

アドホックネットワークにおける不定期データ更新を考慮した キャッシュ無効化について

林 秀樹[†] 原 隆浩[†] 西尾 章治郎[†]

[†] 大阪大学大学院情報科学研究科マルチメディア工学専攻
〒565-0871 大阪府吹田市山田丘 2-1
E-mail: {hideki,hara,nishio}@ist.osaka-u.ac.jp

本稿では、不定期にデータの更新が発生するアドホックネットワークにおいて、古いキャッシュデータを効率的に無効化する二つの方式を提案する。一つめの提案方式では、移動体がデータを更新した際に、その移動体が相互接続している移動体に無効化情報を放送する。二つめの方式では、ある二つの移動体が新たに接続した場合に、自身がもともと相互接続していた移動体の集合に、新たな無効化情報を放送する。これにより、移動体が、すでに更新されている古いキャッシュデータにアクセスする回数を削減すると同時に、無駄なアクセスをした際に生じるロールバック処理の回数を削減できる。

On Cache invalidation Considering Aperiodic Data Update in Ad Hoc Networks

Hideki HAYASHI[†] Takahiro HARA[†] Shojiro NISHIO[†]

[†]Dept. of Multimedia Eng., Graduate School of Information Science and Technology, Osaka University

In this paper, we propose two cache invalidation methods in ad hoc networks where each data item is updated at inconsistent intervals. In the first method, when a mobile host holding an original data item updates the data item, it broadcasts an invalidation report to all connected mobile hosts. In the other method, when two mobile hosts are connected, they rebroadcast invalidation reports received before to newly connected mobile hosts. Our proposed methods reduce the number of accessing invalid cached data items which have been updated and the number of roll backs caused by such invalid accesses.

1 はじめに

近年、無線通信技術の発展と計算機の小型化に伴い、携帯型計算機を持ち歩くことで、いつでもどこでもネットワークに接続することが可能な移動体計算機環境が普及しつつある。特に、移動体にルータの機能をもたせ、移動体のみの一時的なネットワークを形成するアドホックネットワークに関する研究の関心が高まっている [1, 2, 7, 8]。アドホックネットワークでは、移動体の移動によってネットワークが分断された場合に、分断された部分ネットワーク内のデータに対してアクセスできないため、データの利用率が低下してしまう (図 1)。この問題を解決する手法として、データの複製を作成し、オリジナルデータをもつ移動体とは別の移動体に配置することが有効である。

従来のアドホックネットワークに関する研究の大半は、IETF(Internet Engineering Task Force) を中心として、相互接続されている移動体間における通信性能を向上させるためのルーティングプロトコ

ルに関するものである [5, 6, 9, 10, 11]。一方、アドホックネットワークにおけるアプリケーションには、ユーザ同士が直接コミュニケーションをとるものだけでなく、センサーネットワークでのデータ共有や発掘調査などの協調作業のための情報共有など、他のユーザがもつデータにアクセスするものも多い。そのため、アドホックネットワークにおけるデータ可用性の向上を目的として、効率的に複製配置を行うことの重要性は高い。しかし、筆者らの知る限り、アドホックネットワークにおける複製配置の研究は、筆者ら以前には行われていない。

筆者らは文献 [3] において、移動体が限られたキャッシュ領域をもち、データを更新しないアドホックネットワークを想定して、データの複製を配置する方式を提案した。これらの提案方式は、データへのアクセス頻度とネットワークのトポロジを考慮した、ヒューリスティックな方式である。さらに文献 [4] において、データの更新が周期的に起こるアドホックネットワークを想定して、データ利用率向上のための複製配置方式を提案した。これらの提案方

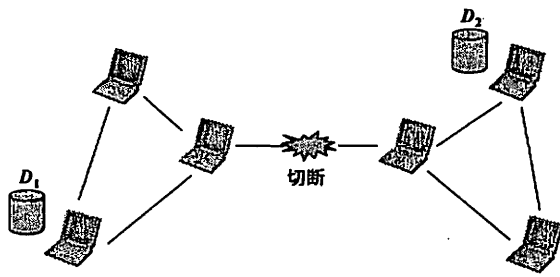


図 1: ネットワークの分断

式では、各データへのアクセス頻度、各データの次の更新までの時間、及びネットワークの接続状態を考慮して複製の配置を決定する。

一方、実環境では、データ更新が不定期に起こることが一般的である。このような場合、移動体が更新発生後の無効な複製（キャッシュデータ）にアクセスする可能性がある。古いキャッシュデータに対するアクセスは、オリジナルをもつ移動体と再接続した際にロールバックされる。このような無駄なデータアクセスや、ロールバック処理は、消費電力が重要な問題となる移動体では好ましくない場合が多い。そこで、本稿では、古いキャッシュデータに対するアクセス回数を削減するために、更新発生時のタイムスタンプ情報を放送し、キャッシュデータを効率的に無効化する方式を提案する。

