

車載無線機器による車車間アドホックネットワーク上の堅牢な経路制御（継続）

山口 弘 純 大阪大学大学院情報科学研究科准教授

1 研究の目的・意義

1-1 研究の背景と概要

近年、高度な交通システム（ITS）の実現を目指し、DSRC や VICS ビーコンなどの狭域通信器が路側や店舗などに設置されつつある。しかし、これら固定インフラの完全な整備と展開には膨大なコストと年月がかかり、機器の更新は安易にできないといった問題点もある。これに対し、安価な車載無線機器による車車間アドホックネットワークを構築し、車両から車両へデータをバケツリレー式に転送することで、インフラがない地域や災害等で利用不能となった場合でもデータ配信を可能とする方式に注目が集まっている。これにより、例えば、(1) 事故現場に停止した緊急車両から現場に流入する車両に事故渋滞映像を配信することで渋滞を回避する、(2) 無線 LAN スポットが提供するインターネットサービスを、道路を走行する車両を介して近隣の駐車場まで拡大する、(3) 災害時に携帯電話基地局が不通となった場合、緊急車両が被災者と避難所との音声通話データの中継する、など多くの応用事例が考えられる。しかし、これらの映像・音声ストリーミングや双方向通信サービスはリアルタイム性が求められるため、中継車両の移動のもとでもデータ遅延および損失が少ない十分堅牢な経路を提供することが求められる。

本研究では、車車間アドホックネットワーク上で堅牢な通信経路を実現するルーティング技術に関する研究を行う。GPS などの測位機器と電子地図を用いて、高確率で同方向に同速度で移動する車両を予測し、その上に通信経路を確立するルーティングプロトコルを設計しており、車車間通信リンクの切断リスクを軽減し、経路の堅牢性を高めることができる。今年度は高精度計算機シミュレーションにより現実的なシナリオのもとでの UDP 転送のパフォーマンスと TCP スループットを確認するとともに、無線アドホックネットワークテストベッドを構築し、位置情報ルーティングの性能評価を行うプラットフォームを実現している。

1-2 設計したプロトコルの概要

以下に設計したプロトコルの概要を述べる。各ノードは近距離無線デバイスを装備し、一意な ID を持つとする。また、GPS などの測位装置と電子地図を装備しているとする。電子地図は有効グラフ $G = (E, V)$ で表される。 E は道路セグメントの集合、 V は交差点の集合とする。道路セグメントは隣接する 2 つの交差点間の道路を表し、各方向ごと 1 セグメントとする。また、各セグメントには国道番号などの道路識別子が（もしあれば）与えられているとし、各ノードは道路セグメントが同じ道路上のものかどうかの判断ができるとする。各交差点にはその位置情報も付随しているものとする。各ノード u は測位装置と電子地図により、自身の位置 $P(u)$ 、自身が存在するセグメント識別子 $S(u)$ 及び道路識別子 $R(u)$ 、そのセグメントの両端の交差点識別子及びそれら交差点の位置情報がわかっているとする。

地理領域はグリッドに分割される。あるグリッドに隣接する周囲 8 グリッドをそのグリッドの隣接グリッドと呼ぶ。グリッドにある各ノードは隣接グリッド内のすべてのノードと通信可能であるよう、グリッドサイズ w と無線範囲 r の関係を $w = 2^{1/2} * r / 4$ と定義するなお、グリッドサイズ w が各ノードにおいて既知の場合、経度 x 分、緯度 y 分の座標のノード v の存在するグリッド ID $G(v)$ を下記のように定義する。

$$G(v) = G_x(v) * W + G_y(v)$$

ただし、 $G_x(v)$ は x のメートル表現の w による商、 $G_y(v)$ は y のメートル表現の w による商、 W は $G_y(v)$ の最大値より大きい定数とする。 W は $G_y(v)$ の最大値より大きい定数であるため、グリッド ID は x および y より一意に決定される。これにより、各ノードは共通パラメータ w 及び W からある位置を含む一意なグリッド番号を計算することができる。

設計したプロトコルは 2 地点間通信経路をオンデマンドに構築し、維持するプロトコルである。始点ノード s と、終点座標 d に対し、このプロトコルは道路経路に沿って、 d を含むグリッド $G(d)$ への経路を探索して構築する。構築した経路が切断した場合、プロトコルは始めに通信経路を構築した道路経路上でのみ経路の再探索を行い、これが失敗した場合、プロトコルは経路を放棄し、再度始点ノード s と終点座標 d に

対し、経路を探索して構築する。プロトコルは経路構築プロセスと経路維持プロセスからなる。

経路構築プロセスでは、LAR のメッセージ削減手法に基づき、メッセージを転送する際、s と G(d) を含む領域を予め設定し、この領域内のみでメッセージを転送する。s から G(d) へ経路を探索する場合、s は G(s) に隣接しておりかつ転送領域内の各グリッドから、1 ノードずつ選択し、RREQ メッセージを転送する。転送した経路探索メッセージ (Route REQuest、以下 RREQ) には s のノード ID、s が存在する道路セグメント ID S(s)、道路 ID R(s) 及びグリッド ID G(s) が含まれている。RREQ メッセージを受信した各ノードは、同様の選択方法で転送領域内の隣接グリッドごとに 1 つノードを選択し、自身のノード ID、自身が存在する道路セグメント ID、道路 ID 及びグリッド ID を追加した RREQ メッセージを転送する。G(d) の隣接グリッド内のノードが RREQ を受信した場合、ノードは G(d) 内のもっともノード ID の若いノード d0 を指定して RREQ メッセージを転送する。d0 を G(d) の代表ノードとよぶ。d0 は最初の RREQ メッセージを受信してから一定時間は他の RREQ メッセージの到着を待ち、受信した複数の RREQ メッセージから後述する基準に基づき最も品質の良い通信経路候補を選択し、その経路を辿り s へ経路決定 (Route REPLY、以下 RREP) メッセージを送信することで経路を決定する。ただし、RREP には選択された RREQ に記録されていたノード ID 列とグリッド ID 列を含める。

RREQ メッセージから得られる道路経路情報から適切な通信経路を選択するために、(i) 道路経路上での 2 ノードの走行距離、(ii) 道路経路の交差点数、及び (iii) それら交差点のうち右左折を必要とする回数、が 2 ノード間のリンクの生存時間にどの程度影響を与えるかを実験により調査している。この目的のため、交通シミュレータを用いて、格子状の道路からなり、交差点間のセグメント長が 50m から 500m まで、50m 刻みで変更した地図を用意した。これらの地図において、0 から 8 個の交差点を持つ道路経路を選んだ。そのため、道路長は最短 50m から最長 4500m となった。また、各交差点では、ノードは一樣な確率で直進または右左折するようにした。すべての地図において、ノード密度は同じになるように設定した。このもとで、2 ノード間のリンク接続率 (2 ノードが道路経路に同時に存在していた時間に対する、2 ノード間が直接通信可能となった時間総和の割合) を観測した。その結果、以下のことがわかった。

