

特別研究報告

題目

**車車間アドホックルーティングプロトコル GVGrid における
経路修復機能の実装と評価**

指導教員

楠本 真二 教授

報告者

濱田 淳司

平成 19 年 2 月 20 日

大阪大学 基礎工学部 情報科学科

平成 18 年度 特別研究報告

車車間アドホックルーティングプロトコル GVGrid における
経路修復機能の実装と評価

濱田 淳司

内容梗概

近年，基地局を用いない車車間マルチホップ通信技術が注目を集めており，私の所属する研究グループにおいても，車車間マルチホップ通信における位置情報ルーティングプロトコル GVGrid を提案している．GVGrid は，車車間マルチホップ通信における既存の位置情報ルーティングプロトコル GPCR と比較し，経路生存時間が長いことがシミュレーション実験により示されている．

本研究では，GVGrid の主要機能である経路修復機能の実装を行った．この実装は，Uppsala University で開発されている AODV プロトコルの実装である AODV-UU をもとに，その経路修復機能部分を追加修正することで実現している．

実マップにおける車両の移動を想定して作成したシナリオに基づき，GVGrid と AODV の経路上に ICMP パケットを連続送出した上で，スループットの比較評価を行った．その結果，GVGrid はより高いスループットを達成でき，かつプロトコルの制御パケット数が少ないことを確認した．

主な用語

モバイルアドホックネットワーク (MANET)，車車間通信，経路修復，AODV

目次

1	まえがき	1
2	車車間ルーティングプロトコル GVGrid	2
2.1	前提条件	2
2.2	経路探索プロセス	3
2.2.1	隣接ノードの選択方法	4
2.3	ルーティングテーブルの更新	6
2.4	経路維持プロセス	6
3	AODV プロトコル	7
3.1	経路構築プロセス	7
3.2	ルーティングテーブルの更新	7
3.3	経路維持プロセス	8
3.4	AODV-UU	9
4	経路維持プロセスの実装	10
4.1	リンク強制切断の実装	10
4.2	経路維持メッセージ作成・送信機能の実装	10
4.3	経路維持メッセージ受信時の処理の実装	11
4.4	経路維持プロセスにおけるルーティングテーブル更新の実装	12
5	評価実験	14
5.1	既存のテストベッドとの位置づけ	14
5.2	MAC 層における filter 機能	14
5.3	車両の移動シナリオと経路切断状況	15
5.4	計測ツール	21
5.5	評価結果	22
6	あとがき	26
	謝辞	27
	参考文献	28

1 まえがき

近年、高度交通システムの普及を目指し、DSRC や VICS などの狭域通信器が徐々に路側や店舗などに設置されつつある。しかし、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必要である。さらに整備維持コストの観点からは、技術の進歩や改新に対する機器の更新が容易でないといった問題点もある。そこで、固定インフラの通信範囲外の地域へ情報を配信するために配信範囲を拡大する補完的な役割や、極めて局所的な情報伝達の手段として、固定インフラを用いない車車間マルチホップ通信技術が注目されている。そのアプリケーションとして、事故情報や渋滞情報のリアルタイム配信や車両へのインターネットサービスの提供などが考えられる。しかし、これらのアプリケーションの実現にはリアルタイム性が求められるため、移動する車両間において出来る限りデータ遅延および損失の少ない安定した通信経路の構築が要求される。

車車間ルーティングプロトコルの 1 つに、私の所属する研究グループで提案している GVGrid[6] がある。GVGrid は、近距離無線基地局や事故現場に停車した緊急車両などの移動しない情報送信元から、交差点などの特定地点を含む領域に存在または通過する車両群への通信経路をオンデマンドに構築する。通信経路を構築する際に車両 (以下ノードと呼ぶ) の移動に伴った通信経路の切断を軽減するために、道路に沿って移動するというノードの移動特性を考慮し、GPS などの測位機器と電子地図を利用してなるべく道路に沿った経路を構築する。

また、GVGrid は、地理領域を正方形の領域 (以下グリッドと呼ぶ) に分割し、初期経路の形状をその経路が通過するグリッド列として記憶する。初期経路は広範囲を探索した結果得られた良質の経路であると考えられるため、経路切断時には記憶したグリッド列上のみで再探索を行い初期経路に近い新たな通信経路を再構築することで、耐切断性の高い通信経路を維持しながらも再構築のためのメッセージ数を削減できるという特長を持っている。

本研究では、GVGrid の主要機能である経路維持機能の実装を行い、また AODV との経路切断における経路修復機能の比較評価を行った。

GVGrid の経路修復機能の実装は、Uppsala University で開発されている AODV プロトコルの実装コードをもとに行った。また、無線 LAN を備えたノート PC10 台程度の小規模テストベッドを構築し、実マップにおける車両の移動を想定して作成したシナリオに基づき、GVGrid と AODV の経路上に ICMP パケットを連続送出した上で、スループットの比較評価を行った。その結果、GVGrid はより高いスループットを達成でき、かつプロトコルの制御パケット数が少ないことを確認した。

2 車車間ルーティングプロトコル GVGrid

GVGrid は、各車両が GPS などの測位装置及び電子地図を利用可能であるという仮定の下で、近距離無線基地局や停止中の緊急車両などの固定送信元から目的地となる終端座標を含むグリッドへの通信経路をオンデマンドに構築し、維持する車車間ルーティングプロトコルである。GVGrid は車両密度が高く路側に高層ビルなどが立ち並ぶ様な都市部を対象としている。その下で道路に沿って移動するという車両の移動特性を利用し、道路に沿った通信経路を探索することで耐切断性の高い通信経路の構築を実現している。GVGrid プロトコルは、初期経路を構築するための経路探索プロセスと、経路切断時に再構築するための経路維持プロセスからなる。

2.1 前提条件

各ノードは一意な ID を持ち、近距離無線デバイス、GPS などの測位装置、電子地図を装備しているとする。道路セグメントの集合を E 、交差点の集合を V とすると、電子地図は有向グラフ $G = (E, V)$ で表される。

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、各方向ごとに 1 セグメントとする。各セグメントにはそれを識別するセグメント識別子が与えられているとする。また、各セグメントには国道番号などの道路識別子が存在するならばそれが与えられているものとし、各ノードは、道路セグメントが同じ道路を構成するものかを判断をすることができる。各交差点には交差点識別子に加え、その位置情報及び信号の有無に関する情報が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 $S(u)$ および道路識別子 $R(u)$ 、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を把握することができる。

GVGrid は地理領域をグリッドに分割する。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。あるグリッド内に存在する各ノードが、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に、グリッドサイズ w と無線範囲 r の関係を $w = \sqrt{2} \cdot r / 4$ と定義する (図 1)。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 、緯度 y の座標に位置するノード v が存在するグリッド $IDG(v)$ を以下のように定義する。

$$G(v) = G_x(v) \cdot W + G_y(v)$$

ただし、 $G_x(v)$ は x の w による商、 $G_y(v)$ は y 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ので、グリッド ID は x と y よ

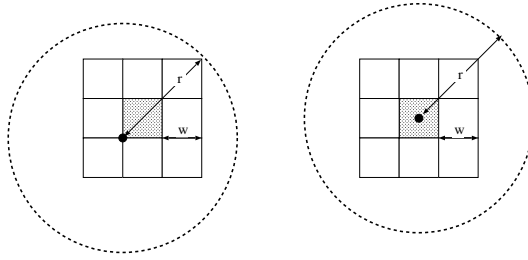


図 1: 無線距離 r とグリッドサイズ w の関係

り一意に決定される. これにより, 各ノードは共通定数 w および W から, ある位置を含むグリッド ID を計算することができる.

