2008－MBL－45（25）
2008 / 5 / 23

Hop by Hop ルーティングプロトコルにおける ノードディスジョイントな経路構築法の検討，評価

森　拓海†，高橋　修‡
†公立はこだて未来大学大学院システム情報科学研究科
‡公立はこだて未来大学

概要　MANET(Mobile Ad-hoc Network)ではデータ到達率の向上や遅延短縮といった目的で複数の経路を用いたルーティングプロトコルが研究されている．複数経路のセキュリティ面の利用としては，IDS などの悪意のあるノード/経路を検出，回避する機構により複数経路を切り替えることで検出したノード/経路を回避する手法がある．経路を適切に切り替え，効率的に検出したノード/経路を回避するために，保持する複数の経路が互いにノードディスジョイントであることが有効である．しかし AODV などの Hop By Hop 方式のルーティングプロトコルの場合，RAM 資源の節約のため経路情報として各ノードは自身の前後のノード情報のみを保持するため，ディスジョイントな経路の保持が困難である．本論文では，ノードディスジョイントな経路の特性を利用し，Hop By Hop 方式のルーティングプロトコルでの効率的な経路構築方法を提案するとともに，実装評価することによって提案手法の有効性を明確にする．

Node-Disjoint Multipath Construction Method on Hop by Hop Routing Protocols and its Evaluation

Takumi Mori†，Osamu Takahashi‡
† The graduate school of Future University-Hakodate, Systems Information Science
‡ Future University-Hakodate

Abstract　Multipath routing protocols aim at improvement in the data arrival rate and shortening the delay time is studied in MANET(Mobile Ad-hoc Network). For the use of the security side of multipath routing, there is what change over route to avoid malicious nodes/paths using method which detect malicious nodes/paths such as IDS. Node disjoint multipath is useful element of route change to avoid the detected node/path effectively. However, in the case of the routing protocol of Hop By Hop methods such as AODV, it is difficult to construct the node-disjoint paths because of maintain routing information is only the node information of before and after of own for the saving of RAM resources. Using the characteristic of node disjoint paths in this article, we evaluated the effective node disjoint path construction method by the routing protocol of the Hop By Hop method. As a result of the evaluation, our proposal method was effective.

1. はじめに

小型の無線通信端末機器のみで構築されるネットワークである MANET(Mobile Ad-hoc Network)では通信経路を形成するためのルーティングプロトコルが IETF（the Internet Engineering Task Force）[1]で数多く研究されている．その中にデータ到達率の向上や遅延短縮といった目的で複数の経路を用いたルーティングプロトコルがある．複数経路はその中の１本を主経路とし，そのほかは代替経路として保持することで，経路切断時に瞬時に通信を再開できる利点を持つ．

セキュリティ面の利用としては，IDS（侵入検知システム）に深い関連がある．IDS により悪意のあるノード/経路を検出したのち，検出したノードを迂回する経路を代替経路より選択，切り替えることで遅延を最小限にとどめることができる．しかし，保持する複数の経路に制約を設けなければ，切り替えた経路に検出したノードが含まれる可能性がある．検出したノードを含まない経路を選択する方法は2つある．1つは，代替経路のノード情報に検出したノードが含まれないかどうか検査する方法である．この方法は単純であるが，保持する経路数によっては多大なオーバーヘッドを発生させる．2 つ目は，保持する経路が互いにノードディスジョイントであることである．ノードディスジョイントであることとは，経路同士が互いに共通の中間ノードを含まないことである．ノードディスジョイントな経路間で経路の切り替えを行えば，主経路上に含まれるノード（検出ノード）が代替経路に含まれないことが保証される．

本論文では，ノードディスジョイントな経路を Hop By Hop 方式のルーティングプロトコルである AODV 上で構築するための手法を提案する．提案手法では送信元ノードの隣接ノードが代理ノードとして送信元ノードに代わり経路構築を行う．シミュレーション評価より経路検索回数が減少することから，AODV と比較しネットワークへの負荷が減少することを確認した．しかし，データパケットの転送遅延が増大する問題も明らかとなった．

以下 2 章ではノードディスジョイントな経路の解説と特徴と問題点を述べ，3 章で実現すべき課題を明確化し代理ノードを用いた提案方式を解説する．4 章にてシミュレーション評価とその結果を考察し，得られた結果から結論を述べ，5 章にてまとめとする．

