

アドホックネットワークにおけるトポロジ変化に適応した複製の再配置について

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

あらまし 移動体のみで構成されるアドホックネットワークでは、移動体の移動に伴い、頻繁にネットワークが分断されるため、従来の固定ネットワークと比較して、データの利用性が低下してしまう。そこで、本稿では、各移動体が自身から N ホップ内の移動体のトポロジ情報と保持するデータ（複製）の情報を管理することで、ネットワークが分断される直前の状況を検出し、分断される前に局所的に複製配置を変更する方式を提案する。

キーワード アドホックネットワーク、複製配置、データ利用性、移動体計算環境

On Replica Relocation Adapting to Topology Changes in Ad Hoc Networks

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

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

2-1 Yamadaoka, Suita-shi, Osaka, 565-0871 Japan

E-mail: †{hideki,hara,nishio}@ist.osaka-u.ac.jp

Abstract In ad hoc networks, network division occurs frequently due to the movement of mobile hosts. Thus, data accessibility in ad hoc networks is lower than that in conventional fixed networks. In this paper, we propose a method that locally changes the replica allocation on some mobile hosts just before a network division occurs by detecting such a bodement. In our proposed method, each mobile host manages topology information on mobile hosts within N hops and that on data items held by them.

Key words ad hoc networks, replica allocation, data accessibility, mobile computing environment

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き、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環境が普及しつつある。特に、ルータ機能をもつ移動体のみで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への関心が高まっている。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、データの利用性が低下してしまう。例えば、図 1 のアドホックネットワークにおいて、中央の 2 台の移動体間の無線リンクが切断された場合、左側の 3 台の移動体はデータ D_2 に、右側の 3 台の移動体はデータ D_1 にアクセスできなくなる。この問題の解決手法として、移動体のもつオリジナルデータを他の移動体に複製として配置することが有効である [7] ~ [9]。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、ユーザ同士が直接通信するものだけでなく、災害時の救助活動やセンサネットワーク等のデータ共有といっ

た他の移動体のもつデータにアクセスするものも少なくない。したがって、アドホックネットワークにおいて、データ利用性の向上のため、効果的に複製を配置することは重要である。

これまでに、筆者らは、文献 [1] ~ [3] において、アドホックネットワークにおける複製配置方式を提案した。文献 [1] では、データ更新の発生しない環境を想定し、移動体が各データのアクセス頻度とネットワークトポロジを考慮して、自身のもつ限られたデータ領域内に周期的（再配置周期）に複製を配置する方式を提案した。文献 [2] では、データ更新が周期的に発生する環境を想定し、各データへのアクセス頻度、各データの次の更新発生までの時間、および複製配置時のネットワークトポロジを考慮して、周期的に複製を配置する方式を提案した。さらに、文献 [3] では、データ更新が不定期に発生する環境を想定して、データの利用性の向上と古い複製へのアクセス回数の削減を目的とした複製配置方式を提案した。この方式は、データ更新の発生間隔を確率的に計算し、各データに設けたタイムアウト時間に基づいて、周期的に各移動体に複製を配置するもの

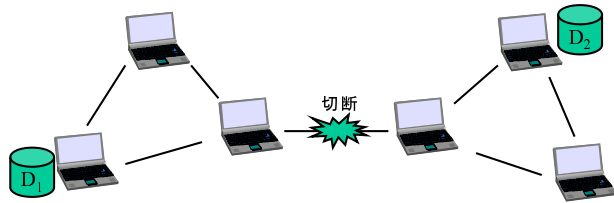


図 1 ネットワークの分断
Fig.1 Network division.

