

# エージェント数制御を行う Ant 型アドホックネットワークルーチング

大藪良祐<sup>†</sup>川合誠<sup>†</sup>野口拓<sup>†</sup><sup>†</sup>立命館大学大学院 理工学研究科

## 1 はじめに

アドホックネットワークでは、移動可能な各端末がデータを中継していくことで離れた端末間の通信を実現する。有線ネットワークと違いネットワークトポロジが絶えず変化するため、効率的なルーチングプロトコルの開発が求められている。

Reactive 型に分類される AODV[1]と Proactive 型に分類される Ant 型モバイルエージェントを用いたルーチングを組み合わせたハイブリッドルーチングプロトコル[2]はお互いの欠点を補い合うプロトコルである。しかし、Ant 型モバイルエージェントを用いたルーチングには、ルーチングの要となるエージェントパケットが、転送中にノードの移動などで失われていく問題が存在する。

本稿ではこの Ant 型モバイルエージェントを用いたルーチングにおいて、エージェント総数の減少によりルーチング性能が大きく低下する問題を明らかにする。その解決案としてエージェント数を自動制御する方法を提案し、シミュレータでの性能評価で、問題解消の確認とエージェント数とルーチング性能の関係について検証する。

## 2 エージェント数減少問題

Ant 型モバイルエージェントを用いたルーチングは、ルーチング情報を持った複数のエージェントパケットが、隣接ノード間を常にユニキャストで転送されていくことでルーチング情報を伝播させる。

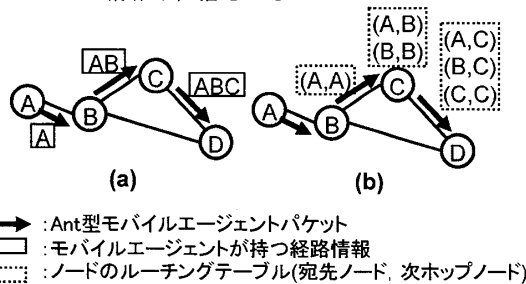


図1: Ant 型モバイルエージェントルーチング

エージェントはネットワーク中を歩き回りながら、通ってきたノードのアドレスを記録する(図 1a)。同時に立ち寄ったノードのルーチングテーブルに、それまで通ってきたノードへの経路を与える(図 1b)。結果、エージェントの辿った逆順が、利用可能な経路として確立される。エージェントはネットワークの動作開始時にパラメータとして定めた総数だけランダムに選ばれた各ノードより生成され、ユニークなエージェント ID を付加される。

Ant-like Ad Hoc Network Routing Controlling the Number of Agents Automatically

Ryosuke Oyabu<sup>†</sup>, Makoto Kawai<sup>†</sup>, Taku Noguchi<sup>†</sup>

<sup>†</sup>Graduate School of Science and Engineering, Ritsumeikan University

生成されたエージェントはそれぞれ独立にネットワーク中を歩き回り、新たにエージェントが生成されることはない。そのため、エージェントが転送中にパケットロスやノードの移動などで失われると、エージェントの総数が減少する。エージェントの総数が減少すると、ノードに新鮮なルーチング情報が伝播される機会が減少するため、ルーチング性能が大きく落ちることになる。

## 3 エージェント数制御

本稿では、エージェントを転送する前にエージェントのクローンを生成することで、パケットロスによりエージェント総数が減少する問題を解決し、パラメータとして定めたエージェント総数を維持する方法を提案する。

エージェントは隣接の次ホップノードへ移動する前に、自分と同じ情報を持ったクローンを送信元ノードに残しておく。クローンエージェントは、親エージェントの次ホップノード到達を確認できれば消滅し、確認できなければ親と同じように自分のクローンをノードに残し、親が移動したものと異なる隣接ノードへ移動する。このようにしてエージェントが次ホップノードへ正常に到達できたかどうか不確かな場合に複製を他の経路で流しておけば、総エージェント数が減少することはない。親エージェントが正常に到達できたかどうかを判断するには、IEEE802.11 の ACK メッセージのような機能を用いる。

次に、エージェント数を制御する方法を示す。エージェントは生成時にユニークなエージェント ID を付加されるため、複製や再生成されない限り個々のエージェントを見分けることは容易である。エージェント数を減少させないためには複製を行わなければならないが、単純に複製するだけでは、複製されたエージェントたちは同じエージェント ID を持つため、見分けることができない。そこで、各エージェントにエージェント ID に加えてクローン ID というフィールドを設定する。エージェントは自分のクローンを生成するにあたり、自分のクローン ID 以外の全ての情報を、新しく生成したエージェントへコピーし、自分のクローン ID に 1 加算した値を、新しく生成したエージェントのクローン ID として設定する。こうして複製されるエージェントは、エージェント ID とクローン ID の組により見分けることができる。

親エージェントがクローンを生成後、次ホップへ到達できたにもかかわらず、クローンがそれを確認できず活動を開始した場合、同じエージェント ID を持つ親エージェントとクローンが余分に生成されることになり総エージェント数が増加する。この現象が度々発生すると総エージェント数が増加し続けていくことになり、ネットワーク帯域の圧迫をひきおこす。これを解決するため、エージェントは立ち寄ったノードに自分のエージェント ID とクローン ID の組を記録しておく。エージェントは各ノードに立ち寄るたびに、エージェントリストを確認

し、自分と同じエージェント ID で自分のクローン ID よりも大きいクローン ID のエージェントが存在する場合、自ら消滅するようにする。このように、同じエージェント ID を持つエージェントたちを淘汰させて、一つのエージェント ID につき一つのエージェントのみが生き残るようにすることで、最初に定めたエージェント数の個数だけエージェントを存在させておくことができる。

