

アドホックネットワークにおけるトポロジ依存機能の動的再配置

鈴木 貴也[†], 石原 進^{††} 水野 忠 則^{†††}

オンラインゲームなどの協調作業アプリケーションでは、アプリケーションに参加する端末間のデータの一貫性や遅延差を調停する機能(調停機能)を持つ端末(調停端末)が必要となる。このようなアプリケーションをアドホックネットワークで利用する場合、ネットワーク上のいずれかの端末が調停端末としての役割を果たさなければならない。このとき、ネットワークの中心に近い端末が調停端末となれば、調停端末とメンバとの間の通信遅延が小さくなると同時にトラフィックも減少し、ユーザの操作に対するアプリケーションの反応がすばやくなるなどアプリケーションの効率が向上する。しかしながら、端末の移動によるトポロジの変化および通信経路の変更の影響により、調停端末がネットワークの中心から大きく外れてしまう可能性がある。本稿では、ネットワークの中心に近い端末が調停端末となるようトポロジ変化に対応して調停機能を再配置する手法を提案する。本手法ではこの処理に協調作業アプリケーションが利用するメッセージ以外の新たなメッセージを発生させない。ルーティングプロトコルとして DSR を用いた場合について、提案手法の有効性をよってシミュレーションによって明らかにした。

Relocation of a Mediation Function on a Mobile Ad Hoc Network

YOSHINARI SUZUKI,[†] SUSUMU ISHIHARA^{††} and TADANORI MIZUNO^{†††}

In applications for cooperative works like online games, the mediation host which mediates the difference of delay or the consistency of data shared among group (total ordering) between hosts that join in the application is required. When such applications are used on mobile ad hoc networks, one of the hosts on the network must have the function of the mediation. If the host which is near the center of the network performs as the mediation host, the communication delay and the traffic will become small, and the efficiency of application programs will be improved. However, a host cannot stay in one place on the network because of the changes of topology by the movement of mobile hosts, the mediation host may locate far from the center of the network. In this paper, we propose a technique for relocating mediation functions corresponding to the topology so that a host near the center of the network may turn into the mediation host. The performance of proposal method was evaluated from view points of communication delay and the number of rearrangement of the mediation host by simulation in case of using DSR (Dynamic Source Routing) as a routing protocol on ad hoc networks.

1. はじめに

近年のモバイル端末の普及にともない、協調作業アプリケーションを携帯端末から利用する機会が増えている。一方、無線 LAN や Bluetooth¹⁾, HomeRF²⁾ などの無線通信技術の発達により、ワイヤレスモバイルアドホックネットワーク環境が実現可能となってきた。

アドホックネットワークとは、移動無線端末のみから構成されるネットワークである。固定の通信インフラを使う必要がないので、端末のみでいつでもどこでもネットワークを構築できるという利点がある。また、アドホックネットワークでは、無線通信距離の制限で相互に通信できない端末間の通信をマルチホップ通信で行うことで、広範囲の通信を可能としている。このような利点を生かし、近い将来アドホックネットワーク環境でも、マルチプレイヤーゲームやオークション、共有ホワイトボードなど空間やオブジェクトを共有する協調作業アプリケーションを利用する機会が増えると予測される。

協調作業アプリケーションでは、参加するメンバ端末間での遅延差の調停や、データの一貫性(因果関係)

[†] 静岡大学大学院情報学研究科
Graduate School of Information, Shizuoka University

^{††} 静岡大学工学部
Faculty of Engineering, Shizuoka University

^{†††} 静岡大学情報学部
Faculty of Information, Shizuoka University
現在、日本電気株式会社
Presently with NEC Corporation

を調停する特定の端末(調停端末)が用いられる場合がある。これらの処理には必ずしも調停端末が必要とされない場合もあるが、トラフィックの削減や個々の端末の負荷低減を目的として利用される機会は多い。たとえば、端末間の遅延格差を調停して、オークションや早押しクイズなどで発生する遅延差に基づく不公平性を解消する ICEGEM³⁾ や、マルチキャスト通信の順序一貫性保証のためにシーケンスサーバを用いる方式⁴⁾ などでは調停端末が使われている。

調停端末を使用する場合、有線ネットワークでは、固定のホストが調停端末の役割を果たすことが多い。一方、アドホックネットワークではネットワーク上のいずれかの端末が調停端末の役割を担うことになる。このとき、ネットワークの中心に近い端末を調停端末とすると、調停端末と協調作業に参加したメンバの端末間の通信遅延やトラフィックが減少するので、アプリケーションの処理効率が向上する。しかしながら、マルチホップの接続を含むアドホックネットワークでは、端末の移動によりネットワークのトポロジが変化し、このため、ネットワークの中心に存在していた調停端末がネットワークの端へ移動してしまう可能性がある。調停サーバがネットワークトポロジの中心から離れると、通信遅延やトラフィックが増加し、アプリケーションの動作が滞る原因となる。

そこで本稿では、アドホックネットワークのトポロジが変化してもネットワークの中心に近い端末が調停端末となるよう、ネットワークのトポロジに応じて調停機能を再配置する手法として REMARK (RElocatin of Mediation functions on a Mobile Ad hoc netwoRK) を提案する。REMARK は、この処理のためにアプリケーションで使用するメッセージ以外に余分なメッセージを必要としない点に特徴がある。

以下、2章でまずアドホックネットワークでの協調作業アプリケーションの問題点を検討し、調停機能の再配置の必要性について明らかにする。次に3章において REMARK における調停機能の再配置手法の詳細について述べ、4章で評価の概要およびシミュレーションモデルについて述べる。5章でシミュレーションの結果に基づく検討を行い、6章でまとめとする。

2. アドホックネットワーク上の協調作業

2.1 アドホックネットワーク

本稿で扱うモバイルアドホックネットワークとは、無線通信機能を持つ移動端末により構成される動的なネットワークであり、以下のような特徴を持つ⁵⁾。

(1) ネットワーク構築に固定の通信インフラを必要

としない。

- (2) 相互に直接通信できない端末どうしは、他の端末を中継することで通信する。
- (3) 端末の移動により端末間の直接リンクが切断されたり新しく作成されたりする。この結果、トポロジの変化やグループの分断、併合が起きる。
- (4) 特定の機能を持つホスト(サーバなど)が存在しない場合が多い。