以下では、2章で想定環境について述べる。3章で本稿で提案するキャッシュの無効化方式について述べ、4章で提案方式に適応したデータアクセス方法について述べる。5章で提案方式の性能評価のために行ったシミュレーション結果を示し、6章で提案方式に対する考察を行う。最後に7章で本稿のまとめを述べる。

2 想定環境

本稿では、不定期なデータ更新が起こるアドホックネットワークにおいて、他の移動体をもつデータに対してアクセスする環境を想定する。各移動体は、自身のキャッシュに、他の移動体をもつデータの複製を作成し、複製を一定の周期（再配置周期）で再配置する。複製の再配置時には、筆者らが文献[3]において提案した次の三つの方式を用いる。

1. SAF(Static Access Frequency) 方式：各移動体がアクセス頻度の高いデータの複製を配置する。
2. DAFN(Dynamic Access Frequency and Neighborhood) 方式：SAF方式で複製を暫定的に配置した後、隣接移動体間で複製の重複を解消する。

3. DCG(Dynamic Connectivity based Grouping) 方式：安定度の高い移動体のグループを作成し、グループ内で複製を共有する。

本稿では、最新のデータにアクセスした場合にのみアクセスが成功し、古いキャッシュデータにアクセスした場合は失敗とみなす。アクセス要求は、自身もしくは相互接続された移動体がオリジナルデータをもつ場合、即座に成功する。なお本稿では、相互接続された移動体とは、1ホップ以上の無線リンクで相互に通信可能な移動体の集合を指す。自身もしくは相互接続された移動体オリジナルをもたず、複製のみをもつとき、その複製に対して暫定的にアクセスする。暫定的なアクセスは、後にオリジナルをもつ移動体と相互接続した際に、成功か失敗かが決定する。これは、オリジナルをもつ移動体において更新履歴を保持し、接続時にその移動体に対して、アクセスを行った複製のタイムスタンプ（バージョン）とアクセス時刻を知らせることで実現できる。暫定的なアクセスが失敗となった場合は、複製にアクセスする前の状態に戻るようロールバック処理を行う。一方、自身もしくは相互接続された移動体がアクセス対象のデータや複製をもっていない場合は、アクセス要求は、即座に失敗する。想定環境のその他の詳細を以下に示す。

- m 個の移動体（識別子： M_1, M_2, \dots, M_m ）が存在し、各々が自由に移動する。
- n 個のサイズの等しい異なるデータ（識別子： D_1, D_2, \dots, D_n ）が存在し、各々が特定の移動体にオリジナルデータとして保持されている。
- 各移動体 $M_i (i = 1, \dots, m)$ は、自身もつオリジナルデータ以外に、データ C 個分のキャッシュ領域をもち、複製を作成する。
- 各移動体の各データに対するアクセス頻度は既知とし、時間的に変化しない。
- 各データは、そのオリジナルをもつ移動体によって不定期に更新される。更新発生後、古いキャッシュデータは無効となる。
- 各移動体は、ネットワーク内に存在する各データの更新時刻の情報（タイムスタンプ）を保持する。この情報の表をタイムスタンプ表と呼ぶ。

3 キャッシュの無効化

前章で述べた想定環境において、古いキャッシュデータに対するアクセス回数を削減するために、古

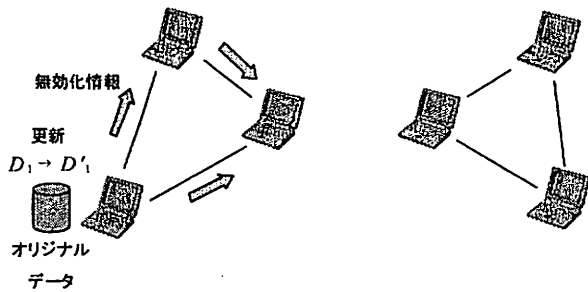


図 2: 更新時放送方式

いキャッシュデータを効率的に無効化する方法を提案する。以下では、提案する二つの方式について説明する。

3.1 更新時放送方式

更新時放送方式では、オリジナルデータをもつ移動体はそのデータに対して更新を行ったとき、自身と相互接続している移動体に無効化情報を放送する(図2)。無効化情報には、次のような情報が含まれる。

