

自律分散型ネットワークのための基盤技術

瀬崎 薫
生産技術研究所

1 概要

従来のインフラストラクチャ型のネットワークを補完代替するための、インフラストラクチャレスなネットワークとして、アドホックネットワークが注目されている。本研究では、アドホックネットワークの性能改善のための諸技術の研究開発を行っている。また、これら技術は無線センサネットワークの基盤技術と共通する物も多いので、これについても合わせて検討を行っている。本年度は、特にゲートウェイ発見方式、省電力化、新しいアドレス体系等の種々の技法について検討を行った。

2 はじめに

モバイルアドホックネットワーク (Mobile Ad Hoc Network: MANET) は、従来のインフラストラクチャ型のネットワークのサービスエリアの補完、あるいは大規模災害時等においては、これを代替する手法として注目を集めている。しかしながら、MANET は基本的には、自律分散的制御を行う上、無線リンクを用いることから、その安定的な運用に向けて解決すべき課題が山積しているのが現状である。当研究室においては、課題解決のための精力的な研究を行ってきた。また、これら MANET の要素技術は無線センサネットワークの研究領域とオーバーラップするため、両者の研究を並行して行っている。本年度の主な研究成果は、

1. 物理的位置情報に基づく新しいアドレス体系である時空間アドレス (Spatio Temporal Address) と、そのアドレス解決手法の提案 [1, 2]
2. MANET の省電力型 MAC protocol [3]
3. 無線センサネットワークの階層的データ集約手法 [4]
4. 歩行者モビリティモデルとその MANET への適用 [5]

5. 適応的ゲートウェイ広告手法

などである。本稿では、このうち主に 1. について紹介する。

3 時空間アドレス：STA

本章では、STA の概要を紹介すると共に、その衝突検知手法についても概説する。

3.1 概要

STA は各端末の時空間情報を基本情報として生成されるため、以下に代表されるような特徴を持つ。

1. 完全な一意性

物理的な物体は同位置同時刻に存在することはないため、自然各物体が保持する時空間情報もまた重複することはない。

2. 自律分散性

時空間情報は各端末の保持する位置同定デバイス及び時計によって得られる情報であるため、取得時に他の端末との通信は要求されず、STA は完全に自律分散的に算出が可能である。

3. 経路制御への利用

現在まで端末の位置情報を利用した経路制御手法が多数提案されており、それらを用いることにより経路制御可能なアドレスとして機能する

以上の特性を鑑みると、アドレス割り当てに際してネットワーク全体に与える負荷が低いことから、ネットワークを利用する時にのみアドレスを使用するといった使い方も可能である。

3.2 STA における衝突検知

以上のように時空間情報をアドレスの基本情報として用いることにより、様々な利点を楽しむこと

が可能である．しかしながらここで用いられる時空間情報はある測定デバイスから得られる情報であり，そこには必ず一定の粒度が存在する．その場合には同一の時空間情報を持つ端末が複数存在することが考えられ，上記(1)の利点が保証されない．よって STA を用いた場合にも衝突検知を行なう必要があるが，ここで時空間情報の局所性を利用することにより，効率的な衝突検知が可能である(図1)．