2.2 経路探索プロセス

経路探索プロセスでは, 始点ノードを s , 終点座標を d とすると, s から d を含むグリッド $G(d)$ まで道路経路に沿った通信経路を構築する. その際, LAR[1] 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 $G(d)$ への経路を探索する場合, s は通信領域内である $G(s)$ の各隣接グリッドから後に述べる方法を用いて 1 ノードずつ選択し, 経路探索 (Route Request, 以下 RREQ) メッセージを送信する. 送信した RREQ メッセージには s のノード ID, s が存在する道路の道路セグメント ID $S(s)$, 道路 ID $R(s)$ およびグリッド 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, 同様の選択方法で通信領域内の各隣接グリッドから 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 およびグリッド 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, $G(d)$ 内の最もノード ID の小さいノード d' ($G(d)$ の代表ノードと呼ぶ) を指定して RREQ メッセージを転送する. d' 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 メッセージから最も品質の良いと思われる通信経路を選択し, その経路を利用して s へ, 選択された RREQ メッセージに記録されていたノード ID 列とグリッド ID 列を保持した経路決定 (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. なお, 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使用される.

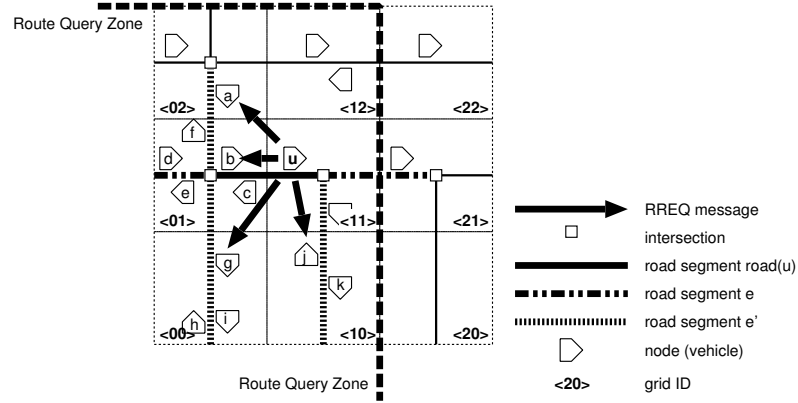


図 2: 隣接ノードの選択例

2.2.1 隣接ノードの選択方法

ノード u が隣接グリッドから RREQ メッセージを転送するノード v を選択する方針は以下の通りである。なお、以下は $S(u)$ と $S(v)$ が交差点を 1 つ以上共有しているような v (すなわち同一、逆方向、あるいは隣接道路セグメントに存在する v) のみを対象とする。

- (1) $S(u) = S(v)$ の場合、2 ノードは同じ道路セグメントにあるため、近い速度で同じ方向に移動していると考えられる (図 2 のノード b)。また、 $S(u) \neq S(v)$, $R(u) = R(v)$, かつ u と v の移動方向が等しい ($Dir(u) = Dir(v)$) 場合、交差点を越えた同じ道路上の同方向の道路セグメントに v が存在する (図 2 のノード d)。これらのノードを最も優先して選択する。なお、 u と v との距離が w に近いほど、経路上のノード間の間隔が一定になり、2 ノード間の距離が大きくなることによる切断が発生しにくいと考えられるため、 u にとっては、隣接グリッド内に理想位置が存在する。上記の条件を満たすノードが複数存在する場合、理想位置に近いノードの優先度を高くする。ただし、 $S(u)$ と $S(v)$ に共有される交差点がグリッド内にあれば、その交差点を理想位置とする。
- (2) $S(u) \neq S(v)$ であるが、 $R(u) = R(v)$ かつ $Dir(u) \neq Dir(v)$ である場合、 v は交差点を共有する同じ道路上の逆方向の道路セグメントと考えられる。この場合、2 ノード間の相対速度が大きく、切断する可能性が高いと考えられるため、2 番目に優先して選択する。上記の条件を満たすノードが複数存在する場合、前述の方法と同様に理想位置に基づき優先度を決定する。
- (3) $S(u) \neq S(v)$ かつ $R(u) \neq R(v)$ である場合は最も優先度を低くする (図 2 のノード a , g , i)。上記の条件を満たすノードが複数存在する場合、前述の方法と同様に理想位置

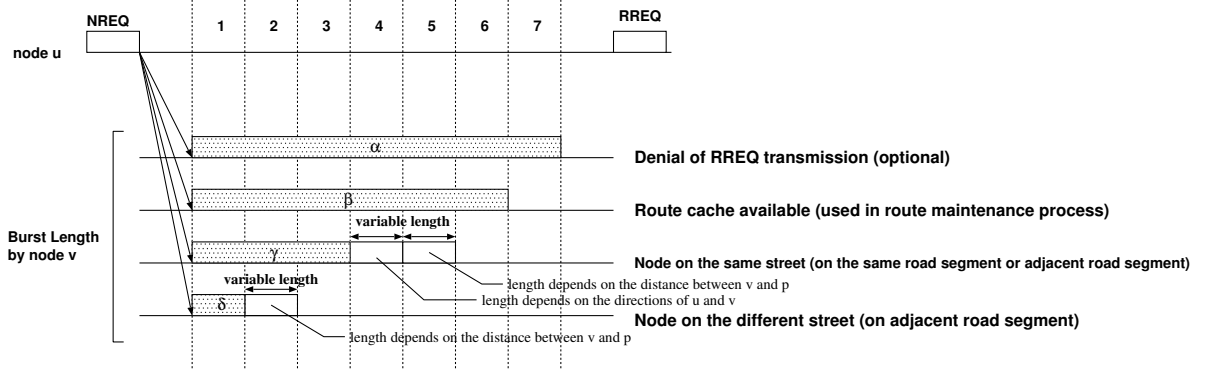


図 3: 隣接ノードのブラックバースト長

置に基づき優先度を決定する。

これらの情報は、一般に Hello メッセージを定期的に隣接ノードと交換することで収集できる。しかし、GVGrid はノード密度の高い都市部での利用を想定しているため、頻繁な Hello メッセージの交換はパケットの衝突を引き起こす可能性が高い。また、Hello メッセージの交換頻度を下げれば、車両の移動速度は大きいので、位置のずれが大きく、経路の品質に影響を与える。

そこで GVGrid では、Hello メッセージを使用せずに、ブラックバースト (*Black burst*[5]) と呼ばれる手法を改良して使用している。ブラックバーストは、もともと IEEE801.11 における音声通信をサポートするためにノードにチャネル利用の優先度を与えるための技法であり、ノード検索要求に対し、各ノードは優先度に比例したバーストを発生する。バースト発生後にチャネルがビジーでなければそのノードがデータを送信できる。

GVGrid ではこのブラックバーストを以下のように改良している。ノード u はあるグリッド g から隣接ノードを選択する際、NREQ メッセージを送信する。NREQ メッセージには対象グリッド ID g 、 u の存在する道路の道路セグメント ID $S(u)$ 、および道路 ID $R(u)$ 、 u の移動方向 $Dir(u)$ 、 u の座標 $P(u)$ を含める。NREQ メッセージを受信した各ノード v は、まず自身の存在するグリッド ID $G(v)$ が g と一致するか確認する。異なっていればメッセージを破棄し、一致していれば以下の方針でバースト長 $B_{len}(v)$ を決定する。以下、 α 、 β 、 γ 、 δ を定数とし、“ $\alpha > \beta > \gamma + 1$ ” 及び “ $\gamma > \delta$ ” を満たすとする。

- $B_{len}(v) = \alpha \cdot T$: 最長バーストであり、グリッドにすでに RREQ メッセージを転送したノードが存在しており、例えばメッセージ数削減の目的のために RREQ メッセージの転送を防止したいなどの場合に与えられる。
- $B_{len}(v) = \beta \cdot T$: 経路維持プロセスにおいて、経路の再探索時に、通信経路の一部を