である。

文献 [1] ~ [3] の方式では、再配置周期ごとのネットワーク全体のトポロジを考慮して、全移動体の複製を再配置する。したがって、再配置周期において、複製の再配置のために大きなトラフィックを生じてしまう。さらに周期的な再配置を行うため、次の周期までにトポロジが大きく変化した場合、その時点の複製配置が効果的ではなくなり、データの利用率が低下してしまう場合がある。

そこで、筆者らは、文献 [5] において、トポロジ変化に適応した複製配置方式を提案した。この方式は、各移動体が隣接移動体のトポロジ情報と保持するデータの情報（保持データ情報）を管理し、ネットワークが分断される直前の状況を検出して、局所的に複製を再配置する。ここで、ネットワークが分断される直前とは、自身と隣接移動体との経路が一つしかなく、これらを接続する無線リンクが切断すると、ネットワークが分断される状況を示す。しかし、この方式では、移動体が隣接移動体の情報しか管理しないため、実際は自身から 2 ホップ以上先の移動体を介して相互接続している隣接移動体を、誤って分断直前の状況として検出する可能性がある。そのため、必ずしも効果的な複製の再配置とはなっていない。そこで、本稿では、文献 [5] の方式を拡張し、各移動体が自身から $N (\geq 1)$ ホップ内の移動体に関するトポロジ情報と保持データ情報を管理して、複製を再配置する方式を提案する。なお、本稿の提案方式において、 $N = 1$ の場合、文献 [5] の提案方式と等価になる。

以下では、2 章で想定環境について述べる。3 章で提案方式について説明する。4 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実験の結果を示す。最後に 5 章で本稿のまとめを述べる。

2. 想定環境

本稿では、アドホックネットワークにおいて、他の移動体のもつデータにアクセスする環境を想定する。各移動体は、自身のデータ領域に、他の移動体がオリジナルとしてもつデータの複製を作成する。移動体のデータアクセスは、自身のデータ領域内にもつオリジナルや複製に対してアクセスした場合、即座に成功とみなす。自身のデータ領域にアクセス対象のデータをもっていない場合、相互接続している移動体に問合せパケットをフラッディングし、それらの移動体のいずれかがそのオリジナルや複製をもつ場合、アクセスは成功とみなす。なお本稿において、相互接続している移動体とは、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。一方、自身、または相互接続している移動体がアクセス対象のオリジナルや複製をもたない場合、アクセスは失敗とみなす。

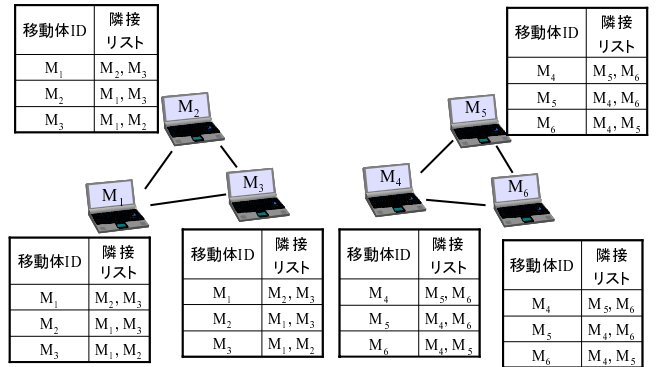


図 2 各移動体のもつ隣接リスト表 ($N \geq 1$)

Fig.2 Neighbor list table held by each mobile host ($N \geq 1$).

表 1 隣接リスト表の管理に用いられるパケット

Table 1 Packets for managing neighbor list table.

パケット名	属性
部分更新	移動体 ID, 接続 (切断) 移動体 ID, 状態, TTL
一括更新	移動体 ID, 隣接リスト, TTL

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。なお、簡単化のため、これらのサイズはすべて等しく、データ更新が発生しないものとする。
- 各移動体 $M_i (i = 1, \dots, m)$ は、自身のもつオリジナルデータ以外に、データ C 個分のデータ領域をもち、そこに複製を配置する。
- 各移動体の各データへのアクセス頻度は既知とし、時間的に変化しない。

3. 局所的な複製配置の変更

本章では、まず、ネットワークが分断される直前の状況を検出する際に必要なトポロジ情報の管理方法について述べる。次に、複製配置を変更する際に必要な保持データ情報の管理方法について述べる。さらに、ネットワークが分断される前に局所的に複製配置を変更する方式について説明する。

