信するごとに M_{seg} 増加させる．これはネットワーク内に残っていたセグメント，つまりは FS 分のセグメントを考慮するためである．

4. あて先端末の広告ウィンドウサイズである r と新しい c を比べ，送信できるセグメントがあれば送信する．

5. 新しい ACK，つまり再送したセグメントに対する ACK を受信すると，c を s_{th} に再設定する．

また，送信したセグメントがタイムアウトした場合には，輻輳ウィンドウサイズを IW に設定し，再びスロースタートを行うことによって輻輳回避を行う．

このように，TCP Reno はセグメント損失の発生により輻輳を検知し，ウィンドウサイズを減少させることで送信速度を低下させ，輻輳回避を行う．一方，アドホックネットワークでは，電波干渉や障害物などの影響によるセグメント損失が多発するが，このような理由による損失も輻輳と誤認される．そのため，ネットワークに余剰帯域が存在している場合でも送信ウィンドウサイズが十分に増加せず，帯域が十分に活用されない可能性がある．

4.2.2 TCP Vegas

TCP Reno は，セグメント損失を利用して輻輳検知を行っているため，セグメント損失の発生直後に輻輳回避が行われることで，輻輳以外によってセグメント損失が発生した場合には，輻輳ウィンドウサイズが必要以上に小さくなってしまいスループットが低下する問題がある．そこで，TCP Vegas[39] では，RTT と送信データ量を基に，期待されるスループットと実際のスループットを計算し，その差をウィンドウ制御に利用する．TCP Vegas の輻輳ウィンドウサイズは以下の式 4.3 を用いて更新される．

$$c_{k+1} = \begin{cases} c_k + M_{\mathrm{seg}} & (\delta_k \leq \alpha) \\ c_k & (\alpha < \delta_k \leq \beta) \\ c_k - M_{\mathrm{seg}} & (\beta < \delta_k) \end{cases} \tag{4.3}$$

また，ここでは

$$\delta_k = R_{\min} \times \left(\frac{c_k}{R_{\min}} - \frac{c_k}{R_k} \right) \tag{4.4}$$

とする．ここで α，β は定数であり通常は $\alpha = M_{\mathrm{seg}}$，$\beta = 3M_{\mathrm{seg}}$ に設定されている．また，スロースタート時のウィンドウサイズの更新は以下の式 4.5 ように行われる．

$$c_{k+1} = c_k + \frac{M_{\mathrm{seg}}}{2} \tag{4.5}$$

TCP Vegas でのスロースタートは，δ_k が $\alpha < \delta_k \leq \beta$ を満足するまで継続される．これらのウィンドウ制御が理想的に動作した場合，送信ウィンドウサイズは一定値に固定され，セグメント損失が発生せずに安定したスループットを得ることができる．また，TCP Vegas の再送制御は TCP Reno と同様であるが，RTT が RTO (Retransmission Time Out) より大きい場合には，重複 ACK を受信すると三つ目の重複 ACK を待たずに再送を行う．

このように，TCP Vegas では RTT，及び輻輳ウィンドウサイズから，スループットの期待値と実測値を算出し，ネットワーク内の残留セグメント数を予測し，それを基にウィンドウ制御を行っている．しかし，ネットワークの状態が随時変化するアドホックネットワークでは，RTT が大きく変動するため，残留セグメント

数予測の基準となる最小 RTT を正確に把握することは困難である．そのため，正確な残留セグメント数を把握することができずに，適切な送信速度で通信を行うことができない場合がある．

4.2.3 TCP Westwood

TCP Westwood[38] では，セグメント損失発生時の輻輳ウィンドウサイズを帯域推定を用いて決定する．TCP Reno ではセグメント損失が発生するとウィンドウサイズを半分程度に減少させるが，余剰帯域がある場合にはネットワークの利用効率を低下させる要因となる．そこで，TCP Westwood ではセグメント損失時に ACK の到着間隔を用いて帯域推定を行い，輻輳ウィンドウサイズ再設定値を算出する．帯域推定は以下の式 4.6，式 4.7 を用いて行われる．

$$b_k = \frac{d_k}{t_k - t_{k-1}} \tag{4.6}$$

$$\bar{b}_k = \alpha_k \bar{b}_{k-1} + (1-\alpha_k)(\frac{b_k + b_{k-1}}{2}) \tag{4.7}$$

ここで t_k は ACK 受信時刻，t_{k-1} は一つ前の ACK 受信時刻，d_k は $t_k - t_{k-1}$ の間にあて先端末が受信したデータ量を示す．また，$\bar{b}_k$ は係数 α を用いて平滑化した帯域幅である．α は，推定帯域幅の急激な変動を抑制するためのローパスフィルタとしての役割をもち，式 4.8 にて定義される．

$$\alpha_k = \frac{2\tau - \Delta t_k}{2\tau + \Delta t_k} \tag{4.8}$$

ここで $\frac{1}{\tau}$ はローパスフィルタの周波数である．

セグメント損失が発生し，輻輳ウィンドウサイズを減少させる際には，以下の式 4.9 を用いて更新ウィンドウサイズを算出する．

$$c_k = \bar{b}_k \times R_{\min} \tag{4.9}$$

ここで算出したウィンドウサイズは，スロースタートと輻輳回避フェーズの閾値である s_{th} としても用いられる．

このように，TCP Westwood では帯域推定を用いてセグメント損失時の輻輳ウィンドウサイズ再設定値を算出しているが，経路の状態が不安定なアドホックネットワークでは必ずしも正確な帯域推定を行えるとは限らない．

4.2.4 輻輳ウィンドウサイズの上限値を制限する手法

文献 [40] では，輻輳ウィンドウサイズの上限値を帯域推定を用いて制限し，過剰な輻輳ウィンドウサイズの増加を抑制する手法が提案されている．また，本章では，文献 [40] で提案されている手法を TCP UB（Upper Bound）と呼ぶこととす

る．TCP Reno では，セグメント損失が発生するまで ACK を受信するたびに輻輳ウィンドウサイズを増加させるが，無線アドホックネットワークでは，干渉やバッファ溢れを引き起こす可能性がある．そこで，TCP UB では，TCP Westwood と同様の仕組みを用いて帯域推定を行い，それを基に輻輳ウィンドウサイズを制限する．輻輳ウィンドウサイズの上限値は以下の式 4.10，4.11 を用いて算出される．

$$R_{\mathrm{ratio}} = \frac{R_{\mathrm{ave}}}{R_{\mathrm{min}}} \tag{4.10}$$

$$c_{\mathrm{ub}} = \frac{b \times R_{\mathrm{min}}}{M_{\mathrm{seg}}} \times \left(\alpha + \frac{\beta}{R_{\mathrm{ratio}}} \right) \tag{4.11}$$

ここで R_{ave} は平均 RTT，c_{ub} は輻輳ウィンドウサイズの上限値，b は推定帯域幅，α，β は定数であり，それぞれ 1，3 に設定されている．輻輳ウィンドウサイズが上限値を超えた場合には，上限値へ輻輳ウィンドウサイズを再設定し，それ以外では通常の TCP Reno と同様のウィンドウ制御を行う．

このように，TCP UB では帯域推定を用いて輻輳ウィンドウサイズの上限値を制限しているが，TCP Westwood と同様に，無線アドホックネットワークでは必ずしも正確な推定帯域を算出できるとは限らない．また，輻輳ウィンドウサイズの上限値が必要以上に低く見積もられた場合には，セグメント損失時の輻輳ウィンドウサイズ縮小によって，必要以上に輻輳ウィンドウサイズが減少し，通信効率が低下する可能性がある．

4.2.5 TCP Vegas-A

TCP Vegas-A[41] は，TCP Vegas を基本とし，輻輳ウィンドウ制御の閾値 α，β を動的に変更することで，経路変更時の RTT の変化に対応した RTT 駆動形のトランスポートプロトコルである．TCP Vegas では，コネクション内での最小 RTT を用いて残留セグメント数を予測しているが，経路変更が頻発する無線アドホックネットワークでは，選択された経路によって最小 RTT が大きく異なる．そのため，残留セグメント数の予測が正確に行えない可能性がある．これに関連して，文献 [42] などの経路変更に対応した手法が提案されているが，パラメータによる設定が煩雑であり，適切な設定値とすることが困難である．そこで，TCP Vegas-A では，動的にパラメータを変化させることで，煩雑性を回避し，適切な制御を行う．TCP Vegas-A では，ウィンドウ制御の閾値である α，β を実測スループットの変化と δ に応じて更新する．

このように，TCP Vegas-A ではウィンドウ制御の閾値を動的に変更することで，経路変更時の RTT の変化に対応しているが，δ を算出する式に変更が加えられていないため，最小 RTT が大幅に変化した場合には対応することが困難であると考えられる．

4.2.6 無線アドホックネットワーク向け トランスポートプロトコル

無線アドホックネットワークでは，端末の移動によってトポロジーが随時変化することや無線通信の特徴などから，有線と異なる通信方式を用いる必要がある．そこで，TCP-ELFN [43] では，ELFN（Explicit Link Failure Notification）を用いて，経路切断情報を送信元端末へ通知し，セグメント送信を一時的に休止する手法が提案されている．しかし，ELFN を用いるためには，エンドエンド間だけでなく，経路上の端末の機能も拡張する必要があるという問題点がある．

TCP-BuS [44] では，無線アドホックネットワークのために，ELFN を用いた拡張を行っている．経路切断時には ERDN（Explicit Route Disconnection Notification）が，修復時は ERSN（Explicit Route Success Notification）がそれぞれ送信元端末へと通知される．また，それらのメッセージを確実に送信元へ届けるために，定期的に制御メッセージを送信し，経路の生存確認を行う．そのほかにも，タイムアウトの延長，選択確認応答（SACK：Selective Acknowledgement）の導入，不必要な高速再送の回避などの機能を有している．しかし，TCP-BuS には，前述した TCP-ELFN と同様の問題のほかに，制御や生存確認メッセージの頻繁な交換によってネットワーク負荷が増大するという問題がある．

ATCP [45] では，ネットワーク層とトランスポート層の間に ATCP 層を定義し，TCP の状態とネットワークの状態を監視した制御を行う手法が提案されている．この手法では，通常，輻輳，損失，切断の四つ状態をもち，通常時には何もせずに通常の TCP と同様に動作する．また，ECN（Explicit Congestion Notification），ICMP（Internet Control Message Protocol）メッセージをそれぞれ輻輳検知，及び経路切断情報取得のために用いる．しかし，ATCP では，ネットワーク層とトランスポート層の間に新たな層を追加しているため，トランスポートプロトコルの変更だけでなく，ルーチングプロトコルにも変更を加える必要がある．

ATP [46] では，ウィンドウベースではなくレートベースの送信速度制御が用いられ，更に，クイックスタートを用いることで通信開始時の送信レート向上を図っている．また，ATP では，あて先や中継端末からのフィードバックを制御のために利用する．フィードバックには，フロー制御や信頼性確保，輻輳制御の情報が内包され，送信端末へ通知される．しかし，送信速度制御のためには，各リンクでの伝搬遅延や端末における処理遅延を計測する必要があり，端末の機能拡張や計算負荷の増大などの問題がある．更に，フィードバック送信による負荷の増大が問題となる．

4.3 往復遅延時間の変動を用いた適応形トランスポートプロトコル

無線アドホックネットワークでは，端末の移動性が高いことや，無線通信の特徴などから，有線通信と比較して不安定な通信環境となる．そのため，経路構築やフレーム送信の際には，それらの特性を考慮したプロトコルが用いられている．そこで，トランスポートプロトコルを設計する際にもアドホックネットワークでの通信特性を考慮し，より効率的な通信を可能にする必要がある．提案するトランスポートプロトコルでは以下の点を考慮し，制御を行う．

1. 実効通信帯域が限られており，ネットワーク資源が有線通信と比べて乏しい
2. 端末が携帯電話など十分な計算資源を備えていない場合がある
3. 無線通信の特徴から，ビット誤りが発生する
4. 電波干渉，障害物などがセグメント損失の要因となる
5. 端末が高い移動性をもつことから，通信経路が変動する
6. 端末の相対的な位置関係が変化することで，通信成功率が変化する
7. ネットワークの状態が変動するため，通信環境の把握が困難である

これらの特徴に対して提案手法では次のように対処する．1. に関しては，送出されるセグメント数を必要最小限とすることで，ネットワークに与える負荷を低減する．2. に関しては，プロトコルの計算量を低減することによって，端末負荷の低減を図る．3., 4. に関しては，セグメント損失の発生と送信速度制御を切り離すことによって，セグメント損失による影響を排除する．5.〜7. に関しては，エンドエンド間の状態を送信元端末が把握する必要がある．下位レイヤの情報を用いることによって，輻輳の発生や経路切断，変更などを知ることができるが，汎用性を考慮した場合にはトランスポートプロトコル単体で動作することが望ましい．そこで，提案手法では，ネットワークの状態を把握した送信速度制御を行うとともに，ネットワークや端末に与える負荷を低減することを目的とする．

4.3.1 RTT の変動に基づくウィンドウ制御

提案手法におけるウィンドウ制御は，TCP Vegas と同様に RTT によって駆動される．また，提案手法における RTT は，図 4.8 に示すように，セグメントを送信し，その ACK が返答されるまでの時間のことを指す．提案手法では，より正確なネットワークの状態把握のため，平滑化した RTT と測定した RTT との相対的

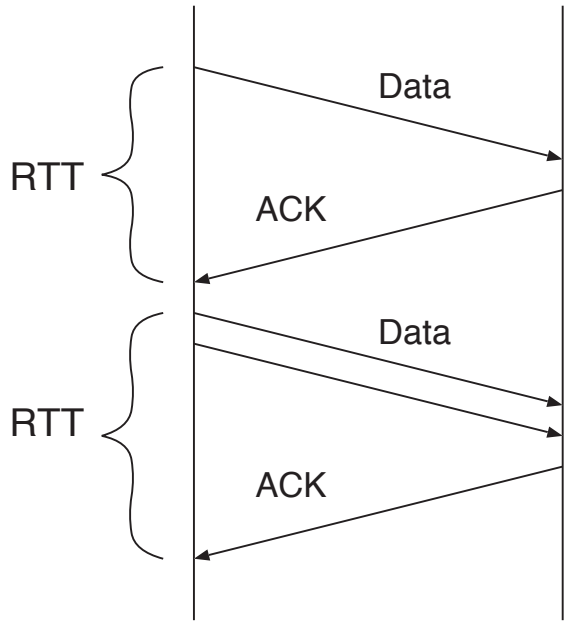

図 4.8 RTT 計測

な変化量を用いる．ここで相対的な変化量を用いるのは，絶対的な変化量を用いた場合には，変化量がRTT の大小に左右され，制御のための閾値などを適宜変更する必要があり，RTT が大幅に変動するアドホックネットワークには適さないと考えられるためである．平滑化したRTT は，加重平均を用いて算出され，算出時と比較して少し前の状態の平均的なRTT を表している．そのため，通信環境の変化によってRTT が変動している場合でも，平滑化したRTT を基準とし，測定したRTT との相対変化量を算出することによって，現時点でのネットワークの状態を把握することが可能となる．つまり，提案手法では，基準を少し前の状態とするこで，現時点での状態変化の傾向を把握し，それに応じた制御を行うことができる．

RTT が増加傾向にある場合には，図4.9 に示すように，ネットワーク内の残留セグメントが増加して混雑している状態と認識し，送信速度を低下させて輻輳の発生を抑制する．また，RTT が減少傾向にある場合には，ネットワークに余裕がある状態と認識し，送信速度を増加させて通信効率の向上を図る．このようにRTT を用いることで，経路上の状態を把握した制御が可能となり，ネットワーク資源を有効に利用することができ，通信効率が向上すると考えられる．

提案手法ではウィンドウ制御のためにタイムスタンプオプション [47] を有効にし，正確なRTT 計測を行う．また，RTT を計測するごとに以下の式 4.12 を用いて平滑化したRTT を算出する．

$$\overline{R}_k \;=\; (1-\alpha)\overline{R}_{k-1} + \alpha R_k \tag{4.12}$$

ここで α は定数である．そして，計測したRTT と平滑化したRTT を用いて，以

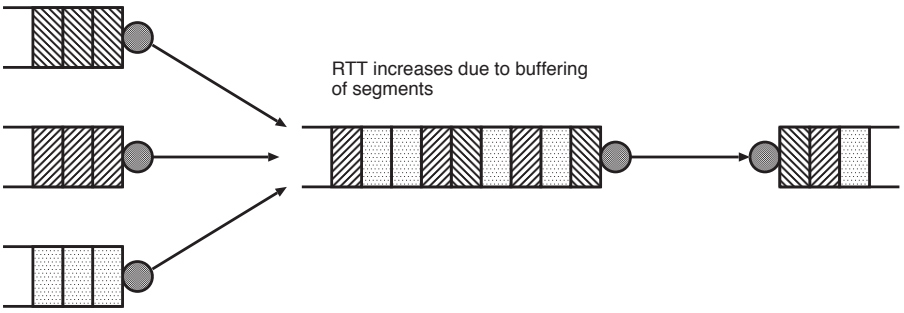

図 4.9 RTT 増加の要因

下の式 4.13，式 4.14 を用いて輻輳ウィンドウサイズ更新を行う．

$$\Delta_k = \frac{\overline{R}_k}{R_k} - 1 \tag{4.13}$$

$$c_{k+1} = \begin{cases} c_k - M_{\mathrm{seg}} & (\Delta_k \leq -\beta) \\ c_k & (-\beta < \Delta_k \leq \beta) \\ c_k + M_{\mathrm{seg}} & (\beta < \Delta_k) \end{cases} \tag{4.14}$$

ここで，Δ_k は計測した RTT に対する平滑化した RTT の割合，α，β は定数である．また，前述した定数 α，β によってウィンドウ制御の感度を調整することができるが，本章ではシミュレーションによって得られた最適値として，$\alpha = 0.1$，$\beta = 0.3$ を用いる．ネットワークが混雑し，RTT が増加傾向にある場合には，輻輳ウィンドウサイズを減少させることで単位時間あたりの送出セグメント数が抑制され，送信速度が低下する．反対に，通信環境が向上し，RTT が減少傾向にある場合には，ネットワークに余剰帯域があると認識して輻輳ウィンドウサイズを増加させることで送信速度を増加させる．これらの制御により，経路上の通信状態を考慮したウィンドウ制御が可能となり，ネットワーク資源の有効活用を図ることが可能である．また，通信開始時には，TCP Reno と同様に ACK を受信するたびに M_{seg} だけ輻輳ウィンドウサイズを増加させるスロースタートを用いることで，ネットワーク容量に合わせた送信速度の初期値を決定する．経路再構築が行われた場合には，再びスロースタートから通信を再開することによってネットワーク容量に合わせた送信速度を決定する．平滑化した RTT は，スロースタート中に新しい経路の RTT によって更新されるため，経路変更時にも新しい経路に対応した制御が可能となる．

提案手法では，ネットワークの状態に応じたウィンドウ制御を行うことで公平性を確保する．通信を行っているリンク上に他の端末からのフローが流入した場合には，バッファリング遅延によって RTT が増加することから，送信速度が低下し，余剰帯域が発生する．流入したフローが発生した余剰帯域を獲得することによって，帯域を共有した通信を実現する．また，TCP Reno と提案手法が混在した環境下では，TCP Reno と交互に輻輳ウィンドウサイズの増減を行うことで公平

性の確保が行われる．TCP Reno が輻輳ウィンドウサイズを増加させている間は，RTT の増加によって提案手法の輻輳ウィンドウサイズが減少し，セグメント損失の発生によって TCP Reno が輻輳ウィンドウサイズの縮小を行った場合には，RTT が減少し，提案手法では輻輳ウィンドウサイズの増加を行う．このように，提案手法では TCP Reno と交互に輻輳ウィンドウサイズの増減を繰り返すことによって，平均的に輻輳ウィンドウサイズが等しくなり，公平性が確保される．

また，提案方式における RTT の相対変化量を算出する式 4.12 は，TCP Vegas で利用される式と類似している．実際，式 4.4 を変形すると次式 4.15 が得られ，TCP Vegas においても RTT の相対変化量がウィンドウ制御に利用されていることが分かる．

$$\delta_k = \left(1 - \frac{R_{\min}}{R_k}\right) \times c_k \tag{4.15}$$

TCP Vegas では，輻輳ウィンドウサイズの大小によって RTT の変化量に対するウィンドウ制御の感度が変化する．すなわち，c_k が大きい場合には輻輳ウィンドウサイズが増加しにくく，小さい場合には減少しにくくなる．一方，提案手法では輻輳ウィンドウサイズの大小に関わらず，RTT が一定以上変動した場合にウィンドウ制御が行われるため，通信状態の変化に基づいた柔軟な制御が可能になると考えられる．

更に，提案手法ではセルフクロッキングを積極的に利用することによってネットワークの自律的なフロー調整を促す．セルフクロッキングとは ACK の到着間隔に合わせてデータ転送を行う機構であり，自律的に送信速度を調整する機能を有している．図 4.10 にセルフクロッキングの例を示す．送信元端末は ACK を受信することによって新しいセグメントを送信するため，経路上の最も低速なリンクの送信速度に応じた間隔で ACK を受信することになる．そのため，特別な制御を

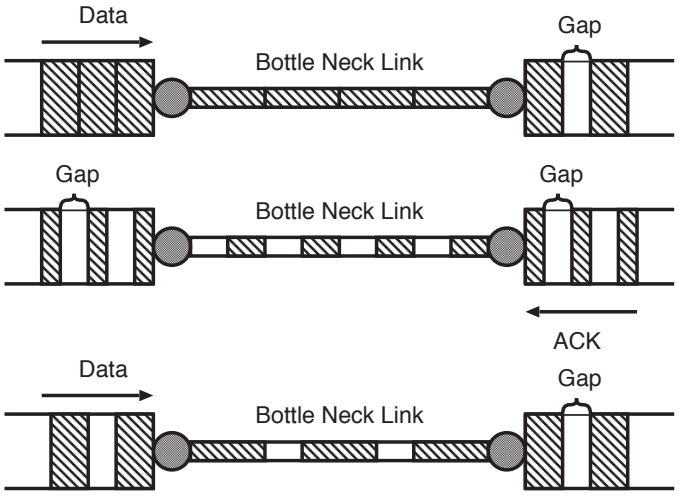

図 4.10 セルフクロッキング

用いることなく，送信元端末でのセグメント送信速度が自律的に調整される．提案手法では輻輳ウィンドウサイズを TCP と比べて大きな変動幅で調整を行っているが，セルフクロッキングを利用することで複雑な送信速度制御の機能を廃している．このことによって，自律的なフロー制御を促すとともに，端末負荷の低減を図る．