さらに、ネットワークが移動無線端末のみで構成されることから、アドホックネットワークでは、通信にともなう電力消費をできるだけ抑える必要があるという要件がある。

以下、このような環境での協調作業アプリケーションに求められる要件について検討する。

2.2 ネットワークの分断にともなう一貫性管理

マルチプレイヤーゲームやオークション、共有ホワイトボードなど空間やオブジェクトを共有する協調作業アプリケーションをアドホックネットワーク環境で利用する場合、まず問題となるのは上記の(3)で示されるネットワークのトポロジの変化である。

協調作業アプリケーションでは、共有データ全メンバで同一に保たれるように、各メンバが行った処理内容あるいはその結果を同一の順序で全メンバに伝える必要がある。これを一貫性の管理と呼ぶ。ところが、アドホックネットワークでは、端末の移動の結果、ある時点まで協調作業を進めていたグループのメンバどうしが突然複数の別のネットワークに分断されてしまうことが起こりうる。また、分断の結果できたネットワークが、再度1つのネットワークに併合されることもある。アドホックネットワーク上での協調作業アプリケーションでは、このような状況においても、協調作業が矛盾なく進むように一貫性の管理を行う必要がある。さらに、ネットワークが分断された状況でも、別のネットワークにされた各グループで継続して作業が行えることが望ましい。文献6)では、エージェントを用いたサーバレスのグループ管理手法が提案されている。

ただし、本稿ではグループの分断・併合に関する問題については取り扱わない。以下の議論においては、トポロジの変化が発生してもグループの分断が永続的に継続する状況を考えないことにする。すなわち、一時的にすべての端末が相互に接続されない状況になったとしても、有限の時間の後にすべての端末が相互に接続されるものとし、分断によって孤立した1台ないし複数台の端末によって新たにグループを作成することがないものとする。この仮定は、明確な意思を持った

グループが同一地域に集まって作業を行う場合には達成可能なものとする。たとえば、大きな会議室に集まった人々がグループを作り作業を行う場合や、ゲームを楽しむ場合がこれにあてはまる。

2.3 一貫性管理にともなうトラフィックの削減

グループの分断を含まない場合、協調作業における一貫性管理の手法は、従来型のネットワーク上の協調作業を対象に多く提案されてきた。これらの手法は、特定のサーバを用いる手法と、サーバを用いない方法の2つに大別される。サーバを用いる場合、サーバに負荷が集中するという問題があるものの、サーバを介してメンバからのメッセージを集約してから配信することができ、システム全体のトラフィックを削減することが可能である⁴⁾。一方、サーバを用いない場合、特定の端末に負荷が集中することはないものの、個々のメンバがメッセージを発するたび、マルチキャストあるいはブロードキャストによってすべてのメンバへ因果関係保持のための情報を含んだメッセージを配信する必要があるため、システム全体のトラフィックが増大する。特に IEEE802.11b などの無線 LAN はブロードキャスト時にリンクごとの到着確認が行われないので、メッセージ発生頻度が高い場合に著しく通信の信頼性が低下する。

一貫性管理に特定の端末を使うかどうかは、個々の端末からの通信にともなうトラフィックおよび電力消費と、特定ホストの負荷とのトレードオフ問題となる。アドホックネットワーク上で協調作業を行うような環境では、ネットワーク中に特別に能力の高いホストが存在する可能性は必ずしも高いといえないので、ホストの負荷の問題は重要である。一方、トラフィック増大の問題も各端末の電力消費に関わり、同様に重要である。

本稿では、システム全体のトラフィック削減を第1の目標として、サーバを使って一貫性管理を行う場合を考えるものとするが、このサーバの機能を必要に応じて、異なる端末に移転させることをも想定する。以下の議論のため、本稿ではこのようなサーバ機能を一時的に担う端末を調停端末と呼ぶことにする。

2.4 トポロジの変化にともなう端末間遅延の変化への対応

トラフィック削減を目的に一貫性管理をネットワーク上の特定の端末で行うとすると、この端末のネットワーク上の位置が問題になる。調停端末がネットワークトポロジの末端付近にあると、調停端末とメンバ間のホップ数が最大となって、トラフィックおよび遅延の面から効率が良くない。効率を上げるには、調停端

末をネットワークの中心付近に配置するのが望ましい。

このように調停端末をトポロジ上の適切な位置に配置することで効果があるのは、一貫性管理を行うサーバだけではない。たとえば文献 3) では、オークションや早押しクイズ、空間共有型のマルチプレイヤーゲームなど、早いもの勝ちの論理が働くアプリケーションにおいて、端末間の遅延差にともなう不公平を解消するための遅延格差調停機構を提案している。このような処理を行う調停端末はネットワークの中心にある方が遅延格差調停にともなう遅延を少なくすることが可能なほか、調停端末からメンバ端末への遅延のばらつきが小さくなるので、遅延差を利用した不正の働きも小さくすることができる。

アドホックネットワークではノードの移動にともなうトポロジが変わるので、一時ネットワークの中心にいた調停端末が、ある時間後にネットワークの末端に移動する可能性がある。こうなると調停端末をネットワーク中心に置いた場合の利点が得られなくなってしまふ。このため、性能を維持するために、調停端末の機能をトポロジの変化に応じて、適切な端末へ動かす仕組みが必要である。

3. トポロジ依存機能の動的再配置

3.1 REMARK の特徴

前章で述べたように、アドホックネットワークでの協調作業において、トラフィックの増大を避ける場合には、調停端末が必要になり、そのネットワーク上の位置はトポロジの中心付近が望ましい。端末によって得られる効果はそのネットワークのトポロジに依存するともいえる。そこで本稿において提案する REMARK では、調停端末がネットワークトポロジの変化を察知し、その時点におけるトポロジに応じてネットワークの中心に近い端末が調停端末となるように調停機能を適切な端末に再配置する(図1)。これにより、トポロジ変化にともなうアプリケーションのメッセージ交換効率の低下を防止する。

