

インフラ管理型アドホックネットワークルーティングの スケーラビリティ評価

小林 基成 鈴木 俊博 カーン アシック 森田 正範

(株)NTT ドコモ ネットワーク研究所 〒239-8536 神奈川県横須賀市光の丘 3-5

E-mail: {kobayashi, toshi, ashiq-khan, morita}@netlab.nttdocomo.co.jp

あらまし 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では、ルーティングに必要な制御パケット量(総データ量)が、端末数の2乗に比例して増加するため、規模の大きいネットワークを構築することができない。そこで我々は、アドホックネットワークのルーティング計算はセルラなどのインフラネットワークで行い、端末に通知されたルーティング情報をもとに、アドホックネットワーク上でデータ伝送する方式を検討している。本稿では、提案方式において、アドホックネットワークでの制御パケット量と、インフラネットワーク上のルート管理サーバでの計算時間を評価し、提案方式における収容可能端末数を示す。

キーワード アドホックネットワーク、ルーティング、スケーラビリティ

Scalability Evaluation of Infrastructure Controlled Ad-hoc Network Routing

Motonari KOBAYASHI Toshihiro SUZUKI Ashiq KHAN and Masanori MORITA

Network Laboratories, NTT DoCoMo 3-5 Hikarinooka, Yokohama, Kanagawa, 239-8536 Japan

E-mail: {kobayashi, toshi, ashiq-khan, morita}@netlab.nttdocomo.co.jp

Abstract In Ad-hoc Network routing protocols, the amount of control packets for routing increases in proportion to the second power of the number of terminals. Therefore, it is difficult to construct large scale of Ad-hoc Networks. To solve this problem, we study the routing method where data traffic is transmitted on Ad-hoc Networks and routing information is managed in an infrastructure network. In this paper, we evaluate the amount of control packets and route calculation time of our proposed method. We derive the maximum number of terminals that can be accommodated in the network when using our proposed routing method.

Keyword Ad-hoc Networks, Routing, Scalability

1. はじめに

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は、基本的に制御パケットをフラッディングすることで情報交換を行い、ルート制御を行っている。AODV[1]においては Route Request (RREQ)の送信にフラッディングが用いられている。RREQをフラッディングするためには、ネットワーク中の端末の数だけの中継を必要とするため、ネットワーク中に送信されるパケット量は端末数に比例する[2]。さらに、端末数に比例した頻度で通信要求が発生するため、RREQパケットの量は端末数の2乗に比例($O(n^2)$: 端末数)することとなる。よって、端末数が多くなると、通信帯域の大部分がRREQによって占められてしまうため、大規模なアドホックネットワークを構築することは難しい。OLSR[3]では、TCメッセージにフラッディングが用いられているが、multipoint relays(MPR)を用いてTCメッセージを中継する端末を限定することで、ネットワーク中に送信されるパケット量を削減している。しかし、TCパケットの量は基本的に $O(n^2)$ であることに変わり

は無く、アドホックネットワークの大規模化には限界がある。アドホックネットワークを階層化することで制御パケット量を $O(n \log^2 n)$ [4]にすることが可能であることが知られているが、やはり端末数が増加する以上に制御パケット量が増加してしまう。

アドホックネットワークをスケーラブルに拡張するためには、ネットワークに参加する端末の増加によって、他の端末が送信する制御パケットが増加しないことが要求される。つまり、ネットワーク全体の制御パケット量が $O(n)$ であることが必要である。CAMA[5]、CBM [6]では、インフラネットワークでルーティングを行うことで、この条件を満たしている。これらの方式では、各端末は Hello パケットにより、隣接する端末とのリンクの存在を確認する。そして、インフラネットワークに存在するサーバにリンクの情報を登録する。サーバでは、登録されたリンク情報を用いて、任意の端末間のルートを計算する。ここで用いられている制御パケットは、Hello パケット及びリンク情報の登録パケットであり、これらの制御パケットはマル