2. 関連研究 ノードディスジョイント

複数経路を構築するアドホックルーティングプロトコルではディスジョイントと呼ばれる経路特性がある．おもに効率的な通信経路分散を行うことを目的として用いられる．

送信元ノードと宛先ノード間で複数経路がディスジョイントである場合，ノードディスジョイントとリンクディスジョイントの 2 種類存在する．ノードディスジョイントとは経路同士が中間ノード（送信元，宛先ノードを除いたノード）を共有しないことである．一方リンクディスジョイントはある経路上のリンク間がノードディスジョイントであることを指す．リンクディスジョイントでは経路分岐が発生し，中間ノードを共有することがある．セキュリティ方式としてディスジョイントな経路を用いる場合は，リンクディスジョイント

な経路では経路分岐制御が必要な上，中間ノードが重複するため不適切である．したがって本論文ではノードディスジョイントについて着目する．

2.1. 経路特性と経路構築法

ノードディスジョイントな経路の特徴としては，以下の 4 つがある．

(1) 中間ノードを共有しない
(2) 効率的な通信経路分散
(3) 経路縮小の発生
(4) Hop By Hop 方式での経路発見が困難

(1)中間ノードを共有しないとは，複数経路を保持する上で互いの経路に同じ中間ノード（送信元ノードと宛先ノードを除いたノード）を含まないことである．この特徴を持つ経路は送信元ノード以外で経路が分岐しない．(2)効率的な経路分散とは，特定のノードがボトルネックになることがない経路を構築できることである．一般的に，最短距離解決などの手法で形成された複数経路は宛先ノードの 1 つ前の中間ノードを共有する場合が多い．しかしノードディスジョイントな経路同士は中間ノードを共有しないため，確実に送信元ノードから宛先ノードまで中間ノードを共有しない経路を構築することができる．(3)経路縮小の発生とは，最短経路解決手法などで形成される複数経路数よりも確保できる経路数が減少することである．ノードディスジョイントな経路では，特に送信元ノードと宛先ノードの隣接ノード数以上の経路を形成することができない．したがって，それらの隣接ノードが極端に少ない場合は確保できる経路が減少する．(4)Hop By Hop 方式での経路発見が困難とは，Hop By Hop 方式ではあるノードが保持する情報が宛先ノードと次ホップノードのみであるため経路全体を把握することができず，経路同士が中間ノードを共有しているかの判断が難しいことである．

(1), (2)の特徴に関しては，セキュリティに応用することが可能である．IDS（侵入検知システム）などで悪意のあるノード/経路を検出するセキュリティ機構の場合，検出したノード/経路を回避するように経路を切り替える必要がある．切り替え経路候補が最短経路解決などの手法により確保されている場合は中間ノードを共有することがあり，再度検出したノード/経路を含む経路に切り替えてしまう可能性がある．切り替え経路候補にノードディスジョイントな経路を保持していれば，検出したノード/経路を通らないことが保証される．

ノードディスジョイントな経路形成法として NAF（Node Association Factor）値を用いた経路形成法を紹介する．Node Disjoint である条件を定式化した NAF 値の詳細は上野裕介他「ルートの独立性を考慮したマルチパスルーチングプロトコルの提案とその評価」[2]を参照されたい．

NAF 値は複数経路間で，中間ノードの共有数を数値化したものである．全ノード集合を N とし，注目するノードをN_kとする．ノードN_kの順序集合であるルートR_iは以下のように表わされる．

$$R_i = (N_k, \leftrightarrow), \{N_k | N_k \in \mathbb{N}, k \in \mathbb{Z}\}$$

↔は順序集合の各要素間にリンクが存在することを示す．一回のルート構築プロセスでルート候補として保持されるルート数を n とし，n から選ばれるルート数 m とする．すると m,n 間には$n \geq m$が成り立ち，保持されるルート候補R_iの集合 R は次の式で表すことができる

$$R = \{R_1, \cdots R_i, \cdots R_n\}(1 \leq i \leq n)$$

ここで，2 つのルート(R_i, R_j)における NAF 値 は次のように表せる．

$$NAF(R_i, R_j) = |R_i \cap R_j| - 2$$

送信元ノードと宛先ノードは必ず共有されるため 2 を引くことで，共有する中間ノード数となる．

この NAF 値を利用したノードディスジョイントな経路を構築するためには，すべての経路に対する NAF 値が 0 である必要がある．ノードディスジョイントな経路集合R_{ND}は次のように表すことができる．

$$R_{ND} = \{R_1, \cdots R_i, \cdots R_m\}\left(\sum_{\substack{1 \leq i \leq m \\ 1 \leq j \leq m}} R(i,j) = 0\right)$$

