社団法人　情報処理学会　研究報告
IPSJ SIG Technical Report

2005－MBL－35（12）
2005－ITS －23（12）
2005／11／17

アドホックネットワークにおけるSelfish Node対策の評価

横山 信[†,‡]　　中根 由和[†]　　高橋 修[‡]　　宮本 衛市[‡]
†日本情報通信コンサルティング株式会社　　‡公立はこだて未来大学

近年のモバイルコンピューティングの急速な発展に伴い無線インタフェースを標準装備した機器が普及しつつある。そして今後ユーザのコミュニティにおいて相互協力的にパケットを中継する形態のアドホックネットワークが一般的になると考えられる。この中に非協力的で利己的に振舞うノード(Selfish Node)が出現するとネットワークの公平性が損なわれ、最終的にネットワークが利用不能となる恐れがあるため、これはセキュリティの領域に属する重要な問題である。そこで Selfish Node の影響について整理し、その検出方法と対策について検討を行い、シミュレーション評価を行った。

Evaluation of a Countermeasure Against Selfish Node in Ad Hoc Network

Shin Yokoyama[†,‡]　Yoshikazu Nakane[†]　Osamu Takahashi[‡]　Eiichi Miyamoto[‡]
†Nippon information Technology Consulting Co.,Ltd.　‡FUTURE UNIVERSITY-HAKODATE

Recently portable electronic devices have a wireless interface as standard equipment become popular accompanied a rapid development of mobile computing. Henceforward, it will be a common type of an ad hoc network, which relays packets among users' nodes in the network mutually without base stations. However, selfish nodes will emerge in an ad hoc network, and the fairness of the network will be in danger. Eventually the network service might be unavailable for the users. It is a serious problem in the security system. This is the evaluation based on the simulations of their impact on ad hoc network for a suitable detection method and countermeasure.

1. はじめに

本稿での前提としては、各ユーザの端末ノードが互いにパケットを中継しあうことによって通信が成立するという形態のネットワークを想定し、ルーティングプロトコルとしては、AODV を想定している。

そのようなコミュニティベースのアドホックネットワークを正常に利用できるようにするためには、Selfish Node の問題を解決することが重要である。

この問題のセキュリティ上の位置づけは、センサネットワークにおける DoS 攻撃について述べている文献[1]の分類によると、ネットワーク･ルーティングのレイヤに属する、Neglect and greed となる。

以下、Selfish Node の定義と動作について述べ、本稿で検討した 4 通りの利己的な動作の影響と、Selfish Node の検出と対策のアルゴリズム、およびシミュレーション評価の結果について述べる。

2. Selfish Node の定義と動作

本稿において Selfish Node とは、自己の通信のためにネットワーク（他のノード）を利用する一方、他のノードの通信のために自己のリソースが利用されることを拒否するノードである。ネットワークを利用することが目的であるため、ネットワークが利用不能になるような積極的な妨害を行うものではない。Selfish Node という用語は[2]にみられる。

以下、Selfish Node の具体的な動作について 4 通り挙げる。

（ア）「RREQ を中継しない」
通常の動作ではルート探索要求 RREQ を中継するが、この動作では RREQ を中継しない。Selfish Node はこれにより、自身が他のノード間のルートに含まれないようにし、他のノードの為の中継を避ける。

（イ）「HELLO (RREP)を送信しない」
通常の動作では、ノードは互いに定期的な RREP メッセージ(HELLO)を送信して隣接ノードの存在とリンク状態を管理するが、この動作では自身がデータを送受信するまで、HELLO を送信しない。また、（ア）と同様に RREQ を中継しない。Selfish Node がこの動作をとると、Selfish Node 自身のパケット送信のタイミングまでは他のノードから存在を検知することができないため利己的な動作を疑われることがなく、他のノードの為の中継を避けることができる。

（ウ）「RREQ を遅れて中継する」
前述（ア）の動作、及び、（イ）の動作でパケ

ットの送受信をした後は、1 ホップ前のノードから、次のノードが RREQ を中継しているか観察することにより当該動作を容易に検出できる。しかし RREQ をプロトコル上のタイムアウトの限界まで遅らせて中継すると、自身の他に中継が可能なノードがある場合は、そちらが RREQ を先に中継するために、自身は極力中継を避けることができる。他に中継可能なノードが無ければ Selfish Node 自身がパケットを中継することになるが、プロトコルの規定値以内で中継をしているために、他のノードから利己的な動作を断定することが難しい。