- ・ ノードの走行距離の増加に応じて、リンク接続率はおおよそ線形的に減少する。これは 2 ノードの走行距離が短い間は、信号などでノード間距離が若干増加しても互いの通信範囲内にとどまっている可能性が高いが、走行距離が長くなるにつれて、それが通信範囲より大きくなる可能性も増加することから、おおよそ自然なものである。
- ・ 交差点数の増加によるリンク接続率の低下は線形的である。
- ・ 右左折回数が少ない場合にリンク接続率が非常に高く、右左折数増加に対する接続率の低下率がかなり大きい。

以上の結果から、道路経路上の実際の走行距離を除けば、右左折数が支配的な要素であることがわかる。経路探索メッセージから通信経路を決定する場合、それに付随する道路経路の右左折数が最も少ない通信経路を優先すればよいと考えられる。なお、RREP メッセージを転送する際、経路上の各ノードは、RREP メッセージ内に記録されていたノード ID 列から、自身の前方及び後方のノードをルーティングテーブルに記録する。また、グリッド ID 列も記録する。グリッド ID 列は道路経路形状を表しており、経路維持プロセスで用いられる。

経路維持プロセスは、経路が切断した場合、経路探索で発見された通信経路が存在する道路経路を含むグリッド上に存在しないノードを経路から切り離し、それらのグリッド上のみから代替ノードを探すことで、その道路経路に沿った通信経路を復元させる。このグリッド列を道路経路グリッド列と呼ぶ。ノード s から G(d) 内のノード d0 までの通信経路は、中間ノードの移動や、d0 の G(d) からの離脱などにより切断される場合がある。経路上の各ノードは、自身の前後ノードとのリンク切断が発生した場合、自身が現在存在しているグリッドが道路経路グリッド列に含まれないと判明すれば、自身と接続しているノードとのリンクを強制切断する。これにより新たなリンク切断が発生し、道路経路グリッド列上に存在しないノードは同様の動作を行う。その結果、道路経路グリッド列から離脱したノードはすべて通信経路から外される。前方ノード (経路上での d0 側のノード) とのリンクが切断されたが道路経路グリッド列上にあるノードは、以下に述べる経路維持プロセスを開始する。各ノードは道路経路グリッド列における自身のグリッドの前方グリッドから代替ノードを探し、自身の記憶した道路経路グリッド列を含む経路維持メッセージ (Route Re-PaiR、以降 RRPR) を送信する。その際、切断した通信経路の残存ノードが存在すれば、そのノードを優先的に選択する。これにより、道路経路グリッド列上にある残存経路を最大限に利用できる。再構築が不可能な場合、そのノードは経路エラーメッセージ (Route ERRor メッセージ、以下 RERR) を送信ノード s に送信する。この場合、ノード s は経路構築プロセスを実行し、新たな経路を探索する。

2 研究の方法

2-1 シミュレーションによるデータ転送性能評価

(1) 環境ならびにパラメータ設定

Settings	(A)	(B)	(C)	(D)	(E)
Map. Type	Single Street	Manhattan Street			Osaka Downtown
S-D locations	on the same street			on the different streets	
Max. Num. of Lanes	2	2	6	6	6
Field Size ($m \times m$)	1,500	1,500×400		1,000×1,000	1,500×1,700
Speed Limit	60km/h				
Ave. Inter-vehicle Distance per Lane (i.e. Density)	60m (low density) / 20m (high density)				
Radio Range	200(m)				
Grid Length	70(m)				
Simulation Time	300(sec.)				
Link Capacity	2Mbps				

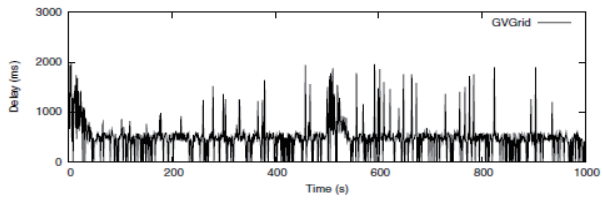
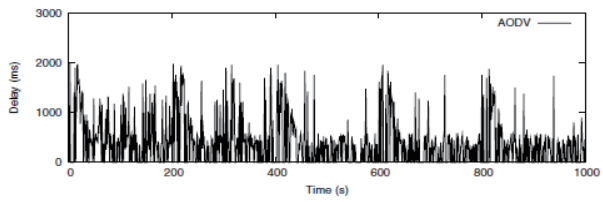
図 1 実験シナリオの設定

車車間ネットワークにおけるルーティングを評価するためには、車両の移動モデルに関する信頼性が重要になる。最近では、都市環境における車両の移動モデルの現実性が車両速度に与える影響ならびに、それがどの程度ルーティング性能に影響を与えるかを調査した研究がある。本研究では、車両の現実的な移動を再現できるトラフィックシミュレータを利用している。このシミュレータでは、レーン変更や車間距離に応じた速度調整などをモデル化している。我々はこのトラフィックシミュレータのトレース出力を利用し、ネットワークシミュレータ GTNetS に与えることで、設計したプロトコルの現実的な環境での評価を行っている。このトラフィックシミュレータにおいて、以下の3種類の道路地図モデルを利用している。

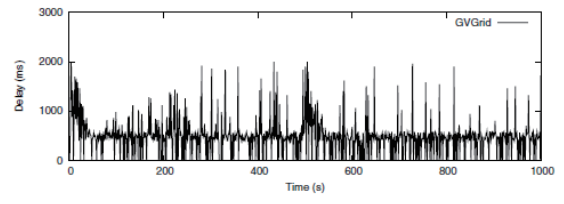
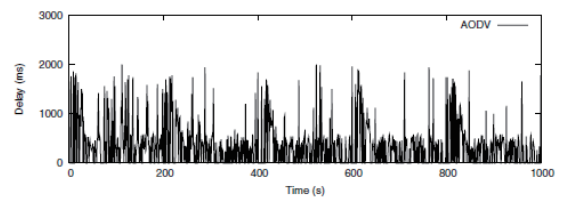
- (a) 単純な1道路モデル (single street model) : 高速道路のような直線道路を想定している。
- (b) 格子状道路モデル (the Manhattan street model) : 東西方向と南北方向の道路とそれらの交差点からなる、典型的な都市の格子状道路を想定している。
- (c) 大阪中心部モデル (the Osaka downtown model) : 大阪市天王寺駅周辺の主要道路をモデル化している。