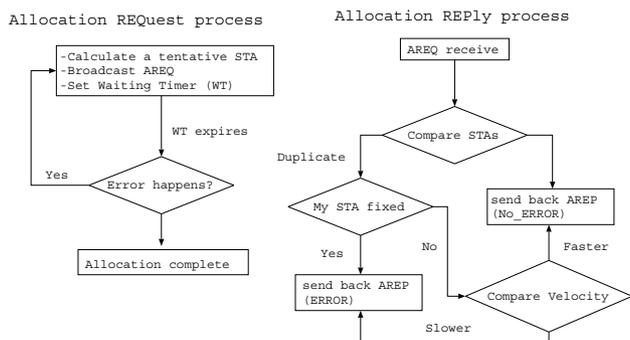


図 1: 衝突検知のフローチャート

STA を取得する端末(スタータ)はその時点における位置情報及び時刻情報を算出し，初期 STA を決定する．次に初期 STA の一意性を確認するために周辺端末に対して割り当て要求(Allocation REQuest, 以下 AREQ)をブロードキャストする．時空間情報はある程度粒度が存在した場合においても，それ以上の距離が各端末に存在すれば同一の値をとることがないため(時空間情報の局所性)，衝突検知は周辺端末との間でのみ実行すればよく，効率的な衝突検知が実現できる．AREQ をブロードキャストした後は待ち時間を設定し，待ち状態に入る．待ち時間が終了した時点で改めて衝突の存否を検証し，衝突がない場合には初期 STA を設定し，動作を完了する．また衝突が発生していた場合にはその時点で再度現在地及び現時刻情報から STA を算出し，同様の動作を繰り返す．

AREQ を受信し衝突検知を依頼された端末(リゾルバ)は，現在設定されている自身の STA との比較を行ない，重複していない場合には即座にエラービットを設定しない割り当て応答(Allocation REPLY, 以下 AREP)を返信する．重複していた場合には次の段階に進み，自身の STA が既に決定されたものか否かを検証する．既に決定された STA を保持していた場合には完全に優先権があるため，エラービット

を設定した AREP を返信する．STA が未確定の場合には，さらに次の段階に進み，スタータとの速度の比較を行なう．ここで，より高速で移動する端末は新規 STA をより迅速に取得することができるため，低速度で移動する端末に STA 割り当ての優先権を与える．つまり，リゾルバがより低速度で移動している場合にはリゾルバに STA 割り当ての優先権を，逆にスタータがより低速度で移動している場合にはスタータに優先権を与える．以上の機構により，より迅速な STA の収束が実現可能となる [1]．

3.3 時空間情報の粒度

STA を用いることにより自律分散的なアドレス割り当てが実現可能であるが，本節では基本情報となる時空間情報について考察する．まず，STA にもやはりアドレス長という制限が存在するため，自然，そこに含まれる時空間情報にも粒度という制限が存在する．時空間情報の粒度に制限が存在した場合，その粒度にも依るが，同一の時空間情報を保持する端末が複数存在する可能性がある．さらに，時空間情報を取得するために用いられるデバイスにも自ずと限界があり，そもそもが情報として一定の粒度が存在する．よって，STA が無限に詳細であることはあり得ず，同一の時空間情報が複数の端末で保持されることは十分に発生し得る．

そこで低粒度の時空間情報を基に STA を生成した場合に，その一意性にどのような影響が生じるかについて検討を行なった [2]．位置情報の粒度を 1.0m, 10.0m, 100.0m, 時刻情報の粒度を 1.0 秒, 10.0 秒とし，これらを変化させることによって発生する重複 STA の総数を計測した．シミュレータとして NS-2[6]を用い，端末数は 30, 50, 70, 100, それぞれをランダムに配置し Obstacle Mobility Model[7] に従って移動させる．ある時刻において空間に存在する全端末が一斉に STA を取得する．

結果，やはり位置情報の粒度が 100.0m, 時刻情報の粒度が 10.0 秒の場合に最も多数の重複 STA が発生した．また傾向を詳細に検討した結果，時刻情報に比べて位置情報の粒度が STA の一意性に大きな影響を与えることが確認された．さらにスタータの待ち時間をより長く設定することにより，重複 STA 数を大幅に削減可能であることも確認された．これ

はスタータの待ち時間内に適切な重複検知が不可能であったことが一因であると考えられ、その対処方として一度 STA の一意性を確認したと考えられる。スタータが STA を設定後、その STA を再度周辺端末にブロードキャストして報告する STA レポート機能を導入した。結果、さらに大幅な重複 STA の削減が確認された。また、一度設定した STA を保持し続けることによる重複の発生も確認されたため、各端末の移動度に応じた STA 更新機能も導入し、最終的にネットワーク全体で迅速に一意的な STA を割り当て可能であることも確認した。

しかしながら上記の STA レポート機能は重複 STA が発生した場合に近隣端末に STA の再設定を強請する機能であり、そこには冗長な遅延や制御パケットの発生などが予想される。そこで次章では STA の収束時間及び発生した AREQ 数に関する計算機シミュレーション及び検討を行なう。