- データの識別子
- 更新時刻 (タイムスタンプ)

無効化情報を受信した各移動体は、自身のタイムスタンプ表を参照して、自身のもつ複製が無効かどうかを判断する。

具体的にはまず、受信した無効化情報に含まれるタイムスタンプと自身のタイムスタンプ表の情報を比較して、前者の方が新しい場合は、自身のもつタイムスタンプ表の情報を無効化情報に含まれるタイムスタンプに置き換える。これと同時に、受信した無効化情報を、自身と隣接している移動体に転送する。さらに、そのデータの複製をキャッシュ領域に保持している場合は、それをキャッシュ領域から破棄する。破棄された複製のためのキャッシュ領域は、空けたままにしておく。その後、自身のもつタイムスタンプ情報と同じ、もしくはより新しいデータにアクセスが成功したとき、空けておいた領域にそのデータの複製を配置する。

一方、無効化情報に含まれるタイムスタンプと自身のタイムスタンプ表の情報が同じ場合は、同一の無効化情報を再度受信しているため、隣接移動体には転送せず、無効化情報を破棄する。

この方式では、移動体がオリジナルを更新する場合にのみ、無効化情報を放送するので、ネットワーク全体のトラヒックは小さい。また、オリジナルをもつ移動体と相互接続している移動体は、その複製

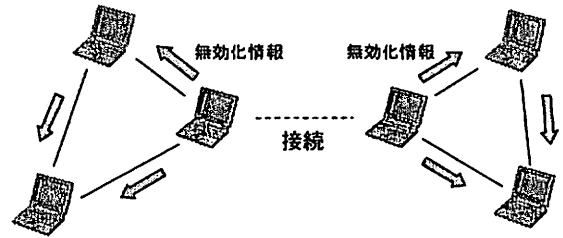


図 3: 接続時再放送方式

に関して最新のタイムスタンプをもつ。しかし、移動体の移動に伴い、リンクの接続や切断が頻繁に起こる環境では、相互接続している移動体が異なるタイムスタンプ情報をもつ場合が生じる。そのため、このことを考慮したデータアクセス方法が必要となる。

3.2 接続時再放送方式

接続時再放送方式では、オリジナルをもつ移動体が更新を行った際に無効化情報を放送するだけでなく、新たに接続(隣接)した移動体どうしが、次の手順に基づいて、タイムスタンプ表の情報を更新し、無効化情報を再放送する(図3)。

1. 識別子が M_i と M_j ($i < j$) である二つの移動体が新たに接続したとき、識別子の添字が大きい移動体 (M_j) が、小さい移動体 (M_i) に自身のタイムスタンプ表を送信する。
2. 移動体 M_i は、受信した移動体 M_j のタイムスタンプ表と自身のもつタイムスタンプ表の各項目を比較して、タイムスタンプ情報を更新する。その後、次の処理を行う。

- 移動体 M_i のもつタイムスタンプの方が古かったデータに関しては、新しいタイムスタンプ情報(無効化情報)を M_i がもともと相互接続していた移動体に放送する。
- 移動体 M_j のもつタイムスタンプの方が古かったデータに関しては、新しいタイムスタンプ情報を M_j に送信し、 M_j から M_j がもともと相互接続していた移動体の集合に放送する。

放送された無効化情報を受信した移動体は、更新時放送方式と同様に、複製の無効化を行う。

この方式では、移動体間の接続が新たに起こる度に無効化情報を放送するので、相互接続している移動体は、同一のタイムスタンプ表をもつようにな

る。また、オリジナルをもつ移動体と相互接続していなくても無効化情報が伝播するため、広範囲で複製の無効化が可能となり、古いデータにアクセスする回数を削減できる。しかし、ネットワークのトポロジ変化が頻繁な場合は、無効化情報を放送する頻度が増えるので、更新時放送方式と比較すると、トラヒックが大きくなる。

4 データアクセス

本章では、提案方式を用いる際のデータアクセス方法について説明する。

前述の通り、アクセス要求をした移動体が、オリジナルにアクセスした場合は、即座にアクセスが成功する。しかし、複製に暫定的にアクセスした場合は、最新の複製であるとは限らないので、アクセスが失敗となる可能性がある。