(エ) 「ユーザデータを中継しない」

AODV のルーティングプロトコルとしては通常の動作をするが、その後中継すべきデータパケットを受信してもそれを中継しない。通常のノードの場合でもデータパケットのロスは起こりうるためその区別をして検出することが困難になる可能性がある。

これらの動作により、Selfish Node は以下のような利得を得る。

- 他のノードのパケットを中継しない分、利用可能な帯域が増加する場合がある。
- 他のノードのパケットを中継しない分、電力を節約できる。

但し、前者の起こる場合はまれであり、主に後者の効果が大きい。

一方、ネットワークに以下のような悪影響が及ぶ[3]。

- Selfish Node により、ホップ数の増加やスループットの低下が見られる。ネットワーク中のノード密度が低い時に顕著である。
- Selfish Node の割合が増加すると、通信の効率が悪化したり、最終的に通信が不能になったりする可能性がある。

3. Selfish Node の検出アルゴリズム

以下に、検出のアルゴリズムを述べる。

(ア) 「RREQ を中継しない」動作に対して

前提として、検出を行うノードから、隣接のノードの存在を知ることが必要である。1 ホップ前のノードが RREQ を中継する時、TTL が 2 以上であり、次のノードが宛先ノードでないならば、次のノードも一定時間内に RREQ を中継するはずである。もし一定時間内に RREQ の中継が確認できなかった場合、Selfish Node であると判定する(図 1)。