4.3.2 ネットワーク負荷を低減する再送制御

提案手法では通信の信頼性確保のために，損失セグメントの再送を行う．提案手法における再送制御は，TCP Reno の SACK オプション [2] と同様に，損失セグメントのみを再送することによってネットワーク負荷を最小限に抑える．また，TCP Vegas などの RTT 駆動形プロトコルでは，ウィンドウ制御がセグメント損失の有無に影響されないという特徴を有している．そのため，提案手法でもセグメント損失による輻輳ウィンドウサイズの縮小を行わず，セグメントの送信を継続する．また，軽度の輻輳が発生している場合にもセグメント損失が発生する可能性があるが，このような場合には同時に RTT も増加するため，再送後の通常のウィンドウ制御によって輻輳ウィンドウサイズが減少することで輻輳回避が行われる提案手法におけるセグメント損失時の制御は，前述した式 4.13，式 4.14 によるウィンドウ制御を一時的に停止し，以下の式 4.16 を用いて行われる．

$$c_a = \begin{cases} c + l & (\text{if segment loss}) \\ c & (\text{otherwise}) \end{cases} \tag{4.16}$$

ここで，c_a は送信可能範囲を示す輻輳ウィンドウサイズ，c は通常の輻輳ウィンドウサイズ，l は損失したセグメントサイズとする．つまり，c_a は通常のウィンドウサイズと再送が必要なセグメントサイズの和である．重複 ACK を受信し，セグメント損失を検出したことにより $c_a \neq c$ となった場合の送信側ウィンドウサイズには c_a を用い，再送と通常のセグメント送信を並行して行う．再送は，損失したセグメントに対して個々に行うことによって，無駄な再送を低減する．一方で，再送タイマが起動し，タイムアウトが発生した場合には，経路切断や重度の輻輳が発生していると認識し，スロースタートから通信を再開する．

以上を用いることによって，輻輳ウィンドウサイズを減少させることなく再送を行う．更に，セグメント損失に伴う輻輳ウィンドウサイズの減少を回避することによって，輻輳ウィンドウサイズを保ったまま通信を継続することが可能となり，通信効率の向上が期待できる

4.3.3 状態遷移の簡略化

提案手法では，コネクション確立時と切断時の状態遷移を簡略化することによって，送出されるコントロールセグメント数を削減する．提案手法における状態

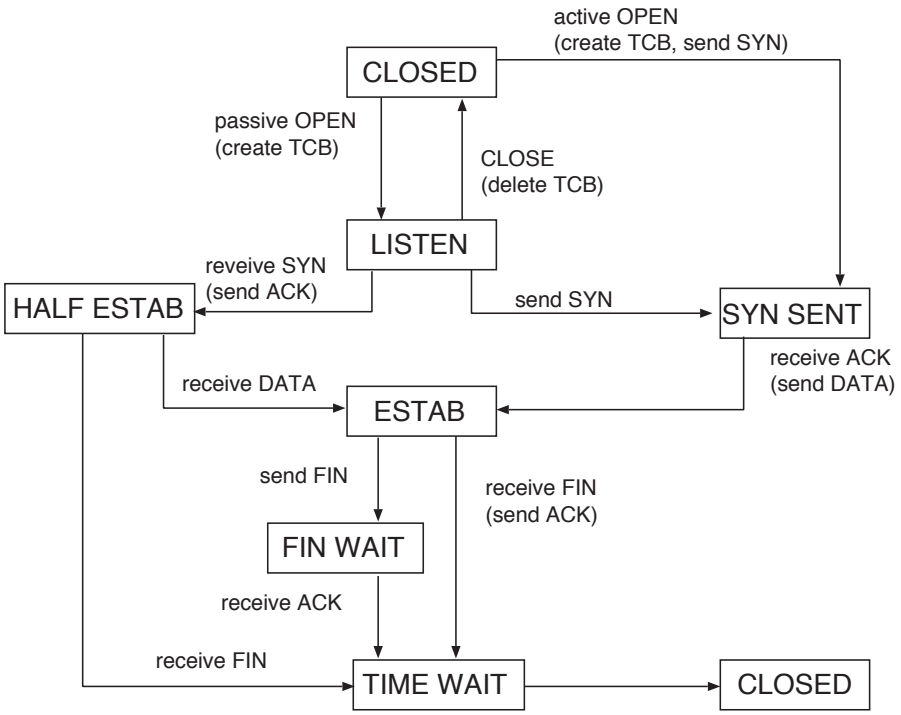

図 4.11 状態遷移図

遷移図を図 4.11 に示す．更に，図 4.12 に提案手法でのコネクションの確立の動作例を示す．

1. 送信元端末は，コネクションを確立したい相手に向けてSYN フラグを立てたセグメントを送信する．

2. SYN フラグの立ったセグメントを受信した端末は，自身の状態がLISTEN であれば状態を HALF ESTAB へ遷移し，ACK を返信する．

3. 送信元端末は，自身が送信したSYN フラグを立てたセグメントに対するACK を受信すると，状態を ESTAB へと遷移し，データの送信を開始する．

4. データセグメントを受信した端末は，HALF ESTAB から ESTAB へと遷移し，コネクションを確立する．

また，図 4.13 にコネクションの切断の動作例を示す．

1. データを送信し終えた送信元端末は，相手に向かってFIN フラグを立てたセグメントを送信し，FIN WAIT へ遷移する．

2. FIN フラグの立ったセグメントを受信した端末は，自身の状態をTIME WAIT へと遷移し，ACK を返信する．

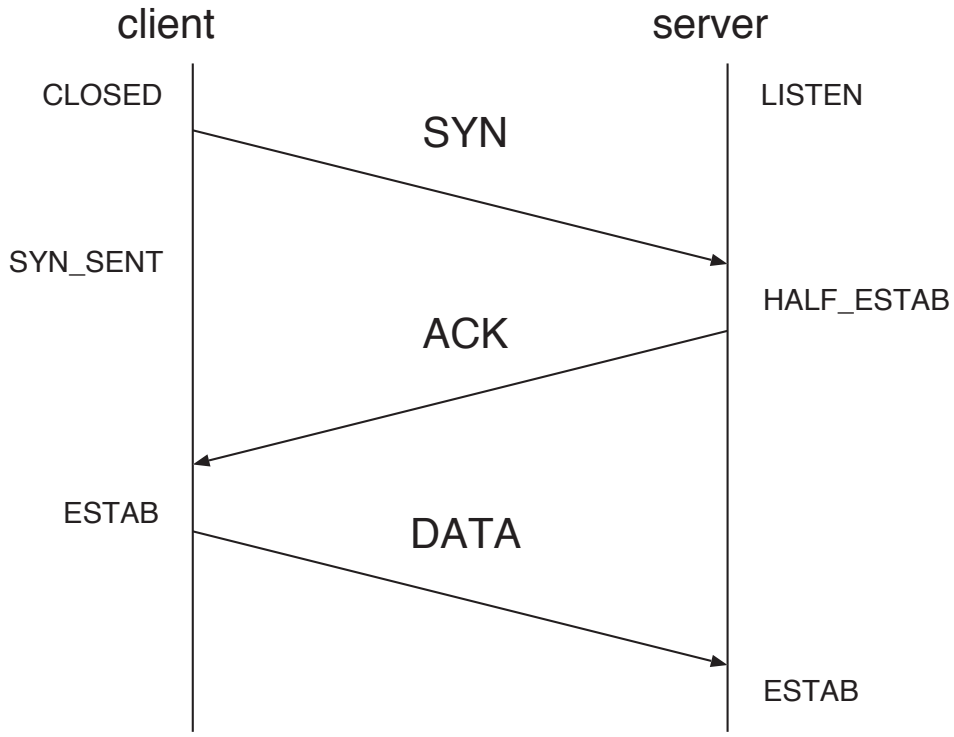

図 4.12 コネクション確立

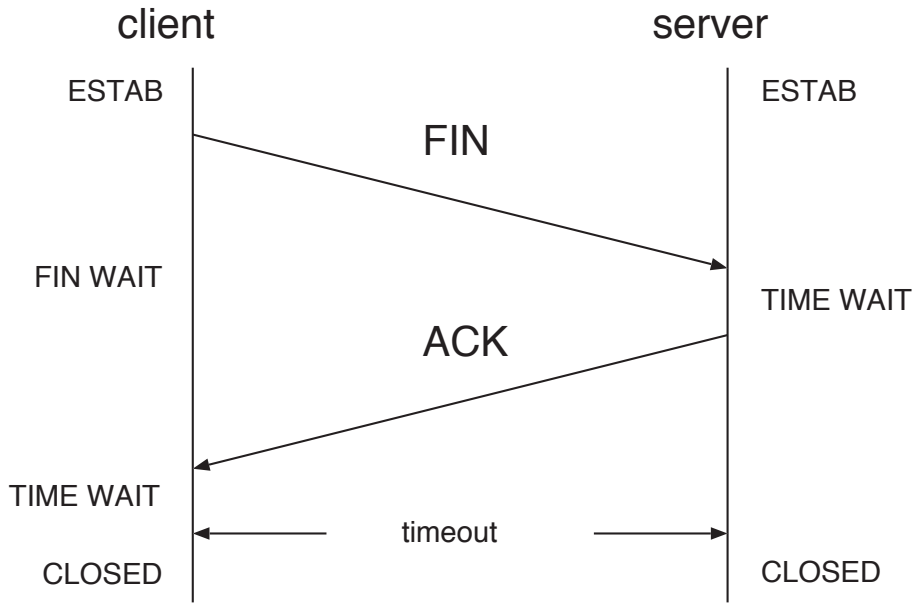

図 4.13 コネクション切断

3. 送信元端末は，自身が送信したFINフラグを立てたセグメントに対するACKを受信すると状態をTIME WAITへと遷移する．

4. サーバ，クライアント側ともに，設定したタイマがタイムアウトすると状態をCLOSEDへと遷移し，コネクションを切断する．

4.4 シミュレーション評価

4.4.1 シミュレーション条件

本章では，以下の三つのコンピュータシミュレーションを用いて，TCP Reno，TCP Vegas，TCP Westwood，TCP UB，TCP Vegas-Aを利用した通信との性能比較を行う．なお，**4.2.6**で述べたトランスポートプロトコルは，ネットワーク層，及び端末の改良を必要とするので比較対象とはしない．各シミュレーションに共通する条件は以下のとおりである．シミュレータにはGloMoSim 2.03[48]を用いる．無線通信方式にはIEEE 802.11bを用い，有効電波半径を100m，伝送速度を11Mbpsとする．ルーチングプロトコルとしてAODVを用いる．また，M_{seg}は1,460Byteとし，正確なRTT計測のためタイムスタンプオプションを有効にする．更に，TCP Vegas，TCP Vegas-Aでは，$\alpha = M_{\mathrm{seg}}$，$\beta = 3M_{\mathrm{seg}}$とし，提案手法では，$\alpha = 0.1$，$\beta = 0.3$とする．

（1）シミュレーション1

シミュレーション1では，TCP Reno，TCP Vegas，TCP Westwood，TCP UB，TCP Vegas-A，提案手法の性能をそれぞれ独立に比較する．100m四方から1,000m四方の領域に同一のプロトコルを実装した端末100端末をランダムに配置し，全ノードの中から送信元端末とあて先端末をランダムに選択する．各送信元端末は平均生起間隔30秒で平均1MByteのデータをあて先端末へ送信する．ノードの移動はランダムウェイポイントを利用し，ウェイティングタイムを20秒，各端末の移動速度を0m/sから10m/sまで変化させる．

（2）シミュレーション2

シミュレーション2では，共有リンクを介して通信を行った場合の端末間のスループットの公平性を評価する．シミュレーションでは，TCP Reno，TCP Vegas，TCP Westwood，TCP UB，TCP Vegas-A，提案手法のうちいずれかのトランスポートプロトコルを使用し，図4.14に示すトポロジーを用いて，端末0〜端末2がそれぞれ端末5〜端末7へ，共有リンク3–4を介して20Mbyteのデータ送信を行う．

（3）シミュレーション3

シミュレーション3では，共有リンクを介して通信を行った場合のプロトコル間のスループットの公平性を評価する．シミュレーションでは，TCP RenoとTCP

Vegas，TCP Westwood，TCP UB，TCP Vegas-A，提案手法のうちいずれかのトランスポートプロトコルを使用し，図 4.15 に示すトポロジーを用いて，端末 0，端末 1 がそれぞれ端末 4，端末 5 へ，共有リンク 2–3 を介して 20Mbyte のデータ送信を行う．

シミュレーション 1 の評価指標として，セグメント損失率，再送回数，及びスループットを用いる．セグメント損失率は，送信セグメントに対する受信セグメント数を示すもので，通信の信頼性を表す指標として用いる．再総回数は全送信セグメントに対して実際に再送が必要となったセグメントの数を示すもので，ネットワークに与える負荷，及び通信の信頼性を示す指標として用いる．スループットは，単位時間あたりの転送データ量から再送セグメントやコントロールセグメントなどの重複分を差し引いた実際の転送効率（グッドプット）を表す．以上の指標を用いることによって全体的な通信効率の評価を行う．

シミュレーション 2，3 では，端末間の公平性評価のため，端末の平均スループットのほか，Stability Index [49] と Fairness Index [50] を用いる．Stability Index,

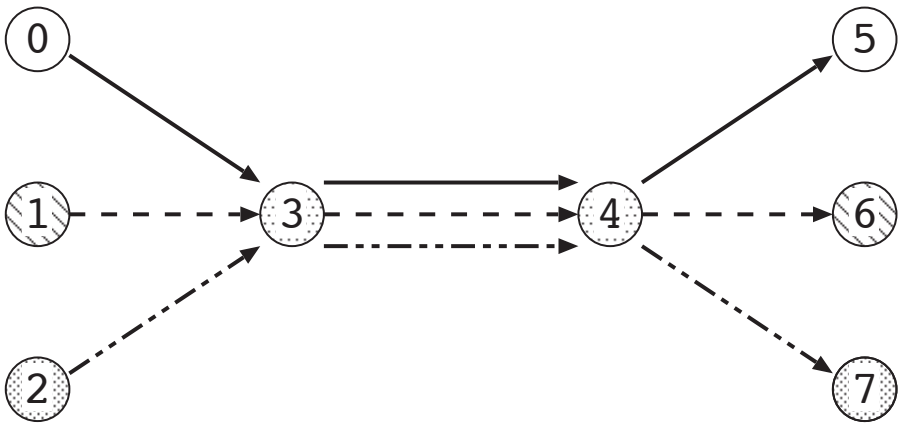

図 4.14 シミュレーション 2 のトポロジー

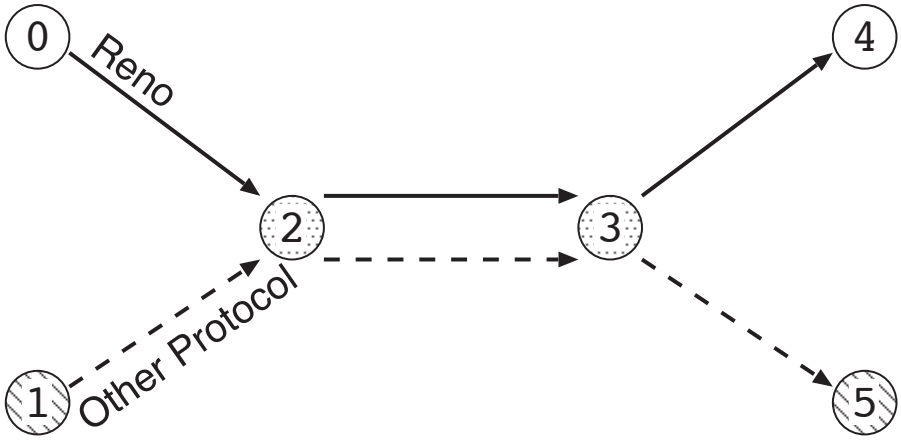

図 4.15 シミュレーション 3 のトポロジー

Fairness Index はそれぞれ通信の安定性と公平性を表す指標で，Stability Index は 0 に近いほど安定したスループットであることを示し，Fairness Index は 1 に近いほどフロー間の公平性が高いことを示す．Stablity Index, Fairness Index はそれぞれ，以下の式 4.17，式 4.18 を用いて算出される．

$$S_i = \frac{1}{\overline{x}_i}\sqrt{\frac{1}{m-1}\sum_{k=1}^{m}(x_i(k)-\overline{x}_i)^2} \tag{4.17}$$

$$F = \frac{\left(\sum_{i=1}^{n}\overline{x}_i\right)^2}{n\sum_{i=1}^{n}\overline{x}_i{}^2} \tag{4.18}$$

ここで $\overline{x}_i$ は，端末 i の平均スループット，n はフロー数，$x_i(k)$ は端末 i の 5 秒ごとの平均スループット，m は平均スループットのデータ数を表している．

4.4.2 シミュレーション結果

(1) シミュレーション 1

図 4.16〜図 4.18 にシミュレーション 1 の結果を示す．これらの図は，TCP Reno を 1 とした場合の相対的な割合を示している．

図 4.16 は，全送信セグメントに対する受信セグメントの割合を示している．結果から，提案手法では，TCP Reno 以外の手法と比較して，セグメント損失を低減している．しかし，TCP Reno と比較してセグメント損失率が増加している．こ

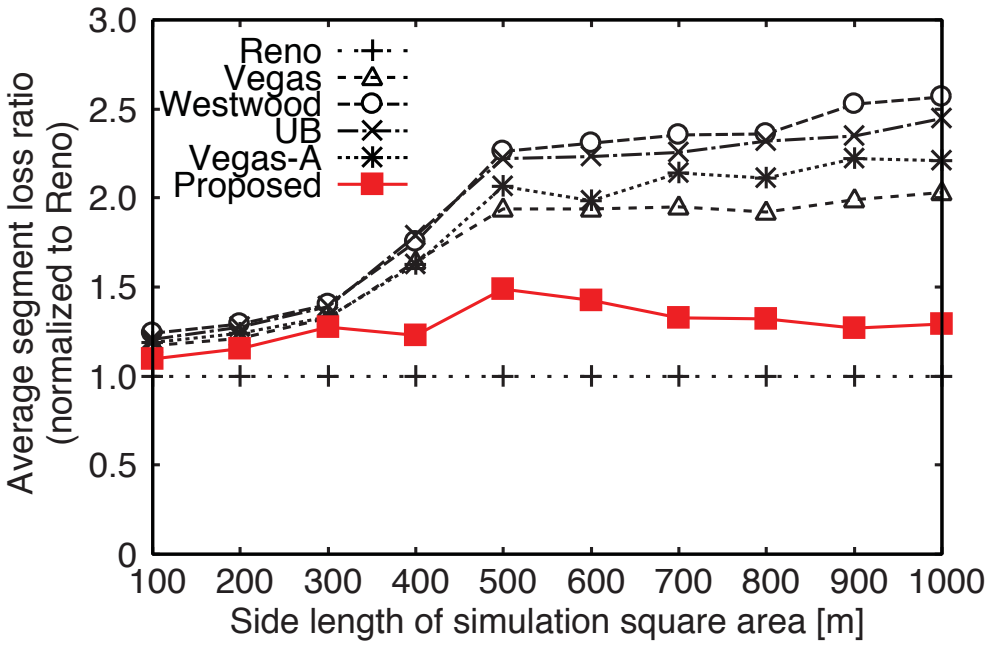

図 4.16 シミュレーション 1 におけるセグメント損失率 〈発表文献 1.2〉

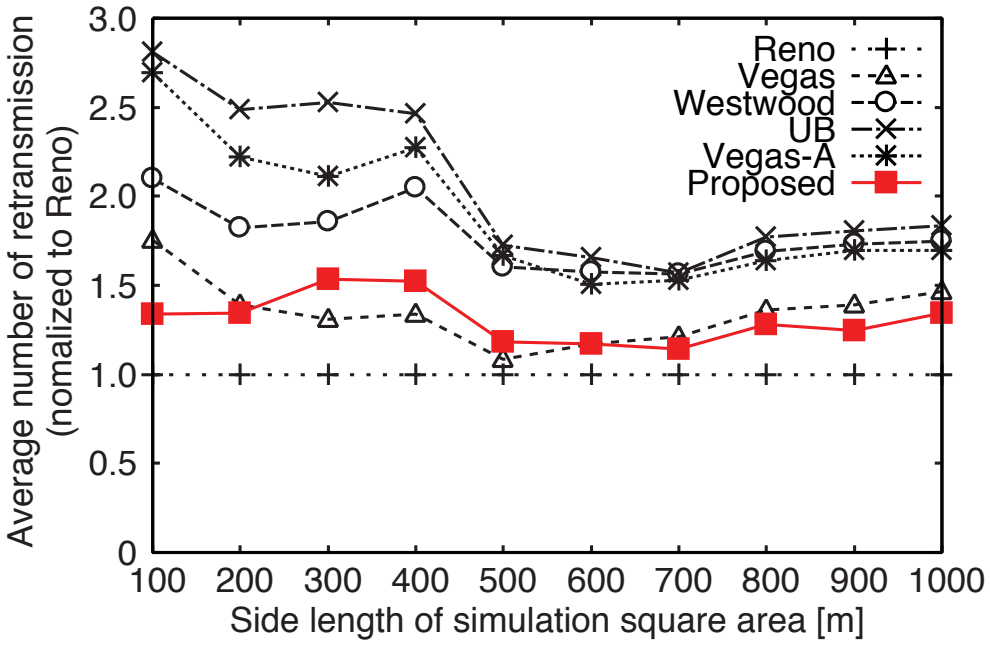

図 4.17 シミュレーション 1 における再送回数 〈発表文献 1.2〉

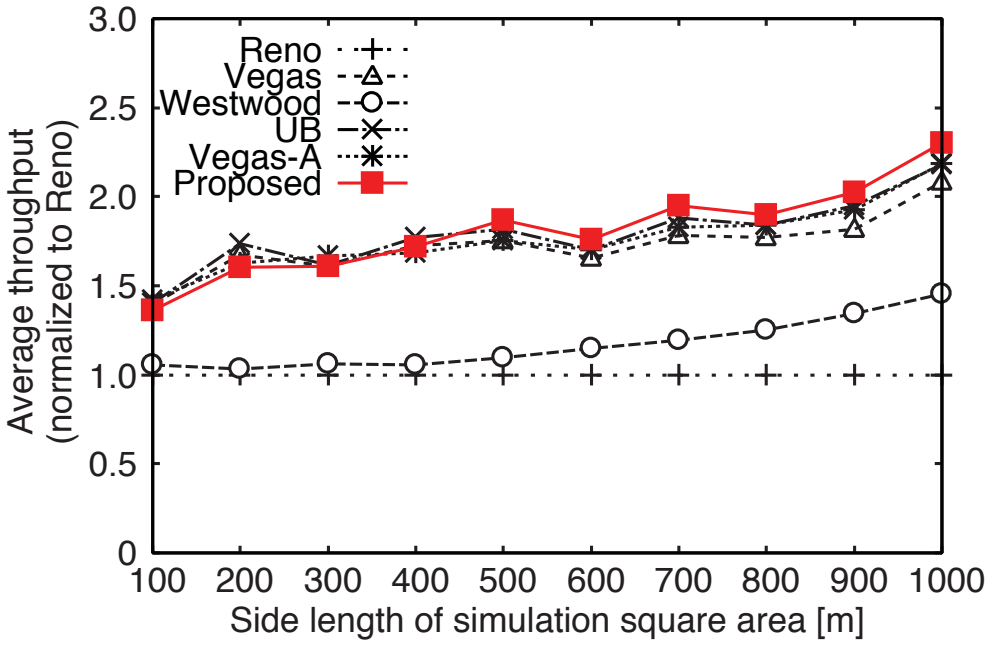

図 4.18 シミュレーション 1 におけるスループット 〈発表文献 1.2〉

れは，TCP Reno 以外の手法では，帯域推定や RTT の実測値を用いた制御を行っているが，通信環境が随時変化するアドホックネットワークでは正確な状態把握が困難であるため，適切な制御が行われずに，過剰なセグメントを送出している場合があるためと考えられる．また，TCP Reno では，セグメント損失が発生した場合には輻輳ウィンドウサイズを減少させ，送信速度を低下させることから，スループットが低下し，干渉や輻輳などの発生が他の手法と比較して減少したと考えられる．