3.1 トポロジ情報の管理

ネットワークが分断される直前の状況を検出するために、各移動体は、他の移動体の隣接移動体に関する情報 (隣接情報) を保持する。ここで、各移動体がネットワークに存在する全移動体の隣接情報を保持する場合、その管理に要する通信トラフィックと計算オーバーヘッドが大きくなり、移動体の消費電力が大きくなる。これは、バッテリー容量に制約のある移動体計算環境では好ましくない場合が多い。そこで、本稿では、各移動体が自身から N ホップ内の移動体の隣接情報を隣接リスト表として保持するものとする。隣接リスト表には、自身、または自身から N ホップ内の移動体の識別子 (移動体 ID) とその移動体に隣接する全移動体の識別子のリスト (隣接リスト) が含まれる。図 2 に、 $N \geq 1$ の場合の各移動体の隣接リスト表を示す。隣接リスト表は、二つの移動体間で接続や切断が起こるたび

に更新される必要がある．表 1 に，そのために用いるパケットを示す．表中の部分更新パケットの“移動体 ID”はその送信元となる移動体の識別子，“接続（切断）移動体 ID”は接続（切断）した移動体の識別子，“状態”は接続（切断）移動体 ID が示す移動体と接続もしくは切断したか，“TTL”はパケットの転送ホップ数を表す．また，一括更新パケットの“隣接リスト”は，移動体 ID が示す移動体に隣接する全移動体の識別子のリストを表す．以下に，二つの移動体 (M_i と M_j) の接続時と切断時における M_i の動作について説明する．

【接続時の動作】

(1) M_i は，隣接リスト表における自身の隣接リストに M_j を挿入する．

(2) M_i は，もともと N ホップ内に接続していた移動体のもつ自身の隣接リストを更新するため， N ホップの範囲で部分更新パケットをフラディングする．このパケット内の移動体 ID には M_i ，接続移動体 ID には M_j ，状態には“接続”，TTL には N が含まれる．これを受信した移動体は，隣接リスト表における M_i の隣接リストに M_j を挿入した後で，TTL を 1 減らし，隣接移動体にパケットを再放送する．この動作は，TTL が 0 になるまで繰り返される．ただし，一度受信したパケットと同じものを受信した移動体は，TTL が 0 でなくても，再放送は行わない．

(3) M_j と M_j にもともと相互接続していた移動体は， M_i と M_i にもともと相互接続していた移動体の隣接情報を知らない可能性がある．そのため， M_i は，もともと自身から $n (< N)$ ホップ内で接続していた移動体の隣接情報を，TTL が $N - n$ に設定された一括更新パケットに含めて， M_i に送信する． M_j は，受信したパケットに含まれる隣接情報を隣接リスト表に挿入した後，TTL を 1 減らし， M_i 以外の全隣接移動体にパケットを転送する．さらに，これを受信した移動体は， M_j と同様の動作で，隣接リスト表を更新した後，TTL を 1 減らし，隣接移動体にパケットを再放送する．この動作は，TTL が 0 になるまで繰り返される．

以上の動作は， M_j においても同様に行われる．図 3 に，図 2 の状態から M_3 と M_4 が接続したときの $N = 2$ の場合における M_3 の動作を示す．図中の点線の矢印は部分更新パケットの送信，実線の矢印は一括更新パケットの送信を表し，各パケット内に含まれる情報を表 1 に記述した属性の順で括弧内に示す．ここで，隣接リスト表における灰色の領域は，移動体が新しく挿入した情報を表す．

【切断時の動作】

(1) M_i は，隣接リスト表における自身の隣接リストから M_j を， M_j の隣接リストから M_i を削除する．

(2) M_i は，隣接リスト表を用いて， N ホップ内の全移動体を確認する．具体的には，隣接リスト表における自身の隣接リストに含まれる識別子を始点として， N 回以内で辿れた識別子を記録する．その後，隣接リスト表において， N ホップ内で接続していない移動体の隣接リストを削除する．

(3) M_i は， N ホップ内の移動体のもつ自身の隣接リストを更新するため，これらの移動体に部分更新パケットを送信する．このパケット内の移動体 ID には M_i ，切断移動体 ID には M_j ，状態には“切断”，TTL には N が含まれる．これを受信