REMARK では調停端末の選択およびその伝達のために新たなメッセージを必要としない。この処理に必要なメッセージは、協調作業を行うグループ間で定期的に交換されるメッセージに必要なに応じてピギーバックされる。調停端末の再配置やトポロジ把握にアプリケーションで使用されるメッセージやルーティングプロトコルの特性を利用する。ただし、調停端末が変わる場合に、古い調停端末と新しい調停端末のために確実にプログラム実行状態を受け渡すときには相応のメッセージを必要とする。

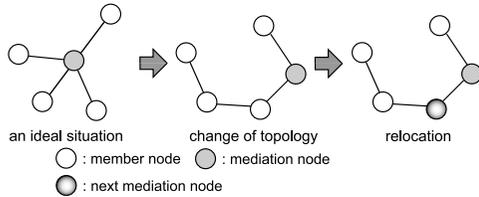


図1 REMARK の概念図
Fig.1 Concept of REMARK.

REMARK で行う調停機能の再配置は、リーダー選出問題の1つともとらえることができる。リーダー選出問題とは、複数のプロセス、あるいはノードから唯一のリーダーを選ぶ問題であり、従来より多くの研究が行われている。リンク障害を含む固定ネットワーク上でのリーダー選出問題を扱った研究としては文献 7), 8) などがある。また、特に、アドホックネットワーク上のリーダー選出問題を扱った研究としては文献 9)~11) などがある。文献 9) は3次元空間を考慮したモデルにより、アドホックネットワーク上のリーダー選出問題を解決するアルゴリズムを示している。文献 10) はアドホックルーティングプロトコル TORA¹²⁾ を応用したリーダー選出アルゴリズムを提案している。また、文献 11) では AODV でのアドホックネットワーク上のマルチキャストで必要となるリーダー選出の方法について説明している。上記のいずれも分断および併合されたグループ内での唯一のリーダーを選出するアルゴリズムを提案しているものの、ネットワークのトポロジならびにノード間のホップ数に関しては考慮されていない。

一方 REMARK では、リーダーである調停端末とメンバ端末との間でのアプリケーション上の通信で得られるトポロジ情報から、ネットワークトポロジの中心付近の端末を新しいリーダー(調停端末)とする点で違いがある。

なお、以下の説明で、REMARK では初期状態においてトポロジにかかわらずグループ内で唯一の端末が調停端末として動作することを仮定しているが、その選出に関しては特に手法を定めていない。また、本稿ではネットワークが長時間分断される場合の動作に関しては考慮に入れていないので、この場合の調停役の選出に関しても定めていない。これらに関しては、上記の文献 9)~11) の手法を用いることで、解決できると考える。

3.2 想定アプリケーション

REMARK では、以下のような特徴を持つアプリケーションでの利用を想定する。具体例としては、マルチプレーヤオンラインゲームやオンラインオーク

ションがあげられる。また実装によっては、共有ホワイトボードなどもこれに含まれる。

- 調停端末とすべてのメンバ端末との間で定期的なメッセージ交換が頻繁に発生する。
 - メンバ端末は調停端末から送信されるメッセージの受信をきっかけに調停端末へメッセージを送信する。
 - メッセージの交換間隔の短縮により効率が向上する(ユーザの操作に対する反応が早くなるなど)。
- 想定するアプリケーションにおけるメッセージ交換の流れは次のようになる。

- (1) 調停端末がすべてのメンバ端末へメッセージを送信する。
- (2) すべてのメンバ端末は調停端末へメッセージを返信する。
- (3) すべてのメンバ端末からの返信を受信した後、調停端末は次のメッセージをすべてのメンバ端末へ送信する。

3.3 制御メッセージ

REMARK ではアプリケーションが発生する2種類のメッセージを使用して調停端末の再配置を行う。1つを下りメッセージと呼び、もう1つを上りメッセージと呼ぶ。

下りメッセージは、調停端末からメンバ端末へマルチキャストないし複数のユニキャストにより送信されるメッセージで、最新の調停端末のアドレスが含まれる。上りメッセージは、下りメッセージに対する応答としてメンバ端末から調停端末へユニキャストにより送信されるメッセージであり、メッセージを送信した端末の持つトポロジ情報が含まれる。下りメッセージはすべてのメンバ端末からの上りメッセージ到着後に送信され、上りメッセージは下りメッセージ受信後に送信される。いずれのメッセージもアプリケーションにより発生する調停端末とメンバ端末との間のメッセージにピギーバックして送信される。よって、REMARK により通常のメッセージ交換に加えて大きなオーバーヘッドが発生することはない。

調停端末は上りメッセージに含まれる経路情報を利用してトポロジ変化を察知し、新しい調停端末を選択する。上りメッセージに含まれるトポロジ情報の例としては、端末の持つルーティングテーブルや送信端末から調停端末までの経路などがあげられる。

3.4 新しい調停端末の選択

調停端末では、すべてのメンバ端末からの上りメッセージを受け取ると、以下の手順で新しい調停端末を選択する(図2)。

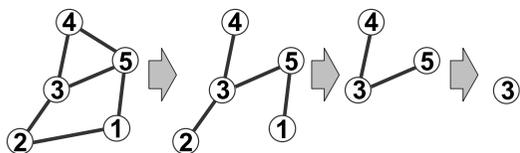


図 2 新しい調停端末の選択
Fig. 2 Selection of a new mediation node.