図 4.17 に各手法の再送回数を示す．結果から，TCP Reno 以外の各手法では，TCP Reno と比較して再送回数が増加している．これは，TCP Reno ではセグメント損失が発生した場合には輻輳ウィンドウサイズを半分に減少させるが，他の手法ではそれと比較して大きな輻輳ウィンドウサイズでの通信が継続される．そのため，経路切断などが起きた場合には切断中により多くのセグメントが送信され，セグメント損失が増加し，より多くの再送が必要となるためであると考えられる．

図 4.18 は，エンドエンド間のスループットを示している．RTT 駆動形プロトコルである TCP Vegas，TCP Vegas-A，提案手法では，ACK 駆動形プロトコルである TCP Reno と TCP Westwood を上回るスループットを達成している．これは，RTT 駆動形プロトコルでは，RTT の変化から状態を把握し，ウィンドウ制御を行うため，セグメント損失が発生した場合でも輻輳ウィンドウサイズの縮小を行わないためである．一方，TCP Reno と TCP Westwood では，セグメント損失の発生によって輻輳ウィンドウサイズが減少するため，余剰帯域がある場合でも送信速度を増加させることが困難となる．また，TCP UB では，輻輳ウィンドウサイズの上限を制限しているため，輻輳ウィンドウサイズを増加し続けることによるセグメント損失が低減される．そのため，輻輳回避による輻輳ウィンドウサイズの縮小回数が減少し，スループットが向上したと考えられる．

以上のことから，TCP Reno 以外の手法より低いセグメント損失率としながら，より高いスループットを達成していることが分かる．また，シミュレーション領域が拡大した場合には，従来手法と比較して提案手法のスループットが最大となることが示された．

（2）シミュレーション 2

図 4.19〜図 4.24，表 4.1〜表 4.6 にシミュレーション 2 の結果を示す．図 4.19〜図 4.24 は，それぞれのトランスポートプロトコルを利用した場合の端末ごとのスループットの変化を示している．また，表 4.1〜表 4.6 は，それぞれのトランスポートプロトコルを利用した場合の平均スループット，Stability Index，及び Fairness Index の値を示している．

図 4.19，図 4.21，表 4.1，表 4.3 から，基本動作を同じとする TCP Reno と TCP Westwood では，スループットの変動が非常に大きいことが分かる．また，図 4.22，

表 4.4 から，TCP UB では，輻輳ウィンドウサイズの上限値を制限しているため，TCP Reno や TCP Westwood と比較して，輻輳や干渉に伴うセグメント損失が低減され，比較的安定したスループットで通信を行っている．しかし，TCP UB では帯域推定によって輻輳ウィンドウサイズの上限値を制限しているが，端末ごとに推定帯域が異なる場合があり，公平性が低下の要因となっている．一方で，図 4.20，図 4.24，表 4.2，表 4.6 から，RTT 駆動形プロトコルである TCP Vegas と提案手法では細かいスループットの変動はあるが，ほぼ一定のスループットを保った安定的な通信を行っていることが分かる．これは，通信環境に変動がない場合には一定のスループットを保って通信を行うという TCP Vegas を基本とするプロトコルの特徴に起因している．そのため，Stability Index も比較的小さな値になっている．しかし，図 4.23，表 4.5 から，TCP Vegas-A では，非常に大きくスループットが変動していることが分かる．これは，TCP Vegas-A では，ウィンドウ制御の閾値を動的に変化させるため，TCP Vegas と比較して，輻輳ウィンドウサイズの更新が頻繁に行われる可能性があるため，送信速度の変化が大きくなっているためであると考えられる．

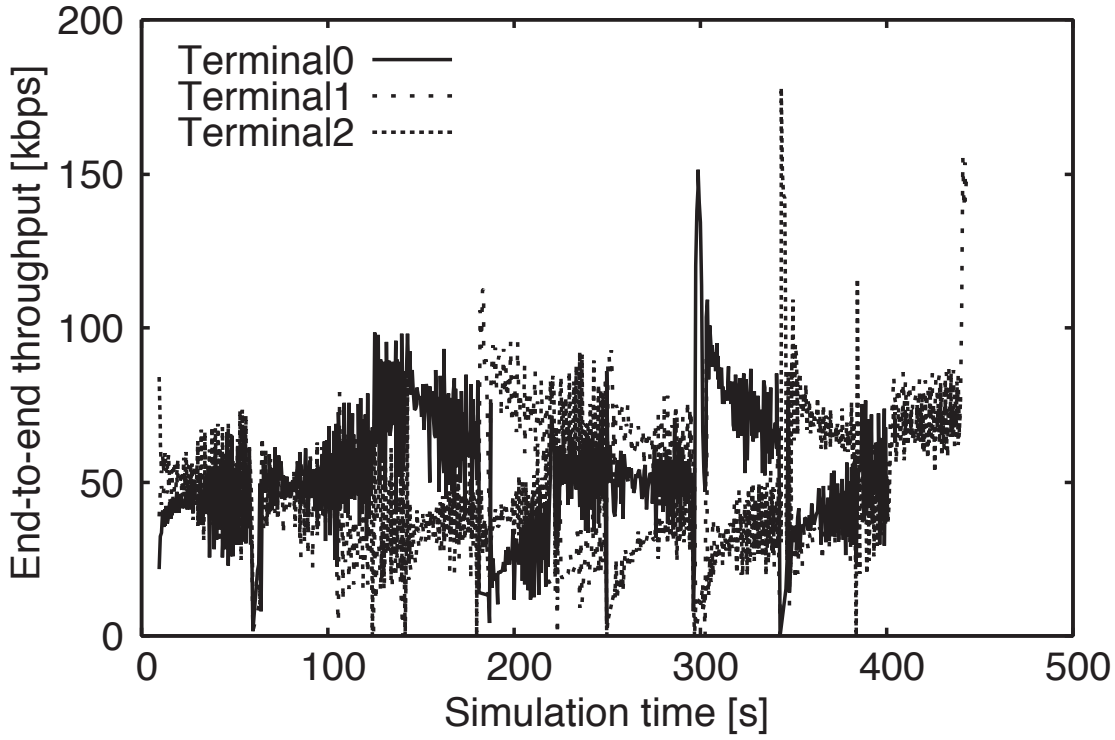

図 4.19 シミュレーション 2 におけるスループット（TCP Reno） 〈発表文献 1.2〉

表 4.1 シミュレーション 2 における評価値（TCP Reno） 〈発表文献 1.2〉

	Terminal 0	Terminal 1	Terminal 2
Throughput	47.665 kbps	41.684 kbps	47.016 kbps
Stability Index	0.28699	0.66477	0.42676
Fairness Index	0.99653		

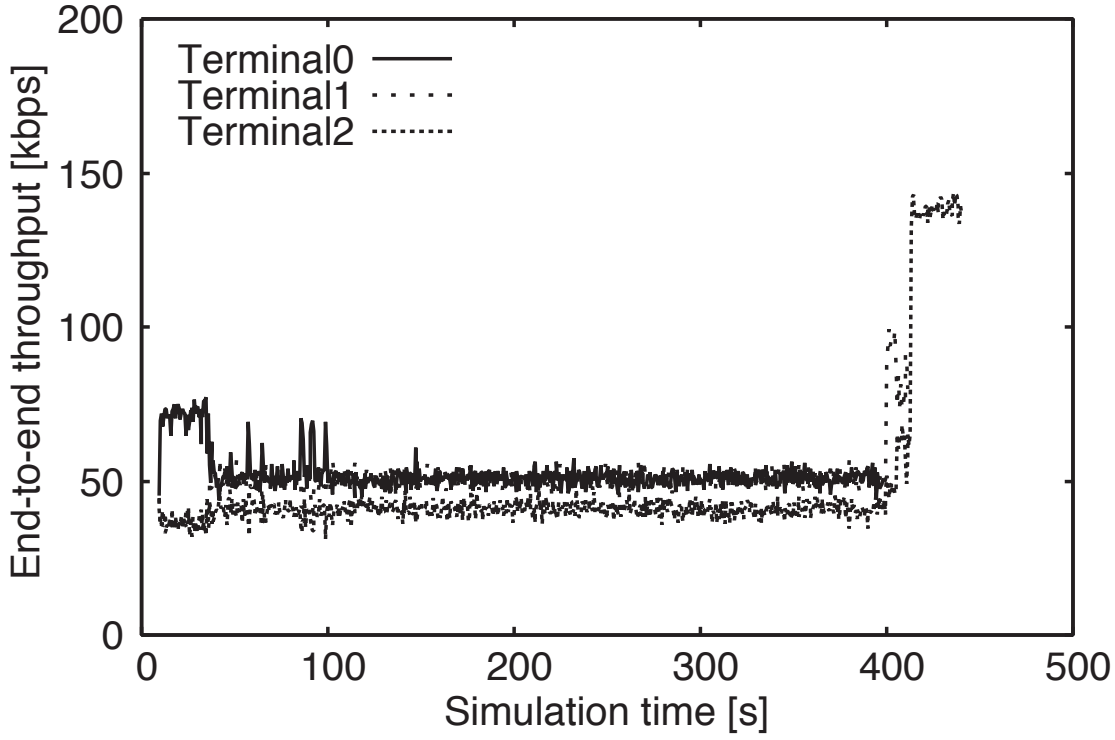

図 4.20 シミュレーション 2 におけるスループット（TCP Vegas）〈発表文献 1.2〉

表 4.2 シミュレーション 2 における評価値（TCP Vegas）〈発表文献 1.2〉

	Terminal 0	Terminal 1	Terminal 2
Throughput	42.125 kbps	42.480 kbps	49.771 kbps
Stability Index	0.26638	0.15015	0.088972
Fairness Index	0.99385		

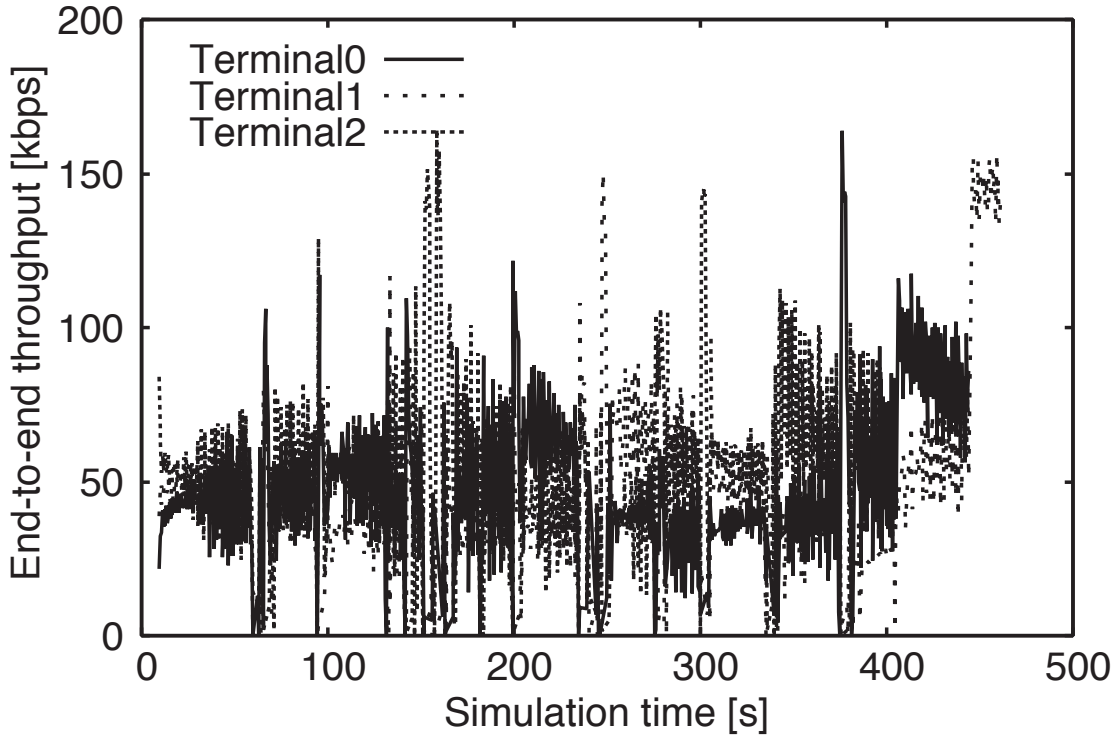

図 4.21 シミュレーション 2 におけるスループット（TCP Westwood） 〈発表文献 1.2〉

表 4.3 シミュレーション 2 における評価値（TCP Westwood） 〈発表文献 1.2〉

	Terminal 0	Terminal 1	Terminal 2
Throughput	42.425 kbps	47.237 kbps	38.278 kbps
Stability Index	0.75836	0.60251	1.8080
Fairness Index	0.99268		

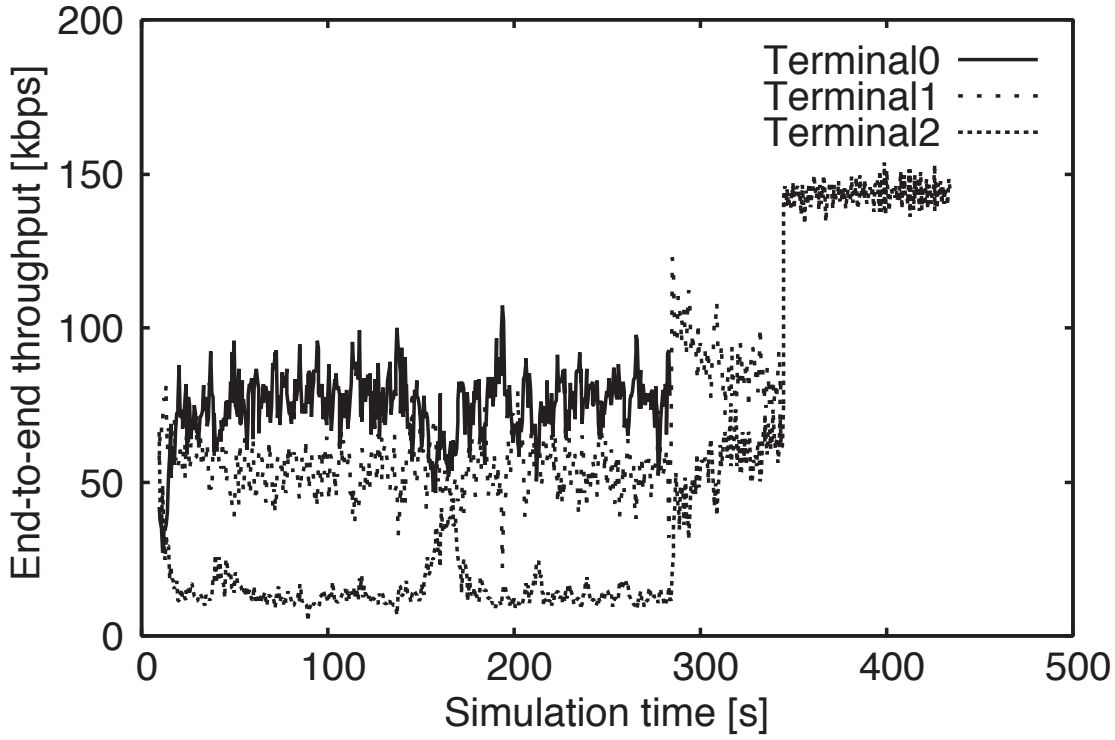

図 4.22 シミュレーション 2 におけるスループット（TCP UB） 〈発表文献 1.2〉

表 4.4 シミュレーション 2 における評価値（TCP UB） 〈発表文献 1.2〉

	Terminal 0	Terminal 1	Terminal 2
Throughput	71.984 kbps	59.415 kbps	47.015 kbps
Stability Index	0.12278	0.23749	1.2220
Fairness Index	0.97145		

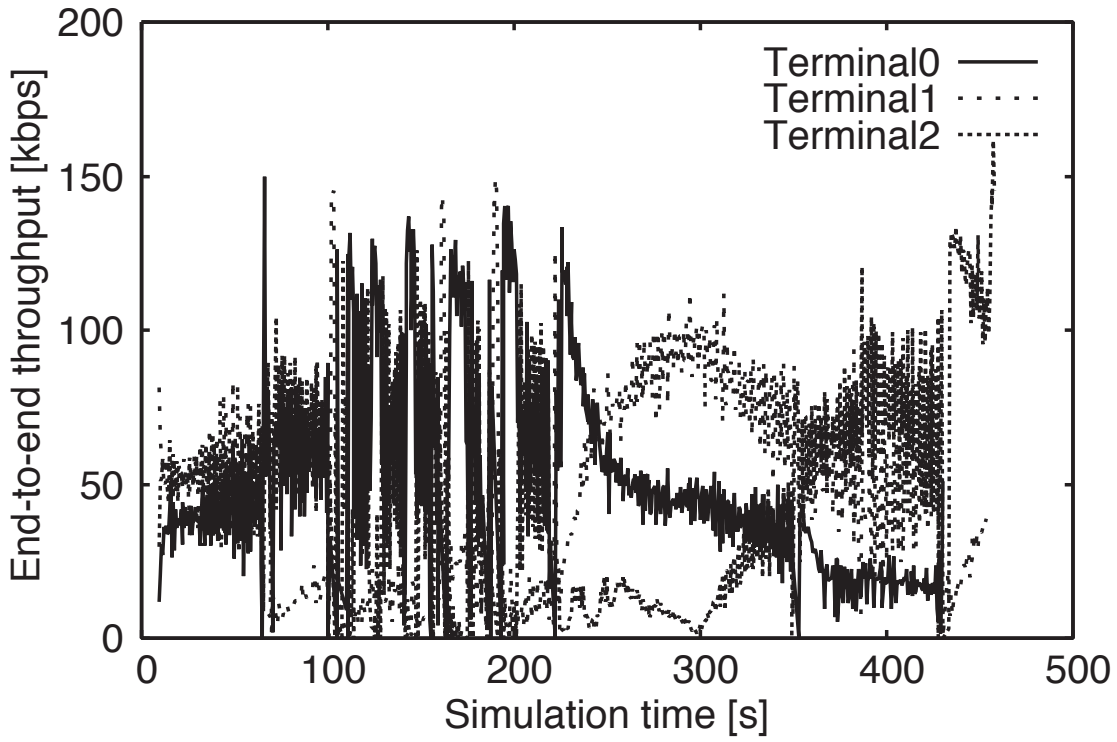

図 4.23 シミュレーション 2 におけるスループット（TCP Vegas-A） 〈発表文献 1.2〉

表 4.5 シミュレーション 2 における評価値（TCP Vegas-A） 〈発表文献 1.2〉

	Terminal 0	Terminal 1	Terminal 2
Throughput	47.383 kbps	44.969 kbps	44.584 kbps
Stability Index	0.59453	0.62337	0.68666
Fairness Index	0.99926		

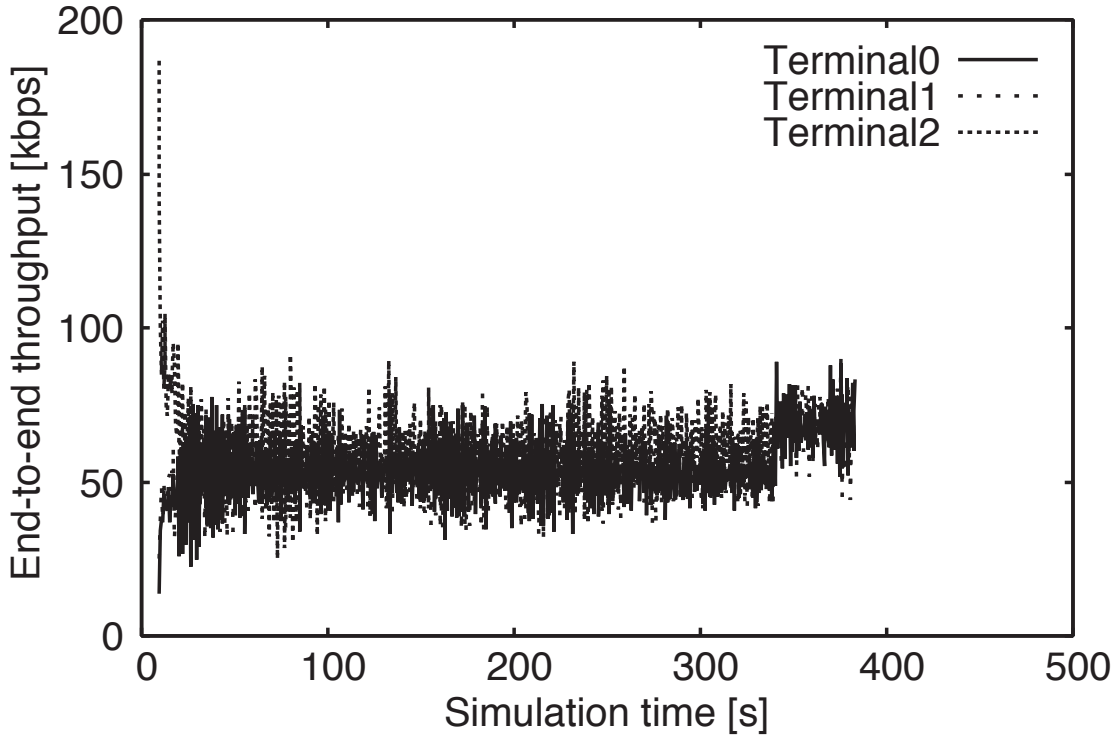

図 4.24 シミュレーション 2 におけるスループット（提案手法） 〈発表文献 1.2〉

表 4.6 シミュレーション 2 における評価値（提案手法） 〈発表文献 1.2〉

	Terminal 0	Terminal 1	Terminal 2
Throughput	53.248 kbps	53.344 kbps	61.528 kbps
Stability Index	0.13162	0.12941	0.10174
Fairness Index	0.99522		