4 衝突検知における遅延及び制御パケットの発生

本章では、STA の重複検知に STA レポート機能を付加した場合における STA の収束時間及び発生 AREQ 数の検討を行なう。シミュレーション環境は前章に挙げたものと同様とし、位置情報の粒度は重複 STA が最も多く観測された 100.0m とした。まず図 2 に時刻情報の粒度が 10.0 秒であった場合の割り当て収束時間と発行 AREQ 数の分布を示す。

図 2 より、端末数がいずれの場合にも待ち時間が 1.0 秒の場合に発生した AREQ 数が少数であることが確認できる。待ち時間が長いということは、スタータにおけるタイムアウトまでの時間が長いことを意味し、そのため発生 AREQ 数が削減できたと考えられる。また前章で述べたように待ち時間の延長によって重複 STA 数が大幅に削減できることからスタータが AREQ を発行する必要性も大幅に削減されたと考えられる。次に収束時間について考察する。待ち時間が 0.1 秒と 1.0 秒を比較した場合、待ち時間が 0.1 秒の方がより多くのタイムアウトを迎えるため、迅速な STA の収束が期待できる。図 2 からその傾向は見て取れるが、詳細にその収束時間を検討してみた場合、表 1 のようになり、端末数 100 の場合を除き、大きな差異は見られなかった。

端末数	割り当て収束時間	発行 AREQ 数
30 (WT:0.1s)	10.146528s	111
30 (WT:1.0s)	11.01484s	21
50 (WT:0.1s)	20.196297s	173
50 (WT:1.0s)	20.001184s	25
70 (WT:0.1s)	20.159875s	146
70 (WT:1.0s)	25.035602s	25
100 (WT:0.1s)	4.508358s	74
100 (WT:1.0s)	30.001424s	31

表 1: 割り当て収束時間と発行 AREQ 数 (時刻情報の粒度: 10 秒)

端末数が 100 の場合には [2] から待ち時間 0.1 秒の場合に非常に多数の重複 STA が発生することが確認されており、反面待ち時間 1.0 秒の場合には大幅に重複 STA を削減可能であることからより大きく STA レポートによる性能向上が測られたと考えられる。

次に時刻情報の粒度を 1.0 秒とより詳細にした場合の同様の計算機シミュレーションの結果を表 2 に示す。

端末数	割り当て収束時間	発行 AREQ 数
30 (WT:0.1s)	3.101264s	31
30 (WT:1.0s)	4.0s	5
50 (WT:0.1s)	2.3s	37
50 (WT:1.0s)	5.0s	7
70 (WT:0.1s)	4.1s	41
70 (WT:1.0s)	5.0s	8
100 (WT:0.1s)	4.329051s	48
100 (WT:1.0s)	6.0s	6

表 2: 割り当て収束時間と発行 AREQ 数 (時刻情報の粒度: 1 秒)

図 2, 表 1, 表 2 より、全般的に収束時間及び発行 AREQ 数の双方が大幅に削減できたことが分かる。また、待ち時間が 0.1 秒の場合と 1.0 秒の場合における全体的な収束時間の差異は時刻情報の粒度が 10.0 秒の場合に比べてさらに減少しており、時刻情報の粒度が 1.0 秒の場合には待ち時間が 1.0 秒であれば、迅速かつ効率的な重複検知が可能であることが確認できる。