上記の計算は全経路上のノード情報が必要となるため，ソースルートを利用する．NAF 値計算およびノードディスジョイント判定は送信元ノードまたは宛先ノードが行う．

2.2. Hop By Hop 方式における利点，欠点

アドホックルーティングプロトコルには，大きく分け 2 種類存在する．1 つは OLSR[3]に代表される，経路を通信要求以前に形成するプロアクティブ型ルーティングプロトコルである．通信要求時に直ちに通信が開始できる半面，通信を行わない状況でも経路維持を行わなければならないため，移動性が高く，通信トポロジ変化が頻発するアドホックネットワークには適さない．もう 1 つが AODV[4]，DSR[5]に代表されるリアクティブ型ルーティングプロトコルである．経路形成を通信要求時に行うことで，プロアクティブ型のように経路維持の必要がない．しかし経路形成が完了するまでは通信できないため，通信開始までに遅延が発生する．一般的に通信トポロジ変化に強く，アドホックネットワークに適している．

複数経路を管理する場合，DSR のような Source Routing 方式では各中間ノードがソースルートを保持するため，経路管理のために多くの情報が利用可能である．特に経路分岐が発生する場合，経路識別にソースルートを用いることが一般的である[6][7]．しかし，AODV のような Hop By Hop 方式では自ノードの前後のノード情報のみを保持するため，経路分岐を制御することが困難となる．

ノードディスジョイントな経路はその特性上，送信元と宛先ノード以外で経路分岐することがない．したがって，中間ノードでは自ノードの前後ノード情報のみの保持で経路維持が可能である．ノードディスジョイントな経路を Source Routing と Hop By Hop 方式のルーティングプロトコル上に展開した場合の両方式の記憶容量の違いを述べる．

2.3. 経路維持に必要な記憶容量

Source Routing (SR)方式と Hop by Hop（HH）方式のルーティングプロトコル上でノードディスジョイントな経路を保持するために必要な記憶容量を比較する．N を全経路上のノードとし N={S,D,X} S:送信元, D:宛先, X:中間ノードと定義する．MaxPath は保持される最大経路本数とし，Np(node)をあるノード(node)の経路保持数とする．Hops(S,D)はノード S-D 間のホップ数とし，容量は Hops(S,D)+2 と定義する．最後に Cn をコネクション数とした．

以下の計算で想定するノードディスジョイントな経路は送信元ノード S から宛先ノード D までの経路同士のホップ数は変化しないものとした（図 1）．

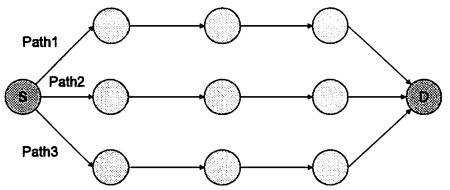

図 1 想定するノードディスジョイントな経路

次にノードディスジョイントな経路を保持するのに必要な順・逆経路の記憶容量を求める．はじめに，Source Routing (SR)について，必要な記憶容量は次のようになる．

$$SR{:}MemND = \{Np(S) + Np(D) + Np(X)\cdot(N\text{-}2)\}\cdot Cn$$

$$=\{MaxPath\cdot 2 + (Hops(X,S)+Hops(X,D)+4)\cdot(N\text{-}2)\}\cdot Cn$$

次に Hop By Hop(HH)方式で同様に求める．

$$HH{:}MemND = \{Np(S) + Np(D) + Np(X)\cdot(N\text{-}2)\}\cdot Cn$$

$$=\{MaxPath\cdot 2 + (2\cdot(N\text{-}2))\}\cdot Cn$$

次に SR と HH で S-D 間のホップ数増加に対する記憶容量の増加量を比較する．記憶容量の増加比率を比較する式を以下のように定義する．

$$\frac{SR{:}\,Mem_{ND}}{HH{:}\,Mem_{ND}}$$

上記式を計算すると，

$$\frac{SR{:}\,Mem_{ND}}{HH{:}\,Mem_{ND}} = \frac{\{MaxPath\cdot 2 + (Hops(X,S)+Hops(X,D)+4)\cdot(N-2)\}\cdot Cn}{\{MaxPath\cdot 2 + (2\cdot(N-2))\}\cdot Cn} = \frac{(Hops(X,S)+Hops(X,D)+4)}{2} = \frac{Hops(S,D)+3}{2}$$