（3）シミュレーション 3

図 4.25，表 4.7 から，TCP Reno では帯域を共有した通信を行っていることが分かるが，TCP Reno では，セグメント損失の発生によってウィンドウサイズが減少し，送信速度が低下する．そのため，アドホックネットワークでは帯域を最大限まで利用することができないと考えられる．また，図 4.27，表 4.9 から，TCP Westwood では，公平性が低下していることが分かる．これは，TCP Westwood では，TCP Reno と比較して，セグメント損失時の輻輳ウィンドウサイズが大きいため，TCP Reno より高い送信レートで通信を行うためであると考えられる．更に，図 4.28，表 4.10 から，TCP UB では，スループットが大幅に低下していることがわかる．これは，TCP Reno からのフローが帯域を占有してしまうことで，推定帯域が必要以上に減少し，輻輳ウィンドウサイズを拡大することができないためであると考えられる．一方で，図 4.26，図 4.29，表 4.8，表 4.11 から，TCP Vegas を基本とするプロトコルでは，大きく公平性が低下している．これは，TCP Reno では，ACK を受信するごとに輻輳ウィンドウサイズを増加させるが，TCP Vegas では，ネットワーク内の残留セグメント数によって輻輳ウィンドウサイズを制御するため，TCP Reno によるセグメントが帯域を占有してしまうためである．

しかし，図 4.30，表 4.12 から，提案手法では TCP Reno と高い公平性を保って通信を行っていることが分かる．これは，提案手法では，RTT の相対的な変化を用いてウィンドウ制御を行うため，RTT の実測値に影響を受けることなく，RTT の変化傾向に合わせてウィンドウ制御を行うことが可能なためである．更に，TCP Reno の輻輳ウィンドウサイズの増減に合わせて，提案手法では輻輳ウィンドウサイズを減少，増加させることで，等しく帯域を共有し，効率を高めた通信を実現している．

以上のことにより，提案手法では従来手法と同等のセグメント損失率，再送回数を維持し，スループットを向上させた通信を実現できた．また，端末間，及びプロトコル間の公平性を保った安定的な通信を行うことで，ネットワーク帯域や端末資源の乏しいアドホックネットワークでの通信効率を向上させることができたと結論できる．

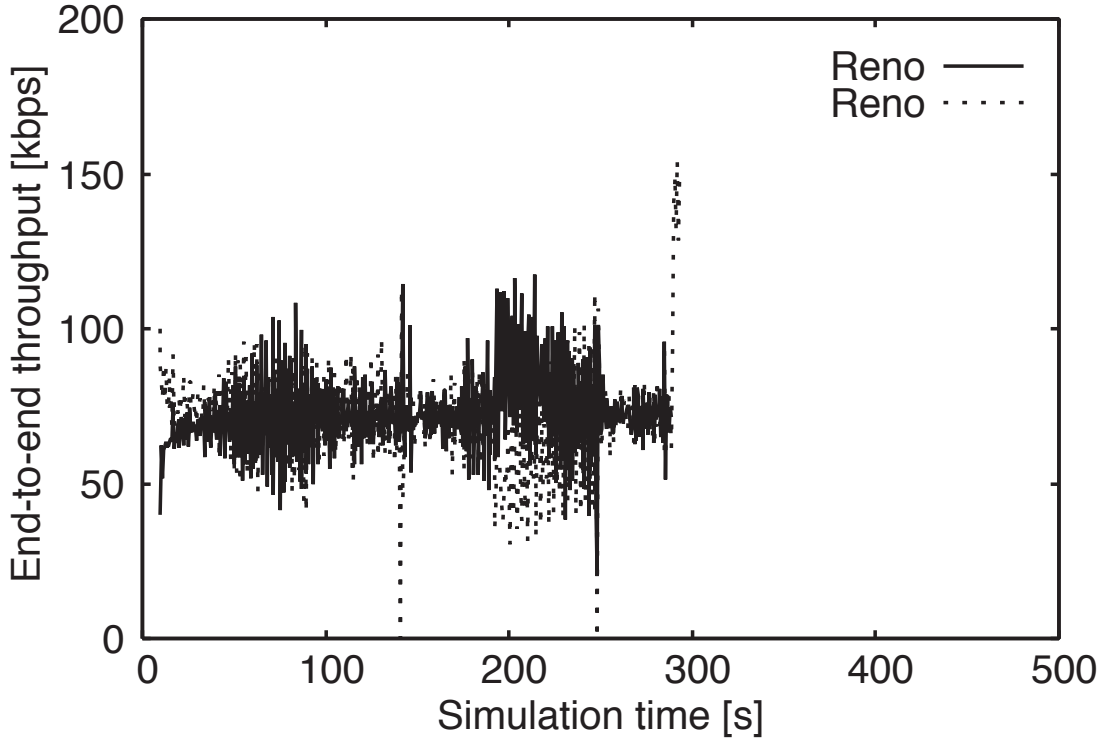

図 4.25 シミュレーション 3 におけるスループット（TCP Reno）〈発表文献 1.2〉

表 4.7 シミュレーション 3 における評価値（TCP Reno）〈発表文献 1.2〉

	TCP Reno	TCP Reno
Throughput	70.700 kbps	69.626 kbps
Stability Index	0.18368	0.17109
Fairness Index	0.99994	

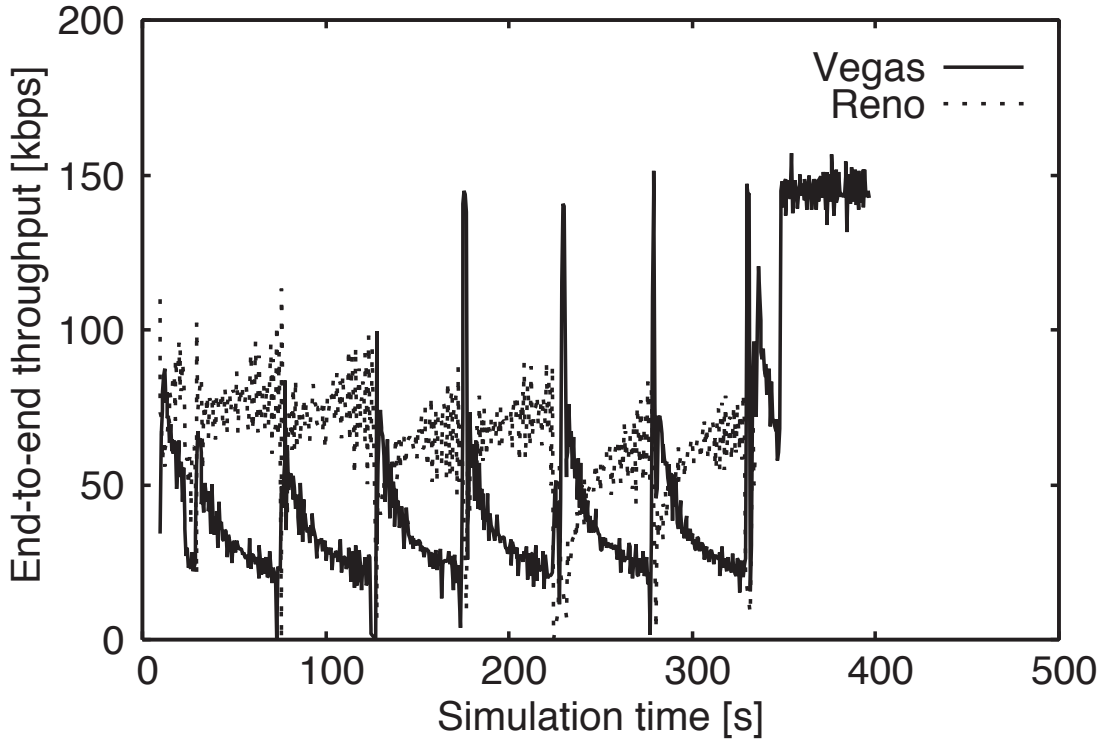

図 4.26 シミュレーション 3 におけるスループット（TCP Vegas）〈発表文献 1.2〉

表 4.8 シミュレーション 3 における評価値（TCP Vegas）〈発表文献 1.2〉

	TCP Vegas	TCP Reno
Throughput	51.410 kbps	61.058 kbps
Stability Index	0.77916	0.24786
Fairness Index	0.99269	

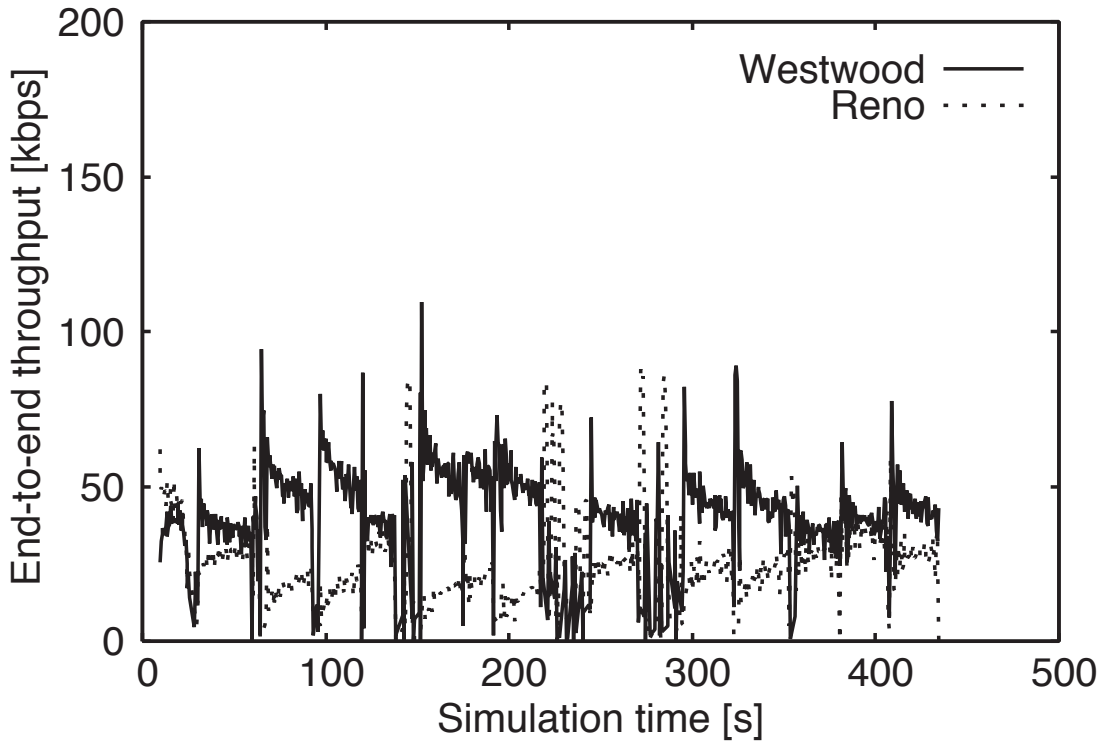

図 4.27 シミュレーション 3 におけるスループット（TCP Westwood） 〈発表文献 1.2〉

表 4.9 シミュレーション 3 における評価値（TCP Westwood） 〈発表文献 1.2〉

	TCP Westwood	TCP Reno
Throughput	48.203 kbps	40.319 kbps
Stability Index	0.25208	0.55337
Fairness Index	0.99213	

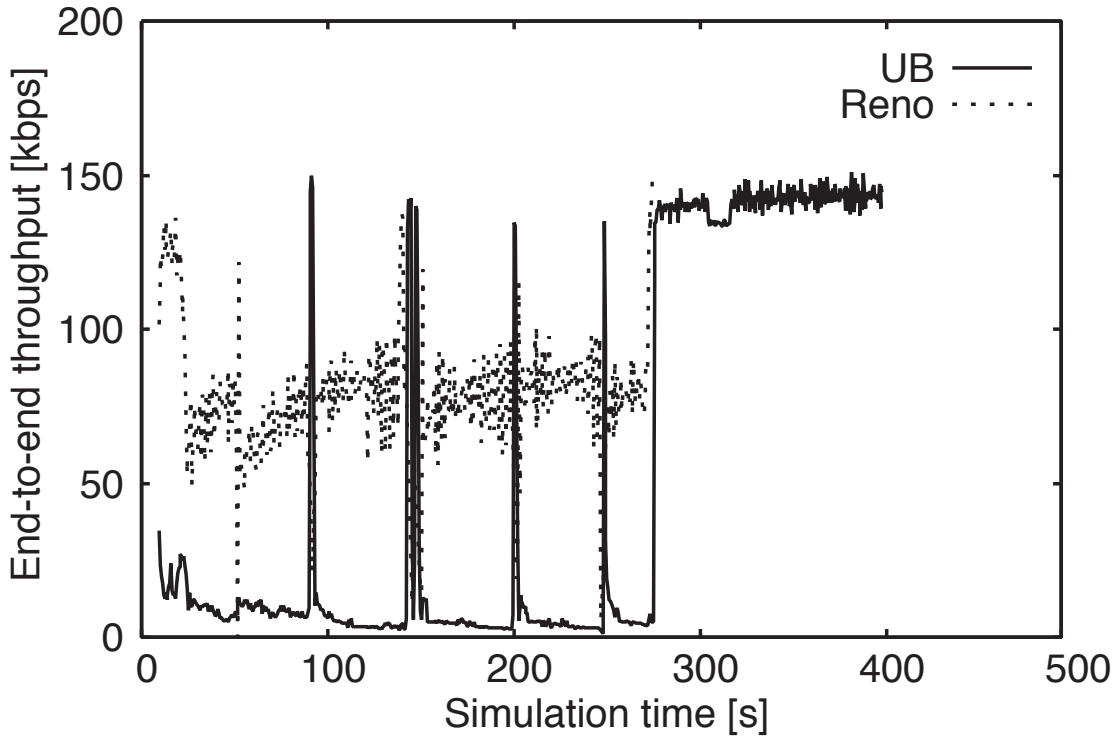

図 4.28 シミュレーション 3 におけるスループット（TCP UB） 〈発表文献 1.2〉

表 4.10 シミュレーション 3 における評価値（TCP UB） 〈発表文献 1.2〉

	TCP UB	TCP Reno
Throughput	51.307 kbps	74.363 kbps
Stability Index	1.2117	0.24600
Fairness Index	0.96743	

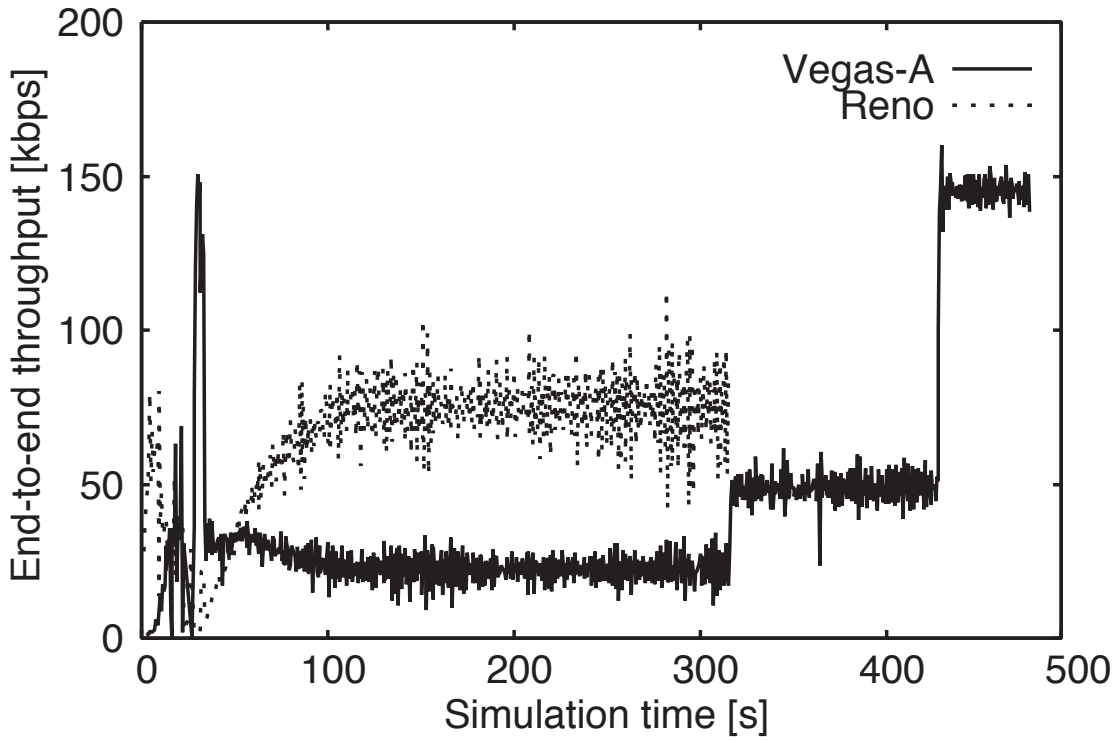

図 4.29 シミュレーション 3 におけるスループット（TCP Vegas-A） 〈発表文献 1.2〉

表 4.11 シミュレーション 3 における評価値（TCP Vegas-A） 〈発表文献 1.2〉

	TCP Vegas-A	TCP Reno
Throughput	51.668 kbps	66.149 kbps
Stability Index	0.95656	0.21662
Fairness Index	0.98512	

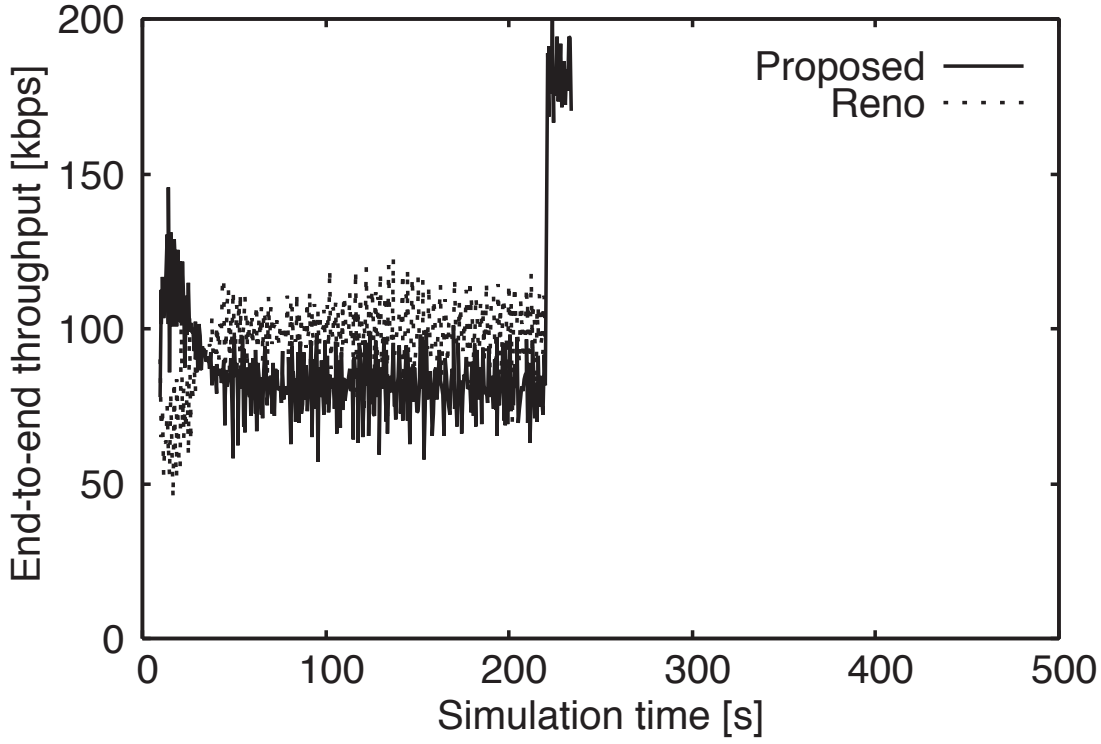

図 4.30 シミュレーション 3 におけるスループット（提案手法） 〈発表文献 1.2〉

表 4.12 シミュレーション 3 における評価値（提案手法） 〈発表文献 1.2〉

	Proposed method	TCP Reno
Throughput	87.198 kbps	92.846 kbps
Stability Index	0.30045	0.30700
Fairness Index	0.99901	

4.5 むすび

本章では，アドホックネットワークにおける通信特性を考慮したRTT駆動形トランスポートプロトコルを提案し，性能評価を行った．従来手法であるTCP Renoなどの ACK 駆動形プロトコルでは，セグメント損失を輻輳の発生と誤認し，送信レートが過剰に低下するという問題があった．また，TCP Vegas などの RTT 駆動形プロトコルでは，ネットワークの状態が随時変化する無線アドホックネットワークにおいて正確なスループットの見積もりや状態把握を行うことが困難であった．更に，ネットワーク帯域，端末資源の乏しい無線アドホックネットワークでは，資源の有効活用のため，送信セグメント数を低減し，公平性を保った通信を行うことが期待される．そこで提案手法では，これらの問題に対して，RTT の相対変化量を用いたウィンドウ制御方式，再送制御の改良によりネットワークの状態を把握した送信速度制御を実現することに成功した．更に，端末間，及びプロトコル間の公平性を確保した安定的な通信を実現できていることを，コンピュータシミュレーションにより示した．

第5章　送信速度制御を用いた負荷分散手法

5.1　まえがき

端末の移動性が高い無線アドホックネットワークでは，ネットワークトポロジーが大きく変化するため，あらかじめ経路表を作成しておくOLSRなどのプロアクティブ形ルーチングプロトコルではなく，AODVやDSRなどのリアクティブ形ルーチングプロトコルを利用した経路探索行う．しかし，これらのルーチングプロトコルでは，通信環境が安定している経路を優先的に利用するため，ネットワーク内の特定の箇所や端末に通信が集中することとなる．そのため，通信が集中した箇所ではより多くの送受信や転送が行われ，電力や無線資源を消費することとなり，ネットワーク内で公平性が低下する問題がある．