チホップすることはない。そのため、各端末が送信する制御パケット量は他の端末の存在に関係なく、ネットワーク全体の制御パケット量は $O(n)$ にすることができる。さらに我々の提案する MARIONet[7][8]においては、リンク情報の登録パケットを削減することで、端末-インフラネットワーク間の無線及びサーバへの負荷を軽減している。

本稿では、提案する MARIONet のスケーラビリティの評価として、アドホックネットワークとして構成可能な最大端末数を算出する。MARIONet などインフラネットワークでルート計算を行うシステムにおいては、アドホックネットワーク内の制御パケット量に加えて、サーバのルート計算負荷がアドホックネットワーク拡大のためのボトルネックとなる。そこで、ボトルネックとなる制御パケット量、ルート計算負荷を算出することで、アドホックネットワークを構成可能な最大端末数を算出する。

2. MARIONet

2.1. MARIONet の概要

図 1 に MARIONet におけるネットワーク構成を示す。MARIONet は、ルーティング情報をコアネットワークで管理、計算し、データ伝送はアドホックネットワークを使用する。インフラネットワーク上のサーバは、アドホックネットワークのトポロジを管理し、通信要求に応じてルートを計算し、端末に通知する。

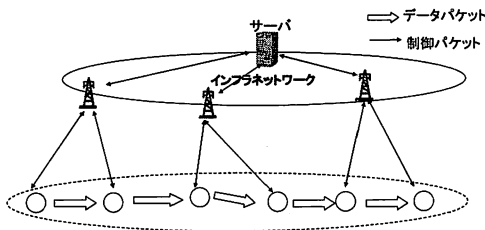


図 1 MARIONet

ルーティングの処理手順を図 2 に示す。

定期的な処理

- 端末は Hello パケットを交換し、隣接端末との間のリンクの存在を確認する (Hello パケット交換)
- 端末は Hello パケットで得たリンク情報をインフラネットワークに登録する (リンク情報登録)

通信開始時の処理

- 通信要求端末は通信要求をインフラネットワークに送信する (通信要求)
- ネットワークは通信要求に応じて登録されているトポロジ情報を元にルートを計算する (ルート計算)
- インフラネットワークは計算したルートを端末に通知する (ルート通知)

- ルート情報を得た端末はそのルート情報に従ってデータを送信する (データ送信)

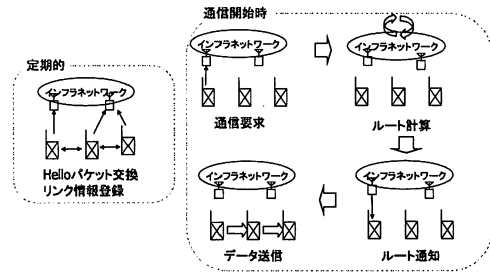


図 2 MARIONet の処理

- 登録端末
- 非登録端末

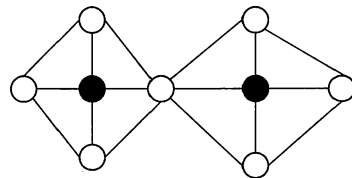


図 3 登録端末、非登録端末によるトポロジ構成

2.2. リンク情報の登録

MARIONet では、アドホックネットワークを構成する端末間のリンクの存在 (リンク情報) をインフラネットワークに登録することで、インフラネットワークでのルート計算を可能にしている。しかし全ての端末が自身の持つリンク情報を送信すると、同一のリンクが複数の端末によって登録されるため、パケット量の観点から望ましくない。そこで MARIONet では、リンク情報を登録する端末を限定することで、アドホックネットワーク中の全リンクを重複することなく登録している。具体的には、端末情報をインフラネットワークに登録を行う「登録端末」と登録を行わない「非登録端末」という 2 つの状態を定義する。各端末は、トポロジ構成により、「登録端末」または「非登録端末」となり、「非登録端末」の場合にはリンク情報を送信しない。「登録端末」、「非登録端末」の状態は以下のルールに基づいて決定される (図 3)。