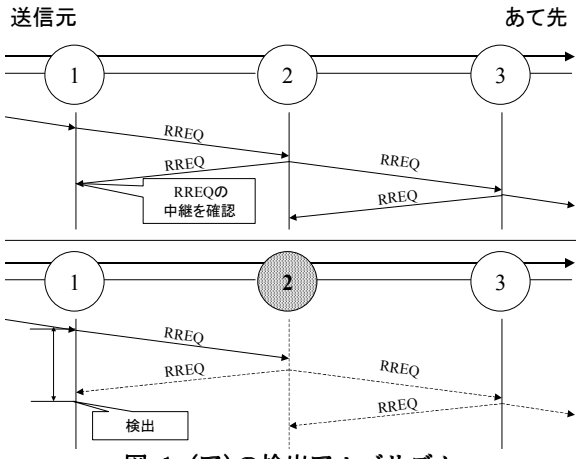

図 1 (ア)の検出アルゴリズム

(イ) 「HELLO (RREP)を送信しない」動作に対して

この動作をするノードは、自身がパケットを送信する時までは無反応なので、それまでは検出することができない。

パケットを送信する際には、その存在が隣接のノードに知られることとなる。その後、前項(ア)同様に、RREQ が中継されているかを確認することができる(図 2)。

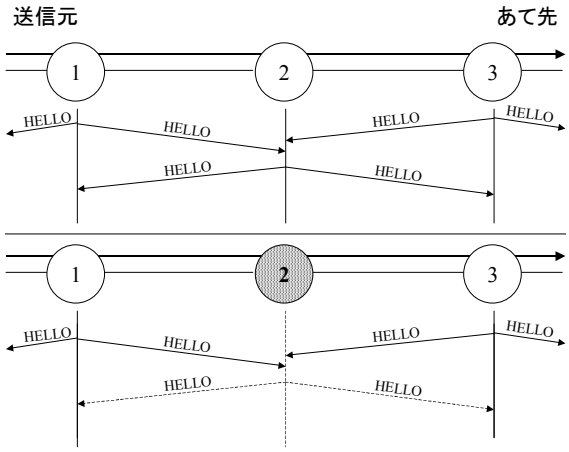

図 2 (イ)の状態

(ウ) 「RREQ を遅れて中継する」動作に対して

RREQ の中継タイミングを 1 ホップ前のノードから観察して、中継のタイミングが遅い場合にこの動作をしていると判定する(図 3)。

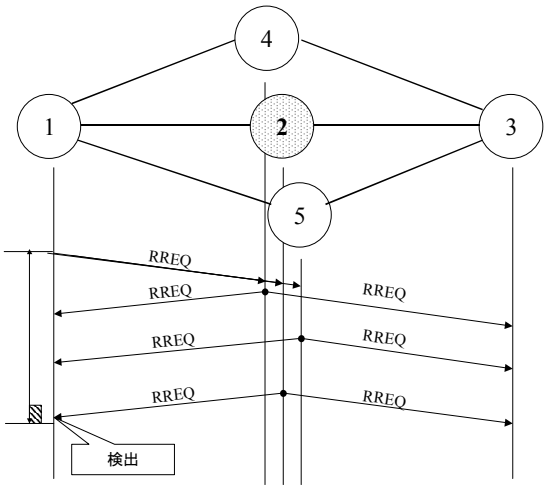

図 3（ウ)の検出アルゴリズム

(エ) 「ユーザデータを中継しない」動作に対して

データを送信した後、それを中継したかを、1 ホップ前のノードから観察することによって検出が可能である(図 4)。

但し通常動作しているノードも、パケットをドロップする可能性はあるので、そのような事象と区別することが難しい。

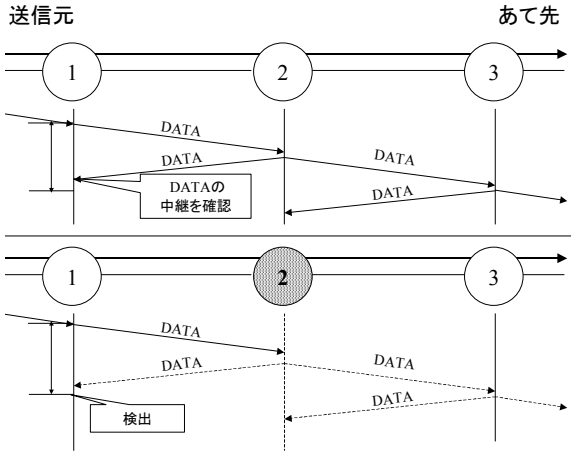

図 4（エ)の検出アルゴリズム

4. Selfish Node への対策アルゴリズム

Selfish Node と判定されたノードを宛先としたパケットもしくは Selfish Node から送信されたパケットの中継を拒否する。また、そのノードからの AODV パケットを処理しない。

対策機能は検出を行ったノードだけで行うのではなく、周辺のノードに通知して中継を拒否する機構を持たせたほうがより高い効果を期待できるが、今回のシミュレーションでは検出を行ったノードでだけ対策を行っている。

5. 評価

シミュレーションはネットワークシミュレータ ns-2.27[4]、AODV-UU 0.8.1[5]を用い、**表 1**の条件で行った。

表 1 シミュレーション条件

ノードの移動	なし
無線方式	802.11 送信レート 2Mbps (Lucent WaveLAN DSSS 無線インタフェースをシミュレート)
無線到達距離	250[m]

また、検出アルゴリズムのパラメータについては、(ア)、(イ) で中継を確認する待ち時間は、AODV プロトコル[6]の NEXT_HOP_WAIT 値に合わせ 50[ms]とした。

(ウ) は AODV-UU の実装で、RREQ の中継の遅延時間が最大 20[ms]に設定されていることより、中継を確認する待ち時間を 20[ms]とした。(エ) のデータパケットの中継を確認する待ち時間は[6]の NET_TRAVERSAL_TIME に合わせ 2800[ms]とした。

5.1. ランダム配置モデルでの評価

ランダム配置モデルのシミュレーションは、**表 1** に加え、**表 2の**条件で行った。図 5では、ランダム配置の状況を示しており、円はある 1 つのノードの無線到達範囲を例示している。