これらの問題に対し，空間的に負荷分散を実現するため，負荷に基づいた経路探索を行う機能を既存のルーチングプロトコルに導入したLBDSR（Load Balanced Dynamic Source Routing）[51]やLBPSR（Load Balance Routing using Packet Success Rate） [52]が提案されている．しかし，これらのプロトコルは通信途中での動的な経路変更に対応していないため，端末の移動に伴うトポロジーの変化や短い周期で変動する負荷に対応することが困難である．一方，通信中に経路変更を行う手法として，通信経路上の中継端末の転送パケット数に基づいて近傍端末と中継機能切換を行う手法[53]が提案されている．しかし，この手法では中継機能切換のために通信が一時的に中断されることや，切換制御のためのメッセージ交換による新たな負荷の増加が問題となる．また，端末密度が低い場合には切換候補となる端末が存在せず，適切な制御が行われないなど，端末密度によって性能が変化する問題もある．更に，時間的に負荷分散を実現する手法として，無線アドホックネットワークでの通信特性を考慮した送信速度制御手法 [39]–[46]，[54]–[61]が提案されている．しかし，これらの手法はスループットの向上を主な目的としているため，負荷分散の観点からは必ずしも適切な制御とならない場合がある．

本章では，無線アドホックネットワークにおけるこれらの問題に対応した負荷分散手法を提案する．提案手法では，通信経路上にある端末の状態を把握すると同時に，経路の近傍端末の通信状態を漏れ聞きによって認識し，通信の送信速度制御を行うことで，経路制御を用いることなく時間的に負荷分散を実現する．また，定期的な送信速度制御によって短い周期で負荷が変動するような環境においても効率的な負荷分散を実現している．コンピュータシミュレーションを用いて，従来手法との評価比較を行うことで，提案手法の有効性を明らかにする．

5.2　無線アドホックネットワークにおける負荷分散

5.2.1　無線アドホックネットワークにおける負荷分散

前述のように，無線アドホックネットワークにはネットワークを統合的に管理する機構が備わっていない．そのため，経路制御や送信速度制御によって自律的に負荷分散を実現する手法が多く提案されている．

経路制御によって空間的に負荷分散を行う手法では，遅延時間を基に制御を行う遅延形 (Delay-based)，トラヒック量などを基に制御を行うトラヒック形 (Traffic-based)，遅延形とトラヒック形を組み合わせたハイブリッド形 (Hybrid-based) の 3 種類が存在する [64]．遅延形の代表的な手法として，AODV を改良し，経路の遅延時間及びホップ数から経路選択を行う LAOR（Load-Aware On-Demand Routing）[65] が提案されている．また，トラヒック形手法として，前述の LBDSR や LBPSR，中継機能切替手法のほかに，ABR （Associativity Based Routing）[66]，LBAR（Load Balanced Ad Hoc Routing）[67]，TSA（Traffic-Size Aware Scheme）[68] などが提案されている．更に，ハイブリッド形手法として，DSR に改良を加え，データリンク層の情報，待ち行列長やホップ数を基に経路選択を行う CSLAR（Contention Sensitive Load Aware Routing）[69] が提案されている．しかし，遅延形の手法では，経路間のホップ数が異なる環境やパケット損失が多く発生するような環境において正確な負荷の見積もりが困難である．また，ハイブリッド形の手法では，煩雑な制御を行うことから，計算負荷が増大し，新たな負荷が発生する可能性がある．

送信速度制御によって時間的に負荷分散を実現する手法としては，無線アドホックネットワーク向けに TCP に改良を加えた手法 [39]–[46]，[54]–[59] やデータリンク層でのスロット割り当てを変更する手法 [60]，[61] が提案されている．ここで，時間的負荷分散とは，過剰なパケット送出やトラヒック集中による衝突やバッファ溢れなどを送信速度制御によって回避することで，通信効率の向上を実現する手法である．アドホックネットワーク向けの TCP では，伝送経路上の残留セグメント数に基づく制御などにより，無線マルチホップ通信に最適化された送信速度制御を行うことで通信効率向上を達成している．そのため，通信効率の向上と同時に不要な負荷の発生を時間的に低減することができると考えられる．一方，スロット割り当てによる手法はデータリンク層で制御が行われるため，通信経路全体や経路近傍など広範囲の通信状態を認識することが困難であり，負荷分散が局所的となる問題がある．

以上の特徴を考慮し，本章では，負荷の見積もりが比較的容易で複雑な制御を必要としないトラヒック形の負荷分散手法と無線マルチホップ通信に最適化された TCP に焦点をあて，検討を行う．

5.2.2　経路制御による負荷分散手法

無線アドホックネットワークでは，**5.2.1** で述べたように経路制御によって空間的に負荷分散を実現する手法が提案されている．LBDSR では，DSR の経路探索を改良し，端末及び経路の状態に基づく経路選択を行うことで負荷分散を実現している．経路を選択する際，DSR が経路長のみを考慮するのに対し，LBDSR は経路の新しさ，残留電力量，トラヒック量，経路長を基に探索された複数経路の優先度を算出し，最も優先度の高い経路を通信経路として選択する．図 5.1 の例では，経路 2 が通信経路として選択されるが，これは各探索経路の優先度 P_i が最も高い経路を通信経路として選択するためである．これにより通信環境の良い経路を優先的に利用することが可能となり，負荷の集中している箇所を避けた経路選択を行う．

LBPSR では，データリンク層とネットワーク層を組み合わせたクロスレイヤ制御によって負荷分散を行う手法が提案されている．LBPSR では，各リンクにおける 1 フレーム送信あたりの RTS 送信回数の逆数を通信成功率（PSR：Packet Success Rate）と定義し，経路全体の通信成功率が最も高い経路を通信経路として利用することで負荷分散を実現している．図 5.2 の例では，最短経路である経路 2 ではなく，通信成功率が最も高い経路 1 が通信経路として選択される．これは，トラヒック集中によって通信負荷が高くなっている箇所では，データリンク層において 1 回の RTS に対して CTS が返答される確率が低くなり，1 フレームを送信するために必要な RTS 送信回数が増加するためである．

文献 [53] では，中継端末における累積転送パケット数に応じて，近傍端末を経由する経路に動的に切り換える手法が提案されている．また，本章では，本手法を LBRN（Load Balancing on Relay Node）と呼ぶこととする．LBRN では AODV を基に，通信途中に中継処理を経路の近傍端末に移譲することで，各端末が転送

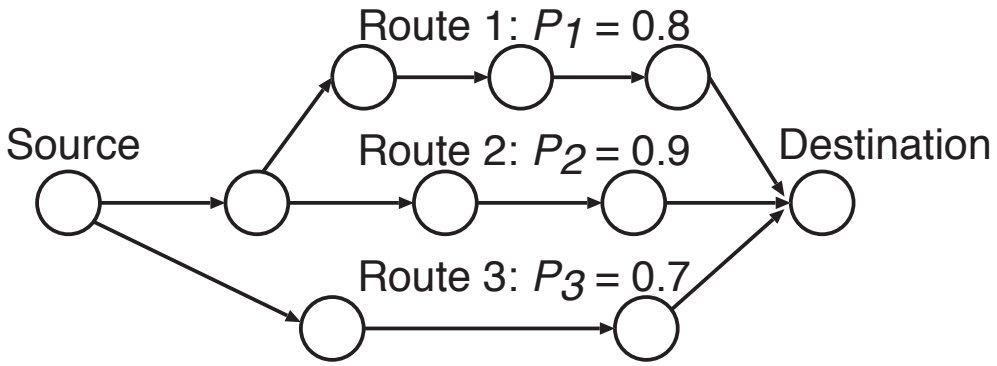

図 5.1　LBDSR での経路探索例 〈発表文献 1.1〉

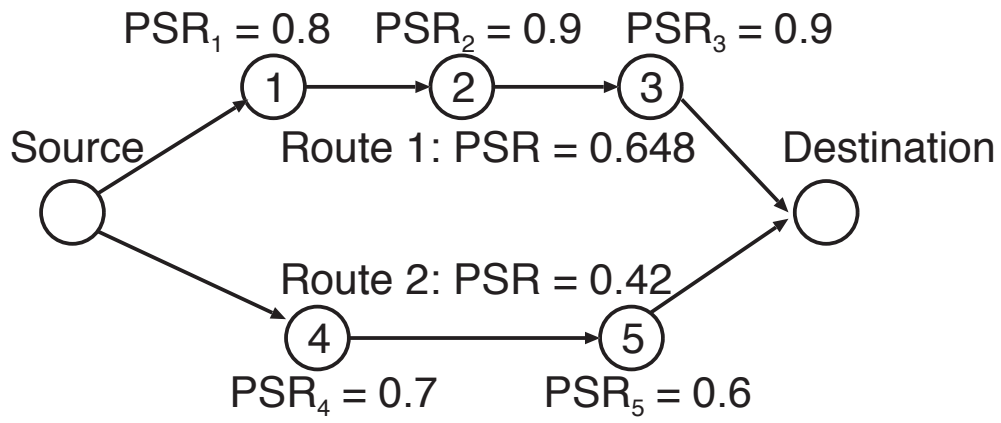

図 5.2 LBPSR の経路探索例 〈発表文献 1.1〉

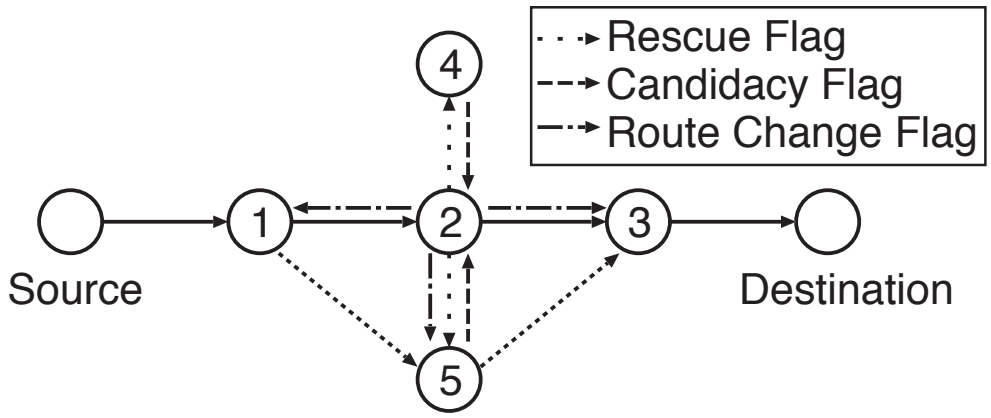

図 5.3 LBRN の動作例 〈発表文献 1.1〉

するパケット数の均一化を行い，負荷分散を実現している．図 5.3 の例では，端末 2 の中継処理を近傍端末である端末 5 に移譲し，1–2–3 の経路から 1–5–3 へと変更している．また，LBRN では経路切換に必要な情報を Hello メッセージの交換によって取得することで，必要最低限のメッセージでの経路切換を実現している．

このように，LBDSR と LBPSR では負荷に基づく経路選択を行うことで負荷分散を実現しているが，経路構築後の動的な負荷分散に対応していないため，負荷が短い周期で変動した場合には対応することが困難である．また，LBRN では，負荷分散効果が端末密度に依存することや，転送パケット数のみを負荷の指標としているため，経路切換によって通信環境が劣化する可能性がある．更に，これらの 3 手法では，通信経路上のみの状態から負荷分散制御を行うため，経路制御が他の近傍端末へ影響を及ぼす可能性がある．

5.2.3 送信速度制御による負荷分散手法

5.2.1 で述べたように，既存の TCP に変更を加えることで，無線アドホックネットワークでの無線マルチホップ環境に最適化した通信を行うプロトコルが数多く提案されている．これらの手法では，無線アドホックネットワークで発生するビット誤りによって輻輳ウィンドウが減少してしまう問題に対し，パケット損失の発生に依存しない輻輳ウィンドウ制御を導入することで送信速度制御を行い，通信効率の向上を達成している．

TCP Vegas では，エンドエンド間の RTT を利用した送信速度制御が提案されている．TCP Vegas では，最小 RTT と観測した RTT から経路上に滞留しているセグメント数を予測し，それらが一定範囲内におさまるようにウィンドウ制御を行う．本方式では，セグメント損失が直接ウィンドウ制御に影響しないため，セグメント損失が頻発する無線アドホックネットワークでは，従来の TCP と比較して，通信効率の向上を実現できる．

文献 [62] では，TCP Vegas を改良し，無線アドホックネットワークの通信特性に適応した TCP Vegas_M が提案されている．TCP Vegas_M では，ファジィ理論を用いて通信状態を評価し，それに応じたウィンドウ制御を行うことで，通信環境に適応した送信速度制御を実現している．そのため，TCP Vegas と比較して，より適切な送信速度制御が可能となり，更なる通信効率の向上を実現している．

TCP–AP [63] では，Adaptive pacing を用いて TCP の送信速度制御を行うことで，電波干渉の発生を低減し，スループットの改善を実現する手法が提案されている．一般に，端末が送出した電波は 3 ホップ先までの通信に影響を及ぼすとされている．そこで TCP–AP では，RTT を用いて 4 ホップ分の伝搬遅延を予測し，送信速度制御を行う手法が提案されている．これにより，電波干渉による再送を低減し，通信効率及び公平性の向上を実現している．

その他，文献 [41]–[46] では，TCP のウィンドウ制御を無線アドホックネットワーク向けに改良することによって，不必要な輻輳ウィンドウサイズの減少や過剰なトラヒックの送出を回避し，通信効率と公平性の向上を達成している．また，文献 [54]–[56] では，RED（Random early detection）や Adaptive pacing を用いた制御の導入が，文献 [57]，[58] では，データリンク層での制御を改善することで TCP の性能改善を行う手法が，それぞれ提案されている．

このように，無線アドホックネットワーク向け TCP では，精度を高めた輻輳制御を行うことで，通信環境に適応した送信制御を実現し，通信効率の向上を達成している．また，適切な送信制御を行うことで，待ち行列長の増加を抑制し，時間的に負荷低減効果を得ることが可能となる．しかし，これらの手法は通信効率，つまりスループットの改善を主な目的として提案されているため，通信経路上以外の通信状態については考慮されていない．よって，可能な限り帯域を確保する

よう動作するため，ネットワーク全体の負荷分散を達成するのは困難であると考えられる．

5.3 経路近傍の通信状態に基づく送信速度制御

前述の通り，従来手法では，主に通信経路上でのみ制御を行っているため，負荷分散の効果が限定的であり，また短い周期での負荷変動に対応することも困難である．そこで本章では，通信経路上の端末だけでなく，経路近傍に存在する端末の通信状態も考慮した送信速度制御を行うことで，通信環境の変化により柔軟に対応可能な負荷分散手法を提案する．

5.3.1 経路負荷の評価

通信経路上で負荷が集中している箇所では，単位時間当たりのトラヒックが多くなるため，無線通信環境では衝突が発生する可能性が高くなる．また，大量のパケット送信が行われるため，中継端末上の待ち行列長が増加する．そこで提案手法では，通信経路上の負荷を衝突発生回数と待ち行列長を用いて検出する．ここで，衝突発生回数は物理層からの情報を基に測定し，待ち行列長はネットワーク層から情報を取得することとする．提案手法では，自端末の衝突発生回数と待ち行列長を Hello メッセージに付加してブロードキャストすることで，定期的に自端末の状態を近傍端末へ通知する．通信経路上の中継端末は，他の中継端末から受信した Hello メッセージ内の衝突発生回数と待ち行列長の情報を基に，Hello メッセージを受信するごとに自端末の次ホップ端末に対する通信負荷の相違度を以下の式 5.1，5.2 を用いて算出する．

$$Z_{i,j} = \max\{|f(C_i, C_j)|, |f(Q_i, Q_j)|\} \tag{5.1}$$

$$f(x, y) = \frac{x - y}{x + y} \tag{5.2}$$

ここで，$Z_{i,j}$ は端末 i の次ホップ端末 j に対する通信負荷の相違度，C_i，Q_i はそれぞれ端末 i の衝突発生回数と待ち行列長を表す．更に，$f(x, y)$ は変動係数を参考にして作成した，2 値の差を $[-1, 1]$ の範囲へマッピングする関数であり，入力された 2 値の差が大きいほど 1，または -1 に近く，差が小さいほど 0 に近い値を示す．算出された $Z_{i,j}$ は，自端末の通信負荷に対する次ホップ端末の通信負荷の差を割合で示している．$Z_{i,j}$ が 1 に近い場合には端末間で通信負荷の差が大きく，0 に近い場合には差が小さいことを表す．また，一般に無線通信では，伝送誤りが衝突発生によるものか雑音やフェージングの影響によるものか区別することが困難なため，文献 [70]–[77] ではタイムスロットの監視や受信信号強度から干渉発生を検出する手法が提案されている．そこで提案手法では，衝突発生を検知する仕組みとして，受動的な衝突検知が可能な信号対干渉雑音比（SINR：Signal to Noise

Interference Ratio）の変化から識別を行う手法 [77] を用いる．このように，提案手法では近傍端末との通信負荷の相違度を算出して，負荷の集中を検出する

5.3.2 近傍の通信状態把握

提案手法では，負荷分散制御のために経路の近傍にある端末の通信状態把握を行う．無線アドホックネットワークでは無線通信によってデータ送信が行われるため，実際の受信端末以外に電波範囲内にある端末へも電波が到達し，衝突の発生を引き起こす可能性がある．また，データリンク層において個々の受信フレームのあて先を判別したり，衝突によって損失したフレームを再送する必要があるため，新たな負荷が発生する可能性もある．そのため，自端末が近傍端末よりも多くのフレームを送信している場合には，自身の通信によって近傍端末へ負荷を与えることとなる．反対に，自端末よりも近傍端末がより多くの通信を行っている場合には，自身の通信によって与える負荷と比較して，近傍端末によって与えられる負荷が相対的に増加する．そこで提案手法では，近傍端末が送信したフレームの漏れ聞きを行い，自身の通信状態と比較することで，近傍の通信状態を評価する．提案手法でのフレーム漏れ聞きには，データリンク層で受信される自端末あてのフレームを除いたフレーム数を用い，一定時間ごとに近傍端末ごとの漏れ聞きパケット数をテーブルに記録する．テーブルに記録した各近傍端末の漏れ聞きパケット数，及び自端末の送信フレーム数から，以下の式 5.3，5.4 を用いて評価値を算出する．

$$E_i = \frac{1}{|N_i|} \sum_{j \in N_i} e_{i,j} \quad (|N_i| \neq 0) \tag{5.3}$$

$$e_{i,j} = f(p_i, \hat{p_j}) \tag{5.4}$$

ここで，$e_{i,j}$ は端末 i の端末 j に対する評価値，p_i は端末 i の一定時間ごとの送信フレーム数，$\hat{p_j}$ は漏れ聞きによって予測した一定時間ごとの近傍端末 j の送信フレーム数，N_i は端末 i に観測された近傍端末の集合，E_i は各近傍端末の評価値の平均を表す．近傍端末の評価は，自端末の送信フレーム数と近傍端末の送信フレーム数を式 5.2 を用いて評価することで，送信フレーム数の差が小さい場合には 0，自端末の方が多い場合には 1，近傍端末の方が多い場合には -1 に近い値を示す．各端末は，一定時間ごとにすべての近傍端末に対して評価を行い，評価値の平均値 E_i を算出する．ここで，E_i が 1 に近いほど自端末が近傍端末と比較してより多く通信を行っていることを表し，-1 に近いほど近傍端末による通信が多いことを表す．提案手法では，算出した E_i を送信速度制御に反映させ，近傍の通信状態に応じた負荷分散制御を実現する．

5.3.3 送信速度制御

提案手法では，**5.3.1** で求めた次ホップ端末との通信負荷相違度と **5.3.2** で求めた近傍端末に対する平均評価値を用いて送信速度制御を行う．そこで，通信経路上の端末は，$Z_{i,j}$ と E_i を定期的に送信元へ通知する．Z_{ij}，E_i を通知するための制御メッセージは，あて先から送信元へ転送され，その際に経路上の端末は自身の情報を付加し，上流端末へ転送を行う．あて先端末 d までの通信経路上のすべての端末からこれらの評価値を受け取った送信元端末 s は，次ホップに対する通信負荷相違度の平均値 Z_{sd}，及び近傍端末に対する評価値の平均値 E_{sd} を以下の式 5.5，5.6 を用いて算出する．

$$Z_{\mathrm{sd}} = \frac{1}{|R_{\mathrm{sd}}| - 1} \sum_{i \in R_{\mathrm{sd}}, i \neq d} Z_{i,n(i,d)} \tag{5.5}$$

$$E_{\mathrm{sd}} = \frac{1}{|R_{\mathrm{sd}}|} \sum_{i \in R_{\mathrm{sd}}} E_i \tag{5.6}$$

ここで，R_{sd} は端末 s を送信元とするあて先端末 d までの経路に含まれる端末の集合，$n(i,d)$ は端末 i からあて先端末 d へパケット送信を行う際の次ホップ端末を表す．また，Z_{sd}，E_{sd} は通信経路上の端末間，及び経路上の端末と近傍端末との相対的な通信量の割合を表し，経路上の過剰なトラヒックの相対割合を示す．送信端末は，これらの値を用いて，以下の式 5.7 に従って TCP の輻輳ウィンドウサイズの上限値 c_{limit} を設定する．

$$c_{\mathrm{limit}} = c_{\max} \{1 - \max(Z_{\mathrm{sd}}, E_{\mathrm{sd}})\} \tag{5.7}$$

ここで，$c_{\max}$ は通常の輻輳ウィンドウサイズの上限値を表す．

図 5.4 に提案手法の適用例を示す．通信負荷相違度の平均値 Z_{sd} が大きい場合には，通信経路上に通信負荷の差が大きい端末が含まれており，負荷集中が発生していると判断できる．この場合には，最大送信速度を $(1 - Z_{\mathrm{sd}})$ 倍することによって過剰なトラヒックを制限し，負荷の低減を図る．反対に，Z_{sd} が 0 に近い場合には，経路上の通信環境に差異がないと判断し，送信速度の制限を緩和することで不要な通信効率の低下を回避することが可能である．更に，近傍端末に対する評価値の平均値 E_{sd} が大きい場合には，通信経路上の端末からの通信負荷が高い状態を表し，通信経路上の通信によって，近傍へ影響を及ぼす可能性があると判断できる．この場合にも，最大送信速度を低下させることによって過剰なトラヒックを制限し，近傍に及ぼす影響を低減する．これらの制御が理想的に動作した場合，電波干渉の発生，及び待ち行列長を最大で $(1 - Z_{\mathrm{sd}})$ 倍，または $(1 - E_{\mathrm{sd}})$ 倍にまで低減することが可能となり，時間的な負荷分散を実現できる．また，それに伴って再送や遅延が減少するため，通信効率の向上も同時に達成できる．

このように，通信経路及び経路の近傍端末の状態に基づいて送信速度制御を行うことで，経路変更を全く伴わない負荷分散制御を実現することが可能となる．また，定期的に送信速度制御を実行することで，通信途中での動的な負荷分散制御も可能となる．そのため，短い周期で負荷が変動する場合や端末の移動性が高い場合にも対応できるという利点がある．