そこで、アクセス対象のデータが、自身のもつオリジナルデータではない場合、ネットワーク内にアクセス要求を放送し、相互接続された移動体のいずれかがそのオリジナルやより新しい複製をもっているとき、それに対してアクセスを行う。オリジナルをもつ移動体と相互接続していないとき、自身を含め、相互接続している移動体のもつ複製の一つに暫定的にアクセスする。このとき、後にオリジナルをもつ移動体と相互接続した際に、暫定的なアクセスが成功か失敗かを調べるため、次のような情報をアクセス履歴として保持する。

- 複製にアクセスした時刻
- アクセスした複製のデータ識別子
- アクセスした複製のタイムスタンプ (バージョン)

ここで、更新時放送方式と接続時再放送方式では、各移動体のタイムスタンプ表の管理法が異なるので、それぞれの方式に適応したデータアクセス方法が必要になる。以下では、両方式のデータアクセス方法を説明する。

4.1 更新時放送方式のデータアクセス

アクセス要求を行う移動体は、相互接続している移動体の中で、アクセス対象のデータをもつものがあるのかを調べるために、次のような情報を含むデータ問合せパケットを放送する。

- 自身の識別子
- アクセス対象のデータの識別子

データ問合せパケットを受信した移動体は、自身がアクセス対象のデータ（複製）をもつ場合、アク

セス要求を行った移動体に、その旨を伝える返信パケット（データ問合せ返信パケット）を送信する。この方式では、相互接続している各移動体が異なるタイムスタンプ表をもつ可能性があるため、データ問合せ返信パケットにタイムスタンプの情報を付加する必要がある。具体的には、データ問合せ返信パケットは、次のような情報を含む。

- 自身の識別子
- アクセス対象のデータの識別子
- オリジナルもしくは複製かを表すフラグ
- データ（複製）のタイムスタンプ

データ問合せをした移動体は、オリジナルをもつ移動体からデータ問合せ返信パケットを受信すると、その移動体にデータ要求パケットを送信し、オリジナルデータにアクセスする。一方、オリジナルをもつ移動体と相互接続していない場合は、データ問合せに対して返信があった移動体の中で、最も新しい複製をもつ移動体にデータ要求パケットを送信する。その後、その複製に暫定的にアクセスする。

4.2 接続時再放送方式のデータアクセス

アクセス要求を行う移動体は、更新時放送方式と同様のデータ問合せパケットを放送する。これを受信した移動体は、アクセス対象のデータ（複製）をもつ場合、次のような情報を含むデータ問合せ返信パケットをアクセス要求を行った移動体に送信する。

- 自身の識別子
- アクセス対象のデータの識別子
- オリジナルもしくは複製かを表すフラグ

この方式では、相互接続している移動体は、必ず同じタイムスタンプ表をもつので、どの移動体も同じバージョンの複製をもつ。そのため、データ問合せ返信パケットに複製のタイムスタンプの情報を含める必要はない。アクセス要求を行った移動体は、オリジナルをもつ移動体から返信パケットを受信すると、その移動体にデータ要求パケットを送信する。それ以外のとき、返信パケットを送信した移動体のいずれかに（通常は最も近い移動体）データ要求パケットを送信する。

5 性能評価

本章では、提案した二つの方式の性能評価のために行ったシミュレーション実験の結果を示す。

5.1 シミュレーション環境

50×50の2次元平面上に、40個の移動体 ($M = M_1 \dots M_{40}$) が存在する。各移動体は、すべての方向に等確率に、0から1の範囲でランダムに決定した速度で移動する。各移動体の無線通信範囲は、半径7の円とする。ネットワーク内に40種類のデータ ($D = D_1 \dots D_{40}$) が存在し、 D_j は M_j にオリジナルデータとして保持されている。各移動体は、最大10個の複製を作成する。文献[3]の複製配置方式を用いて、再配置周期 T で複製の再配置が行われる。各移動体 M_i の D_j へのアクセス頻度は、 $P_{ij} = 0.5 \times (1 + 0.001i)$ とする。データの更新は、平均 U_{avg} の指数分布に基づいた間隔で発生する。