表 2 シミュレーション条件

ノードの配置範囲	1000 x 1000[m]
ノード数	50

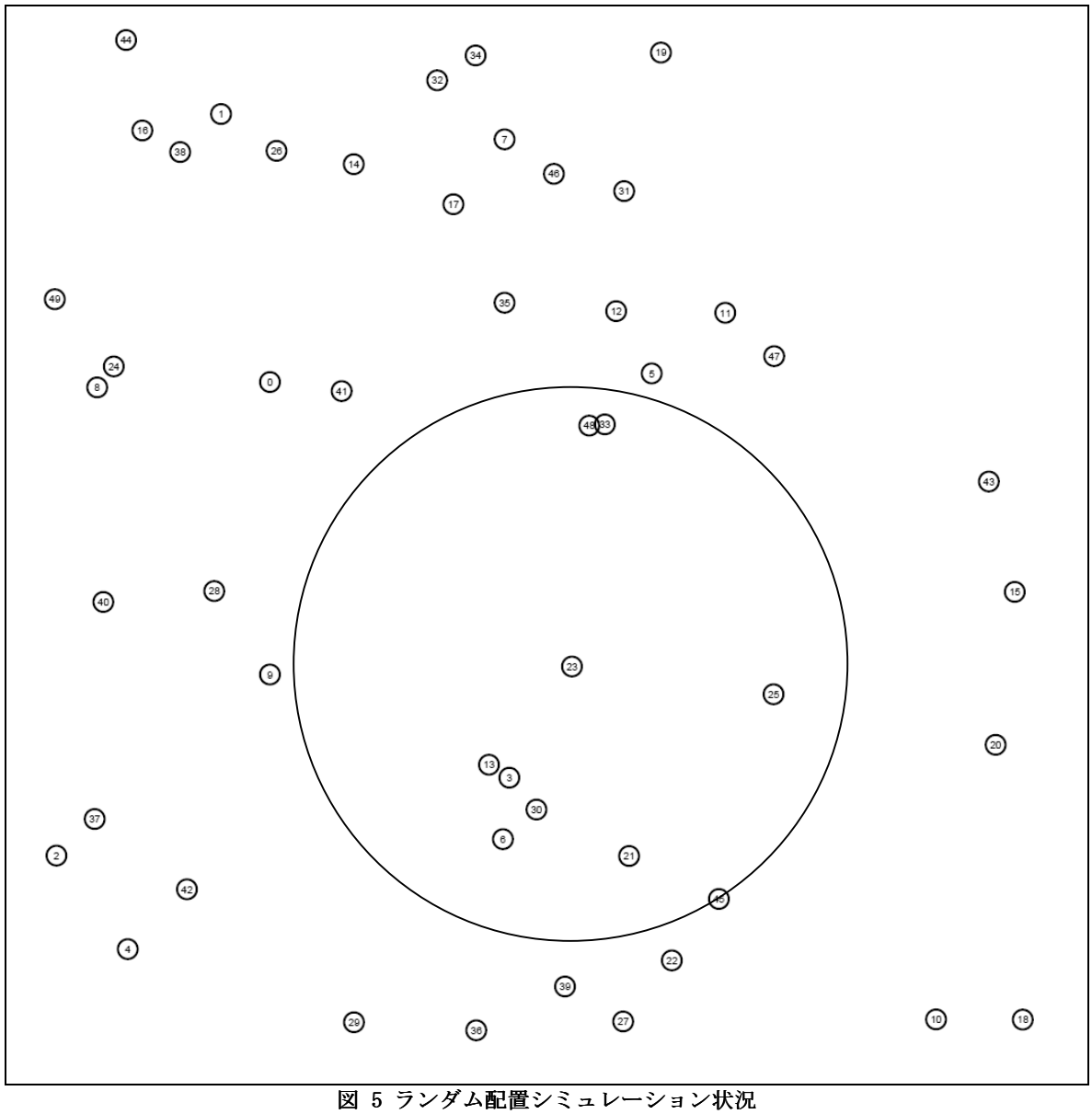

図 5 ランダム配置シミュレーション状況

5.1.1. Selfish Node の影響の評価

まず、比較の対象として Selfish Node が存在しない状態で以下のようにシミュレーションを行った。

- 50 ノードの中からランダムに 2 組（4 ノード）を選び、同時に 2 本の FTP 転送を 10 秒間行う。
- 10 秒ごとに FTP 転送の組を選びなおし、100 秒実行

このシミュレーションにおいて、パケットの中継回数を求めた。中継回数と中継にかかる消費電力はほぼ比例すると考えられるため、これを消費電力の代用の指標とした。結果を表 3の状況 1)の列に記述した。

続いて同様の条件で、50 個のノードの半分が前述の Selfish Node となった場合の影響をみた。但し各組とも通常どうし もしくは Selfish どうしのノードの組とし、通常ノードと Selfish Node の組はないものとした。

（ア）、（イ）、（ウ）、（エ）の 4 通りの動作について、通常のノードと Selfish Node の 1 個あたりの中継回数を求めた。結果を表 3の状況 2)の列に記述した。

5.1.2. 検出機能の評価

Selfish Node あり (Selfish 25, 通常 25)の場合の検出機能の動作について、検出率と誤認率を求めた。但し、対策機能は無効にしたまま行った。