- (1) 調停端末はメンバ端末から送信される上りメッセージに含まれるトポロジ情報を利用して連結グラフを生成する。
- (2) 調停端末は (1) の連結グラフに対して調停端末にあたるノードを始点とした幅優先探索を行うことにより閉路を排除し、全域木を生成する。
- (3) 全域木のうち葉のノード、すなわち 1 つしかリンクを持たないノードをすべて削除する。
- (4) (3) の結果、葉のノードが削除された部分木が残る。この部分木の葉のノードをすべて削除する。この処理を部分木のノード数が 1 となるまで繰り返す。ただし、最後に 2 つのノードが残った場合、IP アドレスなどネットワーク内での ID が小さいほうを残すものとする。
- (5) 最後に残ったノードを調停端末とする。

3.5 調停機能の再配置

調停端末で次の新しい調停端末が選ばれると、以下の手順で調停機能の再配置が行われる (図 3) 。

- (1) 調停端末は下りメッセージを利用し、すべてのメンバに新しい調停端末を通知する。
- (2) メンバは下りメッセージ受信時に新しい調停端末を認識する。
- (3) 新しい調停端末は下りメッセージ受信後に調停を開始する。
- (4) メンバは新しい調停端末へ上りメッセージを送信 (再配置完了) する。
- (5) 下りメッセージより先に上りメッセージが新しい調停端末に到着した場合、新しい調停端末は最初の上りメッセージが届いた時点で調停を開始する。

最初の調停端末はグループで ID (IP アドレスなど) が最小の端末であるとする。これが未知の場合、既存のリーダー選出アルゴリズムを用いて、最初の調停端末を選出する。

調停端末が求めた新しい調停端末のアドレスは、下りメッセージによってメンバに通知される。メンバ端末は、以後、この下りメッセージにより通知された新しい調停端末のアドレスに対して上りメッセージを送信する。よって、新しい調停端末が現在の調停端末

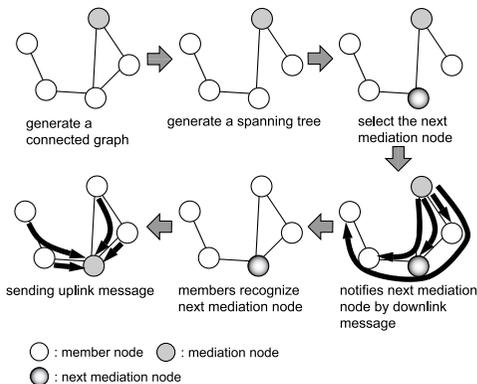


図 3 再配置の手法
Fig. 3 Method of relocation.

と異なる端末であった場合のみ実際の再配置が行われる。

調停機能を再配置する際、現在の調停端末から新しい調停端末へ実行中のプログラムの状態を受け渡す必要がある。状態受け渡しのための具体的手法については検討の必要があるが、本稿では以下のようにして受け渡しが行われるものとする。

すべての端末上でまったく同じプログラムが動作しており、プログラムの状態に関する簡単なパラメータを得ることですぐに調停を実行できるものとする。このとき、調停端末は下りメッセージにプログラムの状態を送信し、新しい調停端末がこの下りメッセージを受信することでプログラムの状態の受け渡しが可能となる。

しかしながら、下りメッセージより先に上りメッセージが新しい調停端末に到着したり、新しい調停端末が下りメッセージを受信できなかったりする可能性もある。そこでメンバ端末は、下りメッセージに付加されていたプログラムの状態を上りメッセージに付加して新しい調停端末へ送信する。以上のような手法を用いることで、新たなトラフィックを生ずることなくプログラムの状態の受け渡しが可能である。なお、より確実に新しい調停端末への機能受け渡しを行うには、新しい調停端末が以前の調停端末に確認応答を行う必要がある。

4. シミュレーションモデル

REMARK の利用により、調停端末とメンバ端末との間の通信遅延の減少が期待される。本稿ではネットワークシミュレータ ns2¹³⁾ を利用したシミュレーションにより REMARK の有効性を検証した。シミュレーションでは複数台の移動端末が定められた範囲の

領域内を移動しながら協調作業を行うシナリオを想定した。このシナリオでは、調停端末が全メンバに下りメッセージを送信すると、これを受け取ったメンバがただちに上りメッセージを送信する。調停端末は、すべてのメンバからメッセージを受け取ると、ただちに新たな下りメッセージを全メンバへ送り出す。

4.1 移動モデル

個々の端末の移動モデルには、以下に示す Random Waypoint モデル¹⁴⁾を用いた。

- (1) 移動領域内から目的地をランダムに選択する。
- (2) 0～最大速度 V_{max} の間から一様分布により移動速度を決定する。
- (3) 目的地へ向けて移動する。
- (4) 目的地へ到着後、pause time の間停止する。
- (5) (1)へ戻る。

4.2 下りメッセージのタイムアウト

本稿では、アドホックネットワークネットワークの分離によるグループの分断を考慮に入れていないが、シミュレーションモデルでは移動端末間の接続が途切れる可能性がある。

そこで、端末間の接続が途切れてもメッセージ交換を継続させるためのシミュレーション上の条件として、調停端末において下りメッセージ送信後、上りメッセージを待ち合わせるためのタイムアウトを設定した。タイムアウト発生時、調停端末は上りメッセージが届いていないメンバ端末に対して下りメッセージを再送する。下りメッセージ再送後に再びタイムアウトが発生した場合、調停端末は新しい調停端末の選択を行わずに次の下りメッセージを送信する。新しい調停端末の選択を行う場合、調停端末はすべてのメンバ端末からの上りメッセージを受信すると同時に新しい調停端末の選択を行い、かつ下りメッセージを送信するものとした。

4.3 パラメータ

以下にシミュレーションの詳細なパラメータを示す。

- 移動領域：一辺の長さ L : 100 [m] ~ 1000 [m] の正方形領域
- 端末数：20
- 移動モデル：Random Waypoint
 - 端末最大移動速度 V_{max} : 2.0 m/s (人が歩く程度の速度), 16.0 m/s (自動車程度の速度)
 - pause time : 1.0 sec
- 移動シナリオ数：移動パターンの異なるシナリオを各 L につき 10 ずつ適用
- シミュレーション時間：500 sec
- トランスポート層：UDP

- MAC 層：IEEE802.11
- 通信範囲：半径 250 m (双方向リンクを仮定)
- 帯域幅：2.0 Mbps
- パケット長：64 bytes
- 下りメッセージのタイムアウト：0.3 sec