キャッシュしているノードがこのバースト長を設定することで、キャッシュを保持するノードが優先的に選択される。

- $B_{len}(v) = \left\{ \gamma + Dir(u, v) - \frac{dist(v, p)}{\sqrt{2} \cdot w} \right\} \cdot T$: 前述の (1) 及び (2) に相当するバースト長を表す。 $dist(v, p)$ は v と理想位置 p の距離を表し、 $-\frac{dist(v, p)}{\sqrt{2} \cdot w}$ によって v と理想位置 p との距離に応じたバースト減少分を表す。 $Dir(u, v)$ は $Dir(u) = Dir(v)$ の場合は 1, そうでない場合は 0 である。この値が 1 であるときは前述の (1) に相当し、そうでなければ (2) に相当する。
- $B_{len}(v) = \left\{ \delta - \frac{dist(v, p)}{\sqrt{2} \cdot w} \right\} \cdot T$: 前述の (3) に相当するバースト長を表す。

図 3 は、 $\alpha = 7, \beta = 6, \gamma = 3, \delta = 1$ とした場合のバースト長の例である。

2.3 ルーティングテーブルの更新

RREP メッセージを転送する際、決定された通信経路上の各ノードは、RREP メッセージ内に記録されているノード ID 列から、自身の前方および後方のノードを自身の保持している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。

2.4 経路維持プロセス

経路維持プロセスは、経路が切断した場合、経路探索時に構築された通信経路が存在する道路経路に対応するグリッド列（以下道路経路グリッド列と呼ぶ）上に存在しないノードを経路から切り離し、道路経路グリッド列上に限定して代替ノードを探すことで、強制的に道路経路グリッド列上に通信経路を復元させる。ノード s から $G(d)$ 内のノード d' までの通信経路は、中間ノードの移動や、 d' 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切断を検出した場合、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に経路から離脱する。下流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存在するノードは、自身のグリッドの前方グリッドから代替ノードを選択し、自身が記録しているグリッド列を含む経路維持（Route RePaiR, 以下 RRPR）メッセージを送信する。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（Route ERRor, 以下 RERR）メッセージを送信ノード s に向けて送信する。 s は RERR メッセージを受信すると経路探索プロセスを実行し、新たな経路を探索する。

3 AODV プロトコル

AODV プロトコル [7] は, MANET 向けのオンデマンド型ルーティングプロトコルである. AODV はブロードキャストによる経路構築・経路維持を行うが, 車両の移動情報などを用いない汎用的な MANET 型ルーティングプロトコルであることから, ノードが高速に移動しトポロジが頻繁に変化する車車間ネットワークでは切断回数が増加してしまう可能性がある.

AODV では, 各ノードがシーケンス番号を管理しており, それをルーティングにおいて利用する. またルーティングテーブルの各経路エントリ内には経路維持プロセスにおいて使用する precursor リストを保持する.

3.1 経路構築プロセス

経路構築プロセスでは, 始点ノード s , 終点ノードを d とし, s から d への通信要求が発生すると, s は隣接するノードへ向けて RREQ メッセージをブロードキャスト送信する. RREQ メッセージを受信したノードは, d への経路を保持していない場合は, RREQ メッセージをさらにブロードキャスト転送する. d が RREQ メッセージを受信すると, RREP メッセージを s へユニキャストで送信する.

AODV では, 各ノードは, シーケンス番号と RREQ メッセージの ID を保持している. シーケンス番号は, 経路がループすることを防ぎ, 新しい経路情報を保持することを保証している. RREQ ID は, ノードの IP アドレスと組み合わせて RREQ メッセージを一意に識別するためのものである. RREQ ID は, RREQ メッセージを送信する度に増加する.

RREQ メッセージを受信した中間ノードは, d への経路を保持しており, そのシーケンス番号が RREQ メッセージ内のシーケンス番号と等しいか, それ以上の場合は, d への新しい経路を保持していると判断し, RREP メッセージをユニキャストで送信する. d への経路を保持しているが, そのシーケンス番号が RREQ メッセージ内のシーケンス番号より小さい場合は, d への新しい経路を構築するために RREQ メッセージをブロードキャスト転送する.

3.2 ルーティングテーブルの更新

ルーティングテーブルの更新は, RREQ メッセージ受信時, RREP メッセージ受信時, HELLO メッセージ受信時に行われる.

AODV では, HELLO メッセージの利用は必須ではないが, 隣接ノードからの HELLO メッセージを受信した場合は, ルーティングテーブルに隣接ノードの情報を記録する.

RREQ メッセージを受信したノードは、送信元 s の情報とその RREQ メッセージを送信した隣接ノードのアドレスを送信元 s への次ホップとしてルーティングテーブルに記録し、逆方向の経路を確立する。なおそれ以降に、同じ RREQ メッセージを受信した場合は破棄される。RREP メッセージを受信したノードは、送信先 d の情報とその RREP メッセージを送信した隣接ノードのアドレスを送信先 d への次ホップとしてルーティングテーブルに記録し、順方向の経路を確立する。これにより、 s と d の間に双方向の経路が確立される。

また各経路エントリは、経路維持プロセスにおいて使用される precursor リストを保持している。precursor リストは、経路エントリにおける次ホップに対する前ホップのノード群からなる。

一般的に、 s から d への RREP メッセージを受信したノードは、RREQ 受信時に作成した s への逆方向の経路を保持している。そしてルーティングテーブルに d への経路エントリを作成する時に、ルーティングテーブルから s 方向への経路エントリを検索し、該当する経路エントリ内の次ホップの情報を、 d 方向への経路の前ホップとして precursor リストに記録する。そして、ルーティングテーブルから検索した s 方向への経路エントリには、RREP を送信してきた隣接ノードを、 s の経路の前ホップとして precursor リストに記録する。

またルーティングテーブルの各経路エントリには、経路タイマが用意されている。経路エントリは、通信が行われなくなってから一定時間後に無効状態になる。さらに無効状態が一定時間継続した後、その経路エントリはルーティングテーブルから削除される。

3.3 経路維持プロセス

有効な通信経路において経路切断が発生すると、経路の切断を検知した上流ノードは、送信シーケンス番号を増やした RREQ メッセージをブロードキャストで送信する。

RREQ メッセージを受信したノードは、自身が目的地への新しい経路を保持していない場合は、RREQ メッセージをブロードキャストで転送し、自身が目的地への新しい経路を保持している場合は、RREP メッセージをユニキャストで送信する。

経路の切断を検知した上流ノードは、経路修復のために RREQ メッセージを送信した後 RREP メッセージを受信するまでの間、送信元から送られてくるデータパケットをバッファリングし、RREP メッセージを受信して経路修復が完了するとバッファしたパケットを目的地へ送信する。ただし、そのノードが、一定の期間内に RREP メッセージを受信できなかった場合は、RERR メッセージを送信元へ向けて送信する。

さらにそのノードは、切断したリンクを利用している全ての送信先を不達送信先として、それらへの経路エントリを全て無効状態にする。そして各不達送信先の経路エントリの precursor リストに対して、そのリストに登録されている全ての送信先を送信先とした RERR のブロードキャスト送信を行い、RERR の送信先に対応する経路エントリを無効状態とする。なお、

RERR の送信先が 1 つの場合はユニキャストで送信する。

RERR を受信した各ノードは、対応する経路エントリを全て無効状態にし、同様の手順で RERR を送信する。この繰り返しにより、RERR を s が受信し、かつ s が d へ通信を継続したい場合は、 s は d へ経路構築プロセスと同じ手順で経路の再構築を行う。