上記式を Hops(S,D)を変数としてプロットすると図 2のようになる．標準的な MANET 上の S-D 間のホップ数である 0～16hops を考慮すると，SR 方式は HH 方式の約 1.5 倍～9.5 倍の記憶容量が必要とされることがわかる．HH 方式では RAM 資源を節約しつつ効率的な経路分散を行うことができる．HH 方式上のノードディスジョイントな経路形成は困難とされるが，次章にて HH 方式上での経路形成法を提案する．

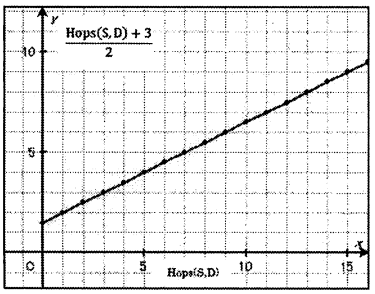

図 2 SR と HH の記憶容量の増加率

3. 提案方式

本章では，Hop by Hop 方式のアドホックルーティングプロトコルでノードディスジョイントな経路形成を行うための課題を明確化する．Hop by Hop 方式のアドホックルーティングプロトコルとしては RFC3561 で標準化されている AODV を対象とする．以下ではノードディスジョイントな経路形成を AODV 上で実現するための課題について議論し，各課題に対し本提案方式で用いた解決法をまとめる．

3.1. 解決すべき課題

AODV 上でノードディスジョイントな経路を作成するための課題としては，以下の 2 点がある．

- RREQ 受信時の経路把握
- RREQ の受信制限

はじめに RREQ 受信時の経路把握について述べる．ノードディスジョイントな経路を発見するには，現在検索している経路の状態を知る必要がある．AODV の Route Request(RREQ)には，前ホップノードの情報しか保持されていない．したがって，RREQ を受信したノードがその経路がノードディスジョイントであることを検査することはできない．そこで RREQ にソースルート記載することにより，経路の状態を RREQ 受信ノードにより把握可能にする．ただし，ソースルートを記録・検査することは RAM および計算資源の消費につながるためソースルートの記録・検査は宛先ノードのみが行うものとする．

次に，RREQ の受信制限について AODV の RREQ フラッディング特性に着目する．AODV の RREQ には 1 検索あたりに一意の ID が割り振られる．AODV では RREQ を受信したノードは以前に同一の ID の RREQ を受信していた場合は RREQ を破棄する．これはループ回避やコリジョン回避のための制約である．この制約により図 3 のように最短経路上のノードが送信する RREQ が優先して処理され，ノードディスジョイントな経路の RREQ は破棄されてしまうため，1 回のフラッディングによりノードディスジョイントな経路を発見することは非常に困難である．仮にこの制約を破棄または緩和した場合，宛先ノードに大量の RREQ が受信される，ネットワーク上に大量の RREQ が浮遊するなど深刻な問題が発生する．

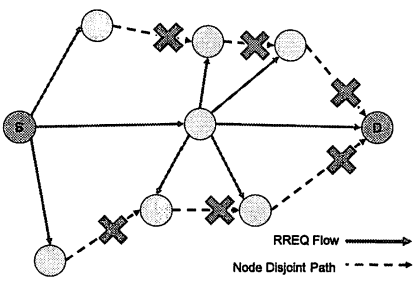

図 3 最短経路のフラッディングによる問題

そこで，複数回経路検索を行う方法を考える．求めるノードディスジョイントの経路数回だけ経路検索を行い，既に Route Reply(RREP)が返信された経路のみ RREQ フラッディング対象から外すことにより経路形成を行う．しかしこの方式では，前回の経路検索が終了するまで次の経路検索を行うことはできないため，求める経路数を確保するまでに多くの時間を要する．また求める経路数分の RREQ フラッディン