検出率・・・通常ノードが、Selfish Node を検出できる確率。全ての Selfish Node のうち、1 つ以上の通常ノードによって検出されたものの割合としている。

誤認率・・・通常ノードが、他の通常ノードを Selfish Node と誤認する確率。全ての通常ノードのうち、1 つ以上の通常ノードによって Selfish Node と判定されたものの割合としている。

（ア）、（イ）、（ウ）、（エ）の 4 通りについてそれぞれ上記の値を求めた。結果を表 3の状況 2)の列に記述した。

5.1.3. 対策機能の評価

Selfish Node あり (Selfish 25, 通常 25)の場合で、さらに各ノードの対策機能を有効にした場合でシミュレーションを行った。

通常のノードと Selfish Node の 1 個あたりの中継回数を求めた。結果を表 3の状況 3)の列に記述した。

5.1.4. ランダム配置の評価のまとめ

以上の結果をまとめると下表のようになる。

	状況 1)	状況 2)	状況 3)
(ア)	○ 167	○ 295	○ 148
		● -	● -
		検出率 96% 誤認率 16%	
(イ)		○ 292	○ 169
		● -	● -
		検出率 24% 誤認率 12%	
(ウ)		○ 256	○ 248
		● 65.0	● 65.5
		検出率 68% 誤認率 56%	
(エ)		○ 0.960	○ 1.24
		● -	● -
		検出率 78% 誤認率 38%	

表 3 ランダム配置の評価のまとめ

凡例
- ○ ・・・ 通常ノード 1 個当たり中継回数
- ● ・・・ Selfish Node 1 個当たり中継回数

状況
1) Selfish Node なし (50 個)
2) Selfish Node 25 個, 通常ノード 25 個 対策機能なし
3) 2)の状態で 対策機能 あり

Selfish Node のない状態では、ノード 1 個あたりの中継回数は 167 回であった。

(ア)の場合、Selfish Node が出現すると通常ノードの中継回数が増し 1 個当たり 295 回となった。対策機能を有効にすると、これは 148 回となった。96%の Selfish Node を検出できたが、誤認率は 16%となっている。

(イ)の場合、Selfish Node は自己の送受信のタイミングまで隠れているために、(ア)と比べて検出率が下がり 24%となった。

(ウ)の場合、誤認率が 56%と非常に高くなっており、効果も得られていない。

(エ)の場合、殆どの組で通信不能になっていた。AODV プロトコルとしては正常動作しているため、データパケットが廃棄されても経路の切替が起こらないためと考えられる。対策機能の効果も得られていない。

6. 考察

(ア)のようなもっとも単純な動作に対しては検出率も高く、検出を行った個別のノードでの対策で効果を上げることができた。

(イ)の動作の場合、隠れている Selfish Node を検出することは原理的に不可能である。しかし、自己のデータ送受信時だけ現れるという動作を検出することによって、対策をとることができると考えられる。

(ウ)の動作に対して、1 回の RREQ の転送による検出は正しくできていないと言えるため、複数回の観察によって判定することが必要である。

また、(ウ)、(エ)の動作に対しては、効果が得られなかった。通知の機構を用いると、効果を得られる可能性がある。

7. 関連研究

文献[2]で、Selfish Node という用語が使われている。Selfish Node を含む不正動作ノードの検出と対策について述べているが、不正動作ノードに対してペナルティを課すような方法はとらず、最も信頼できるパスを選択してスループットを確保するというアプローチである。

文献[7]では、他のノードを評価するカウンタを持たせることによって、相手のパケットを中継する方法を変化させる方法を提案している。 [8]では、パケットの転送にインセンティブを与えて協調を促進するアプローチがとられている。

文献[9][10]では、DSR プロトコルにおいて Selfish Node を特定しネットワークから隔離するアプローチをとっている。

文献[11]では、アドホックネットワークにおけるノードの協調に関する既存の研究をまとめている。現状、特定のレイヤ・特定のプロトコルに注目したものが多く、今後の研究ではプロトコルごとの分析、各レイヤに及ぶ潜在的な影響を明らかにして考える必要があり、協調関係のコントロールにはレイヤ間にまたがるプロトコルのアーキテクチャが有効な方法と考えられるとして NeSt (Network Status) アーキテクチャを提唱している。