3.4 AODV-UU

AODV-UU は、スウェーデンの Uppsala University が中心となって、AODV プロトコルの RFC[9] に基づいて開発されている。

本節では、AODV-UU の動作について述べる。

AODV-UU は、カーネルから送信するデータパケットを受け取ると、ルーティングテーブルにパケットの送信先への経路エントリが存在するか判定する。経路エントリが存在しない場合は経路構築を開始し、経路が構築された後にデータパケットの送受信を行う。そしてデータパケットを受信したノードがデータパケットの目的地であった場合は、受信したデータパケットをカーネルに渡す。

また、中間ノードにおいては受信したデータパケットの目的地への経路を保持している場合は、カーネルに渡さずにそのまま転送する。受信したデータパケットの目的地への経路を保持していない場合は、データパケットをカーネルに渡してバッファリングし、目的地への経路構築を行う。そして経路構築後にバッファリングしていたデータパケットの送信を行う。

AODV-UU は、このようにカーネルとの間でデータパケットをやりとりしている。

4 経路維持プロセスの実装

GVGrid の経路維持プロセスの実装は、AODV-UU の実装コードを基に行った。

GVGrid の経路維持プロセスは、以下の四つの処理からなる。

- 経路の切断を検知したノードが行う処理におけるリンク強制切断処理
- 経路の切断を検知したノードが経路の修復を開始する時における経路維持メッセージ作成・送信処理
- 経路維持メッセージ処理
- 経路の修復に関係するノードが行う処理におけるルーティングテーブル更新の処理

本章では、AODV-UU の実装コードをもとにした、GVGrid の上記の四つの処理からなる経路維持プロセスの実装法について述べる。

4.1 リンク強制切断の実装

GVGrid では、経路の切断を検知したノードは、自身が存在するグリッドが道路経路グリッド列上に存在するか否かを検査し、自身のノードが存在するグリッドが道路経路グリッド列上に存在しない場合は、自身と接続しているノードとのリンクを強制切断し、通信を終了する。

本実装では、各ノードはルーティングテーブル内にグリッド ID 列を保持していないため、経路の切断が発生した場合、経路の切断を検知したノードが強制切断を行うかどうかはシナリオファイルで指定している。

4.2 経路維持メッセージ作成・送信機能の実装

AODV-UU では、経路の切断を検知したノードは、RREQ メッセージを作成してブロードキャストで送信する。

これに対し、GVGrid では、自身のノードが存在するグリッドが道路経路グリッド列にある場合は、RRPR メッセージの作成を行う。なお、GVGrid の RRPR メッセージと AODV-UU の RREQ メッセージのフォーマットは同一である。GVGrid では RRPR メッセージの作成後、自身のグリッドの前方グリッドから代替ノードを選択し、自身が記録しているグリッド列を含む RRPR メッセージを代替ノードに送信するが、現状の実装では位置情報を与えることができていないため、自身が記録しているグリッド列を含めずに、RRPR メッセージをシナリオによって指定されたノードへ送信するようにしている。

4.3 経路維持メッセージ受信時の処理の実装

AODV-UUでは、RREQメッセージを受信したノードは、保持しているルーティングテーブルを検査し、ルーティングテーブル内に目的地への経路エントリを保持していない場合は、RREQメッセージをブロードキャストで転送する。保持しているルーティングテーブル内に目的地への経路を保持している場合は、切断した通信経路の残存ノードであり、この場合は以下のいずれかの処理を行う。

- (1) RREQメッセージを受信したノードが、保持しているルーティングテーブルに目的地への経路エントリを保持しており、かつ目的地への経路が有効であるとき、経路エントリの目的地シーケンスナンバーが受信したRREQメッセージ内のシーケンスナンバーより大きい場合、シーケンス番号が大きいほど最新の経路であると考えられることから、このノードから目的地までの経路においては、経路エントリ内の経路の方がRREQメッセージ内の経路より最新の経路であることがわかる。したがって、RREPメッセージを作成し、経路切断を検知したノードへ送信する。
- (2) RREQメッセージを受信したノードが、保持しているルーティングテーブルに目的地への経路エントリを保持しており、かつ目的地への経路が有効であるとき、経路エントリの目的地シーケンスナンバーが受信したRREQメッセージ内のシーケンスナンバーより小さい場合は、シーケンス番号が大きいほど最新の経路であると考えられるため、このノードから目的地までの経路においては、RREQメッセージ内の経路の方が経路エントリ内の経路より最新の経路であることがわかる。したがって、最新の経路を作成するためにRREQメッセージを作成し、ブロードキャストで転送する。
- (3) RREQメッセージを受信したノードが、保持しているルーティングテーブルに目的地への経路エントリを保持しているが、目的地への経路が無効であるとき、目的地への新しい経路を作成するために、RREQメッセージをブロードキャストで転送する。

これに対し、GVGridはRRPRメッセージ転送時には、シーケンス番号における処理は行わず、目的地方向へのノードを選択してユニキャストで送信する手法をとっているため、AODV-UUを次のように修正した。

GVGridでは、RRPRメッセージを受信したノードは、保持しているルーティングテーブルを検査し、ルーティングテーブル内に目的地への経路エントリを保持していない場合は、RRPRメッセージをシナリオによって指定されたノードへ転送する。保持しているルーティングテーブル内に目的地への経路を保持している場合は、切断した通信経路の残存ノードであるため、以下のいずれかの処理を行う。

- (1) RRPR メッセージを受信したノードが、保持しているルーティングテーブルに目的地への経路エントリを保持しており、かつ目的地への経路が有効である場合は、RREP メッセージを作成し、経路切断を検知したノードへ送信する。
- (2) RRPR メッセージを受信したノードが、保持しているルーティングテーブルに目的地への経路エントリを保持しているが、目的地への経路が無効である場合は、目的地への新たな経路を作成するために、RRPR メッセージをシナリオによって指定されたノードへ転送する。

4.4 経路維持プロセスにおけるルーティングテーブル更新の実装

AODV-UU では、経路維持プロセスにおけるルーティングテーブルの更新は、RREQ メッセージ受信時と RREP 受信時において行われる。

RREQ メッセージを受信したノードは、保持しているルーティングテーブルを検査し、ルーティングテーブル内に目的地への経路エントリを保持していない場合は、送信元の情報とその RREQ メッセージを送信した隣接ノードのアドレスを送信元への次ホップとしてルーティングテーブルに記録し、送信元方向の経路を確立する。

RREP メッセージを受信したノードは、ルーティングテーブルに目的地への経路エントリが存在するか検査し、目的地への経路エントリが存在しない場合は、目的地の情報と RREP メッセージを送信した隣接ノードのアドレスを目的地方向への次ホップとしてルーティングテーブルに記録し、目的地方向の経路を確立する。またルーティングテーブルに目的地への経路エントリが存在し、かつ経路が無効状態である場合は、経路を有効状態にするなどして経路エントリを更新する。

したがって、AODV-UU では双方向の経路の実現に独立した二つの経路エントリを使用している。これに対し、GVGrid では、RREP メッセージ受信時に、RREP メッセージ内に記録されているノード ID 列から、一つの経路エントリへ自身の前方および後方のノードの情報を記録することにより、一つの経路エントリで双方向の経路を実現しているが、今回の経路維持プロセスの実装では位置情報を与える事ができていないので、グリッド ID 列に関する情報を含めていない。そのため今回の経路維持プロセスの実装では、AODV のルーティングテーブルと同様に一つの経路エントリで片方向の経路を実現している。

GVGrid では、経路維持プロセスにおけるルーティングテーブルの更新は、RRPR メッセージ受信時と RREP メッセージ受信時において行われる。

RRPR メッセージを受信したノードは、保持しているルーティングテーブルを検査し、ルーティングテーブル内に目的地への経路エントリを保持していない場合は、送信元の情報とその RRPR メッセージを送信した隣接ノードのアドレスを送信元への次ホップとしてルー