グだけのオーバーヘッドが発生するため，ネットワークへの負荷が非常に大きい．そこで 1 回の経路検索で ID による RREQ 受信制約を緩和しつつ，オーバーヘッドを抑えるため，代理ノードによる経路検索手法を提案する．本方式では，送信元ノードの隣接ノードが代理で経路検索を行う形式を取ることにより，既存のループ・コリジョン回避機構を用いることができる．また，代理ノードを除く中間ノードに送信元からのホップ数を記録させることにより，異なる代理ノードの最短経路上に代理 RREQ がフラッディングされることを抑制する．

次節では以上の機構を備えたノードディスジョイントな経路形成手順を提案する．

3.2. 提案方式 AODV-DMR

本提案方式における経路形成手順は次の 4 つからなる．

(1) Route Request
(2) Alternative Broadcast(Alt Request)
(3) Route Reply
(4) Alternative Reply

送信元ノードからの Route Request に対し，送信元ノードの隣接ノード（以後 代理ノード）が代理で経路検索（Alt Request）を行う．経路形成は代理ノードから宛先ノードまで行われ，代理ノードへ Route Reply が返信される．形成された経路は送信元ノードへ Alt Reply として報告される．

本提案方式 AODV-DMR(AODV-Disjoint Multipath Routing)は Hop By Hop 方式のルーティングプロトコルであり RFC3561 で標準化された AODV[4]をベースとしている．新たに Alt RREP, Alt Reply の 2 つのメッセージを追加した．2 つのメッセージには表 1 のフィールドが追加されている．

表 1 Alt RREQ, Alt RREP の追加フィールド

8bits	Alt Typeフラグ(Alt or Not)
32bits	代理元アドレス
32bits	オリジナルシーケンスNo.
32bits ×hops	ソースルート

3.3. Route Request

送信元ノード S から宛先ノード D への経路形成を考える．はじめに送信元ノードが Route Request(RREQ)をブロードキャストする．受信したノードは自身が送信元ノードの隣接であると，代理ノードどなる．図 4 のような場合はノード A,B,C が代理ノードとなる．

S からの RREQ はホップせず，隣接ノードに受信された時点で破棄される．徐々に検索範囲を拡大していく Expanding Ring Search 機構は代理ノードが行う．また，宛先ノードが送信元ノードと隣接している場合は，直接 Alt RREP を返信する．

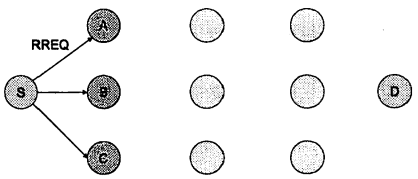

図 4 Route Request の送信

3.4. Alternative Broadcast (Alt Request)

代理ノード A,B,C は代理経路検索を開始する．図 5 のように代理検索要求メッセージである Alt-RREQ を周囲にブロードキャストする．

このとき，代理でブロードキャストされた Alt RREQ には，送信元 S(代理元)の ID とオリジナルのシーケンス番号が保持される．代理ノードによる経路形成におけるループ回避は AODV のものを利用する（図 6）．また，S からのホップ数が中間ノードに記録され，2 度目以降の同一の代理元の代理ノードからの Alt RREQ は，代理元からのホップ数が短ければフォワーディングされ，それ以外は破棄される．

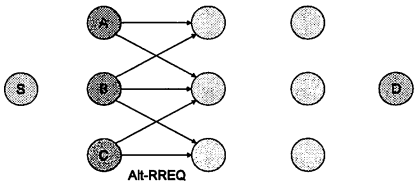

図 5 代理検索-Alt RREQ の送信

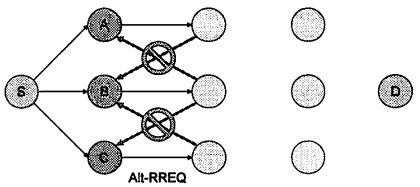

図 6 代理検索-ループ回避

3.5. Route Reply

宛先ノード D に Alt RREQ が受信されると，D は RREP を代理ノードへ返信する（図 7）．D には代理ノードまでのソースルートが記録される．初めて Alt RREQ を受信した場合は即座に RREP を返信する．2 度目以降の受信時には NAF 値の計算を行い，NAF 値が 0 の場合のみ RREP を代理ノードへ返信する．また，既に返信した経路でもその経路を含む短い経路が発見された場合には RREP を返信する．この場合を除き，NAF 値が 0 の経路は中間ノードを共有しないため，RREP が同一のノードに 2 度受信されることはない．