4.4 ルーティングプロトコル

4.4.1 ルーティングプロトコルの選択

REMARK では、独自のメッセージを使わず、アプリケーション上のメッセージがメンバ端末と調停端末間を配送された経路に従って調停端末を決定する。この経路は、そのネットワークが利用するルーティングプロトコルに依存する。したがって、REMARK で決定される調停端末の位置は、上りメッセージの配送のために用いられるルーティングプロトコルに依存する。

モバイル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には DSDV¹⁵⁾, DSR¹⁴⁾, AODV¹⁶⁾, TORA¹²⁾ などがある。DSDV では、各移動端末はルーティングテーブルを保持し、定期的にリンク状態を交換することでルーティングテーブルを更新する。DSR, AODV, TORA は、移動端末はパケットの転送の必要が発生したときに経路を探索するオンデマンド型のプロトコルである。

本稿のシミュレーションではルーティングプロトコルとして DSR を用いた。DSR (Dynamic Source Routing) では、送信されるすべてのデータパケットに送信元から宛先までの経路 (Source Route) が含まれる。Source Route を宛先端末において参照することにより、宛先端末は送信元端末から自端末までの経路を知ることができる。このように、DSR を用いることで上りメッセージを利用したトポロジの把握を比較的容易に実現可能である。よって本稿の評価ではルーティングプロトコルとして DSR を用いることとした。ただし、DSR を用いなくても、適当な間隔で全メンバからのデータパケットに経路記録を付加するにすれば、DSR 以外のプロトコルにおいても経路情報の収集は可能である。

なお、下りメッセージに関しては 1 対多の通信になるので、マルチキャスト通信用のプロトコルを用いるのが適切である。ただし、今回のシミュレーションでは、上りメッセージと同様に DSR によるルーティングを行い、ユニキャストで下りメッセージを配信するものとした。

4.4.2 DSR による影響

ルーティングプロトコルに DSR を使用したときの REMARK での調停端末選出に関する影響について述べておく。

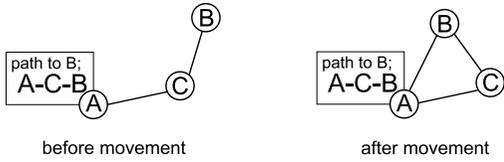


図 4 Source Route が最短ではなくなる例

Fig. 4 Example of a case when a source route is not the shortest route.

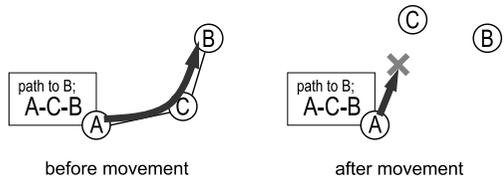


図 5 キャッシュされた経路が利用できない例

Fig. 5 Example of a case when a cached route is not available.

DSR では、宛先端末までの経路が未知であるか、それまで利用していた経路が無効になった場合に、送信元端末は宛先端末までの経路発見を行う。DSR の経路発見では、送信元端末からフラッディングされる経路要求 (Route Request: RREQ) に宛先端末が応答する (Route Reply: RREP) という処理を基本としている。また、宛先端末だけでなく、宛先端末までの経路をキャッシュに保持している端末も RREQ に対して応答する。よって DSR は、

- a) 送信元が知りうる経路が最適な経路であるとは限らない、
 - b) 送信元からの経路から把握可能なトポロジが、実際のトポロジと同一であるとは限らない、
 - c) キャッシュされた経路がつねに利用可能とは限らない、
- という性質を持つ。

a) の例として次のような場合が考えられる。端末 A が通信範囲外にいる端末 B へデータを送るために端末 B への経路を検出した後、移動により B が A の通信範囲内に入ると、A から B への経路は最短でなくなる (図 4)。b) は a) と同様の例で考えることができる。c) は a) とは逆のパターンとなる。つまり、端末 A が端末 B へデータを送信する場合、A がキャッシュしていた経路 (A-C-B) に従って送信しようとする。このとき、端末 C が A の通信範囲の外へ移動していると、A は B へ送信できない (図 5)。

これらの性質により、

- (1) 新しい調停端末は最適なものにならない
- (2) 経路が利用不能であった場合、経路の再発見が

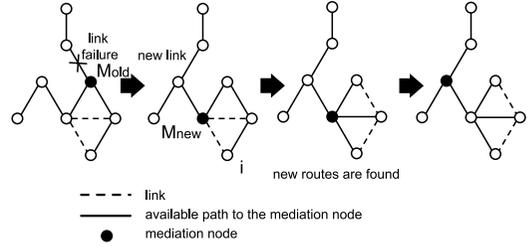


図 6 調停端末の移動の連続発生

Fig. 6 Series of rearrangements of mediation node.

必要となる

という問題点が発生する。これらの問題点により、調停端末がつねに最適な端末であるとは限らなくなり、調停端末とメンバ端末間の通信遅延やトラフィックが増加する、という影響が出ると考えられる。

なお、DSR に限らずオンデマンド型のプロトコルを使用する場合、新しい調停端末への変更が起きたとき、メンバ端末 i がその新しい調停端末への経路を知らないことがある。この場合、新たな経路探索の必要が生じる。この結果、新しい調停端末 M_{new} の選出時に古い調停端末 M_{old} が想定していたトポロジ上の、 $M_{new}-i$ 間のホップ数 M_n よりも少ないホップ数の経路が発見される可能性がある。図 6 のように、 M_{new} 選出時に、 i が M_{new} から最も遠い端末と見なされていた場合、 i からの新しい経路を M_{new} が知ると、新たな調停端末が選出される可能性がある。このような状態が続くと、調停端末の変更が連続して起こり、アプリケーションの進行に影響を与えると考えられる。

4.5 比較対象モデル

REMARK の有効性を検討するため、シミュレーションでは、以下の 3 つの場合について比較した。

- (1) DSR 上での REMARK により調停端末を再配置する場合
- (2) 再配置を行わない場合
- (3) 端末の位置に基づいて調停端末を再配置する場合 (位置に基づく手法)