ティングテーブルに記録し、送信元方向の経路を確立する。

RREP メッセージを受信したノードは、ルーティングテーブルに目的地への経路エントリが存在するか検査し、目的地への経路エントリが存在しない場合は、目的地の情報と RREP メッセージを送信した隣接ノードのアドレスを目的地方向への次ホップとしてルーティングテーブルに記録し、目的地方向の経路を確立する。またルーティングテーブルに目的地への経路エントリが存在し、かつ経路が無効状態である場合は、経路を有効状態にするなどして経路エントリを更新する。



図 4: 構築したテストベッド

5 評価実験

本研究では，GVGrid と AODV の経路修復機能を評価するためのテストベッドを構築した．このとき，マルチホップ通信実現のために，iptables が持つ MAC 層での filter 機能を使用した．

評価対象は，作成したシナリオにおける，GVGrid と AODV の経路修復のスループットである．計測には，パケット分析ツールである Ethereal と Etherape を使用した．

5.1 既存のテストベッドとの位置づけ

MANET のテストベッドとして，MobiEmu[11]，APE[12]，ORBIT[13] などが知られているが，いずれも汎用のハードウェアを用いた実現が困難であるなどの理由から，本研究では汎用の Linux PC のみを用いて容易に実現可能なテストベッドを独自に構築した．

評価用のテストベッドは，IEEE802.11b/g を備えたノート PC 10 台からなる．各ノート PC に搭載している OS は，Vine Linux 3.2 である．

図 4 が構築したテストベッドである．

5.2 MAC 層における filter 機能

MAC 層での filter 機能（iptables の機能）を利用している．これは，指定したノードからのパケットを破棄する機能である．この機能を利用することにより，実験環境において，任意の 2 ノードが直接通信可能な完全グラフトポロジの状態から任意のトポロジを作成することが可能である．

図を用いて，MAC 層での filter 機能を利用したトポロジ作成の様子を説明する．自身の

無線範囲内に複数のノードが存在する場合、無線範囲内に存在するノードは全て通信可能である。図 5(a) は、通信可能な経路を全て記した状態である。この状態では、あるノードがブロードキャストを行うと無線範囲内に存在する全ノードにパケットが届くため、想定したトポロジを作成することが出来ない。そこで iptable の MAC 層における filter 機能を利用する。例えば、図 5(c) のようなトポロジを作成したい場合、ノード A からのパケットをノード B 以外が破棄するようにすればよい。よってノード B 以外のノードの iptable に、ノード A からのパケットを破棄するというルールを記述しておく。これによりノード A がブロードキャストしたパケットをノード B 以外は破棄するため、ノード B のみがそのパケットを受け取る。図 5(b) は、各ノードが図 5(c) のトポロジ作成に不必要な経路をフィルタした様子である。この方法により、無線範囲内に存在する全てのノードが通信可能な図 5(a) の状態から図 5(b) のように不必要な経路をフィルタすることで、作成したい図 5(c) のトポロジを実現できる。

5.3 車両の移動シナリオと経路切断状況

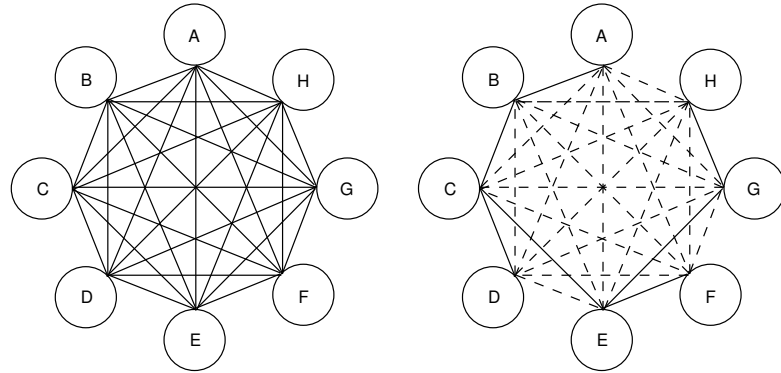
本節では、車両の移動シナリオと GVGrid と AODV における経路切断状況について記述する。

車車間通信ではノードの移動が激しいため、ビルなどの障害物を考慮した場合、構築された通信経路に屈折回数が多い経路は屈折点における切断回数が増える。通信経路の直線部分においては、2つのノードの進行方向が逆である場合は2ノード間距離の増大によりリンクが切断する確率が高く、2つのノードの進行方向が同方向であり、かつ速度が同程度である場合は2ノード間の距離はほぼ一定に保たれるため、切断が発生する確率は低くなる。

GVGrid では、経路の切断が発生した場合、修復を開始するノードは、隣接グリッドに存在するノードを選択して残存経路との接続を行う。直線の道路上におけるノード間距離の増大による切断では、修復前後において経路が大きく変わる可能性は低く、また通信経路が屈折する部分における障害物によるリンク切断では、初期の道路経路上に通信経路を再構築するため、修復前後の経路が大きく変わる可能性は低い。以上のことを考慮して、GVGrid のシナリオを作成した (図 6)。

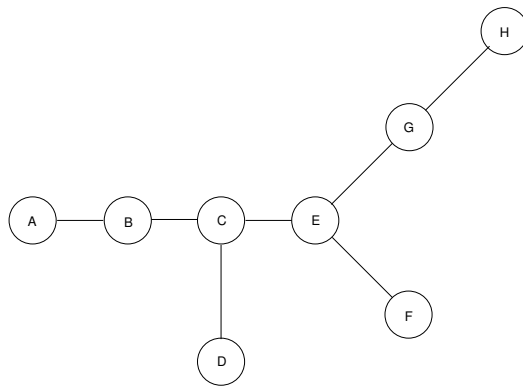
AODV では、経路の切断が発生した場合、RREQ メッセージをブロードキャストによる経路修復を行う。直線の道路上における距離の増大による切断では、修復前後において経路が大きく変わる可能性は低いが、通信経路が屈折する部分における障害物によるリンク切断では、切断前の経路と切断後の経路は大きく変わる可能性がある。また通信経路に屈折する部分が多く含まれる場合、さらに経路の切断が発生し、修復前後で経路が大きく変わる可能性がある。このことを考慮して、AODV のシナリオを作成した (図 7)。

