

ディペンダブルアドホックネットワーク

瀬崎 薫
生産技術研究所

概要

デバイス技術の飛躍的な進歩と小型化・省電力化に伴い、身の回りに無線デバイスが偏在するユビキタスコンピューティング環境が現実のものとなりつつある。ネットワークへのアクセス可能性を最大化・利便化し、真の意味で「ユビキタス」なネットワーク環境を提供するためにアドホックネットワークを利用することが期待されている。しかしながらアドホックネットワークには、信頼性、スケーラビリティ、ネットワーク侵入の容易性等、様々な問題が山積しており、大域でディペンダブルなネットワークとは程遠いのが現状である。本課題ではこれらの諸問題を解決すると共に、屋内外を問わずシームレスに構築されたアドホックネットワーク上で構築されるサービス構造についての研究を行う。

1 はじめに

デバイス技術の飛躍的な進歩と小型化・省電力化に伴い、身の回りに無線デバイスが偏在するユビキタスコンピューティング環境が現実のものとなりつつある。ネットワークへのアクセス可能性を最大化・利便化し、真の意味で「ユビキタス」なネットワーク環境を提供するためにアドホックネットワークを利用することが期待されている。しかしながら現状のアドホックネットワークに対しては、信頼性、スケーラビリティ、ネットワーク侵入の容易性等、様々な問題があることが指摘されており、これらを解決しなければ大域でディペンダブルなネットワークとはならず、本研究ではこれらの諸問題を解決することに取り組んでいる。このようなアドホックネットワークが実現されれば、PDA等の小型携帯端末であり、GPS等の位置情報デバイスと組み合わせて旅行者ガイドシステムを始めとする位置依存型サービスを提供することが期待できる。位置情報デバイスとして、GPSを始めとする各種の物が利用可能であるが、種類のデバイスで屋内・屋外全域をカバーすることは不可能であるため、信頼性の高い位置情報依存型サービスを提供するためには、ユーザの環境によって異なるデバイスを用いる必要がある。そこで本年度は、このように異なる位置同定デバイスをハンドオーバーしつつ、サービスを実現するためのシームレス位置情報適応サービスの枠組みと経路制御手法について検討を行ったのでこれについて報告する。

2 シームレス位置情報適応型サービス

シームレス位置情報適応型サービスでは、既存のサービスのアーキテクチャに加えて、図1に示すように、新たに二つの層が必要となる。

まず、様々な位置同定デバイスから位置情報を収集するため、それらの情報のフォーマットの統一や矛盾解決を行うミドルウェアが必須となる。ミドルウェアは、様々な位置同定デバイスから情報を収集する Reception Layer、受信した情報のフォーマットを統一する Abstraction Layer、統一された情報に関してセンサー間のオーバーラップの処理や不一致の整合性をとるなどの処理を行う Fusion Layer の三つのサブレイヤから成る。

ミドルウェアによってより高精度となった位置情報はさらに上位層に位置するアドレッシング層によってアドレスに変換される。想定しているサービスは位置情報を基に実現されるため、アドレッシングもデバイスの物理的な位置情報に基づいて実現される。

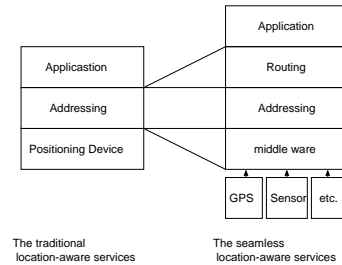


図 1: 既存研究との差異

次に経路制御であるが、サービスをシームレス化することにより、既存のサービスに比べて物理的にカバー可能な範囲が広範囲にわたるため、メッセージはまた、必然的にマルチホップ化する。従って経路制御はサービスシームレス化において、最も核となる技術である。そこで今回はこれらの問題点の内、経路制御に焦点をあて、研究を行った。構成は以下の通りである。まず3章において、地理情報を用いた無線マルチホップ経路制御手法の動向を概説する。4章では、障害物が存在した場合の問題点を述べ、それらの問題点を解決した手法を提案する。5章において計算機シミュレーションによって評価を行い、最後に6章でまとめと今後の課題を述べる。