これらの地図を利用し、5つのシミュレーションシナリオ設定を行った。これらを図1に示す。シミュレーションでは、MAC層にIEEE802.11 (DCF) を仮定している。

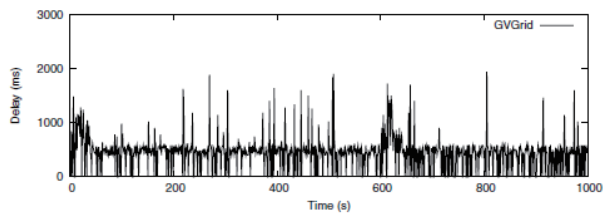
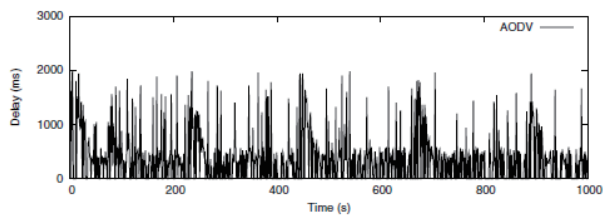
昨年度は、経路の生存時間やパケット到着率など、経路の品質を評価する基本指標の評価を、同じ位置情報ルーティングであるGPCR (C. Lochert, M. Mauve, H. Fusler, and H. Hartenstein, 2005) と比較することで実施している。GPCR はリアクティブ型のVANETルーティングプロトコルであり、位置情報ルーティングGPSR (B. Karp and H. T. Kung, 2000) を利用して設計されている。GPCRでは、地図情報を使わず、交差点に存在する車両を優先的に選択し、GPSRの障害物迂回アルゴリズムを応用して目的地までの経路発見を少ないメッセージで実現する。このGPCRと比較し、本研究で設計開発したプロトコルは、道路情報を有効に活用することで堅牢な経路を実現している。本研究では、より現実的な設定のもとで、UDPパケットの遅延と到着率を測定している。また、TCP上でのスループットを様々な設定条件のもとで測定している。



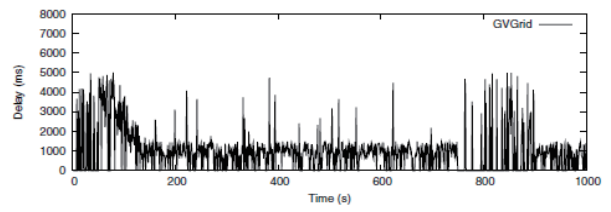
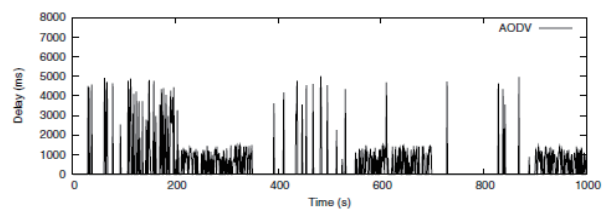
(A)



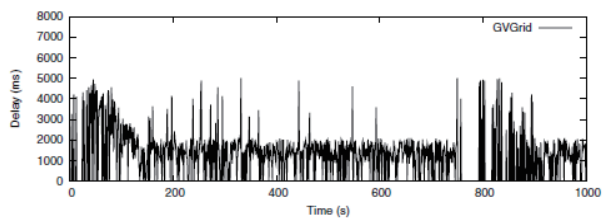
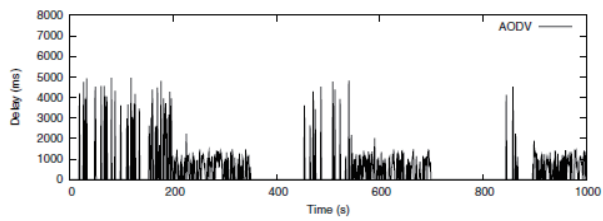
(B)



(C)

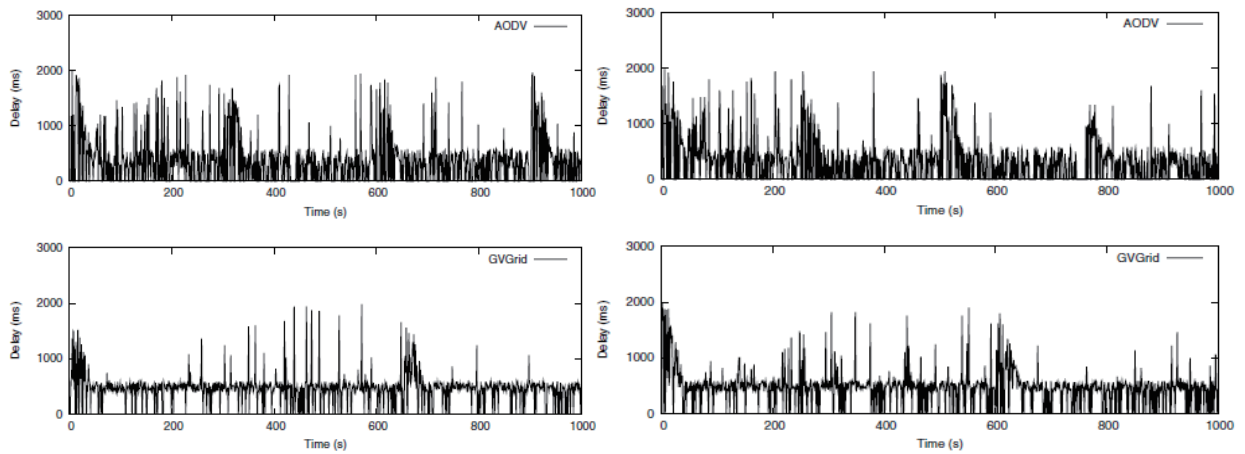


(D)