車両の移動シナリオと経路切断状況は、3 分間のシナリオである。図 6(a) は GVGrid の



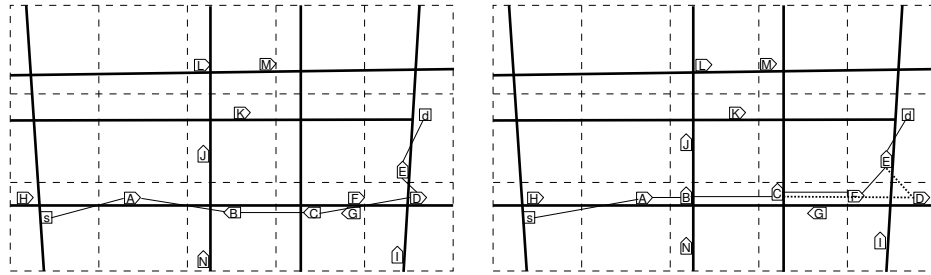
(a) 無線範囲内に複数のノードが存在する場合の通信可能経路

(b) 図 5(c) のトポロジ作成のために、図 5(a) から不必要な経路をフィルタした状態



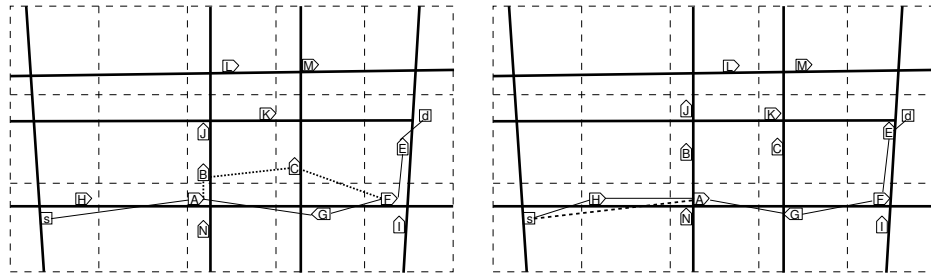
(c) 作成したいトポロジ例

図 5: MAC 層のフィル多機能を利用した例



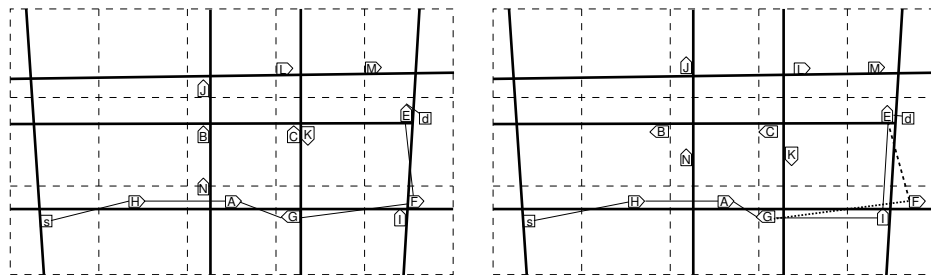
(a)

(b)



(c)

(d)

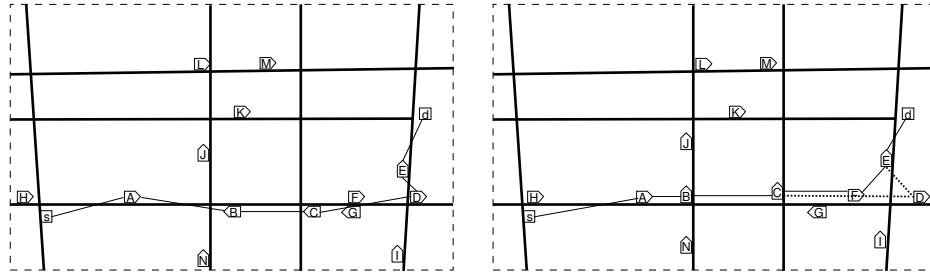


(e)

(f)

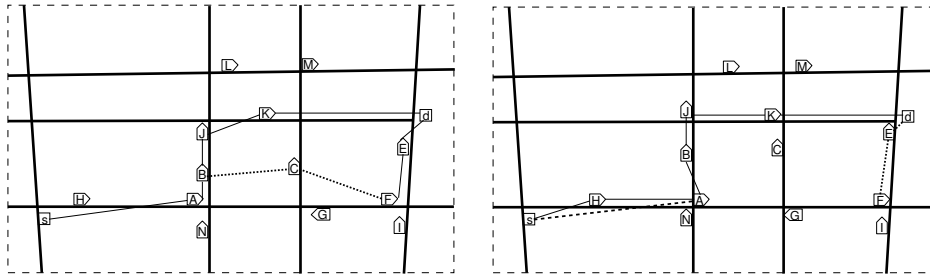
- s : Source
- d : Destination
- : Node
- : Road
- : Link
- : Link break

図 6: 車両の移動シナリオと GVGrid における経路切断状況



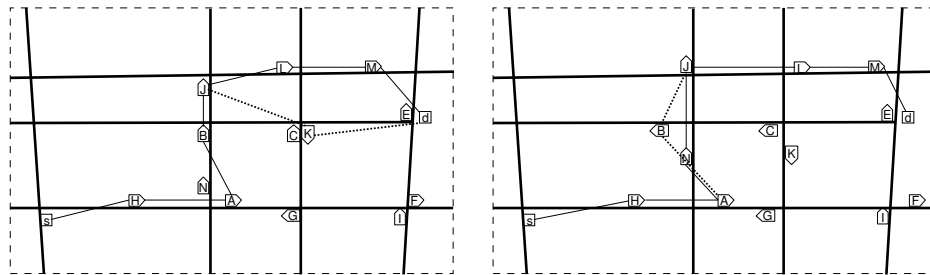
(a)

(b)



(c)

(d)



(e)

(f)

- s : Source
- d : Destination
- : Node
- : Road
- : Link
- : Link break

図 7: 車両の移動シナリオと AODV における経路切断状況

初期経路，図 7(a) は AODV の初期経路とする．以下，GVGrid と AODV の同時間における経路切断と修復による経路の変化について述べる．

図 6(b) と図 7(b) は，ノード C, D 間のリンクは距離が離れたことにより，またノード D, E 間のリンクは障害物により切断した状態を表している．このとき，GVGrid では，リンク切断を検知したノード C が隣接グリッドに存在するノード F を選択して経路修復を開始し，残存経路と接続する (図 6(b))．AODV では，リンク切断を検知したノード C が RREQ メッセージのブロードキャストによる経路修復を開始し，また残存経路を最新の経路と判断して接続する様子である (図 7(b))．

図 6(c) と図 7(c) は，ノード B, C 間のリンクが障害物により，切断した状態を表している．この場合，GVGrid では，ノード B は自身が初期経路から外れたことに気付くため，接続しているノード A, B 間のリンクを強制切断する．そしてノード A は前方グリッドから代替ノード G を選択して経路の修復を開始し，残存経路と接続する (図 6(c))．これに対し，AODV では，ノード B が経路の修復を開始するが，このときノード F, 目的地 d 間に残存経路は最新の経路ではないと判断し，最新の経路を再構築する (図 7(c))．

図 6(d) と図 7(d) は，送信元 s, ノード A 間のリンクが距離が離れたことにより切断した状態を表している．この場合，GVGrid では，送信元 s は前方グリッドから代替ノード H を選択して経路の修復を開始し，残存経路と接続する (図 6(d))．AODV は，リンク切断を検知した送信元 s が RREQ メッセージのブロードキャストによる経路修復を開始し，また残存経路を最新の経路と判断して接続する様子である (図 7(d))．

GVGrid は，経路グリッド列上のノードを選択するので切断は発生していない (図 6(e))．これに対し AODV は，ノード J, K 間が障害物により，ノード K, 目的地 d 間のリンクが距離が離れたことにより切断した状態を表している (図 7(e))．AODV は，リンク切断を検知したノード J が RREQ メッセージのブロードキャストによる経路修復を開始し，最新の経路を再構築する．

図 6(f) は，ノード G, F 間のリンクが距離が離れたことにより，ノード F, E 間リンクが障害物により，切断した状態を表している．この場合，GVGrid では，ノード G は前方グリッドから代替ノード I を選択して経路の修復を開始し，残存経路と接続する．図 7(f) は，ノード A, B 間とノード B, J 間のリンクが障害物により，切断した状態を表している．AODV は，リンク切断を検知したノード A が RREQ メッセージのブロードキャストによる経路修復を開始し，また残存経路を最新の経路と判断して接続する．

次に，構築したテストベッドを用いて，経路の切断とその修復の様子を説明する (図 8)．図 8(a) は，図 6(a) の初期経路が構築された状態である．経路上の各 PC とノード F の PC では，GVGrid プロトコルが動作中である．ここで，ノード D の GVGrid を停止させた場合，ノード C, D 間とノード D, E 間のリンクが切断したことになる．これが図 8(b) の，ノー



(a) 構築されている初期経路 (図 6(a))



(b) ノードの移動により、ノード C,D 間とノード D,E 間のリンクが切断した様子 (図 6(b))



(c) 経路修復後の経路 (図 6(b))