3 地理情報を用いた無線マルチホップ経路制御手法

屋内系を含むシームレスな位置情報適応型サービスを考慮した場合、建物内等3次元空間で実行可能であることが必須条件となる。しかし、従来手法である GPSR[2] は高度情報を無視可能であるという前提条件の下、3次元空間を近似2次元平面上に投影し、その平面上で Perimeter Forwarding を設計している。しかし、現実には高度情報を無視できない場合も多数考えられ、このような仮定は必ずしも成立しない。このため GPSR では三次元空間内では迂回経路を構築することができず、経路構築の成功確率が低下してしまうという問題点があった。そこで、楕円体を利用することによって3次元空間でも迂回経路構築が可能な経路制御手法を提案し、さらに信頼性を高めた経路構築を実現した [1]。

楕円体を用いた手法では送信者と目的地ノードを焦点とする楕円体群を基に隣接ノードを選択する。楕円体群は無数に考えられるが、各隣接ノードを通る楕円体はそれぞれに一つ決定され、結果として隣接ノード数の楕円体を考えることができる。送信者はそれらの楕円体群の中から最も直線に近い形状の楕円体を選択し、その楕円体上に存在する隣接ノードを次ホップノードとして選択する。結果として構築される経路は直線に近い形状の経路となり、効率の良い経路構築が実現される。具体的には次に示す例のようにして経路構築を行う。

図2において、送信者Sは目的地ノードDまでの経路の構築を試みる。今、Sの隣接ノードがノードAとノードBであるとすると、Sはどちらかのノードを次ホップノードとして選択しなければならない。そのため、Sは決定するための値として、次式の値を計算し、隣接ノード V_i の値とする。

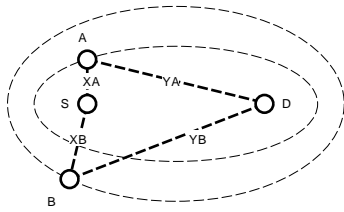


図 2: 楕円体を利用した経路制御

$$V_i = X_i + Y_i$$

ここで X_i は送信者と隣接ノード i との距離、また Y_i は隣接ノード i と目的地ノードとの距離である。この値を各隣接ノード全てに関して計算し、それぞれを比較する。 V が低い値をとる隣接ノードは即ちより直線に近い形状の楕円体上に存在するノードであり、この場合、ノード A が目的に沿った次ホップノードとなる。

4 障害物回避

地理情報を用いた経路制御手法の利点の内、大きなものとしてネットワークが展開される地理環境を考慮できるという点が挙げられる。MANET のような各端末が同時にルータの役割を担うネットワークでは、地理的な環境がネットワーク構成に大きな影響を与える。例えば、図 3 のような環境を想定する。

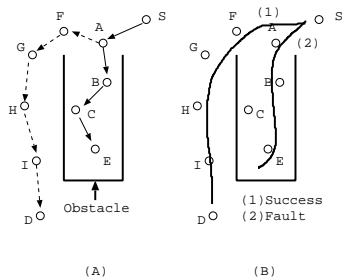


図 3: 障害物のある環境における経路制御

図 3 中ノード B, C, E は障害物（無線電波を遮蔽する）に囲まれている。前節で紹介した楕円体を用いた手法では目的地に向かってより直線に近い軌跡で経路を構築するためノード S からノード D までの経路を構築する場合には、経路 S-A-B-C-E を構築しようとする。しかし、ノード E とノード D は確かに物理的には近い端末であるが、ノード間に障害物があるため直接通信ができず、経路構築失敗となる。この場合、他にも図 3-(B) に示すように物理的な迂回経路（経路 A-F-G-H-I-D）をとることによって D まで経路構築が可能である。しかしながら、純粋にホップバイホップベースで経路構築を行うと、経路の全体像が把握できないため、先に示した例のように経路構築失敗となる。