「非登録端末に少なくとも一つの登録端末が隣接している」

上記のルールに従うと、非登録端末には必ず登録端末が隣接しているため、ある登録端末から見て隣接端末が非登録端末、さらにその隣接端末 (2 ホップ先端末) が非登録端末の場合、2 ホップ先の非登録端末は必ず他の登録端末と隣接していることとなる。つまり、ある登録端末から 3 ホップで必ず他の登録端末にたどり着ける。一方、各端末は Hello パケットにより自身か

ら2ホップ以内のトポロジを保持している。よって、登録端末はその情報を用い、隣接する端末との間のリンク、及び一部の2ホップ先の端末のリンク情報を登録することで、アドホックネットワーク内の全てのリンクを重複無く登録することができる。

3. 制御パケット量の評価

3.1. MARIONet の制御パケット量

MARIONet における制御パケットには以下のものがある。

- Hello パケット
- リンク情報登録パケット
- 通信要求パケット
- ルート通知パケット

このうち通信要求パケット、ルート通知パケットについては通信要求時に発生するものであり、定期的な送信される Hello パケット、リンク情報登録パケットと比較して非常に少ない。そこで MARIONet の制御パケット量としては、Hello パケットとリンク情報登録パケットをカウントするものとする。

図4に Hello パケットのフォーマットを示す。「FI」には Hello パケット送信端末の状態(登録端末または非登録端末)の状態を記述する。また「Link Code」には、リンクの状態(双方向または片方向)と隣接端末の状態情報を含める。「Neighbor Address」は隣接端末のアドレスのリストであるが、各隣接端末に応じた「Link Code」に続いて記述する。「Link Code」は

- 双方向-登録端末
- 双方向-非登録端末
- 片方向

の3タイプ定義する。そのため Hello パケットのサイズは以下の式で表すことができる。

$$P_{hello} = 20 + 4 \cdot n_{neighbor} \quad \dots \quad (1)$$

$n_{neighbor}$: 隣接端末数

いま、サイズ S のエリアに n 台の端末が一様に分布しているとすると隣接端末数は以下のように表せる。

$$n_{neighbor} = \frac{\pi r^2}{S} n \quad \dots \quad (2)$$

r : 通信可能半径

よって、単位時間当たりの Hello パケット送信量は以下のように表せる。

$$Q_{hello} = \frac{n}{T_{hello}} \left(20 + \frac{4\pi r^2}{S} n \right) \quad \dots \quad (3)$$

T_{hello} : Hello パケット送信間隔

0		8		16		24	
Next Header	FI	Reserved	FI	Payload Length			
Option Type	Opt Data Len		Htime				
Link Code	Reserved		Link Message Size				
Neighbor Address							
.....							
Link Code	Reserved		Link Message Size				
Neighbor Address							
.....							

図4 Hello パケットフォーマット

次にリンク情報登録パケット量を算出する。図5にリンク情報登録パケットのフォーマットを示す。リンク情報登録パケットには、隣接する端末のアドレス(Neighbor Address)とその隣接端末に隣接する端末のアドレス(2 hop neighbor)のリストが記述される。これらの情報は、端末間で重複無く選択して記述される[7]。つまり、あるひとつのリンクに対して、ひとつの端末が登録を行うことになるため、各端末のリンク情報登録パケットにリストされる端末の数の平均は、隣接する端末数の1/2となる。よって、リンク情報登録パケットサイズの平均は以下のように表せる。

$$P_{reg} = 8 + 2 \cdot n_{neighbor} \quad \dots \quad (4)$$

Hello パケットと同様に単位時間当たりのリンク情報登録パケット量を求めると以下のように表せる。

$$Q_{reg} = \frac{n}{T_{reg}} \left(8 + \frac{2\pi r^2}{S} n \right) \quad \dots \quad (5)$$

T_{reg} : リンク情報登録パケット送信間隔

0		8		16		24	
Next Header	FI	Reserved	FI	Payload Length			
Option Type	Opt Data Len		Htime				
Link Code	Reserved		Link Message Size				
Neighbor Address							
.....							
Link Code	Reserved		Link Message Size				
Neighbor Address							
2 hop neighbor							
.....							