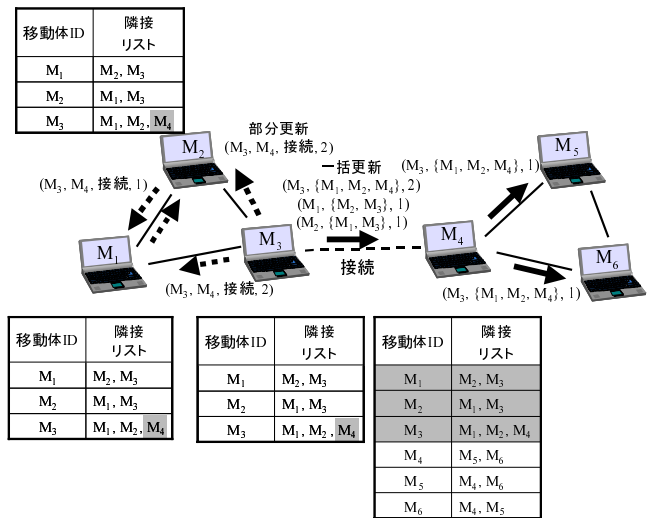


図 3 接続時における隣接リスト表の更新 ($N = 2$)
Fig. 3 Update of Neighbor list tables on connection ($N = 2$).

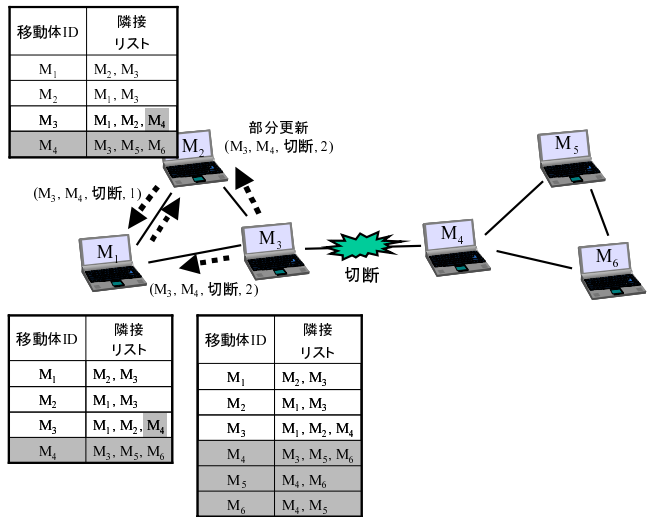


図 4 切断時における隣接リスト表の更新 ($N = 2$)
Fig. 4 Update of Neighbor list tables on disconnection ($N = 2$).

した移動体は，隣接リスト表における M_i の隣接リストから M_j を削除した後で，TTL を 1 減らし，部分更新パケットを隣接移動体に再放送する．また，これらの移動体は， M_i と同様の動作で，自身から N ホップ内で接続していない移動体の隣接リストを削除する．

以上の動作は， M_j においても同様に行われる．図 4 に，図 3 の状態から M_3 と M_4 が切断したときの $N = 2$ の場合における M_3 の動作を示す．ここで，隣接リスト表における灰色の領域は，移動体が削除した情報を表す．

3.2 保持データ情報の管理

提案方式において複製配置を変更する場合，各移動体の保持データ情報が必要になる．複製配置の変更時に，相互接続している全移動体に保持データ情報を問合せると，通信トラフィックが大きくなる．そこで，提案方式では，トポロジ情報と同様に， N ホップ内の移動体に関する保持データ情報を保持データ表として管理するものとする．保持データ表には，自身，または自身から N ホップ内の移動体の識別子（移動体 ID）と，その移