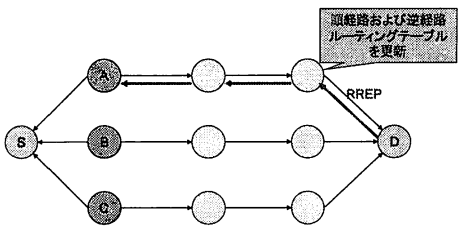

図 7 代理ノードへの Route Reply の返信

このとき，各中間ノードのルーティングテーブルを RREP に従い更新する．これは，より短い経路が発見された場合に経路矛盾が発生しないようにするためである．また更新されるルーティングテーブルの情報は各中間ノードの前後のノード情報であり，ソースルートは記録しない．

3.6. Alternative Reply

図 8 のように代理ノード A に RREP が返信されると，代理ノード A は送信元 S に Alt RREP を返信する．Alt RREP を受信した S はルーティングテーブルに宛先ノード D へのネ

クストホップAという情報のみを記録する．

同様の手順で代理ノードB,Cで経路が形成されるとSにはDへの経路としてネクストホップA,B,Cの3ノードが記録される．

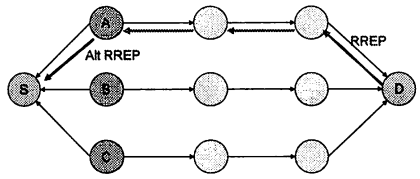

図 8　送信元への経路報告-Alt RREP の返信

以上の手順で，(宛先ノードがNAF値計算のために保持するものを除く）ソースルートを記録することなくノードディスジョイントな経路が構築される．

4. シミュレーション評価

本提案プロトコルとRFC3561に従うAODVとの比較評価を計算機シミュレーションにより行う．シミュレータにNS-2(2.28)[8]，提案方式の実装にはNS-2に付属するAODVを改良し使用した．シミュレーション環境の詳細を表 2に示す．複数経路が十分に形成できるように，トポロジ範囲1000m×1000mに100ノードが存在する環境を想定する．パケットはシミュレーション時間100秒間に990パケットが転送される．ノードのモビリティとして，移動速度(最大)を0～60km/hの間で6パターンを想定する．また，提案方式で形成する経路の最大数を10本とする．さらに通信ペアの初期位置による偏りを減らすため，シミュレーションは任意の通信ペア10組の平均を取る．

表 2　シミュレーション環境

トポロジ範囲	1000m × 1000m
電波到達範囲	半径約200m
ノード数	100ノード
シミュレート時間	100秒
通信	2ノード間の単方向通信。 プロトコル：UDP パケット生成間隔：0.1秒 データパケットサイズ：500byte ｼﾐｭﾚｰｼｮﾝ開始時刻から1秒後に通信開始
通信レート	2Mbbs
ノードの動き	ランダムウェイポイント
ノードの移動速度	0km/h,2km/h,5km/h,7km/h, 20km/h, 60km/h (最大時速) 平均ポーズタイム0秒
最大マルチパス本数 （提案方式のみ）	10本

本提案方式における複数経路の利用法とは，保持する経路の中で最もホップ数の少ない経路を用いて通信する「Shortest Route」，経路の中からランダムで選択する「Random Choose」の2種類を用いる．また，経路のKeep Aliveはオーバーヘッドの増大を考慮し行わない．送信元ノードで経路切断が起こった場合に経路の切り替えを行い，バッファされたパケットを他の経路を用いて送信するPacket Salvage機構を使用する．

4.1. 結果と考察

本節では本提案方式のAODVとの比較シミュレーション結果を示す．図 9にデータパケットの到達率のノード移動速度による変化を示した．次に図 10では，経路（再）形成時に発生するRoute Discoveryイベントの発生回数を示した．最後に図 12では，送信元ノードからデータパケットが送信され宛先ノードに到達するまでの時間（遅延）を示した．

図 8のパケット到達率について着目すると，ノード移動速度が上がるにつれAODVと比較して到達率が徐々に低下する．しかし最大でも遅延が10%未満に抑えられている．しかし 図 9のRoute Discoveryイベント発生回数を見ると，提案方式ではAODVに比べノード移動が高速になるほどイベント発生回数が抑えられる．