図5 リンク情報登録パケットフォーマット

3.2. 制御パケット量のシミュレーション評価

MARIONet の制御パケット量をコンピュータシミュレーションにより測定した。シミュレーションの条件として端末密度一定(端末数が増加すればアドホックネットワークのエリアが広がる)を仮定した。比較のため、一般的に制御パケット量が少ないといわれるリアクティブ型のルーティングプロトコル AODV のパケット量も同条件にて測定した。シミュレータは Qualnet3.7[9]を用いた。表1に諸元を示す。

表 1 シミュレーション諸元

シミュレーション時間	3600s
端末密度	1台/(200m) ²
端末数	16, 64, 144, 256, 400, 576, 784
端末移動モデル	Random Way Point (Pause time =0s, Min Speed=1m/s, Max Speed=5m/s)
Hello/バケット送信間隔(T _{hello})	2s
登録バケット送信間隔(T _{reg})	2s
通信発生頻度(T _{com})	300s/端末
Physical/MAC	IEEE802.11b, 2Mbps
送信電力	15.0dbm
シミュレータ	Qualnet 3.7

図 6 に制御パケット量を示す。端末数が少ないところでは、提案方式の制御パケット量は AODV に及ばない。しかし、AODV の制御パケット量が端末数によって 2 次関数的に増加するのに対して、提案方式では 1 次関数的に増加する。そして、端末数が約 700 台になると、提案方式の制御パケット量の方が少なくなる。提案方式は、端末数が少ないところでは効果は無いが、端末数が多いところにおいて制御パケット量を少なくできるため、非常に大きな規模のアドホックネットワークに適している。

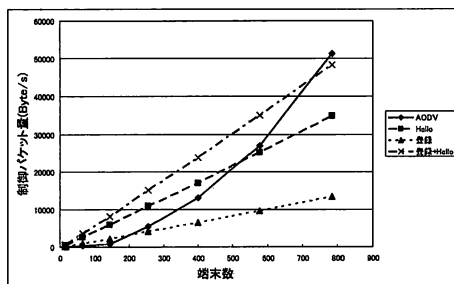


図 6 制御パケット量

図 7 に MARIONet の制御パケット量のシミュレーション値と理論値(式(3), (4))を示す。実環境においては、パケットは確率的に受信に成功するため、式(3), (4)における通信可能半径 r は明確には定めることはできないが、表 1 のシミュレーション諸元において測定した結果から $r=430(m)$ と定めた。図 7 によると、Hello パケット量、リンク情報登録パケット量と共にシミュレーション値と理論値がよく一致していることがわかる。

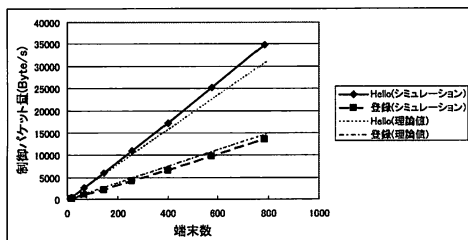


図 7 MARIONet の制御パケット量

4. サーバのルート計算負荷の評価

提案方式においては、アドホックネットワークのルーティングをインフラネットワークにあるサーバが行う。そのため、サーバでのルート計算負荷がボトルネックとなる可能性があり、アドホックネットワークを構成しうる最大の端末数を導出するためには、サーバの負荷を評価する必要がある。端末数が増加すると、サーバに登録されているリンク情報が増加するため、端末間のルートを計算する時間がかかるようになる。そこで、ルート計算時間を評価した。