5.4 シミュレーション評価

5.4.1 シミュレーション条件

本論文では，コンピュータシミュレーションを用いて，負荷分散を行わない通信，従来手法，及び提案手法の性能比較を行う．従来手法には，経路制御による負荷分散手法である LBDSR，LBPSR，LBRN，送信速度制御手法である TCP Vegas, TCP Vegas_M，TCP–AP を用いる．更に，経路近傍の通信状態把握による性能向上を評価するため，経路負荷のみを用いて送信速度制御を行った場合との比較も行う．また，本論文では経路負荷のみを用いた送信速度制御を Proposed_NN と表記する．各シミュレーションに共通する条件は以下のとおりである．シミュレータには GloMoSim 2.03[48] を用いる．端末数は 100 とし，1,000m 四方のシミュレーション領域にランダムに配置する．無線通信方式に RTS/CTS を利用した IEEE

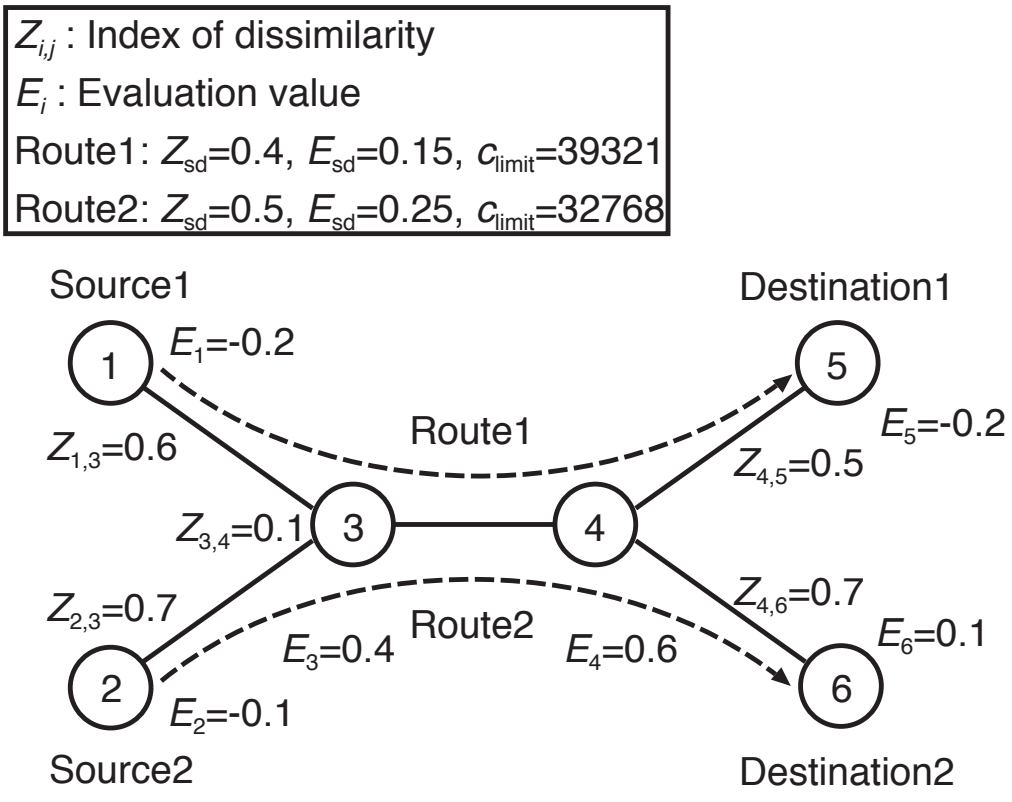

図 5.4 提案手法の動作例 〈発表文献 1.1〉

802.11b を用い，通信半径を 100m とする．ここで，通信半径は 1,024byte のパケット送信時の受信成功確率が 80%となる距離と定義する．また，本シミュレーションで用いる伝送路では，伝搬モデルに 2 波モデルを用いる．本シミュレーションにおける受信信号強度の減衰率変化点であるブレークポイント D_b は約 100m であり，伝送距離 d が $d \leq D_b$ の場合には d^2，$d > D_b$ の場合には d^4 に反比例して減衰し [78]，伝送途中に雑音がランダムに挿入されることで伝送誤りが生じる．また，提案手法での衝突発生検知には，**5.3.2** で述べたように，文献 [77] の手法を用いる．LBDSR 及び LBPSR 以外の手法では，ルーチングプロトコルに Hello メッセージを利用した AODV を用いる．メッセージの送信間隔は，AODV の初期値である 1 秒に設定する．また，提案手法での制御に必要な相違度 Z_{ij}，及び評価値 E_i を送信するための制御メッセージは，Hello メッセージと同様に AODV の機能として実装し，送信周期を 1 秒に設定する． TCP のバージョンは Reno を用い，最大輻輳ウィンドウサイズは 65,535，最大送信セグメントサイズは 1,460byte とする．本シミュレーションでは，情報配信形アプリケーションの利用を想定し，通信負荷，及びトポロジーの変化による性能への影響を調査するため，以下の 2 つのシミュレーションを用いて評価を行う．

(1) シミュレーション 1

シミュレーション 1 では，通信負荷を変化させた場合について評価を行う．全端末内から情報配信元となる 10 端末を選択し，残りの端末へ通信を行う．送信端末は指数分布に従いそれぞれ平均生起間隔 6 秒から 60 秒で，平均 1Mbyte，標準偏差 0.1414Mbyte の正規分布に従うデータをランダムに選択したあて先端末へ送信する．また，各端末の移動にはランダムウェイポイントを利用し，ウェイティングタイムを 20 秒，各端末の移動速度を 0m/s から 10m/s の間でランダムに設定する．シミュレーション 1 により，負荷が集中した環境での各手法の負荷分散効果及び通信効率へ与える影響について調査を行う．

(2) シミュレーション 2

シミュレーション 2 では，端末の移動によって発生するトポロジーの変化が性能に与える影響を評価する．各端末の移動にはランダムウェイポイントを利用し，ウェイティングタイムを 20 秒，各端末の移動速度を 0m/s から 20m/s で固定値として設定し，低速から高速まで幅広い移動速度下での通信特性を評価する．各送信端末は，平均生起間隔 60 秒の指数分布に従って，ランダムに選択したあて先端末へデータ送信を行う．送信データサイズは，平均 1Mbyte，標準偏差 0.1414Mbyte の正規分布に従うものとする．シミュレーション 2 により，トポロジー変化に伴う通信環境変化への各手法の対応能力を観察する．

評価指標として，衝突発生回数，転送パケット数，待ち行列長，スループットを用いる．トラヒック集中によって負荷が高くなっている箇所では，単位時間当たりに送信されるパケット数が増加するため，衝突発生の確率が高くなる．そのため，

衝突発生回数を評価指標として用いることで，どの程度負荷分散が実現されたかを表すことが可能である．転送パケット数は，他端末あての中継パケット数を表し，衝突発生回数と同様に負荷を表す指標として用いる．更に，待ち行列長は中継端末の送信待ち行列長を表し，負荷が集中している箇所では待ち行列長が増加するため，負荷集中を表す指標として用いる．また，負荷分散制御によって経路や送信速度が変化するため，通信効率への影響を評価するためにスループットを評価指標として用いることで，各手法が通信効率へ与える影響を調査する．スループットの算出には，すべてのパケット送信が完了したコネクションのみを用いる．

5.4.2 シミュレーション結果

（1）情報配信形アプリケーションでの評価結果

図 5.5〜図 5.8 に情報配信形アプリケーションを用いた場合の評価結果を示す．なお，送信が最後まで完了したコネクションの割合は，全手法とも約 80%から約 87%程度であった．

結果から，経路選択により空間的に負荷分散を行う LBDSR や LBPSR では，待ち行列長の低減を達成しているが，平均経路長が増加することから衝突発生回数，及び転送パケット数が増加し，スループットが低下している．LBRN では経路切換を行うことで，端末間での転送パケット数を均一化し，待ち行列長の低減を実現しているが，トラヒック量が多い場合には配信端末近傍のリンク利用率が高くなるため，経路切換による負荷低減効果の低下や制御メッセージの増加などの理由により，衝突発生回数が増加している．また，経路切換による遅延の発生やリンク品質の低下を考慮していない制御のため，スループットが低下する．TCP Vegas や TCP Vegas_M などの改良形 TCP では，送信速度制御により過剰なパケット送信を抑制することで時間的な負荷分散効果を得ることができる．そのため，単位時間あたりに送出されるパケット数が減少し，衝突発生を抑制することができている．また，空き帯域を効率的に利用するウィンドウ制御を行うため，スループットの向上も同時に達成している．しかし，経路内に滞留するパケット数が一定となるように送信速度制御を行うため，待ち行列長が増加している．TCP–AP では，Adaptive pacing を用いた伝搬遅延を考慮した送信速度制御によって，スループットの向上を達成している．しかし，シミュレーションで用いたようなランダムトポロジーや負荷が特定の箇所に集中している場合には，経路近傍に存在する端末の通信から大きな影響を受けるため，電波干渉の発生を抑制することが困難である．一方，提案手法では経路上及び近傍端末の通信状態に基づく送信速度制御を行っているため，従来手法と比較して効果的に負荷分散を実現している．従来手法では通信経路上の負荷やエンドエンド間での通信状態から負荷分散制御を行っていたのに対し，提案手法では通信経路近傍の通信状態を統合的に判断し，送信

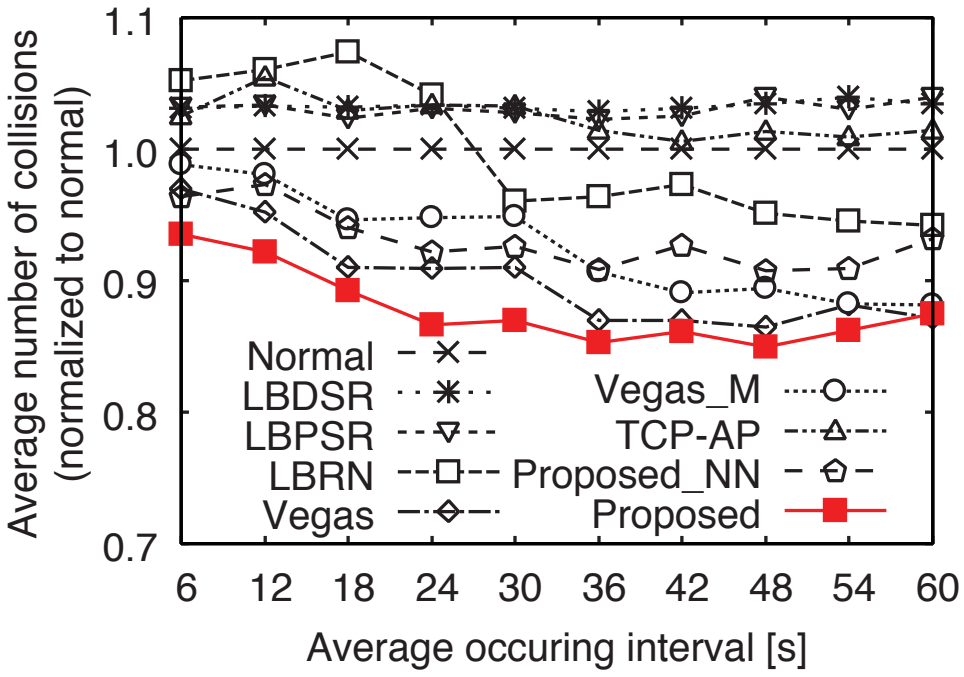

図 5.5 シミュレーション 1 における平均衝突発生回数 〈発表文献 1.1〉

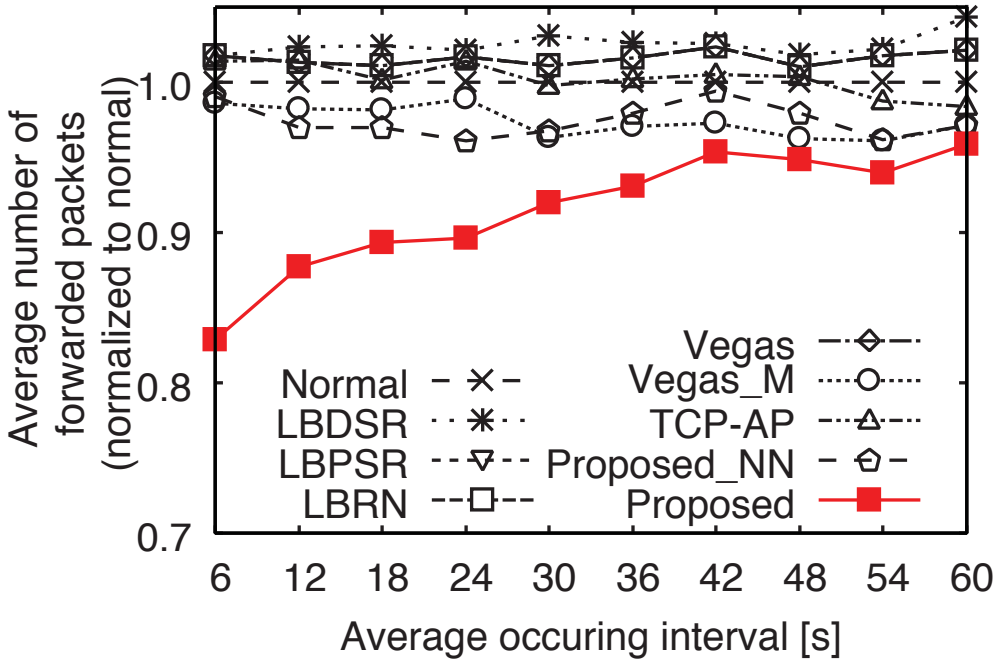

図 5.6 シミュレーション 1 における平均転送パケット数 〈発表文献 1.1〉

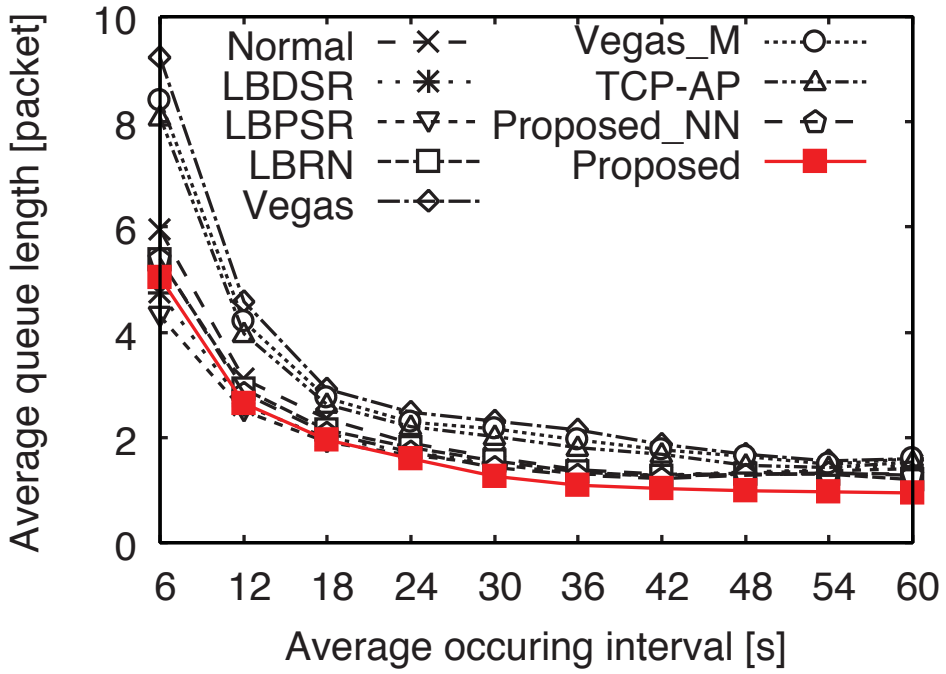

図 5.7 シミュレーション 1 における平均待ち行列長 〈発表文献 1.1〉

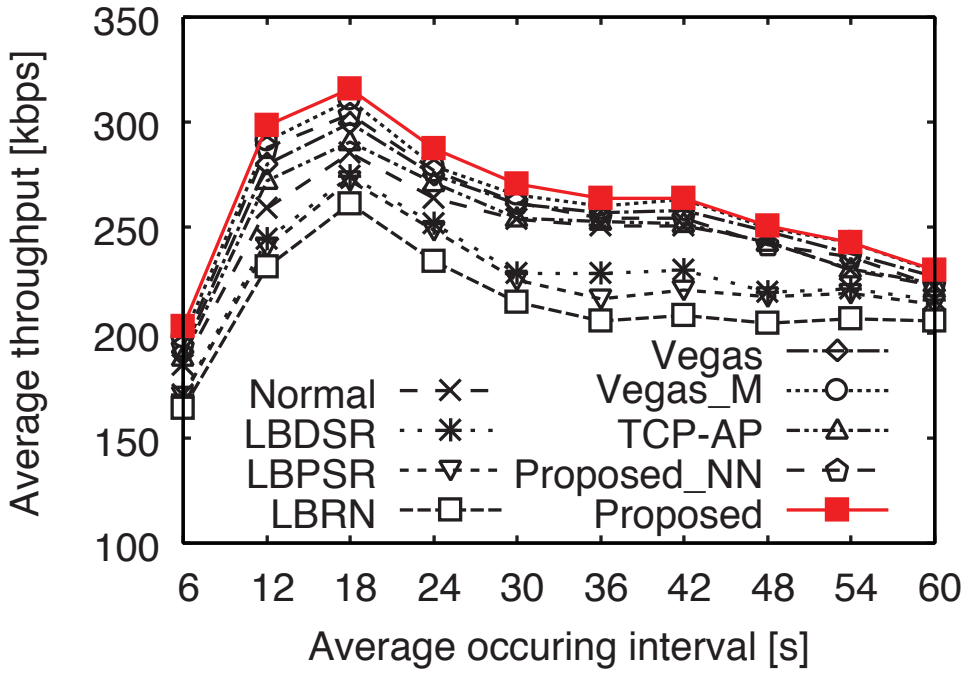

図 5.8 シミュレーション 1 における平均スループット 〈発表文献 1.1〉

速度制御を行うため，従来手法と比較して電波干渉，及びそれに伴う再送を低減している．また，送信速度を通信環境に適応させることが可能となるため，適切な送信速度での通信が可能となり，待ち行列長の低減を実現している．更に，従来の送信速度制御では確立した経路上で可能な限り高速でパケット送信を行うのに対し，提案手法では経路上の通信が近傍端末に与える影響を考慮して送信速度制御を行うため，より広範囲の通信環境に適応した通信が可能となり，通信効率，スループットが向上している．また，近傍の通信状態把握を用いない場合の提案手法においては，従来の送信速度制御と同様に経路上のみの通信状態把握となり，限定的な範囲の通信環境のみに適応した送信速度制御となるため，従来手法と同程度の性能改善となる．

（2）移動速度を変化させた場合の評価結果

図 5.9〜図 5.12 に移動速度を変化させた場合の評価結果を示す．なお，送信が最後まで完了したコネクション割合は，全手法とも約 92%から約 93%程度であった．

結果から，経路選択により空間的に負荷分散を行う LBDSR や LBPSR では，移動速度が上昇するに従ってネットワークトポロジーの変動に対応することが困難となり，負荷分散効果が低下している．また，通常のルーチングアルゴリズムと比較して平均経路長が長くなるという特徴から，移動速度が速い場合には負荷の低い適切な経路探索が困難となり，転送待ちパケットが待ち行列に蓄積される．経路切換による負荷分散を行う LBRN では，移動速度の上昇に伴って切換候補となる端末が自身の近傍に存在する確率が低下し，経路切換を行うことが困難となる．そのため，移動速度が速い場合には，通常のルーチングアルゴリズムとほぼ同等の通信特性となり，十分な負荷低減効果を得ることができない．送信速度制御を行う TCP Vegas と TCP Vegas_M では，移動速度に依らず，一定の負荷分散効果を得ることができているが，シミュレーション 1 と同様の理由から待ち行列長の増加が発生している．また，TCP–AP では，多数の通信が存在する環境では経路外の端末からの電波干渉によって伝搬遅延の見積もりが困難となり，適切な送信速度を設定することができず，十分な改善効果が得られない．一方，提案手法ではシミュレーション 1 の項で述べたように，経路近傍の通信環境も含めた送信速度制御を行っているため，従来手法と比較して，より効果的な負荷分散を実現している．更に，提案手法では経路近傍の通信状態を定期的に確認し，送信速度制御を行っているため，移動速度が上昇し，ネットワークトポロジーが変化した場合でも，変化した通信環境に適応した制御が可能である．ここで，移動速度の増加に従って各手法の待ち行列長が減少しているが，これは送信元端末からのパケット送信が失敗し，呼損が増加したためである．また，シミュレーション 1 の項で述べたように，近傍の通信状態把握を用いない提案手法では，限定的な範囲の通信状態に基づく送信速度制御となるため，従来の空間的負荷分散手法と同程度の性能改善となる．