#### 4 実験と結果

解決案を評価するため、Jist-SWANS ネットワークシミュレータ [3] を用いて既存のプロトコルと比較を行う。

##### 4.1 シミュレーションシナリオ

フィールドサイズ	1500x300 (m)
計測時間	600 (s)
ノード数	50
ノードの移動性	Random way point:0~10(m/s)
PauseTime	0, 30, 60, 120, 300, 600 (s)
物理層伝送距離	250 (m)
チャンネル帯域幅	2Mbps
データリンク層	IEEE802.11
総データ送信レート	20 (パケット/s)
初期エージェント数	10
HIST_SIZE	12

表2: シミュレーションシナリオ

表1のシナリオでPauseTimeについて、0, 30, 60, 120, 300, 600秒の6つの場合でそれぞれ計測する。各ノードはPauseTimeを待機時間とし、待機時間経過後ランダムに選んだフィールド上の目的地へ移動を始め、到達後また待機するという動作を繰り返す。実データパケットは、20個のクライアントノードがそれぞれ1秒ごとに30個のサーバノードからランダムに選び、1パケットずつ送信する。HIST SIZEは各々のエージェントが記録できる経路情報の数を示し、大きい値を取るとノードに与える経路情報も多くなるが、パケットのサイズが大きくなるためネットワーク帯域を圧迫する。

##### 4.2 エージェント数

シミュレーション中のネットワーク内に存在するエージェント数を推定するため、シミュレーションの単位時間ごとに、受信できたエージェントパケットを計測した。

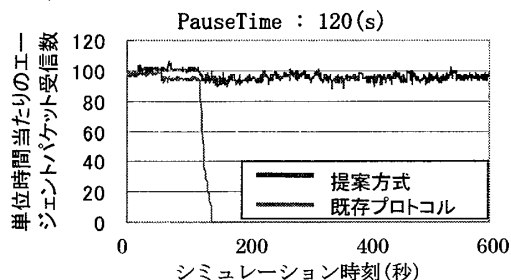


図2. 単位時間当たりのエージェントパケット受信数

図2ではPauseTimeが120秒のときのエージェント数制御を行う場合と行わない場合のプロトコルについて比較している。PauseTimeが120秒のとき、時刻120秒辺りからノードが移動を開始するため、エージェント数制御を行わないプロトコルでエージェントパケットのロスが発生し、しばらくして全ノードが消滅している。一方、本稿の解決案を用いると、ノードの移動に関わらず常に同程度のエージェント数を保っていることがわかる。

##### 4.3 end-to-end 遅延平均

end-to-end 遅延は、送信元ノードにおいて、送信パケット(実データパケット)が発生した時点から、宛先ノードへ実データパケットが到達するまでの時間を表している。これには、送信元ノードが行う経路探索の時間も含み、データリンク層以下の転送時間も含む。

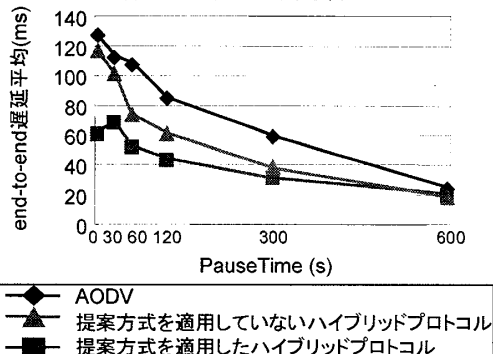


図3. end-to-end 遅延平均

図3では、各PauseTimeのシミュレーション別に、受信した全ての実データパケットのend-to-end遅延の平均値を示している。Ant型モバイルエージェントを用いたルーチングとAODVを組み合わせたハイブリッドプロトコルを評価対象とし、本稿のエージェント数制御を行う場合と行わない場合、また純粋なAODVについても計測し、比較を行った。提案方式を適用してエージェント数制御を行った場合のハイブリッドプロトコルのend-to-end遅延平均が、純粋なAODVのものに比べ半分近くに短縮できていることがわかる。これは、エージェントが常にルーチング情報をネットワークに伝播させているため、フラディングによる経路探索を行わずにデータ配送できることが多いためである。一方、既存のエージェント数制御を行わないハイブリッドプロトコルについて見てみると、ノードの移動性が高いPauseTime:0や30のときには、純粋なAODVに比べて1割ほどしか短縮できていない。これはノードの激しい移動で早々にエージェントパケットのロスが発生し、大部分のシミュレーション時間でエージェントが動作できていなかったことが理由であると考えられる。

#### 5 まとめ

本稿では、Ant型モバイルエージェントルーチングのエージェント数を制御する方法を提案した。シミュレータを用いてエージェント数が減少する問題の解決を証明し、提案方式を適用したハイブリッドルーチングプロトコルの性能評価により、AODVと比較してend-to-end遅延平均を約半分に短縮することを示した。また、本来のAnt型モバイルエージェントの性能を十分に発揮するためには、適正な値として定めたエージェント数を保ち続けることが必要であることを明らかにした。

##### 参考文献

- [1] C. E. Perkins, E. M. Royer and S. R. Das, "Ad Hoc On-Demand Distance Vector (AODV) Routing," in Proc. IEEE Workshop on Mobile Computing Systems and Applications, pp. 90-100, Feb. 1999.
- [2] A. Velmurugan, R. Rajaram, "MANET Internet Access using Reactive Mobile Agent Protocol," in Proc. IEEE, 2005.
- [3] <http://jist.ece.cornell.edu>