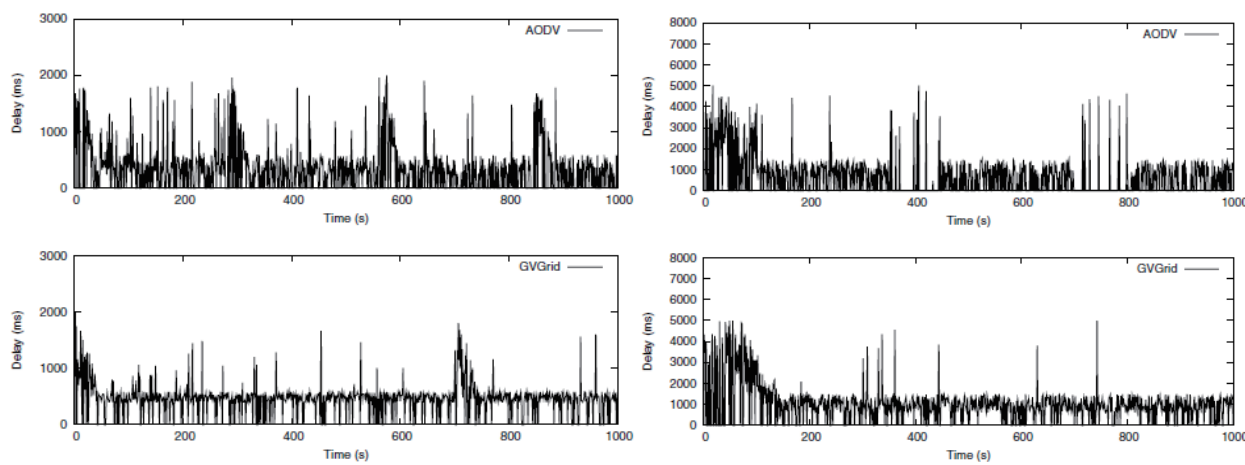
(E)

図 2 パケット遅延 (低車両密度)



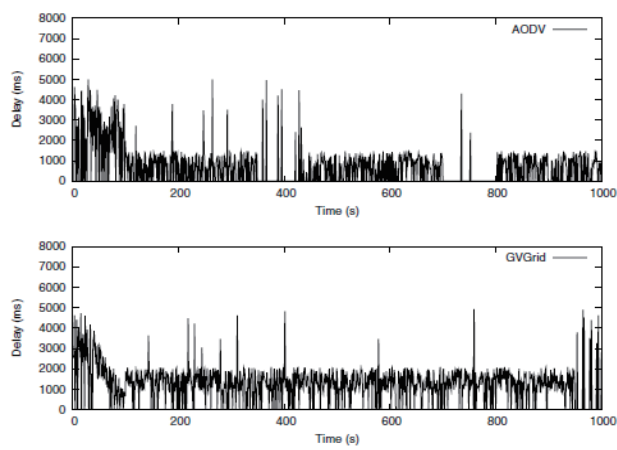
(A)

(B)



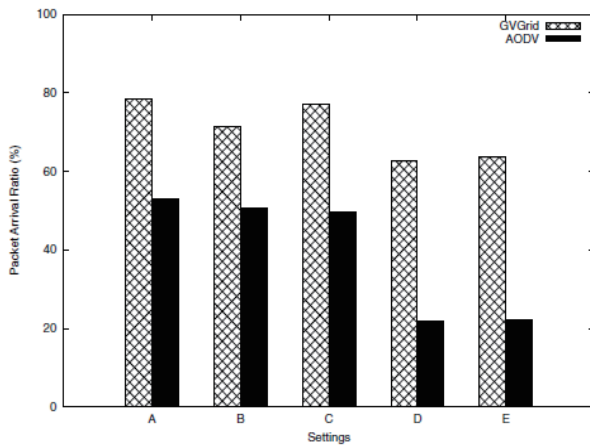
(C)

(D)

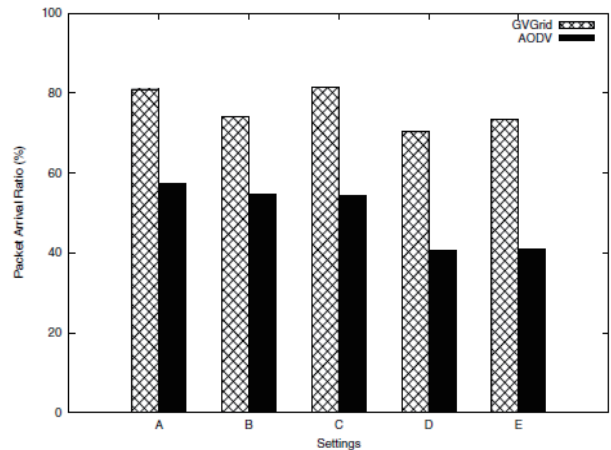


(E)

図 3 パケット遅延 (高車両密度)



(a) Low Density



(b) High Density

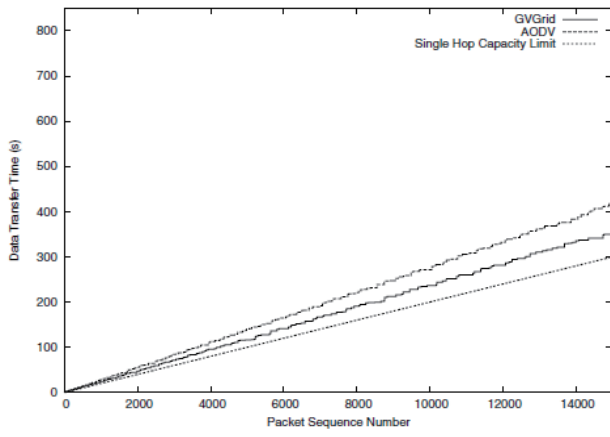
図 4 平均パケット遅延

(2) UDP パケットの転送性能

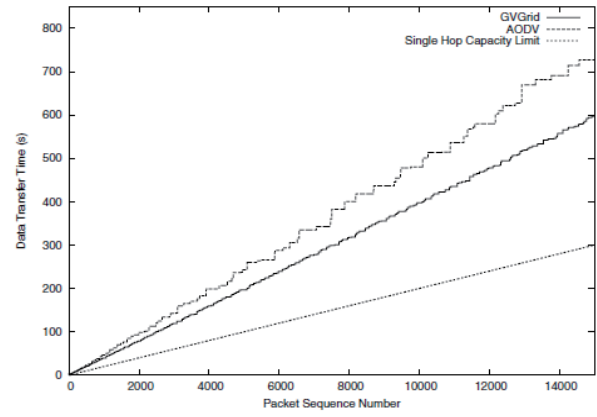
UDP パケットの転送遅延を測定した。経路が切断されるか、経路復元に時間を要する場合、中間ノードバッファでの蓄積またはあふれにより、パケット到着が遅延したり、パケットそのものが棄却される。図 2 および図 3 に、低密度下および高密度下でのパケット遅延をそれぞれ示す。これらは、図 1 に示した(A)~(E) のシナリオ下での、各シミュレーション時刻に送出されたパケットの到着遅延を示したものである。ソースノードは 2Mbps の CBR トラフィックを生成する。リンク容量と空間的制約により、移動がない状況でもある程度のパケット棄却が発生する。なお、図ではパケットロスに遅延ゼロで表現している。また、アプリケーションレベルでのパケット遅延の許容上限も合わせて与える。シナリオ (A)、(B)、(C) において 3 秒を超える遅延、シナリオ (D)、(E) において 5 秒を超える遅延はアプリケーション遅延上限を超えたものとみなし、棄却する。シナリオ (E) を除き、最小および最大遅延は、設計プロトコルおよび AODV とともに類似の傾向を示している。これは、道路形状から、いずれのプロトコルにおいても同じ道路上に経路が生成されるためである。しかしながら、すべてのシナリオにおいて、AODV は深刻なパケットロスを発生している。これは、AODV が車両移動を考慮して設計されているわけではないことに起因する。AODV では、反対方向に移動する車両間でリンクを構築してしまう可能性もある。また、AODV は最小ホップ経路を探索するため、しばしば堅牢性の低い経路を採用してしまう傾向がある。シナリオ (E) における AODV では、最小遅延は設計プロトコルよりも小さいが、頻繁なリンク切断により、高遅延状況が発生したり、パケット棄却が発生したりしている。一方、設計プロトコルでは主要道路上に経路を構築するため、そのような状況を回避できていることがわかる。また、平均パケット到着率を図 4 に示す。設計したプロトコルは車両密度にかかわらず、AODV を大きく上回っていることがわかる。