このような事態を避けるためには、フラッディングベースによって存在する無線リンクを総当りで経路を検索する方法

や、あらかじめ障害物を回避するような地理的な軌跡を設定しておき、それに沿った経路を構築する手法などが考えられる。前者のフラッディングベースによる方式は確かに障害物などを加味する必要はなくなるが、反面膨大な制御パケットを必要とする。そこで、ホップバイホップベースの手法である楕円体を用いた手法に障害物を回避する軌跡情報を与えることにより、この問題点に対処する手法について検討する。

軌跡を利用した経路制御手法

ここではあらかじめ設定された障害物を回避する軌跡を用いる。軌跡を設定し、その軌跡に沿うような経路を構築することにより、障害物を回避した経路が構築される。

参照点を用いた障害物回避手法

本節では、新たに「参照点」を設定することにより、この問題に対処する。ここで「参照点」とはただの地点であり、そこに何かの端末類があるわけではない。送信者はあらかじめ GIS エンジンなどにより、目的地までの障害物を回避した軌跡を知っているものとする。まず経路要求前に軌跡上に複数の RP を設定する。設定された RP は経路要求パケットに記載され、各中継ノードは RP を仮の目的地とみなして経路制御を行う。RP は複数設定されているため、各中継ノードは最もシーケンスの早い RP から順に目的地に設定する。結果、構築された経路は最初に設定された軌跡に沿ったものとなり、障害物の回避が達成される。

先に挙げた例によって上記の流れを示す。

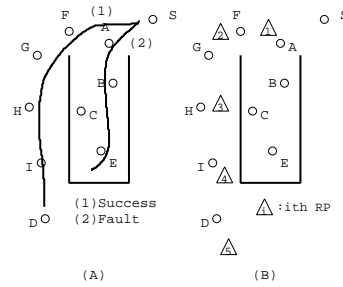


図 4: 参照点の設定

送信者 S はあらかじめ GIS エンジンなどから図 4-(A) 中 (1) の軌跡を取得しているものとする。次に制御パケットに軌跡情報を記載するが、その際にここでは図 4-(B) に示すように RP 1~5 を設定する。経路要求パケットにはこれら RP の位置全てが記載されており、それぞれ順に目的地として設定される。つまり送信者 S はまず RP1 を目的地として設定し、経路要求パケットを送信する。RP1 を目的地とすると、楕円体を用いて次ホップノードを選択すると、まずノード A が選択される。ノード A は自身の隣接ノードと RP1 との距離、及び自身と RP1 との距離を比較する。この場合、ノード A の隣接ノードはノード B 及びノード F であるが、いずれもノード A に比べ RP1 から物理的に遠いため、ノード A は既に RP1 に最も近づいたと判断し、RP を 1 から 2 に変更する。ノード A は次に RP2 を目的地として経路構築を試みるため、再び楕円体を用い、次ホップノードとしてノード F を選択する。ノード F の隣接ノードがノード G とすると、ノード G はノード F に比べ RP2 より近いので、RP の切り替えを行わず、そのま

まパケットはノードGに中継される。ノードGの隣接ノード内ではノードGよりもRP2に近いノードがないため、RPは2から3に切り替えられ、以後、同様にRP4まで中継され、最終的に目的地ノードDに到達する。

5 シミュレーションによる評価

前章までに提案した手法について計算機シミュレーションを行った。以下にその概要と結果、考察を示す。

シミュレーション概要及び結果

空間としては、図5に示すように、仮想的な空間を用意した。

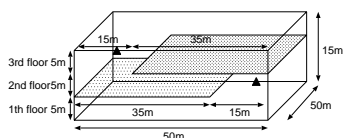


図 5: 仮想3次元空間 (その1)

空間は縦 50 m横 50 m高さ 15 mであり、高さ 5 mの部分と 15 mの部分に障害物として板状の電波遮蔽物を設置した。板は縦 50 m横 35 mであり、設置した箇所は図の通りである。

このような空間内にランダムにノードを配置し、ある任意の一组のノードが経路構築を開始する。ただし、今回は障害物の影響を見ることが目的であるため、ソースノードを1階に、目的地ノードを3階に配置し、直接に通信が行えないようにした。それ以外のノードは空間内にランダムに配置される。