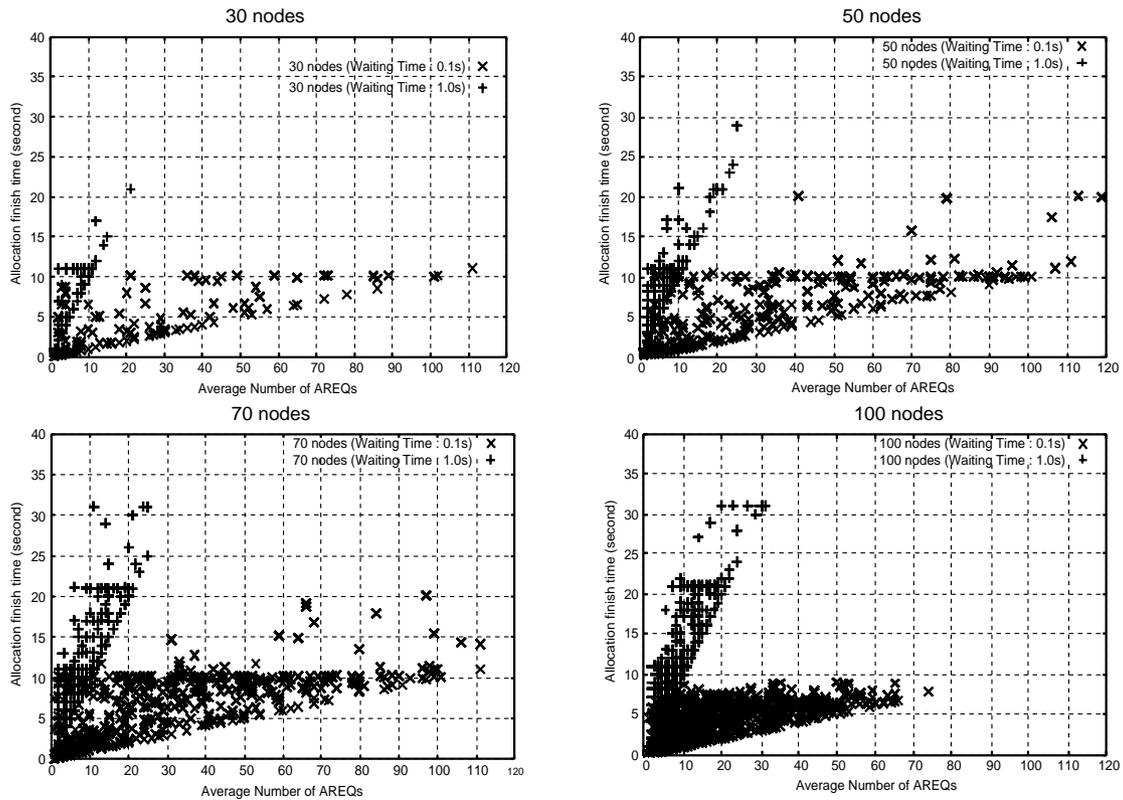


図 2: STA 収束時間及び AREQ 発行回数 (時間情報の粒度: 10 秒)

5 まとめと今後の課題

本稿では主に各端末の時空間情報を基にして生成される時空間アドレス (STA) を紹介すると共に, 時空間情報の粒度が低下した場合にも一意性を確保する手法についても紹介した. またさらにそこで発生が予想される遅延や冗長な制御パケット量についても計算機シミュレーションによって詳細に検討を行った.

今後の課題としては, 既存の IP アドレスを STA に置き換えた場合の通信の検討が考えられる. STA は時々刻々と各端末の移動度に応じて更新される必要があり, そのため既に通信路を確立した端末間では頻りに通信が切断される恐れがある. この点について今後詳細に検討を行なう予定である.

参考文献

- [1] 山崎, 瀬崎: “時空間での唯一性を利用したアドレスリングに関する一検討”, 信学技報 NS (2004).
- [2] 山崎, 瀬崎: “時空間アドレスの粒度に関する検

討”, 信学技報 NS (2004).

- [3] S. Takeuchi, K. Yamazaki, K. Sezaki and Y. Yasuda: “A Proposal of Battery Cost Routing in Consideration of Transmission Power”, Globecom 2004 (2004).
- [4] M. Sekine and K. Sezaki: “Hierarchical Aggregation of Distributed Data for Sensor Networks”, IEEE TENCON 2004 (2004).
- [5] W. Creixell and K. Sezaki: “Mobility Prediction Algorithm for Mobile Ad Hoc Networks using Pedestrian Trajectory Data”, IEEE TENCON 2004 (2004).
- [6] “The network Simulator - ns-2”, <http://www.isi.edu/nsnam/ns>.
- [7] A. Jardosh, E. M. Belding-Royer, K. C. Almeroth and S. Suri: “Towards Realistic Mobility Models For Mobile Ad hoc Networks”, MobiCom'03 (2003).