(3) に示した位置に基づく手法は、最適な再配置手法に近い比較的实现容易な代替手段として用いている。

最適な再配置を行う場合、調停機能はその時点でのネットワークトポロジに基づいて中心に最も近い端末に再配置される。このためには、すべての端末間の接続情報を収集する必要があるが、多大な通信コストと計算コストが必要となる。一方、位置に基づく手法では、すべての端末の位置を把握することにより、簡単な計算で一定の条件下ではあるものの最適に近い再配

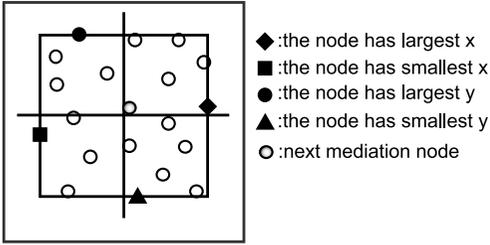


図 7 端末の位置を利用した新しい調停端末の選択

Fig. 7 Selection of a new mediation node using locations of nodes.

置を実現できる。

位置を利用する手法による新しい調停端末の選択は次のような手順で行う。

- (1) 調停端末はすべてのメンバ端末からの上りメッセージ到着後、すべての端末の位置を求める。実環境では、各端末が GPS などによって調べた自らの位置を上りメッセージに含めることで可能である。
- (2) すべての端末の x 座標, y 座標からそれぞれの最大値と最小値 $x_{max}, x_{min}, y_{max}, y_{min}$ を求める。
- (3) 全端末を内包する最小の矩形領域の中心

$$\left(\frac{x_{max} + x_{min}}{2}, \frac{y_{max} + y_{min}}{2} \right)$$

に最も近い端末を新しい調停端末とし、下りメッセージにより通知する。

図 7 のように端末が偏りなく位置している場合、位置に基づく手法により選択される新しい調停端末は最適な端末であるといえる。しかしながら、端末が偏って分布していた場合はネットワークの中心から大きく外れた位置の端末が新しい調停端末として選択される可能性がある。

5. 評価 価

5.1 評価項目

調停機能の再配置が行われていることを確認し、通信遅延の変化を調査するため 4 章において示したシミュレーションモデルを用い、以下の各項目について調べた。

- シミュレーション時間全体にわたる調停機能再配置の実行回数
- 調停端末とメンバ端末との間の最大往復遅延
最大往復遅延は、下りメッセージが送信されてから受信されるまでの遅延と上りメッセージが送信されてから受信されるまでの遅延の合計を端末ごとに計測し

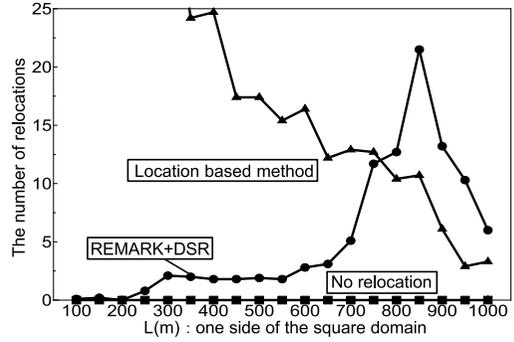


図 8 調停機能再配置の実行回数 ($V_{max}=2.0$)
Fig. 8 Number of rearrangements ($V_{max}=2.0$).

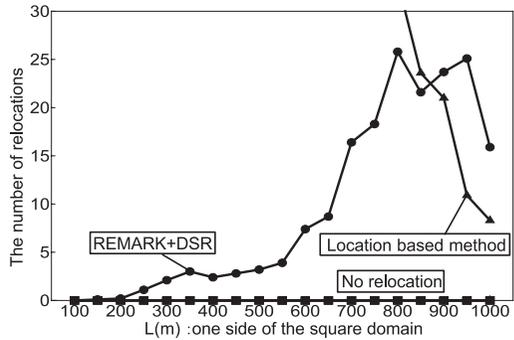


図 9 調停機能再配置の実行回数 ($V_{max}=16.0$)
Fig. 9 Number of rearrangements ($V_{max}=16.0$).

た値の最大値とした。以後に示す各測定値は、端末の初期位置以外が同じ条件のシミュレーション 10 回で得られた平均値である。

5.2 調停機能再配置の実行回数

図 8, 図 9 に、それぞれ $V_{max} = 2.0$ の場合、 $V_{max} = 16.0$ の場合における調停機能再配置の実行回数を示す。これより、領域がある程度大きくなると REMARK によって調停機能の再配置が行われることが分かる。

L が増加するにつれ REMARK を利用した場合の再配置の回数が増加している。このことは、領域が広くなるにつれてマルチホップの経路が生まれやすくなり、それにとまってトポロジ変化が起こりやすくなることを示している。

図 8 において、 L が 100~300 の範囲の場合、REMARK による調停機能再配置の実行回数はほぼ 0 である。これは、端末が互いに通信可能な範囲に密集しており、トポロジが調停端末を中心としたスター型となるので、再配置を行う必要がないためである。

L が 300~700 の範囲では、 L が 100~300 の範囲

の場合と比較して端末の密度が下がる．そのため、メンバ端末と調停端末との通信には2ホップ程度かかるようになり、トポロジ変化が発生するようになる．よって調停機能の再配置も2~3回程度発生する．

L が700~900の範囲では、端末の密度がさらに低下するためメンバ端末と調停端末との通信に要するホップ数は大きくなり、それとともにトポロジ変化も頻繁に発生するようになる．そのため、調停機能再配置の実行回数も多くなる．

ところが、 L が900より大きくなると調停機能再配置の実行回数は減少する． $900 < L$ では端末はまばらに分布する．そのためメンバ端末と調停端末との通信に要するホップ数が大きくなり、それとともにトポロジ変化も非常に頻繁に発生するようになる．しかしながら、同時にリンクの切断も多くなり、下りメッセージのタイムアウト以内に通信に成功する例が少なくなり、結果として再配置の回数も減少してしまっている．

