

アドホックネットワークにおける 多対多マルチキャスト木の構築方法

松井 宏基 佐藤 文明†

† 静岡大学 情報学部

概要: 本稿では、アドホックネットワークでの多対多マルチキャスト木の構築方法を提案する。提案方式ではノードを根とする木を構成する方式を基本にした。そしてそのコアの選択方針を、マルチキャストに参加する全てのノードに公平な木を構築する、とした。マルチキャストに参加する全てのノードに公平な木とは、コアノードから、マルチキャストに参加しているノードへのホップ数が、全て同等である状態とする。提案方式では、コアを根とする1本の木をデータ送信に使用するので、制御パケットが少なくなるという利点がある。今回はシミュレーションにより、マルチキャストに参加する全てのノードが独自に送信木を持つ方式と、提案方式とを比較し、評価を行った。

A construction method of the many-to-many multicasting tree in ad hoc networks

Hiroki Mastui Fumiaki Sato †

† Graduate School of Information, Shizuoka University

Abstract: In this paper, a construction method of the many-to-many multicasting tree in ad hoc networks is proposed. Although there were various systems in a network routing protocol, we proposed a system based on the core as a root of the tree. In the proposed method, a fair tree is built to all the nodes in the multicast group. The fairness to all nodes is the equivalence of the number of the hop from a core node to the node. Since one tree which uses a core as a root is used for data transmission, the proposed system has an advantage that a control packet decreases. The proposed system is evaluated by the simulation in comparing with the system which has a tree for each time.

1 はじめに

近年、モバイル端末(ノートPC、携帯電話など)の普及と、無線通信の発達により、アドホックネットワークという、端末同士が即席に構築できるネットワークが研究されている。また、Bluetooth[4]や、無線LAN等の近距離無線伝送方式に関する研究が行われ、携帯端末同士でアドホックネットワークを構成する環境が整いつつある。アドホックネットワークにおいて、多人数の参加者に同じ情報を送信するときには、マルチキャスト通信が有効となる。また、電子会議や電子ホワイトボードのような応用においては、多対多のマルチキャストが必要となる。この

ようなアドホックネットワークにおけるマルチキャストプロトコルについて様々な研究[1]がされている。その提案されてきた方式には、木をデータ転送の基本とする方式や、メッシュをデータ転送の基本とする方式や、フォワーディンググループをデータ転送に使用する方式などがある。今回は、木をデータ転送の基本とする方式に着目する。本研究では、全てのマルチキャストに参加するノードに対して公平な木を作り、ノード全てに公平性を持たせることを目標とするマルチキャスト木の構築方法を提案する。それによって、オーバーヘッド等のコストを最小限に抑えることが可能である。

2 アドホックネットワークでのマルチキャスト

2.1 送信者を根とする方式

無線アドホックネットワークにおけるルーティングプロトコルのうち最も単純で、送信者が根となる木を構成し、この木に従ってパケットを分配する方式である。この方式は、Distance Vector Multicast Routing Protocol(DVMRP)[5]に代表される。一般にDVMRPはインターネットで使用されているが、無線アドホックネットワークに拡張されている。しかし、マルチキャストに参加する送信者の数だけ木を作る必要があるため、この方式はスケーラビリティに問題がある。

2.2 コアを根とする木を利用する方式

これは送信者を根とする方式の欠点を改善したものである。送信者を根とする方式は、スケーラビリティに限界があるので、この方式では送信者ごとに木を構築するのではなく、共有木を利用する。この場合、コアと呼ばれる幾つかのノードを維持する必要がある。また、送信者から受信者までの最短経路を取らないという欠点がある。更に、木の再構成が頻繁に必要となる。この方式の代表的なものとして、Adhoc Multicast Routing Protocol(AMRoute)[6]が挙げられる。

2.3 マルチキャストメッシュを用いる方式

Call Admission Multicast Routing Protocol(CAMP)[7]は、コア方式をベースに、木よりも接続性が高いグラフを構成し、しかもループを防止する方式として提案されている。他のユニキャストルーティングプロトコルを基盤とすることを前提とした方式である。ODMRP[3][4]は、コアや木に依存せず、マルチキャストパケット転送の責任を持つノードのグループ(フォワーディンググループ)を利用して、最短経路でマルチキャストパケットを転送する方式である。送信者のマルチキャスト広告に対し、参加ノードが加入パケットをフラッドングすることにより、フォワーディンググループが構成される。