(3) TCP スループット

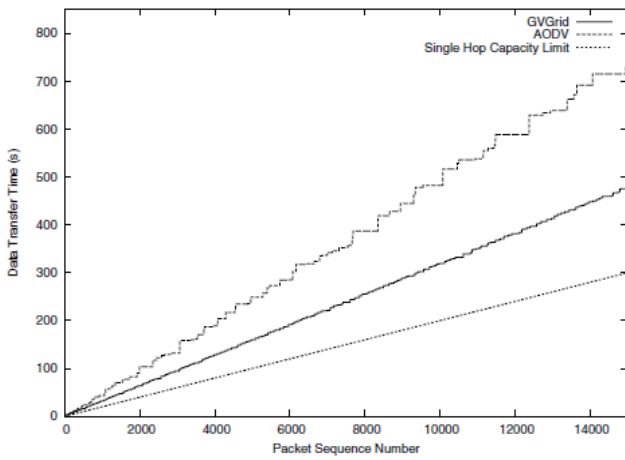
図 5 に、TCP 上でのデータ転送時間を示す。ここでは、低遅延ならびに高遅延での性能が似ているため、低遅延の結果のみを示す。アプリケーション層でのパケット番号 i に対し、パケット番号 i までのすべてのパケットを受信するまでの遅延時間を示している。転送データサイズは 75M バイトであり、各パケットは 5K バイト長でアプリケーション層から送出される。明らかに、提案プロトコルは特に主要道路をその道路地図に含んでいるシナリオ (C)、(D) および (E) において低遅延を実現できている。これは提案プロトコルがそのような道路上の車両からなる経路を構築でき、車両の移動に対しても堅牢であることを示している。図 6 にアプリケーションレベルのスループットを要約して示しており、提案プロトコルの有効性が読み取れる。



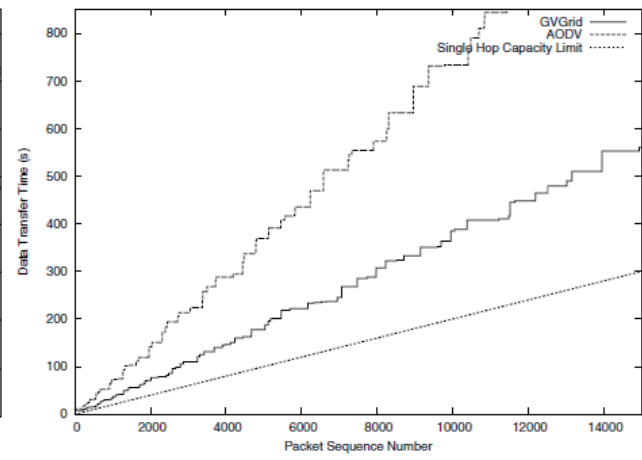
(A)



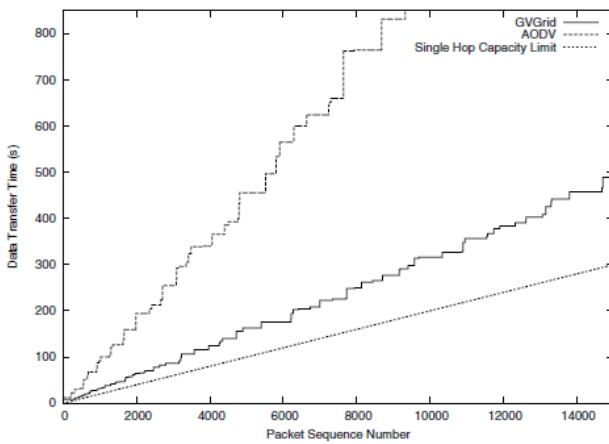
(B)



(C)



(D)



(E)

図 5 TCP パケット遅延

	(A)	(B)	(C)	(D)	(E)
GVGrid (Mbps)	1.6276	1.0026	1.2630	1.0417	1.2370
AODV (Mbps)	1.4063	0.7942	0.7877	0.5598	0.4818

図 6 アプリケーションレベルスループット

2-2 アドホックネットワーク用性能評価環境

(1) 概要

アドホックネットワークのプロトコルやアプリケーションでは、テストベッドを構築し、多数のノードを操作しながら性能評価を行うことが困難である。そこで、アドホックネットワークプロトコルの性能評価を支援する開発環境を設計開発したため、それについて報告する。この開発環境では、シミュレーション用コードスケルトンを実機用コードから自動生成し、実環境とシミュレーション環境を連携させて性能評価することが可能となる。GPSR (B. Karp and H. T. Kung, 2000)、SPEED (T. He, J. A. Stankovic, C. Lu and T. Abdelzaher, 2003)、BIP (J.E. Wieselthier, G.D. Nguyen and A. Ephremides, 2000)、Rumor Routing (D. Braginsky and D. Estrin, 2002) といった位置情報ルーティングプロトコルを実装し、その評価を行った。特に SPEED について実環境とシミュレーションによる性能評価実験を行い、文献 (T. He, J. A. Stankovic, C. Lu and T. Abdelzaher, 2003) で報告されている評価と比較することで、この開発環境の妥当性を示している。