4.1. ルート計算時間の測定

測定方法は、まず、仮想的に端末をランダムに配置し、通信可能半径内に存在する端末間にリンクが存在するとして、アドホックネットワークのリンク情報を生成した。そして、実機にルート計算アルゴリズムを実装し、仮想的なリンク情報を用いてランダムに選んだ 2 つの端末間のルートを計算させ、その時間を測定した。上記のリンク情報生成と、ルート計算を複数回繰り返し平均値を求めた。表 2 に実験諸元を示す。

表 2 実験諸元

測定機器	CPU:Xeon 3.4GHz/1MB L2キャッシュ 800MHz FSB メモリ:8GB DDR2 ECC
計算アルゴリズム	Dijkstra
端末密度	5台/(100m) ²
通信半径	100m
Source, Destination選択	ランダム
端末配置	ランダム

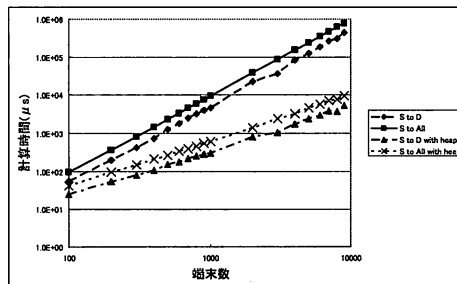


図 8 ルート計算時間

図 8 にルート計算時間の測定結果を示す。Dijkstra 法においては、通常の Dijkstra 法及び heap を用いた Dijkstra 法を評価した。また、Source 端末から全端末までのルートを計算する場合(S to All)と Source 端末から Destination 端末までのルートを求めた時点で処理を終了する場合(S to D)について評価した。Source 端末から全端末までのルートを計算する時間は、ルート計算にかかる時間の最悪値を意味する。図によると、端末数によらず heap を用いたほうが計算速度が速いこと

がわかる。またその場合、端末数 10,000 台のアドホックネットワークにおいて 1 つのルート計算を行うのに最悪の場合でも 0.01s, 平均すると 0.005s 程度で計算可能である。

4.2. ルート計算時間の定式化

実験で得られた結果を元にルート計算時間を定式化する。Heap を用いた Dijkstra 法の計算は以下の処理から構成される。

- insert : Heap への節点を追加する
- decreasekey : 節点のパスの長さを減らす
- findmin : 最もパスの短い接点を見つける
- deletemin : 最もパスの短い接点を削除する

この計算時間これらの処理時間の合計となり、一般に以下のように表すことができる。

$$\alpha \cdot n \cdot \log n + \beta(m-n) \log n + \gamma + \delta n \log n \dots (6)$$

m : リンク数 n : 端末数

今、提案方式においては、リンクの重み付けは行っておらずホップ数を最小になるようにルートを計算している。そのため、「decreasekey」の処理は存在しない。よって提案方式における計算時間は以下のように表せる。

$$T(n, m) = \alpha \cdot n \cdot \log n + \gamma + \delta n \log n + \varepsilon(m-n) \dots (7)$$

$$= an + bm + cn \log n$$

リンク数 m は端末数の 2 乗に比例し、エリア面積に反比例するので式(7)は以下のように変形できる。

$$T(n) = an + d \frac{n^2}{S} + cn \log n \dots (8)$$

端末密度一定(n/S=一定)とすると以下のように表せる。

$$T(n) = en + cn \log n \dots (9)$$

式(9)において、計算時間の実測値を用いて定数 c, e を求める。実測値として、図 8 の heap を用いた Dijkstra 法の Source から全端末へのルートを計算する場合(S to All with heap)のデータを用いた。端末数 100~9000 の区間で最小 2 乗法により定数を同定する(図 9)。図 9 によると、実測値と同定した式は端末数 500 台以上の領域においてほぼ一致している。以下に同定した式を示す。

$$T(n) = -0.878497n + 0.481809n \log n \dots (10)$$

同定した範囲外での式(10)の適合性を評価する。端末数が 14000 台(n=14000)の場合において、計算時間の実測値は、15868μs であった。一方、式(10)によると計算時間は 15668μs となる。これらはほぼ一致しており、