一方、位置を利用する手法による調停機能再配置の実行回数は、 L が小さい間は極端に大きく、 L の増加とともに小さくなっていく．この理由は以下のとおりである．位置を利用する手法では、ネットワークのトポロジをまったく考慮せず、すべての端末を覆う最小矩形の中心に近い端末を新しい調停端末としている(4.5節)． L が小さく端末の密度が高い状態であるほど矩形の中心に近い端末の入れ替わりは発生しやすいので、結果として調停機能再配置の実行回数も大きくなる． L が大きく端末の密度が低い状態であれば、矩形の中心に近い端末の入れ替わりは発生しにくいので、調停機能再配置の実行回数は小さくなる．

$V_{max} = 16.0$ の場合(図9)に関しても、再配置の回数が大きくなることを除いて、図8と同様の傾向が観察できる．再配置の回数が増えるのは、単位時間内のトポロジ変化の回数が増加するためである．

シミュレーション結果での最大の再配置回数は500秒のシミュレーション時間で20回程度であった．下りメッセージの送信頻度は1秒あたり10回程度なので、約250回のメッセージ交換につき1度の調停機能の交換があったことになる．なお、今回のシミュレーションでは、4.4.2項で示した調停端末の移動が連続して発生するような振舞いは観測されなかった．

5.3 メンバ端末と調停端末間の最大往復遅延

図10, 図11にそれぞれ $V_{max} = 2.0$ の場合、 $V_{max} = 16.0$ の場合のメンバ端末と調停端末間の最大往復遅延を示す．

図10より、 L の値が小さい範囲では調停機能の再

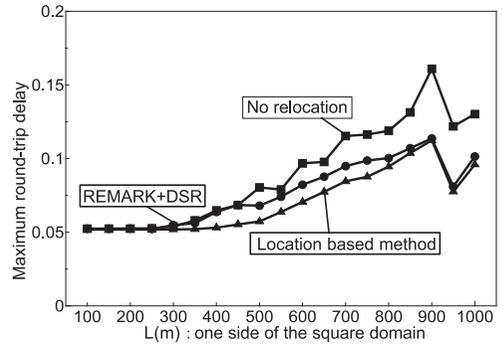


図10 メンバ端末と調停端末間の最大往復遅延 ($V_{max}=2.0$)
Fig. 10 Maximum round trip delay between a member node and the mediation node ($V_{max}=2.0$).

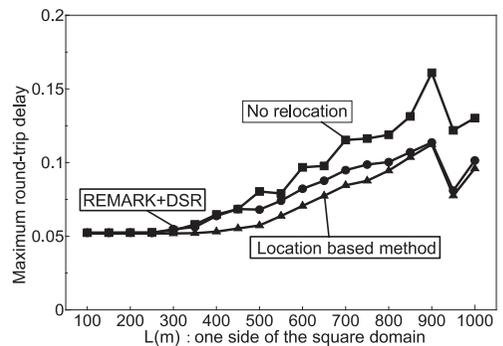


図11 メンバ端末と調停端末間の最大往復遅延 ($V_{max}=16.0$)
Fig. 11 Maximum round trip delay between a member node and the mediation node ($V_{max}=16.0$).

配置の有無による差は見られないが、 L がある程度大きくなると、調停機能の再配置を行わない場合は他の場合と比べて遅延が大きくなることから分かる．このことは、調停機能の再配置を行うことで遅延が小さくなることを示している． L が300より小さい範囲では、ほとんどの端末が互いの通信範囲内に入ってしまう、メンバ端末と調停端末の間のホップ数はたかだか1程度になる．よって再配置の有無や再配置の手法による差は見られない．

REMARKによる再配置が行われた場合と再配置が行われない場合の比較では、0.01~0.04[sec]程度の差が見られる．このとき、 L の値が大きくなるにつれて値の差が大きくなっていくことから、 L の値が大きい、つまり領域が広いほどREMARKによる通信遅延削減の効果が大きくなることから分かる．

一方、位置に基づく手法により再配置される場合と再配置が行われない場合とを比較すると $L < 300$ および $L \geq 900$ の領域では両者の差が見られない．一方、 $L \leq 300 < 900$ の領域では、位置に基づく手法の方が0.02~0.04[sec]最大遅延が短い．この差は、DSR

で得られる経路のみから選出した調停端末が、必ずしもネットワーク上の最適な位置の端末ではなかったことを示している。

L が 900 より大きくなると、再配置の有無にかかわらず最大往復遅延の値が急に小さくなっている。先にも述べたが、 $900 < L$ ではメンバ端末と調停端末の間のホップ数が非常に大きくなるため、トポロジ変化も非常に頻繁に起こるようになり、同時にリンクの切断も非常に多くなる。それにともない、下りメッセージのタイムアウト以内に通信に成功する例が少なくなる。通信に成功した場合、メンバ端末と調停端末の間のホップ数は小さくなっている。ホップ数が少なければ、通信遅延も小さくなる。結果として、最大往復遅延も小さくなってしまったと考えられる。

図 11 でもグラフの傾きが急になり、最大値が大きくなっている点を除いて、図 10 と同様の傾向が観察できる。これらの結果から、端末の移動速度が上昇しても REMARK の通信遅延に対する有効性は変わらないことが分かる。

5.4 検 討

シミュレーションの結果から以下のことが示された。

- REMARK を利用することで調停機能がトポロジに応じて再配置される。
- 調停機能が再配置されたことでメンバ端末と調停端末の間の最大往復遅延が減少する。
- 端末の最大移動速度と端末数が同じであるなら、端末の移動領域が広いほど REMARK による効果を得やすい。
- 端末の移動領域が非常に広くなると、調停機能再配置の実行回数や往復遅延が減少する。

移動領域が非常に広い状態における調停機能再配置の実行回数や往復遅延の減少の理由は、以下のとおりである。

- (1) 移動領域が広い状態では、各メンバ端末と調停端末の間のホップ数が増加するが、同時にリンクの切断が発生しやすくなる。
- (2) リンクの切断が頻発すると、各メンバ端末と調停端末の間で通信が成立しにくくなり、結果として再配置の回数も少なくなる。
- (3) このような状況で通信が成功するのはメンバ端末と調停端末の間のホップ数が少ない場合のみである。
- (4) ホップ数が少なければ通信遅延も小さくなり、結果として最大往復遅延も小さくなる。