シミュレーション実験では、初期位置として各移動体をランダムに配置し、50000単位時間を経過させたときの古い複製にアクセスする回数と、無効化情報によるトラフィックを、各方式で比較する。ここで、無効化情報によるトラフィックとは、50000単位時間内に発生した、無効化情報の放送に要する通信のホップ数に無効化情報のサイズを掛けたものの総和である。無効化情報のサイズは、それに含まれる項目を情報量の単位としたもので、提案方式では、無効化情報の中にデータの識別子と更新時刻が含まれるため、そのサイズは2となる。

5.2 平均更新間隔の影響

T を100に固定して U_{avg} を1から100まで変化させたときのシミュレーション結果を図4と図5に示す。

すべての図において、横軸は平均更新間隔 U_{avg} を示す。縦軸は、図4では古い複製にアクセスする回数を、図5では無効化情報によるトラフィックを示す。グラフ中で、'UB'(Update Broadcast) は更新時放送方式を、'CR'(Connection Rebroadcast) は接続時再放送方式を示す。したがって、'SAF-UB' は、複製配置方式としてSAF方式、キャッシュ無効化方式として更新時放送方式を用いている場合を表している。

図4の結果から、それぞれの複製配置方式において、接続時再放送方式は、更新時放送方式よりも、古い複製にアクセスする回数を大幅に削減できることがわかる。複製再配置方式を比較すると、相互接続している移動体がアクセス対象の複製を保持している可能性が最も高いのがDCG方式であるため[3]、古い複製にアクセスする回数も最も多くなってしまふ。

図5の結果から、接続時再放送方式は、更新時放送方式より無効化情報を多く発生することがわかる。平均更新間隔が小さい場合、無効化情報が頻

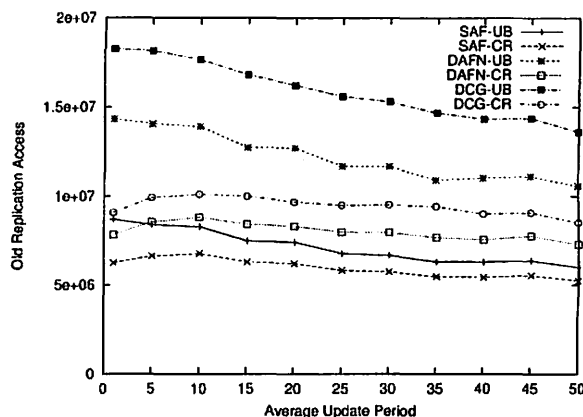


図4: 平均更新間隔と古い複製にアクセスする回数

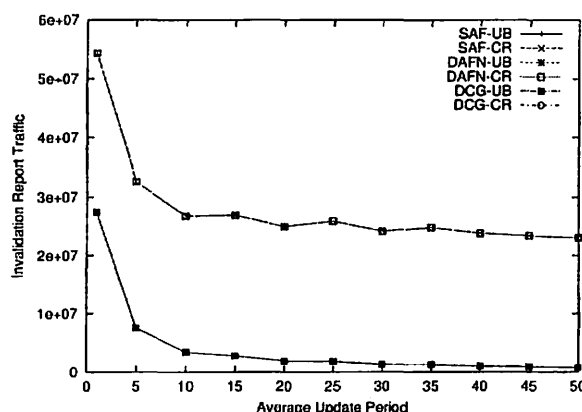


図5: 平均更新間隔と無効化情報によるトラフィック

繁に放送されるため、両手法とも高いトラフィックを示す。特に、接続時再放送方式では、移動体の接続時に、無効化情報を再放送するので、トラフィックがより大きくなる。一方、平均更新間隔が大きくなると、データの更新が発生する頻度が低くなるので、更新時放送方式では、無効化情報によるトラフィックがほとんど発生しなくなる。しかし、接続時再放送方式では、ある値まで低くなるが、無効化情報の再放送の影響から、それ以下には低くならない。また、無効化情報によるトラフィックは、複製配置方式ではなく、ネットワークのトポロジに大きく依存するので、複製配置方式が異なっても、ほぼ同じ値を示している。

6 考察

本章では、提案方式について、いくつかの観点から考察を行う。

6.1 提案方式の特徴

更新時放送方式では、無効化情報を放送するためのトラフィックは小さいが、更新発生時にオリジナル

をもつ移動体と相互接続していなければ、最新の情報を得ることができない。さらに、ネットワークのトポロジが頻繁に変化する場合は、異なるタイムスタンプ表をもつ移動体が相互接続するため、アクセスする複製のタイムスタンプを常に考慮する必要がある。一方、接続時再放送方式では、ネットワークのトポロジが頻繁に変化する環境でも、相互接続している移動体は同じタイムスタンプ表をもつ。さらに、新たに移動体と接続した際に、タイムスタンプ表を更新するため、古いキャッシュデータにアクセスする回数を削減できる。しかし、更新情報を放送するためのトラヒックが大きい。二つの方式は、システム特性に応じて、いずれを選択するかを決定する。