式(10)は同定した範囲外でもよく実測値に合うといえることができる。

Destination 端末までのルートを求めた時点で処理を終了する場合(S to D with heap), Destination 端末は平均すると全端末数の 1/2 回目の検索で見えるので、処理時間は式(10)の 1/2 となる。

$$T(n) = -0.4392485n + 0.2409045n \log n \dots (11)$$

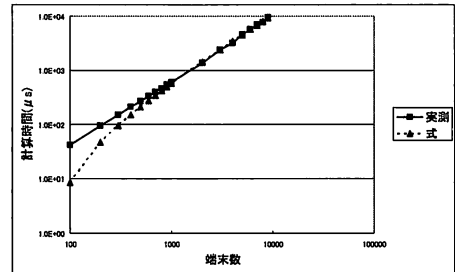


図 9 ルート計算時間の同定

4.3. ルート計算待ち時間

前節では、サーバが 1 つのルートを計算するのに必要な時間を求めた。しかし、サーバは複数の端末から通信要求を受けルート計算するため、ある端末から通信要求を受けたとき、他の端末からの通信要求に対するルート計算を行っている場合がある。よって、通信要求を受けてから実際にルート通知を行うまでの時間を求めるには、待ち時間を考慮しなければならない。サーバで s 個の CPU を持ち s 個のルートを同時計算可能であるとし、M/M/S モデルで表すことができると仮定する。M/M/S モデルにおける平均応答時間(通信要求を受けてからルート通知を行うまでの平均時間は)以下の式で表せる。

$$W = \frac{(s\rho)^{s+1}}{(s-1)!(s-s\rho)^2} P_0 \cdot \frac{T_{com}}{n} + T(n)$$

$$P_0^{-1} = \sum_{n=0}^{s-1} \frac{(s\rho)^n}{n!} + \frac{(s\rho)^s}{s!} \cdot \frac{1}{1-\rho}$$

$$\rho = \frac{n \cdot T(n)}{s \cdot T_{com}}$$

T_{com} : 端末 1 台あたりの平均通信間隔

式(11), (12)を用いて、 $T_{com}=300s$, 表 2 の条件の時のサーバの平均応答時間を図 10 に示す。図によると、端末数が少ないときは、図 8 とほぼ同じ値をとるが、あるところで応答時間が急激に増加することになる。応答時間が急激に増加するところは、サーバの使用率(式(12)における ρ)が 1 近くになっているところである。

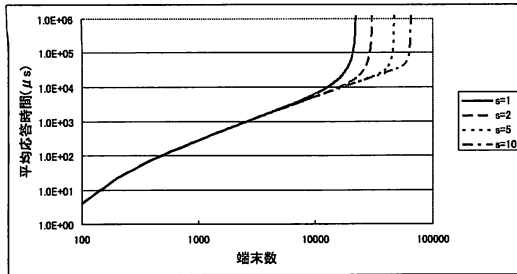


図 10 平均応答時間

5. 端末収容可能数

3 章にて提案方式の制御パケット量の理論式, 4 章においてサーバでのルート計算の理論式を求めた. 本章ではそれらを用いて, 提案方式においてどれほどの数の端末をアドホックネットワークに収容可能か算出する. 評価の諸元を表 3 に示す.