文献[12]では、Selfish Node の利己的通信発生以前にプロアクティブ的に動作する、転送実績に基づ

いた参加促進メカニズム PCOM(Proactive Cooperation Mecanism)を提案している。

文献[13]では、doublehash 認証と呼ばれる、パケットを認証しながら中継していく方式と、Web of Trust の考え方を利用したセキュアなルーティング方式を提案している。多数のユーザの端末同士で構成されるアドホックネットワークにおいては、同様の方法が通知のアルゴリズムにおいて利用可能であると考える。

8. おわりに

アドホックネットワークにおける Selfish Node の問題について述べ、その影響と検出方法及び対策についてシミュレーション評価を行った。

（ア）、（イ）への対策では、Selfish Node の通信を阻止し、通常ノードの中継負荷を減らし一定の効果を得ることができた。

（ウ）、（エ）への対策では、大きな効果が得られなかった。これは、通知の機構を取り入れたときにどのようになるかを評価する必要がある。

利己的な動作は挙げた 4 通りのほかにもあり得るのでさらに整理が必要である。

本稿の方法で、Selfish Node を検出し対策を行うことができるのは、ルート探索時に Selfish Node の 1 ホップ前にあるノードに限られる。この場合 Selfish Node は他の隣接ノードを使用して通信を続ける場合もあるため、効果を上げるためには周辺の他のノードに Selfish Node の存在を通知して中継を拒否する機能が効果的であると思われる。

今回、この通知のプロトコルはできていないが、より詳細に検討する必要がある。

通常のノードが他のノードに Selfish Node として誤検出されると通信を行えなくなるので、その可能性は十分に小さくする必要がある。

また、誤検出の可能性を考慮に入れて、判定を修正できる機構などを考慮する必要がある。特に、Selfish Node を検出してそれに対する対策をとったために別のノードから Selfish Node と検出されてしまうと、問題がある。

さらに、移動性に関する評価も行う必要がある。

参考文献

[1] A. Wood and J. Stankovic, Denial of Service in Sensor Networks, IEEE Computer , Vol. 35, No. 10, October 2002, pp. 54-62.

[2] S. Marti, T. Giuli, K. Lai, and M. Baker. Mitigating Routing Misbehavior in Mobile Ad Hoc Networks. In *Proceedings of the Sixth annual ACM/IEEE International Conference on Mobile Computing and Networking*, pages 255–265, 2000.

[3] 横山、中根、高橋、宮本、宮西、”アドホックネットワークにおけるプロトコル非準拠ノードの検出方法及び対策”, マルチメディア、分散、協調とモバイル(DICOMO2005)シンポジウム、pp. 233-236, 2005.7

[4] UCB/LBNL/VINT, Network Simulator version2 (ns-2).

[5] AODV-UU @ Uppsala University, http://www.docs.uu.se/docs/research/projects/scanet/aodv/aodvuu.shtml

[6] RFC3561, ”Ad hoc On-Demand Distance Vector (AODV) Routing”, http://www.ietf.org/rfc/rfc3561.txt

[7] L. Buttyan, J.P. Hubaux, “Stimulating cooperation in self-organizing mobile ad hoc networks”, MONET, Oct.2003

[8] S Zhong, J Chen, YR Yang, “Sprite: A simple, cheat-proof, credit-based system for mobile ad-hoc networks”, Infocom’03, Mar.2003

[9] Sonja Buchegger , Jean-Yves Le Boudec, Performance analysis of the CONFIDANT protocol, Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing, June 09-11, 2002, Lausanne, Switzerland

[10] Sonja Buchegger and Jean-Yves Le Boudec. Nodes Bearing Grudges: Towards Routing Security, Fairness, and Robustness in Mobile Ad Hoc Networks. In Proceedings of the Tenth Euromicro Workshop on Parallel, Distributed and Network-based Processing, pages 403 – 410, Canary Islands, Spain, January 2002. IEEE Computer Society.

[11] M.Conti, E.Gregori, G.Maselli “Cooperation Issues in Mobile Ad Hoc Networks”, ICDCSW’04, Mar.2004

[12] 鈴木、サットルサヤェン、小林、森田、”アドホックネットワークにおける実績に基づいた参加促進メカニズム”, IEICE Technical Report, AN2005, Jan. 2005

[13] 織田学, 静岡大学情報学部情報科学科 平成15 年度卒業論文 “アドホックネットワークにおけるセキュアルーティング方法”