移動領域が非常に広い状態においても接続性が十分に保たれ、十分な数のメッセージ交換が発生するので

あれば、本稿における結果のように調停機能再配置の実行回数や往復遅延が減少することはないと考えられる。

6. おわりに

ネットワークのトポロジを考慮して、協調作業アプリケーション実行上に必要な調停機能を動的に再配置することにより、アドホックネットワーク上で実行される協調作業アプリケーションの効率低下を防ぐ手法である REMARK を提案した。REMARK では協調作業アプリケーションで交換されるメッセージのみを利用してトポロジの把握とそれによって決定される調停機能の再配置先の伝達を行う。

REMARK について、現実のルーティングプロトコルである DSR の利用を想定したシミュレーションにより性能を評価した。その結果から、REMARK により調停機能が再配置されることを確認し、REMARK による調停機能再配置によりメンバ端末と調停端末の間の最大往復遅延が減少することを確認できた。また、端末の移動速度が同じであるなら、端末の移動する範囲が広いほど REMARK による効果を得やすいことが確かめられた。

再配置にかかわるコストの定式化、再配置にともなう収束時間の検討ならびに調停機能を再配置する際に実行中のプログラムの状態を受け渡す具体的手法については今後検討が必要である。

参 考 文 献

- 1) Bluetooth. <http://www.bluetooth.com>
- 2) HomeRF. <http://www.home-rf.com>
- 3) 石川貴士, 石原 進, 井手口哲夫, 水野忠則: 遅延差のあるネットワークにおけるメンバ間公平性保証方式の特性評価, 情報処理学会論文誌, Vol.42, No.7, pp. 1817-1827 (2001).
- 4) Satoh, F., Minamihata, K. and Mizuno, T.: A Totally Ordered and Secure Multicast Protocol for Distributed Virtual Environment, *Proc. 7th International Conference on Parallel and Distributed Systems: Workshop*, pp.55-60 (2000.7).
- 5) 間瀬憲一, 中野敬介, 仙石正和, 篠田庄司: アドホックネットワーク電子情報通信学会誌, Vol.84, No.2, pp.127-134 (2001).
- 6) 宮越勇樹, 河口信夫, 外山勝彦, 稲垣康善: アドホックネットワークにおける柔軟なグループ管理手法—モバイルエージェントを用いた表現, 情報処理学会マルチメディア, 分散, 協調とモバイル (DICO MO 2000) シンポジウム, pp.1-6 (2000).
- 7) Singh, G.: Leader Election in the Presence

- of Link Failures, *IEEE Trans. Parallel and Distributed Systems*, Vol.7, No.3, pp.157-171 (1996).
- 8) Fetzer, C. and Cristian, F.: A Highly Available Local Leader Election Service, *IEEE Trans. Softw. Eng.*, Vol.25, No.5, pp.603-618 (1999).
- 9) Hatzis, K.P., Pentaris, G.P., Spirakis, P.G. and Tampakas, V.T.: Fundamental Control Algorithms in Mobile Networks, *Proc. 11th Annual ACM Symposium on Parallel Algorithms and Architectures*, pp.251-260 (1999).
- 10) Malpani, N., Welch, J. and Vaidya, N.: Leader Election Algorithms for Mobile Ad Hoc Networks, *Proc. 4th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, pp.96-103 (2000).
- 11) Royer, E.M. and Perkins, C.E.: Multicast Operations of the Ad-hoc On-Demand Distance Vector Routing Protocol, *Proc. 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM)*, pp.207-218 (1999).
- 12) Park, V.D. and Corson, M.S.: A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks, *Proc. IEEE INFOCOM*, pp.7-11 (1997).
- 13) ns-2. <http://www.isi.edu/nsnam/ns/>
- 14) Johnson, D.B. and Maltz, D.A.: Dynamic Source Routing in Ad Hoc Wireless Networks, *Mobile Computing*, Imielinski, T. and Korth, H. (Eds.), Kluwer Academic Publishers (1996).
- 15) Perkins, C.E.: Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers, *Proc. ACM SIGCOMM'94* (1994).
- 16) Perkins, C.E., Royer, E.M. and Das, S.R.: Ad Hoc On-Demand Distance Vector (AODV) Routing, Internet-Draft, draft-ietf-manet-aodv-08.txt (2001.3.2).

(平成 14 年 3 月 25 日受付)

(平成 14 年 10 月 7 日採録)



鈴木 貴也 (正会員)

昭和 52 年生。平成 12 年静岡大学情報学部情報科学科卒業。平成 14 年同大学大学院情報学研究科情報学専攻修士課程修了。同年日本電気株式会社入社。IMS アプリケーション

の開発に従事。



石原 進 (正会員)

昭和 47 年生。平成 6 年名古屋大学工学部電気学科卒業。平成 11 年同大学大学院工学研究科博士後期課程電子情報学専攻修了。平成 10 年度日本学術振興会特別研究員。平成 11 年静岡大学情報学部情報科学科助手。平成 13 年同大学工学部システム工学科助教授。博士 (工学)。モバイルコンピューティング, 無線環境用 TCP/IP に関する研究に従事。電子情報通信学会, ACM 各会員。平成 9 年度電気通信普及財団賞。



水野 忠則 (正会員)

昭和 20 年生。昭和 43 年名古屋工業大学経営工学科卒業。同年三菱電機株式会社入社。平成 5 年静岡大学工学部情報知識工学科教授, 現在, 情報学部情報科学科教授。工学博士。情報ネットワーク, モバイルコンピューティング, 放送コンピューティングに関する研究に従事。著書としては「プロトコル言語」(カットシステム)、「コンピュータネットワーク概論」(ピアソン・エデュケーション)等がある。電子情報通信学会, IEEE, ACM 各会員。当会フェロー。