表 3 諸元

端末密度	5台/(100m) ²
通信半径	100m
通信発生頻度(T_{com})	300s/端末
Helloパケット発生間隔(T_{hello})	2s
登録パケット発生間隔(T_{reg})	2s
Source, Destination選択	ランダム
端末配置	ランダム
ルート計算アルゴリズム	Dijkstra (heap使用)
サーバ計算能力	表2のとおり

式(12)において, $s=1$, $\rho=0.5$ の時, 平均応答時間は約 18.7ms となる. この程度の時間ならアドホックネットワークのルーティングとして十分な性能であると考えられるので, $\rho=0.5$ の時の端末数をサーバ負荷から見た収容可能端末数とする. 図 11 に収容可能端末数を示す. サーバの CPU 数が 1 の時, 約 15000 台の端末の収容が可能である. CPU 数を 10 とすると約 45000 台まで収容可能である.

一方, 制御パケットの観点から見たとき, 提案方式による制御パケット量は端末数に比例する. つまり, 端末一台あたりの制御パケット量は一定で, 式(3), (5)によると 61.1Byte/s となり, 十分処理可能であると考えられる.

よって, 提案方式における端末収容可能数はサーバ負荷によって決定され, その値は数万台程度である.

6. むすび

本稿では, インフラネットワークでアドホックネットワークのルーティングを行う方式において, 制御パケット量が端末数に比例することを示した. また, 制御パケット量, サーバでのルート計算時間を定式化することにより, ルート計算サーバに一般的な PC サー

バ程度の性能のものを用いた場合, 数万台規模のアドホックネットワークを構築できる可能性がある事を示した. 今後, アドホックネットワークとルート計算サーバで負荷を分散する方式[10]を適用し, ルート計算サーバ負荷を軽減することで更なるアドホックネットワークの大規模化を検討していく.

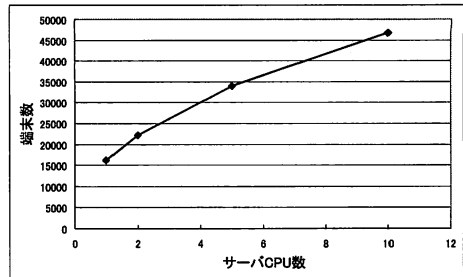


図 11 収容端末数

文 献

- [1] C. Perkins, E. Belding-Royer, and S. Das, 'Ad hoc On-Demand Distance Vector (AODV) Routing', RFC3561, October. 2003.
- [2] Naserian, M., Tepe, K.E., Tarique, M., "Routing overhead analysis for reactive routing protocols in wireless ad hoc networks", IEEE International Conference on Wireless And Mobile Computing, Networking And Communications, 2005.(WiMob'2005)
- [3] T. Clausen, Ed. and P. Jacquet, Ed., "Optimized Link State Routing Protocol (OLSR)", RFC3626, October. 2003.
- [4] Sucec, J., Marsic, I., "Hierarchical routing overhead in mobile ad hoc networks", Mobile Computing, IEEE Transactions on Volume 3, Issue 1, Jan-Feb 2004 Page(s):46-56
- [5] Bharat Bhargava, Xiaoxin Wu, Yi Lu, and Weichao Wang, "Integrating Heterogeneous Wireless Technologies: A Cellular Aided Mobile Ad hoc Network (CAMA)", ACM Mobile Network and Applications, 2003.
- [6] Hui Li, Dan Yu and Hui Chen, "New Approach to Multihop - Cellular Based Multihop Network", IEEE PIMRC 2003, Sept 2003
- [7] 小林基成, 鈴木俊博, サツルサヤエン ラタボン, 森田正範, "Hybrid Multi-hop Network におけるルーティングのためのトポロジ管理方式", 信学技法 NS2004-129, 2004年10月
- [8] M.Kobayashi, T.Suzuki, R.Satrusajang and M.Morita. "Performance Evaluation of Link Registration for Network Assisted Routing", 電子情報通信学会総大会, 2005年3月
- [9] Scalable Network Technologies, Inc., QualNet network simulation software, <http://www.scalable-networks.com/>
- [10] 小林基成, 鈴木俊博, カーン アシック, 森田正範, "アドホックネットワークとルーティング制御サーバ間の負荷分散方式", 電子情報通信学会ソサイエティ大会, 2005年9月