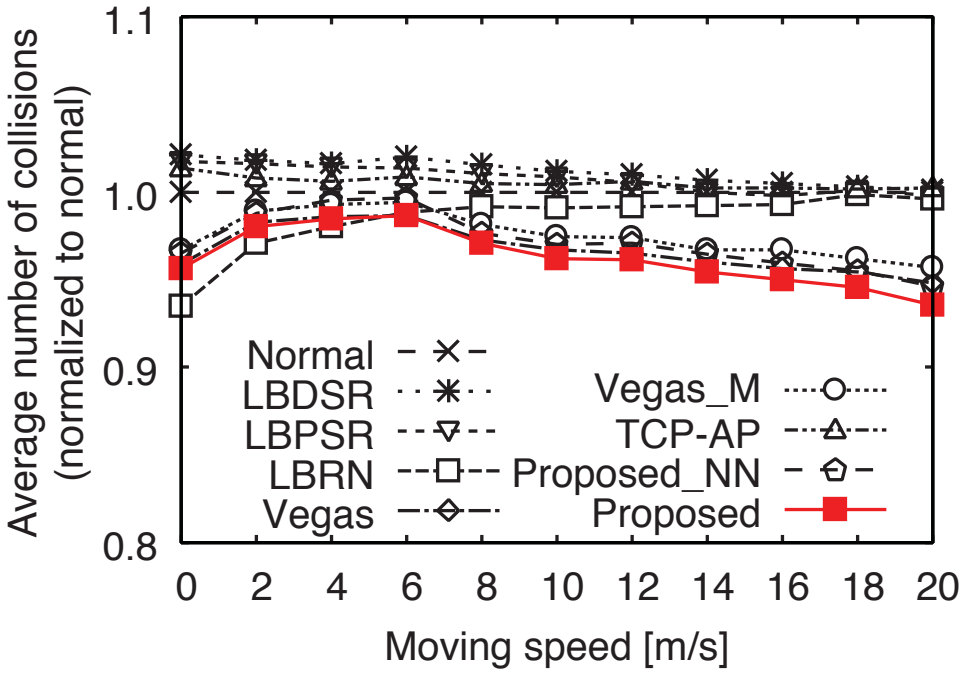

図 5.9 シミュレーション 2 における平均衝突発生回数 〈発表文献 1.1〉

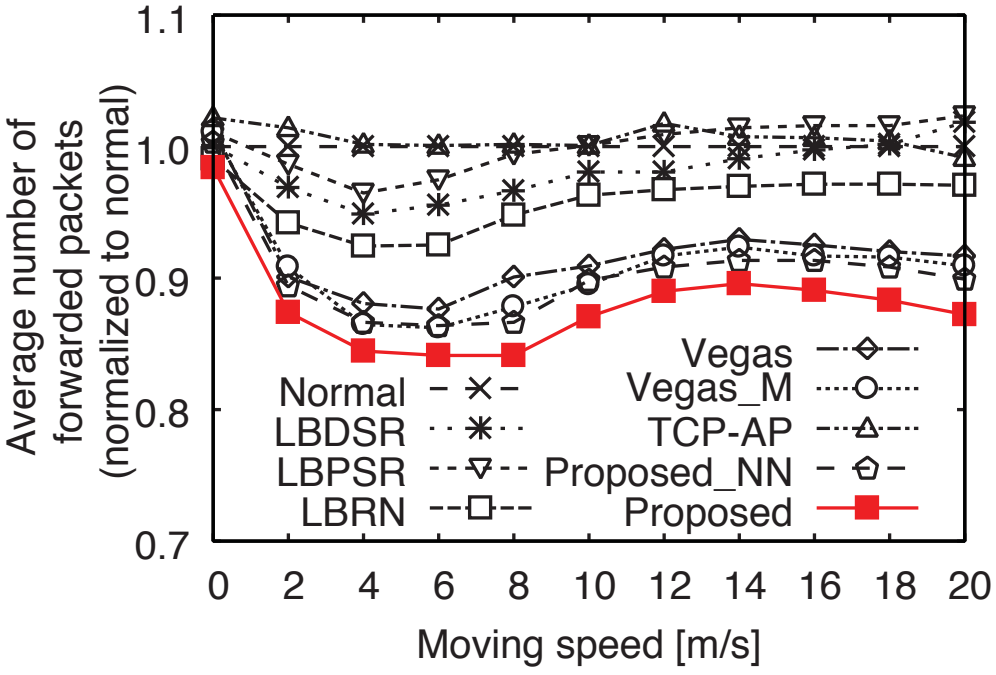

図 5.10 シミュレーション 2 における平均転送パケット数 〈発表文献 1.1〉

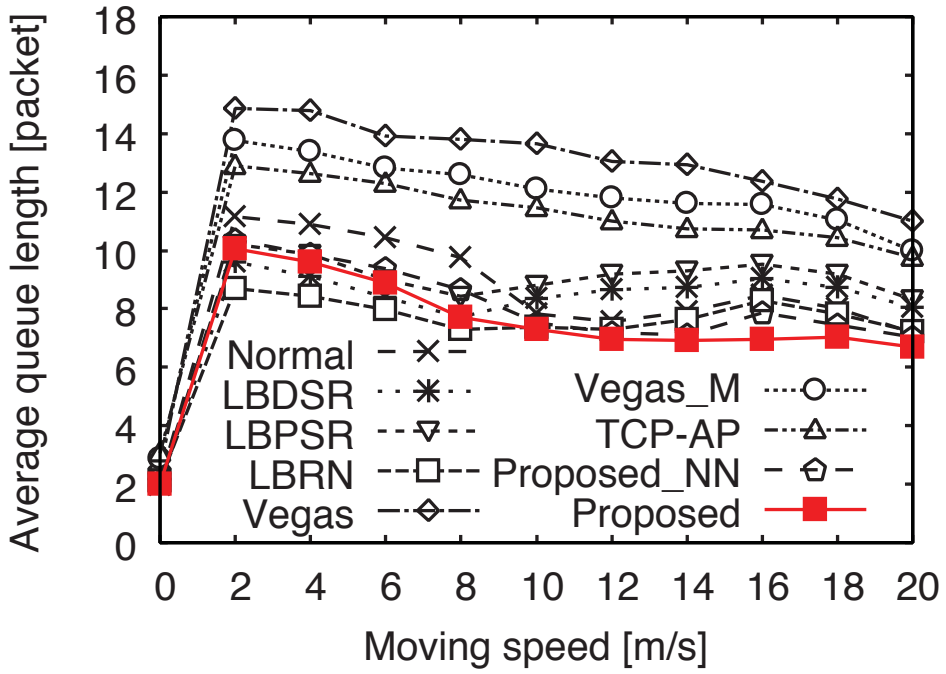

図 5.11 シミュレーション 2 における平均待ち行列長 〈発表文献 1.1〉

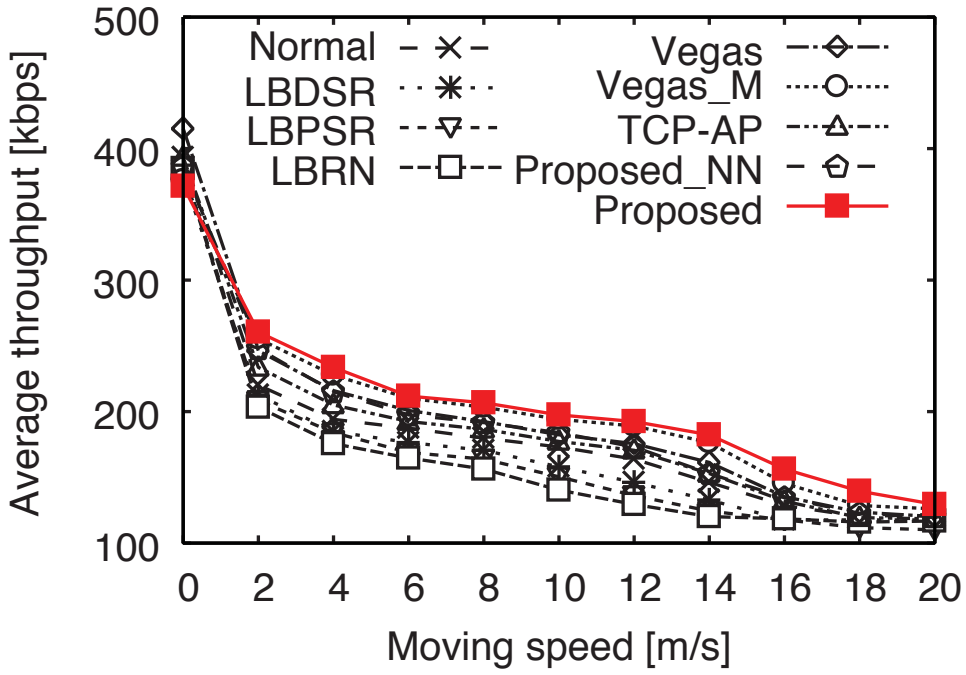

図 5.12 シミュレーション 2 における平均スループット 〈発表文献 1.1〉

5.5　むすび

本章では，無線アドホックネットワークにおいて送信速度制御によって時間的に負荷分散を実現する手法を提案し，性能評価を行った．従来の負荷分散手法では，主に通信経路上の状態を基に制御が行われていたため，ネットワーク全体では適切な負荷分散とならない問題があった．そこで提案手法では，この問題に対し，通信経路やその近傍の通信状態から負荷の集中を検出し，送信速度制御を行うことで負荷分散を実現した．

第6章　結論

6.1　本研究の主たる結果

近年，スマートフォンなどの小形端末に無線デバイスが搭載されており，それらのデバイスを利用することで無線マルチホップネットワークなどを構築することが技術的に可能になってきた．そのため，無線マルチホップネットワークの一種である無線アドホックネットワークにおいて，サービス実現のための効率的な通信方式の研究が盛んに行われている．無線アドホックネットワークでは，特定のインフラを利用することなく，端末同士で自律分散的にネットワークを構築することが可能であるため，街中などの情報配信サービスなどの他に，災害時などの通信手段が途絶したような環境での一時的なネットワーク構築に応用されることが想定されている．しかし，無線アドホックネットワークで実用的な通信を実現するためには，多くの課題が存在し，実用化には至っていない．そこで，本論文では，無線アドホックネットワークでの問題として，高いデータ損失率とそれに伴う通信性能の劣化，自律制御による負荷集中に着目し，研究を行った．以下では，本論文で論じた研究内容について述べる．

(1) 近傍端末を用いた自律分散再送制御手法

第 3 章では，無線アドホックネットワークで頻発するデータ損失に対し，データリンク層で再送制御を行うことで高信頼，低遅延な通信を実現する手法を提案した．従来，伝送途中にデータ損失が発生した場合，送信元端末が ACK の有無を利用してデータ損失の発生を検知していたが，ACK 受信のタイムアウトを待つ必要があることや伝送に失敗したリンクを再送のために再度利用することから，非効率な再送制御が行われたいた．これに対し，近傍端末がリンク上の通信を傍受することで通信の可否を判断し，損失が発生した場合に自律的に再送する手法が提案されている．しかし，従来手法では，フレーム単位で通信の成否を確認するため，連続したフレーム損失時に遅延が増加するなどの問題がある．また，一時的に経路を変更することでこれに対応した手法が提案されているが，中継回数の適切な設定が困難である問題があった．そこで，提案手法では，従来手法と同様に近傍端末を利用した再送制御を行うが，対象リンクの利用可能性を自律的に判断し，リンクが利用不能な場合のみ再送制御を行うことで，不必要な再送制御を行うことなく，通信効率を向上させることができた．

(2) 適応形トランスポートプロトコル

第 4 章では，無線アドホックネットワークで頻発するセグメント損失によって TCP の性能が低下する問題に対し，セグメント損失とウィンドウ制御を独立させた制御手法を提案した．一般的に用いられている TCP では，ACK を通信の確認

だけでなく，輻輳検知にも用いている．これは，TCP が有線通信などの安定した通信環境での利用が想定されているため，輻輳以外に起因するセグメント損失が発生する確率が低いためである．一方，無線アドホックネットワークでは輻輳以外にも端末の移動によるリンク切断や干渉などによってセグメント損失が発生することから，TCP による不必要な輻輳回避が行われることで，通信効率が低下する問題がある．これに対し，従来手法として，RTT や帯域推定を用いたウィンドウ制御手法が提案されている．しかし，従来手法では，時間に伴って通信環境が大きく変化した場合に，十分に性能を発揮することができないという問題があった．そこで，提案手法では，通信環境の変化を RTT の相対的な変化量から検出し，それに基づきウィンドウ制御を行うことで，無線アドホックネットワークでの通信効率向上を実現した．また，提案手法では，ウィンドウ制御とセグメント損失を切り離すことで，不必要な輻輳回避を抑制することが可能となり，帯域を有効に活用した通信を実現することができた．

(3) 送信速度制御を用いた負荷分散手法

第 5 章では，無線アドホックネットワークの自律分散的制御に起因する負荷集中に対し，送信速度制御を用いることで時間的に負荷分散を実現する手法を提案した．無線アドホックネットワークでは，経路制御や通信制御などが端末のみで自律分散的に行われるため，ネットワーク全体の通信状態を一元的に管理することは困難である．そのため，既存のルーチングプロトコルなどでは，通信環境が優れているリンクを優先的に利用するよう経路構築が行われるため，特定の端末やリンクに負荷が集中することとなる．この問題に対し，経路構築時に通信負荷の低い端末を優先的に中継端末として選択する手法や中継機能を経路の近傍端末と交換することで負荷分散を実現する手法が提案されている．しかし，経路選択による負荷分散では経路構築後に動的な制御が困難であることや，中継機能交換による手法では，負荷分散効果が端末密度に依存するなどの問題がある．そこで，提案手法では，従来手法のように空間的に負荷分散を実現するのではなく，送信速度制御によって時間的に負荷分散を実現する手法を提案した．提案手法では，通信経路及びその近傍端末の通信負荷に応じて定期的に送信速度制御を実行することにより，特定の端末に過度の通信負荷がかかることを回避し，時間的負荷分散を実現することができた．

6.2 今後の展望

無線アドホックネットワークは，研究段階の技術であり，実用化のためには様々な技術的課題を克服する必要がある．本論文では，それらの課題に対し，データリンク層，トランスポート層，及びクロスレイヤ制御による解決を図った．本論文では，三つの異なる視点から通信効率向上を実現したが，各手法を連携させる

ことで，より高い改善効果を得ることができると考えられる．しかし，各手法を連携させる場合には，パラメータ設定や連携方法などを新たに検討する必要がある．一方，無線アドホックネットワークは様々な場面に応用することが可能であるため，現在検討されている課題の他にも多くの実用上解決すべき課題が存在する．例として，センサネットワークは無線アドホックネットワークと同様のネットワーク形態，通信方式で構築可能であるが，センサネットワークでは端末の維持時間，つまり電力消費が重要な要素となってくる．そのため，従来研究が進められていた無線アドホックネットワークをそのまま適用することは困難であり，センサネットワークの通信特性，要求に適するように変更を加える，若しくは新たな手法を提案する必要がある．

一方，近年ではより柔軟に様々なネットワーク形態に対応するため，インターネットなどで用いられている TCP/IP の枠組みを超え，無線アドホックネットワークや DTN などでの利用を想定した，レイヤ横断形の通信制御方式の検討も進められている．このような統合的な通信方式を実現するにあたっても，基礎となる技術は従来より研究されてきている技術を基礎とするため，本論文の検討による結果が，無線アドホックネットワーク，及び類似技術の実用化の一助になれば幸いである．

謝 辞

本研究を進めるにあたり，数多くのご指導・ご助言を賜り，また日頃の打合せなど様々な場面において懇切丁寧なご指導をして頂きました，指導教員の田中良明教授に深く感謝致します．また，学会発表・学会活動の機会を与えてくださったとともに，原稿執筆に至るまでご指導を頂いたことも深く感謝致します．

芝浦工業大学システム理工学部電子情報システム学科の三好匠教授には，学部の頃から現在に至るまで懇切丁寧なご指導並びに有益なご助言を賜りましたこと，深く感謝致します．また，研究のみならず，研究者としての心構えや進路に関するご意見など，様々なご意見を頂戴致しましたこと深く御礼申し上げます．

国立情報学研究所の山田茂樹教授，朝日大学経営学部経営情報学科の矢守恭子准教授，情報通信研究機構の徐蘇鋼氏，工学院大学の水野修教授，九州工業大学情報工学部の Marat Zhanikeev 准教授には，研究室でのゼミや毎年恒例の夏期合同合宿にて貴重なご意見を頂戴致しましたことを深く御礼申し上げます．

早稲田大学大学院国際情報研究科の先生方，助手・助教の皆様，事務所の皆様にお世話になったこと，心より御礼申し上げます．特に助手，助教の皆様には，日頃の生活の中で大変お世話になったこと，深く感謝致します．なた，早稲田大学大学院国際情報研究科田中研究室の皆様には，日頃より様々なお話を頂き，研究の支えとなったこと，深く感謝致します．特に，共に研究に取り組んで頂いた王キ氏，楊博氏，呂陽氏，王知越氏，Oluwatuyi Rufus ABIDAKUN 氏，樊星氏，黄稚沐氏，白井達也氏に深く御礼申し上げます．

芝浦工業大学システム理工学部電子情報システム学科三好研究室の皆様には，公私ともにお世話になったこと，深く感謝致します．私の研究を引き継ぎ，研究を行った吉沢剛氏，山崎託氏のみならず，全ての方と共に研究生活を送れたことを感謝致します．

最後になりましたが，私を支えてくださった両親，姉，親戚，友人の皆様に心より感謝致します．

2013 年 5 月 16 日

山本　嶺

参 考 文 献

[1] J. Postel (Ed.), "Transmission control protocol," IETF, RFC 793, Sept. 1981.

[2] M. Mathis, J. Mahodavi, S. Floyd, and A. Romanow, "TCP selective acknowledgement option," IETF, RFC 2018, Oct. 1996.

[3] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance vector routing for mobile computers," ACM SIGCOMM 1994, pp.234-244, Sept. 1994.

[4] C. C. Chiang, H. K. Wu, W. Liu, and M. Gerla, "Routing in clustered multihop mobile wireless networks with fading channel," IEEE Singapore Int. Conf. on Netw., pp.197–211, April 1997.

[5] C. Perkins and E. Royer, "Ad-hoc on-demand distance vector routing," 2nd IEEE Workshop on Mobile Comput. Syst. and Applications, pp.90–100, Feb. 1999.

[6] D. B. Johnson and D. A. Maltz, "Mobile computing," Kluwer Academic Publishers, 1996.

[7] J. Postel, "User datagram protocol," IETF, RFC 768, Aug. 1980.

[8] S. Biswas and R. Morris, "Opportunistic routing in multi-hop wireless networks," ACM SIGCOMM 2004, vol.34, issue 1, pp.69–74, Jan. 2004.

[9] S. Biswas and R. Morris, "ExOR:opportunistic routing in multi-hop wireless networks," ACM SIGCOMM 2005, vol.35, issue 4, pp.133–144, Oct. 2005.

[10] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "Trading structure for randomness in wireless opportunistic routing," ACM SIGCOMM 2007, vol.37, issue 4, pp.169–180, Oct. 2007.

[11] Z. Zhong, J. Wang, S. Nelakuditi, and G.-H. Lu, "On selection of candidates for opportunistic anypath forwarding," ACM SIGMOBILE 2006, vol.10, issue 4, pp.1–2, Oct. 2006.

[12] C. Westphal, "Opportunistic routing in dynamic ad hoc networks: the OPRAH protocol," IEEE Int. Conf. Mobile Ad Hoc and Sensor Syst. (MASS 2006), pp.570–573, Oct. 2006.

[13] Y. Yuan, H. Yang, S.H.Y. Wong, S. Lu, and W. Arbaugh, "ROMER: resilient opportunistic mesh routing for wireless mesh networks," 1st IEEE Workshop Wireless Mesh Netw. (WiMesh 2005), Sept. 2005.

[14] M.S. Nassr, J. Jun, S.J. Eidenbenz, A.A. Hansson, and A.M. Mielke, "Scalable and reliable sensor network routing: performance study from field deployment," 26th IEEE Int. Conf. Compt. Commun. (INFOCOM 2007), pp.670–678, May 2007.

[15] S. Jain and S.R. Das, "Exploiting path diversity in the link layer in wireless ad hoc networks," 6th IEEE Int. Symp. World of Wireless Mobile and Multimedia Netw. (WoWMoM 2005), pp.22–30, June 2005.

[16] M. Zorzi and R.R. Rao, "Geographic random forwarding (GeRaF) for ad hoc sensor networks: multihop performance," IEEE Trans. Mobile Comput., vol.2, issue 4, pp.337–348, Oct.–Dec. 2003.

[17] M. Zorzi and R.R. Rao, "Geographic random forwarding (GeRaF) for ad hoc and sensor networks: energy and latency performance," IEEE Trans. Mobile Comput., vol.2, issue 4, pp.349–365, Oct.–Dec. 2003.

[18] B. Zhao, R.I. Seshadri, and M.C. Valenti, "Geographic random forwarding with hybrid-ARQ for ad hoc networks with rapid sleep cycles," IEEE Conf. Global Telecommun. (GLOBECOM 2004), vol.5, pp.3047–3052, Dec. 2004.

[19] Y. Yan, B. Zhang, H.T. Mouftah, and J. Ma, "Practical coding-aware mechanism for opportunistic routing in wireless mesh networks," IEEE Int. Conf. Commun. (ICC 2008), pp.2871–2876, May 2008.

[20] T. Cui, L. Chen, and T. Ho, "Energy efficient opportunistic network coding for wireless networks," 27th IEEE Int. Conf. Comput. Commun. (INFOCOM 2008), pp.361–365, April 2008.

[21] J. Wu, M. Lu, and F. Li, "Utility-based opportunistic routing in multi-hop wireless networks," 28th IEEE Int. Conf. Distributed Comput. Syst. (ICDCS 2008), pp.470–477, June 2008.

[22] K.H.L. Xu and I. Rhee, "Binary increase congestion control (BIC) for fast long-distance networks," 23rd IEEE Int. Conf. Comput. Commun. (INFOCOM 2004),pp.2514–2524, March 2004.

[23] S. Ha, I. Rhee, and L. Xu, "CUBIC: a new TCP-friendly high-speed TCP variant," ACM SOGPOS, vol.42, issue 5, pp.64–74, July 2008.

[24] D. Leith and R. Shorten, “H-TCP: TCP for high-speed and long-distance networks,” Protocols for Fast Long-Distance Networks, Feb. 2004.

[25] R. Firrincieli and C. Caini, “TCP Hybla: a TCP enhancement for heterogenous networks,” Int, J. Satell. Commun. Network. vol.22, pp.547–566, Sept. 2004

[26] T. Kelly, “Scalable TCP: improving performance in highspeed wide area networks,” ACM SIGCOMM 2003, vol.33, issue 2, pp.83-91, April 2003.

[27] A. Baiocchi, A. P. Castellani, and F. Vacirca, “YeAH-TCP: yet another high-speed TCP,” PFLDnet-07, pp.37–42, Feb. 2007.

[28] S. Liu, T. Basar and R. Srikant, “TCP-Illinois: a loss and delay-based congestion contorl algorithm for high-speed networks,” 1st Int. Conf. Performance Evaluation Methodologies and Tools (VALUETOOLS 2006), vol.65, no.6-7, pp.417-440, Oct. 2006.

[29] K. T. J. Song, Q. Zhang and M. Sridharan, “Compound TCP: a scalable and tcp-friendly congestion control for high-speed networks,” PFLDnet-06, Feb. 2006.

[30] C. P. Fu and S. C. Liew, “TCP Veno: TCP enhancement for transmission over wireless access networks,” IEEE Commun. vol.21, pp.216–228, Feb. 2003.

[31] T. Clausen, P. Jacquet, “Optimized link state routing protocol (OLSR),” IETF, RFC 3626, Oct. 2003.

[32] A. Nasipuri and S. R. Das, “On-demand multipath routing for mobile ad hoc networks,” IEEE Int. Conf. Comput. Commun. and Netw. (ICCCN 1999), pp.64–70, Oct. 1999.

[33] 山本 嶺, 三好 匠, “アドホックネットワークにおける経路の近傍端末を利用した分散再送制御方式,” 信学技報, NS2006-215, vol.106, no.577, pp.295–298, March 2007.