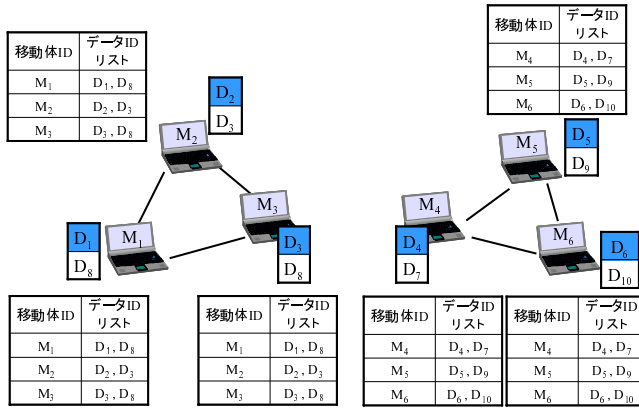


図 5 各移動体の保持データ表 ($N \geq 1$)

Fig. 5 Holding data table held by each mobile host ($N \geq 1$).

表 2 保持データ表の管理に用いられるパケット

Table 2 Packets for managing holding data table.

パケット名	属性
部分更新	移動体 ID, 配置 (破棄) データ ID, 状態, TTL
一括更新	移動体 ID, データ ID リスト, TTL

移動体が保持するデータの識別子のリスト (データ ID リスト) が含まれる。図 5 に、図 2 と同様のアドホックネットワークにおける $N \geq 1$ の場合の各移動体のもつ保持データ表を示す。図中の四角は移動体のデータ領域、その中の灰色がオリジナル、白色が複製を表す。

保持データ表は、 N ホップ内の移動体に関する情報を管理するため、隣接リスト表と同様に、二つの移動体間で接続や切断が起こるたびに更新する必要がある。また、移動体が複製配置を変更した場合にも更新する必要がある。保持データ表の管理に用いるパケットを表 2 に示す。表中の部分更新パケットの“移動体 ID”はその送信元となる移動体の識別子、“配置 (破棄) データ ID”は移動体 ID が示す移動体が配置 (破棄) したデータの識別子、“状態”はデータ ID が示すデータを配置もしくは破棄したか、“TTL”はパケットの転送ホップ数を表す。また、一括更新パケットの“データ ID リスト”は移動体 ID が示す移動体のもつデータの識別子のリストを表す。以下に、二つの移動体 (M_i と M_j) の接続時と切断時における M_i の動作、および M_i の複製配置の変更時の動作について説明する。

【接続時の動作】

二つの移動体が新しく接続した場合、前節で述べた隣接リスト表の更新が行われた後で、保持データ表の更新が行われる。

M_j と M_i にもともと相互接続していた移動体は、 M_i と M_i にもともと相互接続していた移動体の保持データ情報を知らない可能性がある。そのため、 M_i は、もともと自身から $n (< N)$ ホップ内で接続していた移動体の保持データ情報を、TTL が $N - n$ に設定された一括更新パケットに含めて、 M_i に送信する。 M_j は、受信したパケットに含まれる保持データ情報を保持データ表に挿入した後、TTL を 1 減らし、 M_i 以外の全隣接移動体に一括更新パケットを転送する。さらに、これを受信した移動体は、 M_j と同様の動作で、保持データ表を更新した後、TTL を 1 減らし、隣接移動体にパケットを再放送する。この動作は、 M_j においても同様に行われる。図 6 は、図 5 の状態

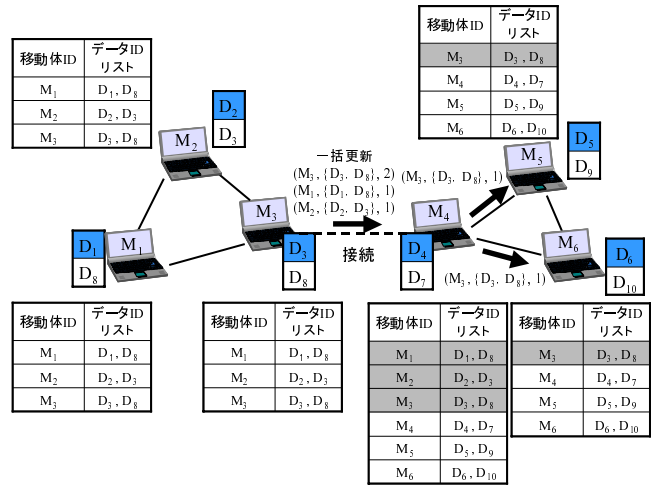


図 6 接続時における保持データ表の更新 ($N = 2$)

Fig. 6 Update of holding data tables on connection ($N = 2$).