6.2 データアクセスの成功、失敗

本稿では、移動体がアクセス要求を行うとき、最新のタイムスタンプをもつデータにアクセスしなければ、失敗とみなされる。しかし、実環境では遺跡発掘調査での発掘状況データのように、データが多少古くても、ある程度の有効性をもつ場合がある。このようなアプリケーションでは、無効化情報を送信する頻度を調節して、トラヒックを抑える方式も考えられる。

6.3 更新情報の放送

本稿では、キャッシュの効率的な無効化を目的とし、無効化情報をネットワーク内で放送することを想定した。しかし、頻繁にアクセス要求を発行する移動体に対しては、無効化情報ではなく、更新情報そのものを送信することで、複製を効率的に最新の状態に保つことができ、データアクセス時のトラヒックを削減できる。

7 まとめ

本稿では、アドホックネットワークにおける不定期なデータ更新を考慮して、古いキャッシュデータを効率的に無効化するために、更新時放送方式と接続時再放送方式を提案した。更新時放送方式では、オリジナルをもつ移動体が更新した際に、自身と相互接続している移動体に無効化情報を放送する。接続時再放送方式では、新たに接続した移動体どうしでタイムスタンプ表を比較し、更新したタイムスタンプ情報を改めて放送する。また、キャッシュデータの無効化方式により、更新したタイムスタンプ表の管理法が異なるため、各方式を用いる際のアクセス方法について述べた。

シミュレーション実験の結果から、接続時再放送方式が古いキャッシュデータにアクセスする回数を

より削減できるが、無効化情報によるトラヒックが大きくなることを確認した。

今後は、更新情報そのものを放送する方式について検討する予定である。

謝辞

本研究の一部は、文部科学省科学技術振興調整費「モバイル環境向 P2P 型情報共有基盤の確立」および日本学術振興会若手研究 (B)(13780330) の研究助成によるものである。ここに記して謝意を表す。

参考文献

- [1] D.J. Baker, J. Wieselthier, and A. Ephremides, "A distributed algorithm for scheduling the activation of links in a self-organizing, mobile, radio network," Proc. IEEE ICC'82, pp.2F6.1-2F6.5, 1982.
- [2] J. Broch, D.A. Maltz, D.B. Johnson, Y.C. Hu, and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," Proc. Mobicom'98, pp.85-97, 1998.
- [3] 原 隆浩, "アドホックネットワークにおけるデータ利用性向上のための複製配置," 電子情報通信学会和文論文誌 B, Vol. J84-B, No. 3, pp. 632-642, 2001.
- [4] 原 隆浩, "アドホックネットワークにおける周期的なデータ更新を考慮した複製配置方式," 電子情報通信学会和文論文誌 B, Vol. J84-B, No. 7, pp. 1391-1395, 2001.
- [5] M. Jiang, J. Li, and Y.C. Tay, "Cluster based routing protocol(CBRP)," Internet Draft, draft-ietf-manet-cbrp-spec-01.txt, 1999.
- [6] D.B. Johnson, "Routing in ad hoc networks of mobile hosts," Proc. IEEE Workshop on Mobile Computing Systems and Applications, pp.158-163, 1994.
- [7] S. Lee and C. Kim, "Neighbor supporting ad hoc multicast routing protocol," Proc. MOBI-HOC'2000, pp.37-44, 2000.
- [8] S. Lee, W. Su, J. Hsu, M. Gerla and R. Bagrodia, "A performance comparison study of ad hoc wireless multicast protocols," Proc. INFOCOM'2000, pp.565-574, 2000.
- [9] M.R. Pearlman and Z.J. Haas, "Determining the optimal configuration for the zone routing protocol," IEEE Journal on Selected Areas in Communications, Vol.17, No.8, pp.1395-1414, 1999.
- [10] C.E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," Proc. ACM SIGCOMM'94, pp.234-244, 1994.
- [11] C.E. Perkins and E.M. Royer, "Ad hoc on demand distance vector routing," Proc. IEEE Workshop on Mobile Computing Systems and Applications, pp.90-100, 1999.