図 8: 構築したテストベッドによる経路修復の例

ドの移動により、ノード C,D 間とノード D,E 間のリンクが切断した状態となる。経路の切断を検知したノード C は、選択ノードとしてノード F を指定して経路修復を開始する。そして経路修復後、図 6(c) の経路が再構築される (図 8(c))。

このように、ノードの両方向の切断はプロトコルを停止させ、片方向の切断は iptable の MAC 層のフィルタ機能を用いて経路をフィルタすることにより、シナリオ上の経路の切断と修復を実現している。

5.4 計測ツール

GVGrid と AODV のスループットの計測は、実験で使用するノードとは別に実験中のパケットをキャプチャするノードが行う (以下パケットキャプチャノード)。

パケットのキャプチャとその解析には、Ethereal を使用した。

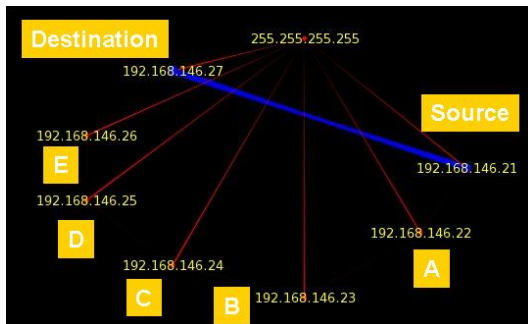
パケットキャプチャノードは、無線 LAN の設定をプロミスキヤスモードにしている。プロミスキヤスモードとは、他の無線を傍受することができるモードである。パケットキャプチャノードは、プロミスキヤスモードを使用し、また Ethereal により実験で使用する全てのノードが送信するパケットをキャプチャすることができる。

本節では、Ethereal と Etherape の機能を紹介する。

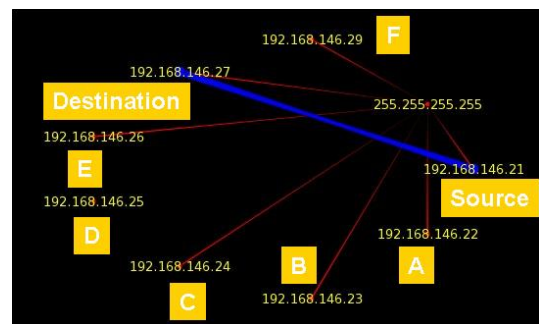
Ethereal は、キャプチャしたパケットを解析する機能を持ち、図 10 のようにノードが送信するパケットのスループットをグラフ化したり、送信されたパケットをプロトコルごとに分類するなどの機能を持つ。本実験における GVGrid と AODV の経路修復の際のスループットの計測には、Ethereal を利用した。

Etherape は、パケットの流れを表すことのできる可視化ツールである。図 9 は、GVGrid におけるシナリオの図 6(a) と図 6(b) におけるスループットを可視化したものである。図 9(a) は、GVGrid におけるシナリオの図 6(a) の状態であり、送信元が s、送信先が d であるパケットが青線の太さの割合で送信されている様子である。なお、制御パケットを表す赤線が存在するノード A、B、C、D、E は制御パケットを送信していることから、中間ノードであることがわかる。図 9(a) からは、送信元 s から送信先 d へ直接パケットが届いているように見える。実際には、送信元 s から送信先 d へは中間ノード A、B、C、D、E を経由しているが、中間ノードはパケットを転送しているだけなので、Etherape 上ではその状況は表されていない。また、Etherape では中間ノードの順番までは表現できない。

次に、図 9(b) は、GVGrid におけるシナリオの図 6(b) の状態である。図 6(b) では、ノード D が経路から外れて切断が発生し、ノード F が代替ノードとして使用される。図 9(b) からは、送信元 s から送信先 d へのパケットは、制御パケットを表す赤線が存在する中間ノード A、B、C、D、F を経由していることがわかる。なお、ノード D は孤立していることから、ノード D がパケットの送信を行っていないことがわかる。



(a) 図 6(a) における送信元 s から送信先 d へのスループット



(b) 図 6(b) における送信元 s から送信先 d へのスループット

線の太さ:スループット

線:経路

赤:制御パケット

青:ICMPパケット

図 9: Etherape のビジュアルツール

5.5 評価結果

ICMP パケットを ping プログラムにより連続送出し、3 分間の実験シナリオを元にした経路の切断と修復における GVGrid と AODV のスループットを計測した。

図 10(b) はシナリオにおける AODV のスループット、図 10(a) は GVGrid のスループットを示している。

経路修復プロセスにおいて、切断が発生してから経路修復が完了するまでの間に送信されたパケットはバッファリングされ、経路修復完了後にバッファリングされていたパケットは送信される。そのため、経路の切断が発生するとデータパケットの送信を停止して経路修復のための制御パケットを送信するためスループットは低下し、経路修復後にはバッファリングしていたデータパケットを送信するため、スループットが大きくなる。

図 10(b) から、AODV では、各経路切断の際にはスループットが著しく低下しており、経路修復後のスループット値は非常に大きくなっていることがわかる。AODV では、経路の修復を開始すると最新の経路を構築しようとするので、部分経路が存在する場合でもその経路が最新でない場合は経路の探索を行う。そのため経路修復にかかる時間が長くなり、経路の切断を検知したノードにおける ICMP パケットのバッファリング数が増えたからだと考えら

表 1: シナリオにおけるパケット数と割合

プロトコル名	制御パケット数	ICMP パケット数
<i>AODV</i>	2503	3450
<i>GVGrid</i>	2201	3453

れる。これに対し図 10(a) より, GVGrid は, 各経路切断の際はスループットは低下しているが, 低下の度合は低く, 経路修復後のスループット値も AODV ほど大きくない。GVGrid は, 経路の修復において部分経路が存在する場合はその経路を再利用するため, 経路修復にかかる時間が短く, 経路の切断を検知したノードにおける ICMP パケットのバッファリング数が少ないためと考えられる。

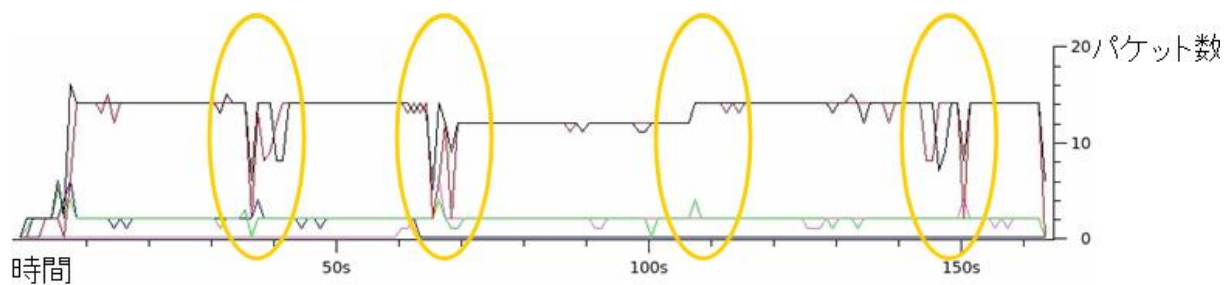
以上のことから, AODV と比較して GVGrid は, 経路修復の間のデータのバッファリングが少なく済むことを示している。これは各ノードのバッファサイズを小さくできる, あるいはビットレートを上げるなどした場合でもデータのバッファ溢れが起こる可能性が減り, 安定したデータ送信を行うことができることを示している。

また一つのノードが送信元と目的地が異なる複数の通信経路を保持している場合, 一つのノードが複数の経路の修復を行わなければならない。このとき異なる目的地への複数のデータをバッファリングすることになる。しかし, ノードのバッファサイズには限界があるので, 一つの経路の修復におけるバッファリングするデータサイズが大きくなると, 複数のデータのバッファリングはバッファ溢れを起こす可能性がある。これは, 一つの経路のデータのバッファサイズを小さくすることにより, 複数の経路修復の間における複数のデータのバッファ溢れの可能性を軽減することができる利点がある。