シミュレーションでは一組のノード間で経路構築が成功したか否かを空間内に存在する端末数を変化させることによって評価する。比較する手法としてはまずフラッディングで経路を構築する場合、次に楕円体を用いた手法、さらにRPを設定することによって障害物を回避する軌跡を設定した上で楕円体を用いる手法である。ここでRPは空間内の障害物情報が既知であるという前提のもと、あらかじめ設定した。座標は(42.5,25,5)と(7.5,25,10)の2点である。これらは図5における▲によって示している。これらを進ることによって障害物を回避した経路構築を実現する。図6に、RPを設定した場合に改善された例を示す。

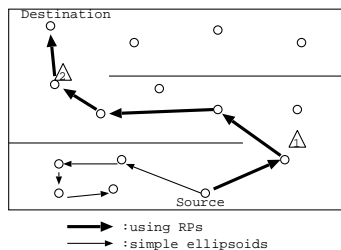


図 6: RP による経路制御の改善

RPを設定せずに経路制御を行った場合には、図中の細線のように経路要求パケットが袋小路に入ってしまい、ループフリー機能によって制御パケットが破棄されてしまう(図中”simple

ellipsoid”)。しかし、RPを順に追うことにより袋小路に入ることなく目的地までの経路制御が可能となる(図中”using RPs”)。

図7にシミュレーション結果を示す。

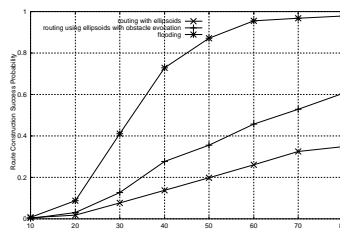


図 7: 経路構築成功確立

図7より、やはりフラッディングによる経路制御が最も成功確立が高い。これは存在するリンク全てに対して経路探索を行うため当然の結果と言える。しかし、フラッディングは以前検証した通り、非常に多くの制御パケットを必要とし、アドホックネットワークに適しているとは言えない[1]。

次にホップバイホップベースである楕円体を用いた手法と比較する。グラフを見ると現在まで確認されている範囲では明らかにRPを設定することによって経路制御を行った場合よりも成功確立が向上していることが見てとれる。しかしながら、今回のような仮想的な3階立ての空間では、送信者と目的地ノードが直線的な経路によって通信可能である場合が5%弱発生した。この場合、純粋に障害物による影響を見たことにならないため、次に障害物の形状を変化させてシミュレーションを行った。設定した仮想空間を図8に示す。

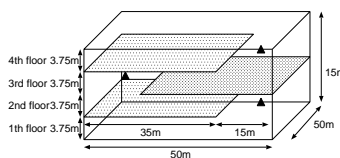


図 8: 仮想3次元空間 (その2)

図8では、図5と同一の空間内に今度は縦 50m、横 35m の遮蔽物を3枚設置した。ここにおいて送信者を1階に、目的地ノードを4階に設置することにより、直線的に通信することは不可能となり、純粋に障害物の影響を測定することが可能となる。この場合、RPは(42.5,25.0,3.75)、(7.5,25.0,7.5)、(42.5,25.0,11.25)の三点となる(図8中”▲”)。

以降では、障害物の形状を変化させた場合の経路構築の成功確率の変化を測定結果を示す。結果を図9及び図10に示す。

障害物を増やすことによって、楕円体を用いた手法及びRPを用いた手法のそれぞれについて成功確率が低下していることが分かる。その低下率もRPを用いた場合が81.22%、用いない場合が83.5544%となっており、若干ではあるが、低下率を抑えることが可能となっている。

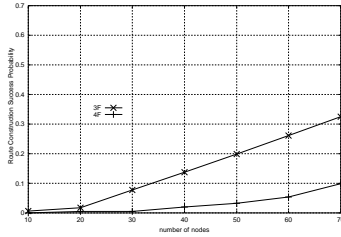


図 9: 経路構築成功確率 (参照点を用いない場合)