(2) 実環境とシミュレーション環境のシームレスな連携による性能評価

シミュレーションと実環境実験との間の連携を行い、性能評価を支援するための機能の提供を以下のように実現した。連携機能として、(1) 2 者間でのコード共有、(2) アニメータによる実環境実験結果の可視化、及び (3) 実環境のログを元にしたシミュレーションのパラメータ設定、が挙げられる。コード共有機能により、各ノードの実機用コードは QualNet シミュレータ用の C++コードのテンプレートを生成することもできる。可視化機能は、実環境のセンサノード群の動作や状態を QualNet アニメータで表示することができる。各ノードのバッテリーの電力残量や LED の点灯状態 (Mica MOTE を想定) を表すグラフィックを提供する。この機能は、実環境で動作させる端末に搭載するプログラムに管理用コンポーネントを追加し、環境依存のメッセージ送受信等のイベント、センサノードの電力残量や位置などの内部状態のログの取得や集約を、それぞれのノード上のコンポーネントが協調して行うことにより実現する。また、パラメータ設定機能は、実環境で得たノードの位置情報や受信メッセージの電波強度のログを元に、実環境での設定や電波状況を継承したシミュレーション環境へのシームレスな移行を実現する。これにより、小規模ネットワークでは実環境で評価し、中規模から大規模ネットワークでは同様の設定でシミュレーションにより評価するといったことが容易に行える。

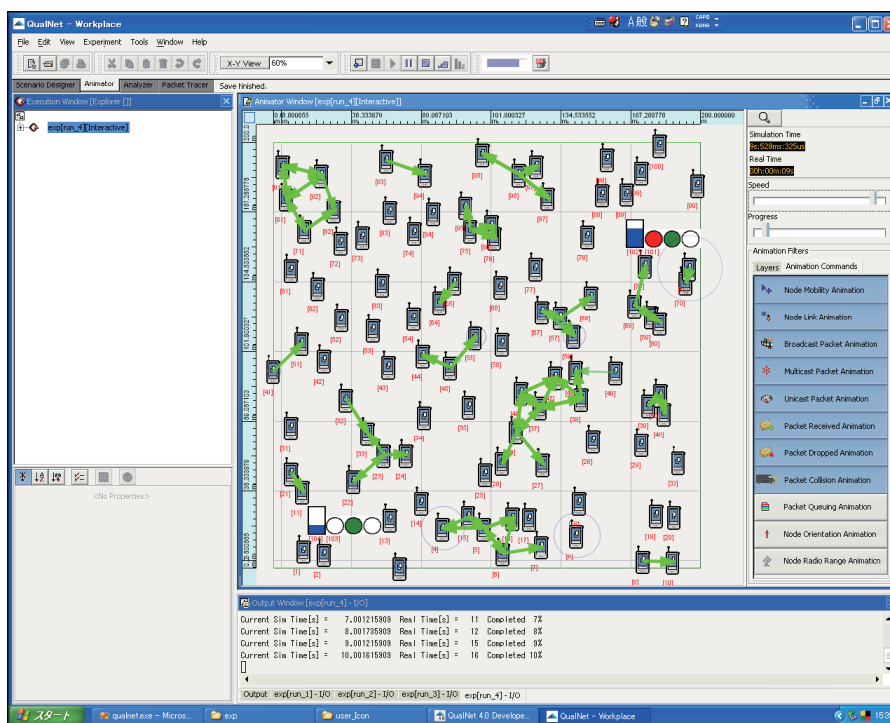


図 7 QualNet シミュレータによるノード表示

(3) 実機ノード管理支援

アドホックネットワークの各ノードにデバッグエージェント、いずれかのノード（これを便宜上、基地局とよぶ）にはデバッグコントローラと呼ばれるモジュールを常駐動作させる。デバッグコントローラとデバッグエージェントとの協調により分散環境でのデバッグを実現する。デバッグ支援では、開発者が、どのノードでどのような条件が成立した場合にネットワーク全体でプログラム実行を停止したいか、また停止後にはどのノードがどのような処理を行うべきかについて記述する。この記述をデバッグシナリオと呼ぶ。具体的には、デバッグシナリオは文の集合からなり、1つの文は複数のノードの内部変数やそれらのノードにおいて発生するイベントについての条件式と、その成立時に実行する処理（アクションと呼ぶ）を記述した動作記述から構成される。

条件式は、基本的にノード名とそのノードで成立判定が可能な部分条件（これを局所条件と呼ぶ）の論理結合の形で記述する。これに対し、各ノード上のデバッグエージェントは、そのノードの局所条件が成立した場合には、基地局のデバッグコントローラへその旨を通知する。基地局上のデバッグコントローラは、条件全体が成立するかを各ノードからの通知に基づき常に判断している。ノード名として\$が指定された局所条件は、すべてのノードがそれぞれその条件判定を行い、成立した場合はその事実をデバッグコントローラに通知する。デバッグコントローラはその通知の送信者を\$に割り当て、アクションを実行する。さらに、ノード名の代わりに、ある条件を満足するノードの集合を指定することもできる。この場合は、各ノードがそのノード集合に自身が含まれるか否かを自身のみで判定できることが前提となる。例えば、ある正方領域に存在する各ノードを局所条件の判定ノードとして指定することもできる。この例では、各ノードが位置情報を保持していれば、その判定は自身で可能である。また、ノードの位置関係、例えばあるノードからのホップ数制限を利用したノード集合指定を行う場合は、位置関係（この例ではホップ数）をあらかじめ何らかの方法で把握していることが前提となる。また、条件式中に実機コード中の変数や関数名を参照して記述することができる。例えば、条件式中に実機コード中の関数名を記述すると、その関数の実行時に値が真となり、ノードの動作状況に応じたデバッグシナリオの実行が可能となる。

以下は、ルーティングにおけるループ検出を行うシナリオ例である。

```
$.on_loop_detected -> |
  nodeIDs = $.received_packet->recent_visit_nodes;
  foreach u in nodeIDs {
    u.refresh_table();
  }
```

このデバッグシナリオでは、1行目が条件を表し、2行目以降が条件成立後のアクションを表す。前述したように、各条件およびアクションでは、変数および関数の前にノードの指定を記述することで、それらを判定および実行するノードを明記する。なお、指定無しの変数や関数は基地局上のデバッグコントローラで保持されていることを意味する。このシナリオは、“\$”で記されるノード（すなわち任意のノード）において関数 `on loop detected` が呼び出された場合（この関数はルーティングメッセージが同じノードを経由する場合に呼び出される）、“\$”ノードの保持する構造体 `received packet` に含まれるノードのリスト `recent visit nodes` を、基地局の持つリスト `nodeIDs` に代入する。そして、`nodeIDs` に含まれる各ノードに関数 `refresh table()` を実行させる。これは、ルーティングプロトコルにおけるアサーションや事後条件の指定に用いることが出来る。