3 共有木による多対多マルチキャストの提案

マルチキャストパケットの配送に、単一の木を用いる共有木方式を検討する。共有木方式は、木を構築するコストが小さく、マルチキャストに参加しないノ

ドへの負担が小さいという利点が考えられる。はじめにある送信者を根とする木を考える。このとき1人の送信者にとって最適な木となるが、次に別の送信者がパケットを送信する際には不都合な場合がある。図1を見ると、ノードAからノードB,Cへパケットを転送する場合は、最短経路を通して転送されるが、BからCへ転送する場合、遠回りをしているのがわかる。そこで、送受信を行うノード全てに公平な木を構

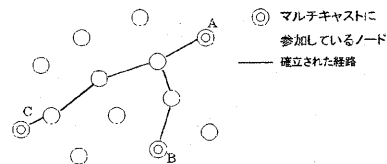


図1: 送信者を根とした木

築することを考える。公平な木とは、マルチキャストに参加しているノード全てが、同等のホップ数で中心のノードに到達している木を指すことにする。この中心のノードをコアノードと呼ぶ。更に、全ての受信者、送信者からのコアノードへのホップ数の平均値が小さいほど良い木とする。また、コアノードは、動的に変化し、木の構築と新しいコアノードを選出することに責任を持つ。公平な木を構築できた場合、マルチキャストに参加している全てのノードは、同等のホップ数でコアノードに到達する。公平な木を構築することで、あるノードから送信されたデータだけが極端に遅れることを防ぐことができる。(図2)

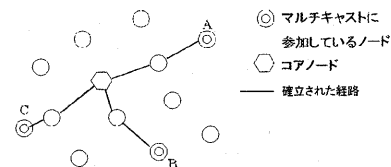


図2: 公平な木を構築

3.1 アルゴリズムの方針

提案する方式は、コアを根とする木を構築する方式に分類されるが、そのコアの選択は、マルチキャストに参加する全ノードから同等なホップ数で到達するノードとする [1][2]。まずマルチキャストに参加する任意のノード(仮のコア)が、他のノードに向けてパケットをフラッドングする。その際に、パケットをフォワーディングするノードは、そのパケットに仮の

コアから自分へのホップ数を加えて次のノードに転送する。そしてそれを受け取ったマルチキャスト参加ノードは、仮のコアにそれを送り返す。その経路によって、仮のコアを根とした木を構築する。次に、仮のコアから最もホップ数の多いマルチキャスト参加ノード間のパスに注目し、パスの中間にあるノード(コア候補)が全てのマルチキャスト参加ノードに対してパケットをフラディングする。このときもホップ数を計測する。計ったホップ数の平均値が、仮のコアが計測したホップ数より小さければ、このノードをコアノードにする。小さくなければ、このノードと仮のコアのパス上の中間のノードをコア候補として同様の処理をさせる。

3.2 構築手順

公平な木を構築する手順を以下に述べる。マルチキャストに参加するメンバとマルチキャスト経路は、コアの要求によって構築・更新される。コアがマルチキャスト経路を構築するので、マルチキャストに参加したノードは、その経路に従ってパケットを転送するだけでよい。木が構築される前は、マルチキャストを開始するノードがコアの役割を果たす。コアは定期的にマルチキャストに参加するメンバを決定し、そのメンバへの経路を構築するために、Join-Query パケットをネットワーク全体にフラディングする。以下に、ノードがコアから Join-Query を受け取った時の処理を示す(図3)。A がマルチキャストを始めた