[34] R. Yamamoto and T. Miyoshi, “Distributed retransmission method using neighbor terminals for ad hoc networks,” 14th Asia-Pacific Conf. Commun. (APCC 2008), Oct. 2008.

[35] 吉沢 剛, 山本 嶺, 三好 匠, “アドホックネットワークにおける一時的経路変更方式,” 信学技報, NS2008-37, vol.108, no.134, pp.67–70, July 2008.

[36] K. Nagao and Y. Yamao, "Cognitive temporary bypassing for reliable multihop transmission in wireless ad hoc networks," IEICE Trans. Commun., vol.E93-B, no.12, pp.3391–3399, Dec. 2010.

[37] QulaNet, Homepage. http://wwww.scalabel-networks.com

[38] M. Gerla, M. Y. Sanadidi, and A. Zanella, "TCP Westwood: congestion window control using bandwidth estimation," GLOBECOM 2001, vol.3, pp.1698–1702, Nov. 2001.

[39] L. Brakmo, S. O'Malley, and L. Peterson, "TCP Vegas: new techniques for congestion detection and avoidance," ACM SIGCOMM 1994, pp.24–35, Aug. 1994.

[40] T. Yamamoto, H. Tode, and K. Murakami, "Enhanced TCP congestion control realizing higher throughput and inter-session fairness in multihop wireless networks," IEICE Trans. Commun., vol.E91-B, no.7, pp.2279–2286, July 2008.

[41] K. N. Srijith, L. Jacob, and A. L. Ananda, "TCP Vegas-A:improving the performance of TCP Vegas," Elsevier Comput. Commun., vol.28, Issue 4, pp.429–440, March 2005.

[42] R. J. La, J. Walrand, and V. Anantharam, "Issues in TCP Vegas," UCB/ERL Memorandum, No.M99/3, Electronics Research Laboratory, Univ. of California, Berkeley, Jan. 1999.

[43] B. Zhang, M. N. Shirazi, and B. Komiyama, "An ELFN-based TCP-freeze scheme using the route information of sender node for ad hoc networks," 10th Asia-Pacific Conf. Commun. (APCC 2004), vol.1, pp.457–461, Sept. 2004.

[44] D. Kim, C-K. Toh, and Y. Choi, "TCP-BuS: improving TCP performance in wireless ad hoc networks," IEEE Int. Conf. Commun. (ICC 2000), vol.3, pp.1707–1713, June 2000.

[45] J. Liu and S. Singh, "ATCP: TCP for mobile ad hoc networks," IEEE J. Select. Areas Commun., vol.19, no.7, pp.1300–1315, July 2001.

[46] K. Sundaresan, V. Anantharaman, H-Y. Hsieh, and A. R. Sivakumar, "ATP: A reliable transport protocol for ad hoc networks," IEEE Trans. Mobile Comput., vol.4, no.6 pp.588–603, Dec. 2005.

[47] V. Jacobson, R. Braden, and D. Borman, "TCP extensions for high performance," IETF, RFC 1323, May 1992.

[48] GloMoSim, website, http://pcl.cs.ucla.edu/projects/glomosim/

[49] C. Jin, D. X. Wei and S. H. Low, "FAST TCP: Motication, Architecture, Algorithms, Performance," 23rd IEEE Int. Conf. Comput. Commun. (INFOCOM 2004), pp.2490-2510, March, 2004.

[50] R. Jain, W. Hawe and D. Chiu, "A quantitative measure of fairness and discrimination for resource allocation in Shared Computer Systems," DEC-TR-301, Sept. 1984.

[51] V.N. Talooki, J. Rodriguez, and R. Sadeghi, "A load balanced aware routing protocol for wireless ad hoc networks," 16th Int. Conf. Telecommun. (ICT 2009), pp.25–30, May 2009.

[52] F. Zou, X. Zhang, X. Gao, D. Shi, and E. Wang, "Load balance routing using packet success rate for mobile ad hoc networks," Int. Conf. Wireless Commun., Netw. and Mobile Comput. (WiCom 2007), pp.1624–1627, Sept. 2007.

[53] 増田修士, 萩谷展之, 松尾太一, 後藤康宏, 阪田史郎, "アドホックネットワークにおける中継ノードのトラフィック量を考慮した負荷分散方式," 信学技報, IN2006-190, vol.106, no.578, pp.61–66, March 2007.

[54] A. Fu, P. Zerfos, H. Luo, S. Lu, L. Zhang, and M. Gerla, "The impact of multihop wireless channel on TCP throughput and loss," 22nd IEEE Int. Conf. Comput. Commun. (INFOCOM 2003), vol.3, pp.1744–1753, April 2003.

[55] S.M. ElRakabawy, A. Klemm, and C. Lindemann, "TCP with adaptive pacing for multihop wireless networks," 6th ACM Int. Symp. Mobile Ad Hoc Netw. and Comput. (MobiHoc 2005), pp.288–299, May 2005.

[56] K. Xu, M. Gerla, L. Qi, and Y. Shu, "Enhancing TCP fairness in ad hoc wireless networks using neighborhood RED," 9th ACM Int. Conf. Mobile Comput. and Comput. (MobiCom 2003), pp.16–28, Sept. 2003.

[57] K. Nahm, A. Helmy, and C.-C. J. Kuo, "TCP over multihop 802.11 networks: issues and performance enhancement," 6th ACM Int. Symp. Mobile Ad Hoc Netw. and Comput. (MobiHoc 2005), pp.277–287, May 2005.

[58] J. Chen, M. Gerla, Y.Z. Lee, and M. Sanadidi, "TCP with variance control for multihop ieee 802.11 wireless networks," IEEE Military Commn. Conf. (MILCOM 2006), pp.1–7, Oct. 2006,

[59] E. Altman and T. Jimenez, "Novel delayed ACK techniques for improving TCP performance in multihop wireless networks," Int. Conf. Personal Wireless Commun. (PWC 2003), pp.237–253, Sept. 2003.

[60] B.J. Wolf, J.L. Hammond, and H.B. Russell, "A distributed load-based transmission scheduling protocol for wireless ad hoc networks," ACM Int. Conf. Wireless Commun. and Mobile Comput. (IWCMC 2006), pp.437–442, July 2006.

[61] J. Yin and X. Yang, "ELQS: an energy-efficient and load-balanced queue scheduling algorithm for mobile ad hoc networks," WRI Int. Conf. Commun. and Mobile Comput. (CMC 2009), vol.2, pp.121–126, Jan. 2009.

[62] H.-F. Liu, L.-J. Li, Z.-Y. Yang, and X.-Y. Huang, "TCP Vegas_M: a novel TCP algorithm in mobile ad hoc network," 6th Int. Conf. ITS Telecommun., pp.629–633, June 2006.

[63] S.M. ElRakabwy, A. Klemm, and C. Lindemann, "TCP with adaptice pacing for multihop wireless networks," 6th ACM Int. Symp. Mobile Ad Hoc Netw. and Comput. (MobiHoc 2005), pp.288–299, May 2005.

[64] C. K. Toh, A.-N. Le, and Y.-Z. Cho, "Load balanced routing protocols for ad hoc mobile wireless networks," IEEE Commun. Mag., vol.47, issue 8 pp.78–84, Aug. 2009.

[65] J.-H. Song, V. Wong, and V.C.M. Leung, "Load-aware on-demand routing (LAOR) protocol for mobile ad hoc networks," IEEE 57th Vehicular Tech. Conf. (VTC 2003-Spring), pp.1753–1757, April 2003.

[66] C. K. Toh, "Associativity-based routing for ad hoc mobile networks," Int. J. Wireless Personal Commun. vol.4, issue 2, pp.103–139, March 1997.

[67] H. Hassanein and A. Zhou, "Routing with load balancing in wireless ad hoc networks," 4th ACM Int. Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Syst. (MSWIM 2001), pp.89–96, July 2001.

[68] A.H. Altalhi and G.G. Richard III, "Load-balanced routing through virtual paths: highly adaptive and efficient routing scheme for ad hoc wireless networks," IEEE Int. Conf. Performance, Comput. and Commun., pp.407–413, April 2004.

[69] L. Yang and M. Hong, “Three load metrics for routing in ad hoc networks,” IEEE 60th Vehicular Tech. Conf. (VTC 2004-Fall), vol.4, pp.2764–2768, Sept. 2004.

[70] J. Peng, L. Cheng, and B. Sikdar, “A wireless MAC protocol with collision detection,” IEEE Tran. Mobile Comput., vol.6, issue 12, pp.1357–1369, Dec. 2007.

[71] P. Shao, A. Matsumoto, S. Aust, T. Ito, and P. Davis, “A collision detection method based on power sensing and time-domain signal processing for wireless LAN,” 11th Int. Symp. Commun. Information Tech. (ISCIT 2011), pp.199–204, Oct. 2011.

[72] D. S. Chan and T. Berger, “Collision detection for carrier sense multiple access in wireless networks,” 16th IEEE Int. Symp. Personal, Indoor and Mobile Radio Commun. (PIMRC 2005), vol.3, pp.1495–1499, Sept. 2005.

[73] J.H. Yun and S.W. Seo, “Collision detection based on transmission time information in IEEE 802.11 wireless LAN,” 4th IEEE Int. Conf. Pervasive Comput. and Commun. Workshop (PerComW 2006), pp.410–414, March 2006.

[74] J.H. Yun and S.W. Seo, “Collision detection based on RE energy duration in IEEE 802.11 wireless LAN,” 1st Int. Conf. Commun. Syst. Software and Middleware (COMSWARE 2006), pp.1–6, Jan. 2006.

[75] J. Kim, S. Kim, S. Choi, and D. Qiao “CARA: collision-aware rate adaptation for IEEE 802.11 WLANs,” 25th IEEE Int. Conf. Comput. Commun. (INFOCOM 2006), pp.1–11, April 2006.

[76] I. Tinnirello and A. Sgora, “A kalman filter approach for distinguishing channel and collision errors in IEEE 802.11 networks,” IEEE Global Telecommun. Conf. (GLOBECOM 2008), pp.1–5, Nov. 2008.

[77] T. Zhou, X. Wang, and W. Hou, “A fast collision detection algorithm in IEEE 802.11 through physical layer SINR monitoring,” 73rd IEEE Vehicular tech. conf. (VTC 2011-Spring), pp.1–5, May 2011.

[78] 電波伝搬ハンドブック編集委員会, “電波伝搬ハンドブック,” リアライズ社, 1999.

図 目 次

表 目 次

発 表 文 献

1. 論文

〈1.1〉 山本 嶺, 三好 匠, 田中良明, “アドホックネットワークにおける送信速度制御を用いた近傍トラヒック適応形負荷分散手法,” 信学論 (B), vol.J96-B, no.7, pp. July 2013.

〈1.2〉 山本 嶺, 三好 匠, 田中良明, “アドホックネットワークにおける往復遅延時間を用いた適応形トランスポートプロトコル,” 信学論 (B), vol.J93-B, no.5, pp.735-746, May 2010.

2. 国際学会

〈2.1〉 R. Yamamoto, T. Miyoshi, and Y. Tanaka, “Neighbour traffic-aware load balancing method in ad hoc networks,” Fourth Int. Conf. Intelligent Netw. and Collaborative Syst. (INCoS 2012), Bucharest, Romania, pp.193-197, Sept. 2012.

〈2.2〉 R. Yamamoto, T. Miyoshi, and Y. Tanaka, “A load balancing method with adaptive transmission rate control in ad hoc networks,” 13th Asia-Pacific Netw. Operations and Management Symp. (APNOMS 2011), Taipei, Taiwan, no.I2-3, Sept. 2011.

〈2.3〉 R. Yamamoto and T. Miyoshi, “Distributed Retransmission Method Using Neighbor Terminals for Ad Hoc Networks,” 14th Asia-Pacific Conf. Commun. (APCC 2008), Tokyo, Japan, Oct. 2008.

3. 研究会

〈3.1〉 山本 嶺, 山崎 託, 三好 匠, 田中良明, “MANET における伝搬遅延推定に基づく Opportunistic Routing,” 信学技報, vol.112, no.463, NS2012-221, pp.325-330, March 2013.

〈3.2〉 山崎 託, 山本 嶺, 三好 匠, 田中良明, “アドホックネットワークにおける近傍ノード協力形再送制御方式,” 信学技報, vol.112, no.350, NS2012-123, pp.37-42, Dec. 2012.

〈3.3〉 山本 嶺, 三好 匠, 田中良明, “アドホックネットワークにおける通信負荷推定を用いた経路制御手法,” 信学技報, vol.111, no.144, NS2011-58, pp.51-56, July 2011.

〈3.4〉 山本 嶺, 三好 匠, 田中良明, “アドホックネットワークにおける近傍トラヒック適応形負荷分散手法（奨励講演）,” 信学技報, vol.111, no.8, NS2011-4, pp.19-24, April 2011.

〈3.5〉 山本 嶺, 三好 匠, 田中良明, “近傍の通信状態に基づくアドホックネットワーク負荷分散手法,” 信学技報, vol.110, no.340, NS2010-118, pp.79-84, July 2010.

〈3.6〉 山本 嶺, 三好 匠, 田中良明, “アドホックネットワークにおける送信速度のクロスレイヤ制御,” 信学技報, vol.110, no.126, NS2010-45, pp.41-46, July 2010.

〈3.7〉 山本 嶺, 三好 匠, 田中良明, “RTT 駆動形ウィンドウ制御手法の特性分析,” 信学技報, vol.109, no.326, NS2009-134, pp.79-84, Dec. 2009.

〈3.8〉 山本 嶺, 三好 匠, “アドホックネットワークにおける公平性を考慮した RTT 駆動形トランスポートプロトコル,” 信学技報, vol.108, no.457, NS2008-175, pp.183-185, March 2009.

〈3.9〉 山本 嶺, 三好 匠, “アドホックネットワークにおける往復遅延時間を用いたフロー制御手法（奨励講演）,” 信学技報, vol.108, no.359, NS2008-124, pp.95-100, Dec. 2008.

〈3.10〉 山本 嶺, 三好 匠, ” アドホックネットワークにおける往復遅延時間を用いたフロー制御手法,” ネットワークシステムワークショップ 2008 予稿集, pp.7-8, Nov. 2008.

〈3.11〉 吉沢 剛, 山本 嶺, 三好 匠, “アドホックネットワークにおける一時的経路変更方式,” 信学技報, vol.108, no.134, NS2008-37, pp.67-70, July 2008.

〈3.12〉 山本 嶺, 三好 匠, “アドホックネットワークにおける適応型トランスポートプロトコル,” 信学技報, vol.108, no.31, NS2008-10, pp.55-60, May 2008.

〈3.13〉 山本 嶺, 三好 匠, “経路の近傍端末を利用したアドホックネットワーク分散再送制御方式,” 信学技報, vol.106, no.577, NS2006-215, pp.295-298, March 2007.

4. 大会

〈4.1〉 山本 嶺, 山崎 託，三好 匠, 田中良明, “アドホックネットワークにおける伝搬遅延に基づくオポチュニスティック経路選択,” 信学総大, 分冊通信 2, B-6-139, p.139, March 2013.

〈4.2〉 山崎 託，山本 嶺, 三好 匠, “アドホックネットワークにおける近傍ノードを利用した連続再送制御,” 信学ソ大, 分冊通信 2, BS-2-3, pp.S-5-S-6, Sept. 2012.

〈4.3〉 山本 嶺, 三好 匠, 田中良明, “アドホックネットワークにおける近傍端末の通信状態に基づく経路制御手法,” 信学ソ大, 分冊通信 2, B-6-16, p.16, Sept. 2012.

〈4.4〉 山崎 託，山本 嶺, 三好 匠, “アドホックネットワークにおける近傍ノードを利用した経路変更手法,” 信学総大, 分冊通信 2, B-6-10, p.10, March 2012.

〈4.5〉 山本 嶺, 三好 匠, 田中良明, “アドホックネットワークにおける自律的負荷評価手法,” 信学総大, 分冊通信 2, B-6-12, p.12, March 2012.

〈4.6〉 山本 嶺, 三好 匠, 田中良明, “アドホックネットワークにおける通信負荷推定に基づく負荷分散手法,” 信学ソ大, 分冊通信 2, B-6-64, p.64, Sept. 2011.

〈4.7〉 山本 嶺, 三好 匠, 田中良明 “アドホックネットワークにおける通信負荷推定に基づく送信速度制御,” 信学総大, 分冊通信 2, B-6-85, p.85, March 2011.

〈4.8〉 山本 嶺, 三好 匠, 田中良明, “アドホックネットワークにおける送信速度制御を用いた負荷分散手法,” 信学ソ大, 分冊通信 2, B-6-73, p.73, Sept. 2010.

〈4.9〉 山本 嶺, 三好 匠, 田中良明 “アドホックネットワークにおけるクロスレイヤ制御を用いた負荷分散手法,” 信学総大, 分冊通信 2, B-6-138, p.138, March 2010.

〈4.10〉 山本 嶺, 三好 匠, 田中良明, “アドホックネットワークにおける公平性を考慮したウィンドウ制御手法,” 信学ソ大, 分冊通信 2, B-6-5, p.5, Sept. 2009.

〈4.11〉 R. Yamamoto and T. Miyoshi, “Flow control method considering fairness for ad hoc networks,” 2008 IEICE General Conference, BS-4-31, pp.S-61-S-62, March 2009.

〈4.12〉 吉沢 剛, 山本 嶺, 三好 匠, “アドホックネットワークにおける近傍ノードを利用した一時的中継制御方式,” 信学総大, 分冊通信 2, B-21-10, p.611, March 2008.

〈4.13〉 山本 嶺, 三好 匠, “アドホックネットワークにおける経路の近傍端末を利用した再送制御方式,” 信学総大, 分冊通信 2, B-21-43, p.632, March 2007.

5. その他

〈5.1〉 B. Yang, R. Yamamoto, and Y. Tanaka, “Dempster-shafer evidence theory based trust management strategy against cooperative black hole attacks and gray hole attacks in MANETs,” ICACT Trans. Advanced Commun. Tech. (TACT), vol.2, issue 3, pp.223-232, May 2013.

〈5.2〉 B. Yang, R. Yamamoto, and Y. Tanaka, “Historical evidence based trust management strategy against black hole attacks in MANET,” 14th Int. Conf. Advanced Commun. Tech. (ICACT 2012), Phoenix park, Korea, pp.394-399, Feb. 2012.

〈5.3〉 M. Zhanikeev, R. Yamamoto, K. Yamori and Y. Tanaka, “A method for side splitting of packet trace,” 13th Asia-Pacific Netw. Operations and Management Symp. (APNOMS 2011), Taipei, Taiwan, no.TS6-1, Sept. 2011.

〈5.4〉 Q. Wang, R. Yamamoto, and Y. Tanaka, “Wormhole attack defense method based on observation report for mobile ad hoc networks,” 26th Int. Tech. Conf. Circuit/System, Comput. and Commun. (ITC-CSCC 2011), Gyeongju, Korea, no.P3-6, pp.1139-1142, June 2011.

〈5.5〉 B. Yang, R. Yamamoto, and Y. Tanaka, “A wormhole attack mitigation strategy based on mobile agent,” IEICE tech. rep. vol.111, no.468, NS2011-206, pp.149-154, March 2012.

〈5.6〉 王キ, 山本 嶺, 田中良明, “MANET における中継優先度を用いた DDoS 攻撃対策手法,” 信学技報, vol.111, no.344, NS2011-144, pp.149-152, Dec. 2011.

〈5.7〉 王キ, 山本 嶺, 田中良明, “隣接ノードの監視情報を用いたワームホール攻撃対策手法,” 信学技報, vol.110, no.448, NS2010-245, pp.455-460, March 2011.

〈5.8〉 O. R. Abidakun, R. Yamamoto, and Y. Tanaka, “Energy Saving Strategy Based on SPIN Routing Protocol in Wireless Sensor Network,” 2012 IEICE General Conference, BS-1-6, pp.S-11-S-12, March 2013.

〈5.9〉 T. Shirai, R. Yamamoto, T. Miyoshi, and Y. Tanaka, “Alternate route construction method using node mobility prediction for ad hoc multicasting,” 2012 IEICE General Conference, BS-1-44, pp.S-86-S-87, March 2013.

〈5.10〉 B. Yang, R. Yamamoto, and Y. Tanaka, “Game theoretic analysis of reputation based IDS for preventing black hole attack in MANETs,” 2012 IEICE General Conference, BS-1-45, pp.S-88-S-89, March 2013.

〈5.11〉 B. Yang, R. Yamamoto, and Y. Tanaka, “A novel mitigation strategy against malicious actions based on risk evaluation aware trust management in MANETs,” 2012 IEICE Society Conference, BS-5-37, pp.S-98-S-99, Sept. 2012.

〈5.12〉 M. Zhanikeev, R. Yamamoto, and Y. Tanaka, “On why distributed monitoring needs distributed PASTA,” 2012 IEICE Society Conference, BS-4-5, pp.S-24-S-25, Sept. 2012.

〈5.13〉 B. Yang, R. Yamamoto, and Y. Tanaka, “Mobile agent based wormhole attack detection in MANET,” 2011 IEICE General Conference, BS-3-12, pp.S-23-S-24, March 2012.

〈5.14〉 王キ, 山本 嶺, 田中良明, “アドホックネットワークにおける中継優先度に基づく DDoS 攻撃対策手法,” 信学総大, 分冊通信 2, B-6-13, p.13, March 2012.

〈5.15〉 M. Zhanikeev, R. Yamamoto, and Y. Tanaka, “Capturing QoS context by alternative flow monitoring in clouds,” 2011 IEICE General Conference, BDS-1-2, pp.S-128-S-129, March 2012.

〈5.16〉 M. Zhanikeev, R. Yamamoto, and Y. Tanaka, “Alternative packet sampling for improved fairness and function,” 2011 IEICE General Conference, BS-3-4, pp.S-7-S-8, March 2012.

〈5.17〉 王キ, 山本 嶺, 田中良明, “MANET における中継ノードの判断に基づく DDoS 攻撃対策手法,” 信学ソ大, 分冊通信 2, B-6-69, p.69, Sept. 2011.

〈5.18〉 B. Yang, R. Yamamoto, and Y. Tanaka, “A trust-aware management strategy against black hole attack in MANET,” 2011 IEICE Society Conference, BS-6-39, pp.S-106-S-107, Sept. 2011.

〈5.19〉 V. Tran-Quang, P. Nguyen Huu, R. Yamamoto, and T. Miyoshi, “A target tracking algorithm considering energy balance in WSNs,” 2011 IEICE Society Conference, BS-6-34, pp.S-96-S-97, Sept. 2011.

〈5.20〉 王キ, 山本 嶺, 田中良明, “MANET における隣接ノードの監視情報に基づくワームホール攻撃対策手法,” 信学総大, 分冊通信 2, B-6-84, p.84, March 2011.