以下の例はシステムのモニタリングや保守を目的としたものである。

```
$.region_square(a,b) && ($.residual_energy < 0.2) -> |
  $.beacon_interval = $.beacon_interval * 2;
```

このシナリオは、地点a、bを対角線上の2頂点とした正方形の領域内に含まれるノード（それらのノードは“`region square(a, b)`”により得られる）であり、その電力残量 `residual energy` の値が20%以下であれば、ノードの持つ変数 `beacon interval` の値を2倍にする。

これらのシナリオは、デバッグコントローラとデバッグエージェントによって実現するが、データや通知の交換や処理実行のプロトコルは更に検討の余地がある。基地局が離れていたり、局所条件が頻繁に成立および不成立を繰り返したりする場合、基地局への通知のためのトラフィックが増大する。したがって可能な場合はノード間協調で条件判定するなど、トラフィックオーバーヘッドを考慮してどのノードが何をすべきかが自動的に決定できることが望ましい。現在、サービス自動分散化技術を応用してこれらを実現する方法を検討中である。

(4) 実環境実験

SPEED プロトコルの性能評価を行い、シミュレーションの性能を文献 (T. He, J. A. Stankovic, C. Lu and T. Abdelzاهر, 2003) の報告と比較した。実験シナリオはこの文献と同様のものを使用した。このシナリオは SPEED プロトコルの輻輳回避の性能のテストを目的としている。実験環境は下表に示す正方領域で、ノードはランダムに配置されている。

	既存報告	シミュレーション	実環境
PHY & MAC	802.11	802.11	802.15.4
帯域	200 Kb/s	200Kb/s, 250Kb/s	250Kb/s
ペイロードサイズ	32 Bytes	32 Bytes	32 Bytes
領域	(200m,200m)	(200m,200m), (20m,20m)	(20m,20m)
ノード数	100	100, 25	25
配置	均一	均一	均一
無線範囲	40m	40m, 8m	8m

ストリームの送信元として領域の左側のいくつかのノードからランダムに送信ノードを選び、ストリームの送信先は領域の右側の基地局となる。送信ノードは 1 packet/sec の一定のビットレートで送信を行う。また、輻輳を発生させるために領域の中央付近から 2 ノードをランダムで選択し、実験時間 150 秒のうち 75 秒から 150 秒の間、その 2 ノード間でフローを発生させる。輻輳度合いの推移による輻輳回避を評価するため、輻輳発生用のフロー (以下、輻輳フロー) は、0 から 100 packets/sec まで 10 刻みの各フローレートを用いて数回ずつのシミュレーションを実行し、基地局へのデータパケットの遅延とロス率を評価する。

実環境実験では端末数に制限があったため、25 ノードによる評価を行った。シミュレーション実験は、比較のために文献 (T. He, J. A. Stankovic, C. Lu and T. Abdelzاهر, 2003) による報告、実環境実験の両方それぞれと同様の設定で行った。MOTE とシミュレータの無線範囲の設定は、それぞれ領域のサイズに合わせた値に設定し、シミュレータではまず揺らぎの無い理想的な電波到達環境を仮定した。

図 8(a) にエンドノード間のパケット到達遅延を示す。100 ノードの実験では、シミュレーション性能と既存報告性能は輻輳フローが 40 packet/sec 時に若干の差が見られるものの、同様の傾向を示していることが分かる。25 ノードの実験では、シミュレーションでは 100 ノード時と同程度の遅延が観測されたが、実環境実験ではシミュレーションに比べ遅延が小さいことが確認できる。図 8(b) はパケットのロス率 (基地局に到達しなかったパケット数) を示している。100 ノードの実験では、シミュレーションの性能と既存報告性能はほぼ同程度であることが分かる。25 ノードの実験でも、ほぼ同等の値を示している。

図 8で見られるように、実環境とシミュレーションの結果を比較すると、実環境の方が遅延が小さく、パケットロス率が高くなっている。これは、実環境では電波の揺らぎが大きいことが原因であると考えられる。実環境では、ノードはより遠くのノードからのビーコンを受信でき、その発信ノードを隣接テーブルに登録する。そのため、より遠くのノードを選択してパケット送信を行うために遅延が小さくなるが、パケットロスの確率は高くなったと考えられる。また、このような実環境実験の結果と分析を踏まえ、実環境実験に合わせたパラメータ設定を行った。具体的には、機器を直に床に設置していたことを踏まえ、アンテナ高を初期値の 1.5m から実際の値である 0.1m に変更し、10m のノード間距離であってもある程度のパケット受信を確認したことを反映し、通信範囲が 8m となる初期値-23.5dBm から、距離が 10m であっても同様に受信可能となる 0dBm へと電波出力を変更した。さらに、パスロスモデルを Flat model から実環境に近い Two Ray へ変更した。また、SPEED プロトコルではパケットの伝送速度に対する閾値を設け、パケットの転送に要する時間や転送先のノードまでの距離などを用いて速度を算出し、閾値を上回るような転送先を選ぶ。この際に用いる値を、実機におけるパケット転送の処理速度を考慮し、100m/s から 500m/s へと設定し直した。この設定を用いて、上表の実環境実験と同様の配置による評価を行った。その結果、図 9のように実環境実験により近づいたシミュレーションの結果を示すことができた。さらに、同様のノード間距離で 40m 四方の領域に配置した 100 ノードのシミュレーションについても、実環境実験に合わせたパラメータ設定を行った場合と行わなかった場合との比較を行った。その結果、図 9のように、25 ノード時に大きな差があったパケットロス率の差が小さくなった。これは、25 ノード時はネットワークを構成するノード数が小さく、各ノードの隣接ノードが少ないため、実環境に合わせた設定を行い電波到達距離が大きくなった

場合、その干渉により輻輳が起こっている中央付近を回避する経路を得られないという状況が起こっていたと考えられるが、100 ノード時には十分な迂回路が得られるために電波到達距離の差の影響を受けにくくなったためと考えられる。このように、実環境実験とシミュレーション実験を平行して行うことで、実環境をより良く再現したシミュレーション環境での評価を行える。また、その環境をより大規模なシミュレーションに適用することにより、環境の差異が様々な規模のネットワークに与える影響を評価できる。