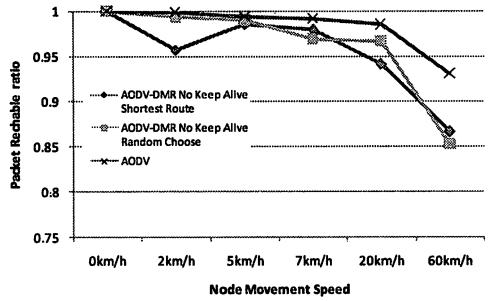

図 9　移動速度別パケット到達率

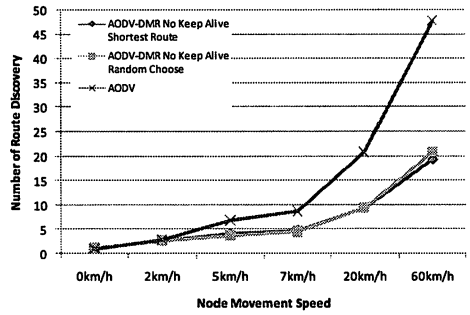

図 10　Route Discoveryイベント発生回数

次に提案方式がAODVに比べ，Route Discoveryイベント一回で形成された経路を用いてどれだけのデータパケットを転送可能であるかを示す．図 11は，提案方式とAODVのRoute Discoveryイベント一回あたりで転送されたデータパケット数の比を表す．提案方式はノード移動速度60km/hで最大AODVの約2.3倍のデータパケットを転送できる．ノード移動速度が速い場合，通信トポロジ変化が頻発するためRoute Discoveryイベントが多発する．Route Discoveryイベントは RREQをブロードキャストするため，ネットワーク全体の帯域負荷が高く，データパケットとのコリジョンを引き起こす．したがって，ノード移動速度が高速なほど場合に転送効率を上がることは同時にデータパケットコリジョンを減少させ，ネットワーク全体への負荷を軽減させるといえる．

実験結果から提案方式には遅延増大の問題が明らかとなった．図 11のEnd to Endデータパケット到達遅延に着目すると，AODVでは高モビリティ環境においても，0.15秒以内のパケット転送を可能にしているのに対し，提案方式では0.3～0.6秒程度の遅延がモビリティに関係なく発生する．これは，保持する経路がノードディスジョイントであるため，最短経路以外の経路のホップ数が増加することに起因する．ま

た，複数形成から時間が経っていてもホップ数の多い経路は使用可能であるが，通信トポロジ変化によりリンク間距離が長くなる．さらに Packet Salvage 機構も遅延増大の原因と考えられる．Packet Salvage では経路が切断されたことを RERR により通知された後，別経路でパケットを再送するため遅延が増大する．

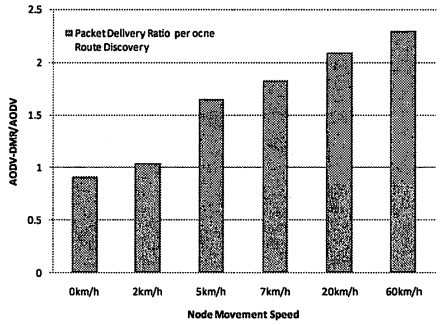

図 11 Route Discovery 一回あたりのパケット転送比

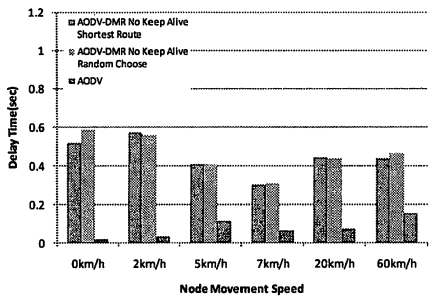

図 12 End to End パケット到達遅延

4.2. 結論

シミュレーション結果よりデータパケットの転送に必要な経路の再検索回数が少なくなることが示された．この傾向はノード移動速度が速いほど顕著である．これはパケット転送に対する制御メッセージのオーバーヘッドを低減させるといえる．しかし，高モビリティ環境（時速 20km 以上）では提案方式のスループットは AODV に比べ若干低下する．これは，提案方式が複数経路を保持・使用するため，高モビリティ環境下では無効経路を使用するためと考えられる．また，送信元ノードと宛先ノード間のデータパケット転送遅延が AODV と比較して非常に大きい．パケット転送遅延の増大は，パケットの到着順序の整合性が取り難くなることや，リアルタイム型のアプリケーションデータを扱う場合に問題となる．セキュリティ面でも，データの同期などの点から好ましくない．これらの理由から，遅延を低減させることが今後の課題である．