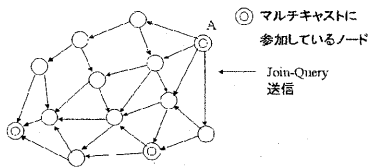


図3: Join-Query 送信

場合、A はコアノードとなり、マルチキャストに参加するノードを確認するために Join-Query をネットワーク全体にフラディングする。それを受け取った各ノードは、マルチキャストに参加する場合は Join-Reply を、Join-Query が転送されてきた経路を遡ってコアに返し、引き続き Join-Query を転送する。(図4) マルチキャストに参加しないノードは引き続き Join-Query を転送する。以上の操作で確立された経路(図5)を使って、次のコアが選ばれる。Join-Reply にはコアからそのノードまでの経路情報とホップ数

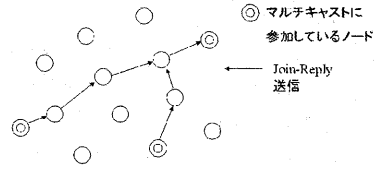


図4: Join-Reply 送信

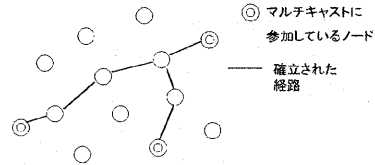


図5: 確立された経路

が記録されている。コアはその情報を元に次のコア候補を決定する。まず、コアから一番ホップ数の多いノードとの経路に注目する。そして、その経路上の中間のノードをコア候補とする。現在のコアは、そのコア候補にホップ数の平均値を調べるように命令する。現在のコアのホップ数の平均値より、コア候補のホップ数の平均値が下回れば、コア候補をコアにし、現在のコアはコアでなくなる。コアが移動した後の図を図6に示す。

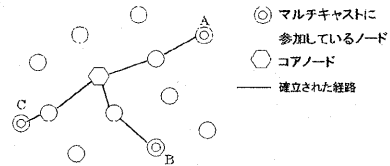


図6: コア移動後

3.3 複数の共有木への拡張

ネットワークの中で複数のコアノードを持つことにより、マルチキャストに参加できる状況を増やすことができる。例えば、マルチキャストに参加したいノードが局所的に集中してしまうと、そのノード群だけに有効なコアノードが選択されてしまう。その結果、そこから大きく離れた場所にあるマルチキャストに参加したいノード群には転送経路が確立されない。そのような状況を防ぐために、大きく離れた場所にいるノード群でもコアノードを選択する。それにより、全体のマルチキャスト参加率を上げるこ

とが可能になる。

具体的な手順として、マルチキャストに参加したいノードは、一定時間コアノードを持つことがなければ Join-Query を発行する。それにより、周囲にマルチキャストに参加したいノードが存在するかどうかを確認することができる。存在すれば、Join-Reply が返ってくるので、Join-Reply に含まれている情報をもとにコアノードを選択し、データ転送木を確立することができる。Join-Reply が返ってこなければ、周囲にマルチキャストに参加したいノードは存在しないということになるので、一定時間経った後に、もう一度 Join-Query を発行すればよい。

3.4 高信頼化

共有木方式は冗長性がないため、途中のリンクが切れると到達率が極端に悪くなってしまふ。これに対して、高信頼化のための2つの方法を検討している。局所的に再送処理を行う方法と、別の転送経路を使用する方法である。

局所的に再送処理を行う方法では、転送経路上のあるリンクが切断した場合、その周囲のノードがデータの転送を行う。あるノードがデータを送った際に、宛先のノードはデータを受け取ったら、受信完了パケットを送信する。送信ノードは受信完了パケットが一定時間送られてこなければ、送りたいデータと宛先を周囲のノードに送信する。周囲のノードは、自分の周囲にそのパケットを送信する。宛先のノードが存在すれば、そのノードから受信確認パケットが返ってくるので、転送を代わりに行ったノードはこれを送信元に返す。以上の操作により、再送処理が行われる。

別の転送経路を使用する方法では、現在のコアノードの転送経路が使用できなくなった際に、1つ前のコアノードの転送経路をバックアップとして使用する。転送経路上のノードは、パケットを受け取れば、受信確認パケットを送信者に返す。このパケットが返ってこなければ、接続が切断したと考えられるので、現在の転送経路を使わず、1つ前の転送経路を使用する。そのためには、新しいコアノードが選択された際に、古い転送経路を使用していたときの情報を破棄しないで、保持しておくだけでよい。