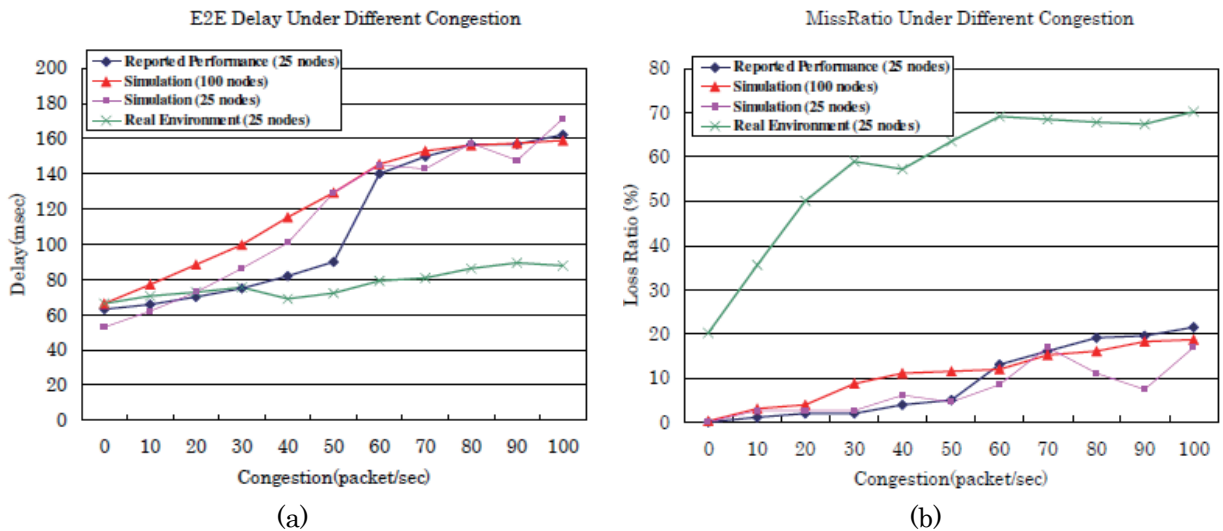


図 8 SPEED プロトコルの性能 (改善前)

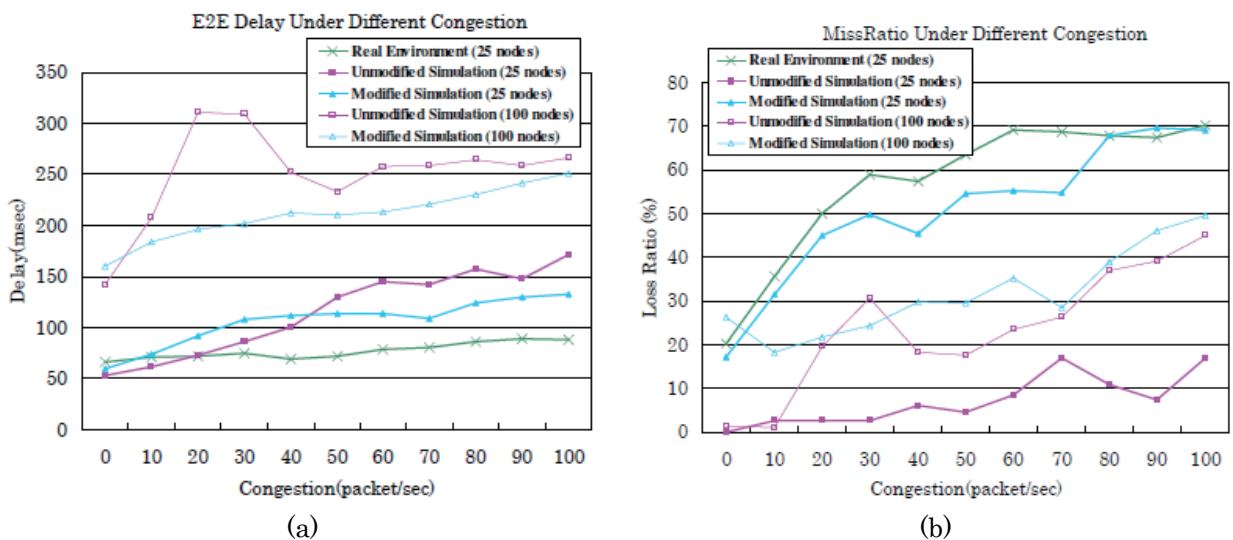


図 9 SPEED プロトコルの性能 (改善後)

【参考文献】

C. Lochert, M. Mauve, H. Fusler, and H. Hartenstein. Geographic routing in city scenarios. ACM SIGMOBILE Mobile Computing and Communications Review, pages 69-72, 2005.

B. Karp and H. T. Kung. GPSR: Greedy perimeter stateless routing for wireless networks. In Proc. of ACM/IEEE Mobicom, pages 243-254, 2000.

Y.-B. Ko and N.H. Vaidya. Location-aided routing (LAR) in mobile ad hoc networks. In Proc. of ACM/IEEE Mobicom, pages 66-75, 1998.

H. Yamaguchi, K. El-Fakih, G. V. Bochmann and T. Higashino. Deriving protocol specifications from service specifications written as predicate/transition-nets, Computer Networks, Vol.51, No.1, pp.258-284, 2007.

T. He, J. A. Stankovic, C. Lu and T. Abdelzaher. SPEED: A stateless protocol for real-time communication in sensor networks, Proc. of the 23rd International Conference on Distributed Computing Systems (ICDCS2003), pp.46-55, 2003.

J.E. Wieselthier, G.D. Nguyen and A. Ephremides, On the construction of energy-efficient broadcast and multicast trees in wireless networks, Proc. of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM2000), pp.585–594, 2000.

D. Braginsky and D. Estrin. Rumor routing algorithm for sensor networks, Proc. of the 1st ACM International Workshop on Wireless Sensor Networks and Applications (WSNA2002), pp.22–31, 2002.

〈 発 表 資 料 〉

題 名	掲載誌・学会名等	発表年月
車車間ルーティングプロトコルによる車両と基地局との通信プラットフォームの提案	情報処理学会マルチメディア, 分散, 協調とモバイル (DICOMO2008) シンポジウム論文集	2008年7月
ワイヤレスセンサネットワークの設計開発支援環境 D-sense	情報処理学会マルチメディア, 分散, 協調とモバイル (DICOMO2008) シンポジウム論文集	2008年7月
車車間ルーティングプロトコルによる車両と基地局との通信プラットフォームの提案	情報処理学会マルチメディア, 分散, 協調とモバイル (DICOMO2008) シンポジウム論文集	2008年7月
D-sense: An Integrated Environment for Algorithm Design and Protocol Implementation in Wireless Sensor Networks	Proc. of the 11th IFIP/IEEE International Conference on Management of Multimedia and Mobile Networks and Services	2008年9月
車車間通信を用いた VICS データ中継手法の提案	第16回マルチメディア通信と分散処理ワークショップ論文集	2008年12月