5. 終わりに

ノードディスジョイントな経路をデータ通信のみに利用する場合，1 回の経路形成に対するパケット転送効率に優れていることを明らかにした．今後は本研究をセキュリティに応用し，悪意のあるノードまたは経路を検出する防御機構で検出情報を安全な経路でフィードバックする手法（図 13）を提案する．

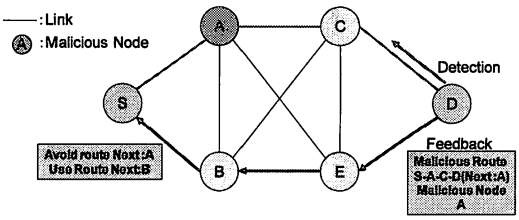

図 13 ノードディスジョイントな経路を用いた検出情報のフィードバック手法

この手法では，送信元ー宛先ノード間のノードディスジョイントな経路を用いて悪意のある経路（ノード）の検出情報をフィードバックする．ノードディスジョイントな経路を用いるため，経路/ノードの検出は宛先ノードが行う．Hop By Hop 方式のルーティングプロトコル上でノードディスジョイントな経路を低記憶容量で形成し，経路をデータ通信のみならずセキュリティ機構にも応用することにより効率的かつ低資源コストで実現する手法を検討する予定である．

参考文献

[1]. Internet Engineering Task Force (IETF), www.ietf.org/

[2]. 上野裕介，撫中達司，小野良司，渡辺尚 「ルートの独立性を考慮したマルチパスルーチングプロトコルの提案とその評価」，情報処理学会論文誌 Vol45 No12, Dec. 2004

[3]. T. Clausen and P. Jacquet. "Optimized Link State Routing Protocol(OLSR). " http://www.ietf.org/rfc/rfc3626.txt, 2003, RFC3626.

[4]. C. E. Perkins, E. Belding-Royer, and S. R. Das "Ad hoc On-Demand Distance Vector (AODV) Routing.", http://www.ietf.org/rfc/rfc3561.txt, July 2003. RFC 3561.

[5]. D. B. Johnson, D. A. Maltz, Y. Hu, and J. G. Jetcheva. "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR)" , draft-ietf-manet-dsr-09.txt, April 2003. IETF Internet Draft.

[6]. 谷山健太，櫻井祐介，甲藤二郎 「アドホックネットワークにおけるディスジョイント・マルチパスルーティング」 , 2004

[7]. 森井健之，谷山健太，甲藤二郎 「アドホックネットワークにおけるオンデマンド型マルチパスルーティングプロトコル実装」,2005

[8]. NS-2 : www.isi.edu/nsnam/ns/

[9]. Ash Mohammad Abbas, "Disjoint Multipath Routing for Mobile Ad hoc Networks" , 2007.April

[10]. Matthew J. Miller, Jungmin So "Improving Fault Tolerance in AODV" , Fall 2002

[11]. Mahesh K. Marina, Samir R. Das "On-demand Multipath Distance Vector Routing in Ad Hoc Networks", ICNP 2001

[12]. Y.Nishibayashi, Y.Sakurai, J.Katto "Multipath Extension of AODV with Enhanced Route Establishment and Proactive Route Management", 2003

[13]. 森拓海，森郁海，高橋修 "アドホックネットワークにおける防御法の分類と耐攻撃性アドホック・ルーティング・プロトコルアーキテクチャの提案",MBL41,2007

[14]. 森拓海，森郁海，高橋修 "AAAr: Anti Attack Ad-hoc routing protocol の提案と実装・評価",DICOMO2007,2007

[15]. 森郁海，森拓海，高橋修 "アドホックネットワークにおける攻撃法・防御法の分類と AODV ベースセキュアルーティングプロトコルの提案",MBL41,2007

[16]. 森郁海，森拓海，高橋修 "AODV ベースセキュアルーティングプロトコルの提案とその実装・評価",DICOMO2007,2007