4 性能評価及び考察

今回は、共有木を複数持たずバックアップを行わない基本的な提案方式で評価を行った。性能評価は、

提案方式と、送信者が独自にそれぞれの木を持つ方式についての比較を行う。送信者が独自にそれぞれの木を持つ方式とは、マルチキャストに参加するノード全てが、他の参加するノード各々に対して木を持つ方式である。

4.1 評価方法

送信者が独自にそれぞれの木を持つ方式と、提案方式についてをシミュレーションを行い、比較し評価する。幾つかのパラメータを変化させて、木を構築・維持するのに必要だったパケット数と、パケット到達数を比較する。

4.2 シミュレーションモデル

シミュレーションは距離 1000m × 1000m のフィールドを想定して、そこに N 個のノードをランダムに配置させた。ノードの中から 5 つ、マルチキャストグループに参加するノードを抽出し、グループの構築と、マルチキャストパケットの配送を行う。これにより、木を構築・維持するのに必要なパケット数と、マルチキャストパケット到達数を 1000 秒で評価した。

4.2.1 移動モデル

各ノードの移動モデルを説明する。T 秒間だけ時速約 10km で方向 θ (8 方向) に移動する。T 秒間経過するとノードは、新たな目的地方向 θ をランダムで決めるか、もしくはその場に留まるかを決定する。移動する場合は T 秒間その方向に、ある一定速度時速約 10km を平均とする速さで移動し、留まる場合は T 秒間その場に留まる。今回のシミュレーションでは、時間 T は 5 秒に設定した。

4.2.2 パラメータ設定

パラメータとして、データが 1 ホップするための時間を平均 0.01 秒、各ノードのデータ処理時間を平均 0.01 秒とする。マルチキャストパケットの送信時間間隔は平均 1 秒、各ノードの通信可能範囲を 100m とする。

4.3 シミュレーション結果

シミュレーション結果を以下に示す。

4.3.1 木を構築・維持するためのパケット数

ノード数の変化における木を構築するのに必要なパケット数を、提案方式と送信者ごとに独自の木を持つ方式で比較した。結果のグラフを図 7 に示す。図 7 から、提案方式は送信者ごとに独自に木を持つ方式に比べ、ノード数が増加しても、木を構築・維持するのに必要なパケット数が少なくてすむということがわ

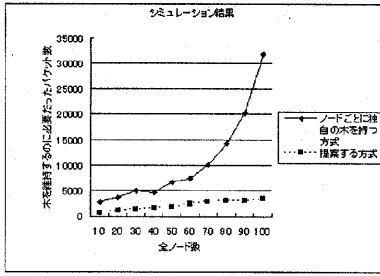


図 7: ノード数に対する、木を構築・維持するのに必要だったパケット数

かる。オーバーヘッドの面では、提案方式は送信者ごとに木を持つ方式をはるかに上回ることがわかる。

4.3.2 ノード数における、マルチキャストパケット到達数

提案方式と、送信者ごとに独自の木を持つ方式のノード数におけるマルチキャストパケット到達数の比較を行った。結果のグラフを図 8 に示す。図 8 か

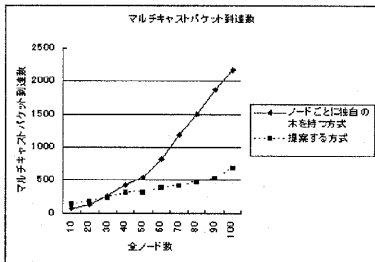


図 8: マルチキャストパケット到達数

ら、ノード数が 10 と 20 までは、提案方式の方が送信者ごとに木を持つ方式よりもマルチキャストパケット到達数が多い。しかし、ノード数が 30 からは提案方式ではマルチキャストパケット到達数はあまり伸びていない。ノード数が 30 を超えると、送信者ごとに木を持つ方式の方が信頼性が高いことがわかる。

4.4 考察

送信者ごとに独自の木を持つ方式では、提案方式より木を構築・維持するのに必要だったパケット数が、非常に多い。提案方式では、木を構築・維持する制御パケットを発行するのはコアだけなので、制御パケットは少なくすむ。送信者ごとに独自の木を持つ方式では、マルチキャストグループに参加するメンバごとに木を作るので、メンバそれぞれが自分の木を構築・維持するための制御パケットを発行しなければなら