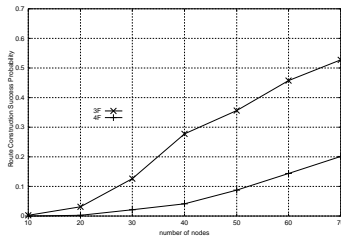


図 10: 経路構築成功確率 (参照点を用いた場合)

6 今後の課題

経路長の最適化

今回は RP の切替えに関して、自分よりも RP に近い隣接ノードが存在しない場合に次の RP に目的地を変更する手法をとったが、このような場合、RP 付近で無駄なホップ数をかさいでしまう (図 11)。

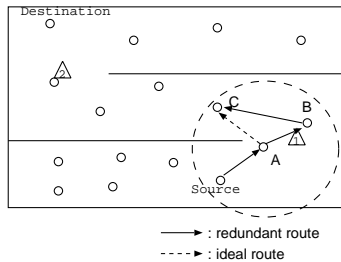


図 11: 冗長な経路

行ったシミュレーション中にもしばしば見られた現象であるが、図 11 中ノード A が隣接ノードであるノード B とノード C から次ホップノードを選択する際に、ノード B の方がノード A よりも RP1 に近いため、RP の切替えを行わず、次ホップノードとしてノード B を選択する。パケットを受信したノード B は自身よりも RP1 に近い隣接ノードが存在しないため、RP を 1 から 2 に切替える。よってノード B の次ホップノードは C となり、図に示すように冗長な経路が構築される (図 11 中、実線)。しかしノード C はノード A の隣接ノードでもあり、仮に

ノード A において RP の切替えが実行されておれば、次ホップノードとして選択されより効率のよい経路が構築されていたことになる (図 11 中、破線)。

この点については、さらに検討する必要がある。現在、RP を切替える手法として、自身が RP よりも一定の距離内に入った場合に、RP を切替える手法を考えている。この一定の距離は判定する端末 (その時の送信者) の通信半径でよいのではないかと考えられる。図 11 の場合、RP1 が A の通信半径内であるため、A は RP を 2 に切替え、次ホップノードとして C を選択する。本手法についても、シミュレーションを行う予定である。

RP 数の最適化

ここでは、RP は既知のものとして、あらかじめ与えてシミュレーションを行ったが、RP の数はパフォーマンスに大きく影響を与える。

シミュレーションによっても明らかになったが、参照点を導入せずとも、楕円体を用いることによってある程度の障害物は回避が可能である。よって、たとえ障害物が存在しても RP を設定せずとも問題がない場合も多く考えられる (図 12)。

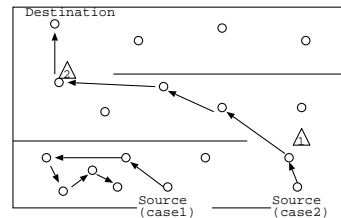


図 12: RP 数の最適化に関する検討

図 12 中、送信者がケース 1 の場所にいる場合には、まず RP1 を目的地としなければ袋小路に入り込んでしまい経路構築失敗となるが、ケース 2 の場所にいた場合には RP を設定せずに単純に楕円体を用いることのみで経路構築が可能となる。そのような場合に冗長な RP を設定することは制御パケットの肥大化を招き、オーバーヘッドの原因となり得る。そのため、どのような場合には RP を設定せずとも経路構築が可能であるのかを明らかにし、参照点の設定の際に考慮しなければならない。

以上より、今後の課題としては、さらに多様な形状の障害物を設定しつつシミュレーションを行い、この問題を明らかにすることが挙げられる。恐らく直進方向からどの程度の角度で障害物を回避すればよいかによって、その点に RP を設定するかどうか決定されるため、その相関性が確認できれば RP の数の最適化が可能であると考えられる。今後はその相関性の確認及び RP 数の最適化アルゴリズムを検討する予定である。

参考文献

- [1] 山崎 浩輔, 瀬崎 薫
“位置情報適応型サービスに向けた地理的経路制御手法の提案” 電子情報通信学会和文論文誌 VOL.J85-B NO.12 p2129-2137
- [2] Brand Karp and H.T.Kung
”GPSR:Greedy Perimeter Stateless Routing for Wireless Networks” Mobicom 2000