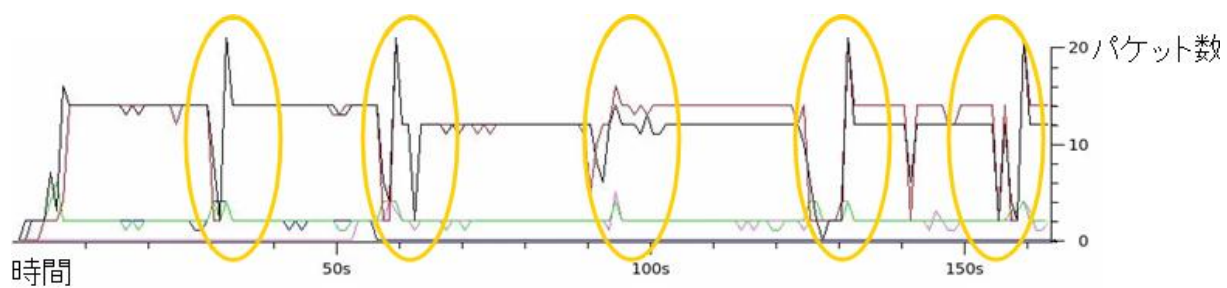
表 1 は, GVGrid と AODV のシナリオにおけるパケット数を示している。GVGrid と AODV は, 同程度の ICMP パケット数を送信しているのに対して, 制御パケット数にはかなりの差が表れている。

AODV は経路の切断が発生した場合, RREQ メッセージをブロードキャスト送信して経路の修復を行い, また部分経路が存在する場合でもその経路が最新でないときは, 制御メッセージを利用して経路の探索を行う。これに対し, GVGrid では経路の切断が発生した場合, RRPR メッセージを特定のノードへ送信して経路の修復を行い, また部分経路が存在する場合はその経路を再利用している手法を採っている。

AODV と比較して GVGrid で制御メッセージが減った要因は, 経路修復において, AODV では, 残存経路が最新でない場合, 修復された部分経路以外に最新の経路探索においても制御パケットが使用されているが, GVGrid は, 修復された部分経路でのみ制御パケットが使



(a) GVGrid



(b) AODV

— Source
 — Destination
 ○ 経路修復部分

図 10: シナリオにおけるスループットのグラフ

用されているからである.

6 あとがき

本研究では、所属する研究グループにより提案されている車車間マルチホップ通信における位置情報ルーティングプロトコル GVGrid の主要機能である経路修復機能の実装を行い、アドホックルーティングプロトコル AODV との比較評価を行った。GVGrid の経路修復機能の実装は、Uppsala University で開発されている AODV プロトコルの実装コード AODV-UU をもとに行った。また、無線 LAN を備えたノート PC10 台程度からなる小規模テストベッドを構築し、作成したシナリオに基づいて、GVGrid と AODV の経路の修復によるスループットの評価を行った。その結果、AODV と比較し、GVGrid は ICMP においてより高いスループットを達成し、また制御パケット数が少ないことを確認した。

今後の課題としては、GVGrid の経路構築プロセスの実装、テストベッドの機能強化が挙げられる。テストベッドの機能強化に関しては、トラフィックシミュレータから得られる車両位置情報の利用、ストリーミング実験 (リアルタイム配信) を通した動画品質、TCP 通信遅延などの評価、ビルなどの障害物の影響による電波が不安定な状況の再現などが挙げられる。

謝辞

本研究を行うにあたり，常に適切な御助言を頂きました大阪大学大学院情報科学研究科コンピュータサイエンス専攻 楠本 真二 教授に深く感謝申し上げます。

本研究において，常に適切な御助言を頂きました 岡野 浩三 助教授に深く感謝申し上げます。

本研究の全過程において，終始熱心な御指導および適切な御助言を頂きました 同 山口 弘純 助手に深く感謝申し上げます。

本研究の全過程において，終始御協力を頂きました 大阪大学大学院情報科学研究科コンピュータサイエンス専攻 博士後期課程の 孫 為華 氏，同 修士課程の 行政 幸治 氏に深く感謝申し上げます。

最後に，様々な御助言，御協力を頂いた 大阪大学大学院情報科学研究科コンピュータサイエンス専攻 楠本研究室の皆様に深く感謝申し上げます。

参考文献

- [1] Y. B. Ko and N. H. Vaidya, "Location-aided routing (LAR) in mobile ad hoc networks," Proceedings of ACM Mobicom, pp.66-75, 1998
- [2] B. Karp and H. T. Kung, "GPSR:Greedy perimeter stateless routing for wireless networks," Proceedings of ACM Mobicom, pp.243-254, 2000
- [3] G. Korkmaz, E. Ekici, F. Ozguner, and U. Ozguner, "Urban Multi-Hop Broadcast Protocol for Inter-Vehicle Communication Systems," Proceedings of the 1st ACM International Workshop on Vehicular Ad Hoc Networks (VANET2004), pp.76-85, 2004
- [4] M. Saito, J. Tsukamoto, T. Umedu, and T. Higashino, "Evaluation of inter-vehicle ad-hoc networks," Proceeding of IEEE 19th Int. Conf. on Advanced Information networking and Applications 2005 (AINA2005), pp.78-83, 2005
- [5] J. L. Sobrinho and A. S. Krishnakumar, "Distributed multiple access procedures to provide voice communications over IEEE 802.11 wireless networks," Proceedings of IEEE GLOBECOM, pp.1689-1694, 1996
- [6] 孫為華, 山口弘純, 楠本真二, "安定性の高い経路を構築する車車間ルーティングプロトコル," 情報処理学会論文誌, Vol.47, No.7, pp.2141-2150, 2006
- [7] C. K. Toh 著, "アドホックモバイルワイヤレスネットワーク," 共立出版, 2003
- [8] S. Corson and J. Macker, "Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations, " RFC 2501, 1999
- [9] C. E. Perkins and E. M. Belding-Royer and S. Das, "Ad Hoc On Demand Distance Vector (AODV) Routing, " RFC 3561, 2003
- [10] Uppsala University, " AODV-UU," <http://core.it.uu.se/core/index.php/AODV-UU>
- [11] Yongguang Zhang and Wei Li, "An Integrated Environment for Testing Mobile Ad-Hoc Networks, " Proceedings of the Third ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'02), Lausanne, Switzerland, June 2002.
- [12] Erik Nordstrom, Per Gunningberg, Henrik Lundgren, "A Testbed and Methodology for Experimental Evaluation of Wireless Mobile Ad hoc Networks, " Proc. of IEEE Tridentcom 2005

- [13] Maximilian Ott, Ivan Seskar, Robert Siracusa, Manpreet Singh, “ORBIT, ” Testbed Software Architecture: Supporting Experiments
- [14] R. Morris, J. Jannotti, F. Kaashoek, J. Li, and D. De-couto, “CarNet:A scalable ad hoc wireless network system,” Proceeding of SIGOPS European Workshop, 2000
- [15] M.-T. Sun and W.-C. Feng and T.-H. Lai and K. Yamada and H. Okada and K. Fujimura, “GPS-Based Message Broadcasting for Inter-Vehicle Communication, ” Proc. of Int. Conf. on Parallel Processing, pp.279-287, 2000
- [16] Tomoyuki Yashiro and Tempei Kondo and Hirotaka Yagome and Masafumi Higuchi and Yuyuka Matsushita, “A Network Based on Inter-Vehicle Communication, ” Proc. of IEEE Int. Conf. on Intelligent Vehicles, pp.234-250, 1993