ない。つまり、木の本数はメンバの数だけ作られるので、それだけオーバーヘッドが大きくなってしまう。

マルチキャストパケットが到達しない原因として次のことが考えられる。1 つは、データ転送経路上のノードが移動してしまったために、そこで経路が途切れてしまっている場合である。もう 1 つは、木を構築する際に転送する制御パケットがマルチキャストに参加するメンバまで届いていないため、コア(送信者)からメンバまでの経路が確立されていない場合である。リンクが切れることによって、木が分断される確率が同等であるとする、1 本の木に全メンバが依存する共有木は到達率が低下する。また、評価では木を構築するための制御パケットが受信者全員に到達せずに、部分木になっていても、その上でのマルチキャストパケットの到達数をカウントしている。そのため、共有木ではコアが領域の端の方にある場合、部分木が小さくなって到達数が著しく減少する場合があります、複数の木を用いる場合と比べて不利になった。(図 9,10)。

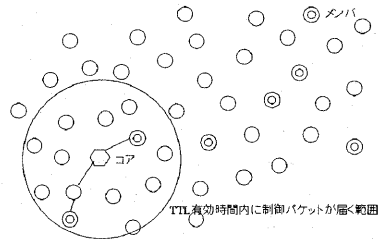


図 9: 効率的なコアが選択できない場合

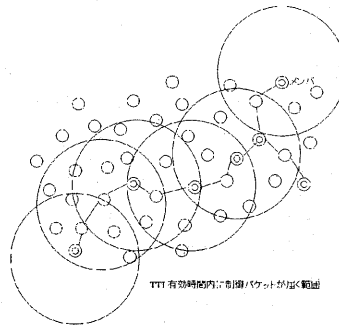


図 10: 送信者ごとに独自の木を持つ方式のカバー状況

5 まとめ

本研究では、アドホックネットワークでの多対多のマルチキャスト木を構築する際に、共有木を使った方式を提案し、木の中心となるコアノードの選択について、動的に行う方法を研究した。シミュレーションによって、送信者から別々の木を構築する方法と比較して、オーバーヘッドが少ないことを確認した。しかし、マルチキャストパケットの到達数は、送信者ごとに独自に木を構築する方式よりも少ないという結果が出た。

問題に対しての具体的な改善策として、コアをネットワーク全体で複数持つようなアルゴリズムが考えられる。マルチキャストパケットの転送経路が途切れた時の問題には、途切れたノードの付近のノードが、そのデータを代わりに転送して、経路を復元する方法が考えられる。

今後の課題としては、上で述べたような方法を加えて信頼性を向上させることが挙げられる。他に、木の構築間隔のパラメータなどの評価を行う。また、更に詳細な設計を行って実装し、評価する。

参考文献

- [1] 鈴木貴也 石川貴史 石原進 水野忠則：モバイルアドホックネットワークにおけるメンバー間公平性保証方式の基礎検討：マルチメディア, 分散, 協調とモバイル (DICOMO 2000) シンポジウム pp.781-786 (2000)
- [2] Sung-Ju Lee William Su Mario Gerla :On-Demand Multicast Routing Protocol (ODMRP) for Ad Hoc Networks IETF MANET Working Group INTERNET-DRAFT
- [3] <http://citeseer.nj.nec.com/bae00unicast.html>, "Unicast Performance Analysis of the ODMRP in a Mobile Ad hoc Network Testbed"
- [4] <http://www.ibsjapan.com/Bluetooth.htm>
- [5] D.Waitzman,C.Partridge,BBN STC,S.Deering:"Disatance Vector Multicast Routing Protocol",IETF (November 1998).
- [6] Mingyan Liu,Rajesh R.Talpade,Anthony McAuley:"AMRoute:Adhoc Multicast",CSHCN T.R. (January 1999).
- [7] Elena Pagani,Gian Paolo Rossi:"Call Admission Multicast Protocol(CAMP) for End-to-end Quality-of-Service" (September 2000)