から、 M_3 と M_4 が接続したときの $N = 2$ の場合における M_3 の動作を示す。図中の実線の矢印は一括更新パケットの送信を表し、パケット内に含まれる情報を、表 2 に記述した属性の順で括弧内に示す。ここで、保持データ表における灰色の領域は、移動体が新しく挿入した情報を表す。

【切断時の動作】

二つの移動体が切断した場合、前節で述べた隣接リスト表の更新が行われた後で、保持データ表の更新が行われる。

M_i は、保持データ表から、 N ホップ内で接続しなくなった移動体に関する保持データ情報を削除する。この動作は、 M_j においても同様に行われる。

【複製配置の変更時の動作】

M_i がデータ領域に保持していた D_k の複製を D_l の複製に変更した場合、下記のように保持データ表が更新される。

(1) M_i は、保持データ表の自身のデータ ID リストから D_k を削除し、 D_l を挿入する。

(2) M_i は、 N ホップ内の移動体のもつ保持データ表における M_i のデータ ID リストから D_k を削除するため、 N ホップの範囲で部分更新パケットをフラッディングする。このパケット内の移動体 ID には M_i 、破棄データ ID には D_k 、状態には“破棄”、TTL には N が含まれる。

(3) M_i は、 N ホップ内の移動体のもつ保持データ表における M_i のデータ ID リストに D_l を挿入するため、 N ホップの範囲で部分更新パケットをフラッディングする。このパケット内の移動体 ID には M_i 、配置データ ID には D_l 、状態には“配置”、TTL には N が含まれる。

図 7 は、図 6 の M_3 が複製を D_8 から D_9 に変更した際の、 M_3 による部分更新パケットの送信の様子 ($N = 2$) を示す。図中の点線の矢印は部分更新パケットの送信を表し、パケット内に含まれる情報を、表 2 に記述した属性の順で括弧内に示す。ここで、保持データ表における灰色の領域は、移動体が更新した情報を表す。

3.3 複製配置の変更

移動体の複製配置の変更は、ネットワークが分断される前に行うのが有効である。そこで、本稿では、移動体が隣接移動体

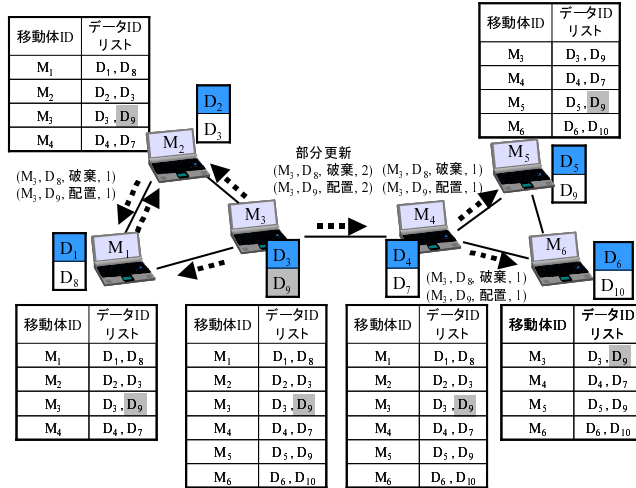


図 7 複製配置の変更時における保持データ表の更新 ($N = 2$)
 Fig. 7 Update of holding data tables on changing replica allocation ($N = 2$).

への経路を一つしか検出できない場合、つまり自身と隣接移動体を含む閉路を検出できない場合にネットワークが分断される直前であるとみなし、複製配置を変更する。この検出には、移動体のもつ隣接リスト表が用いられる。具体的には、この表における隣接移動体の隣接リストを始点として、他の移動体の隣接リストを順に辿っていき、終点として自身の識別子に到達できれば、隣接移動体と閉路を形成していることがわかる。その際、各移動体の識別子は、それぞれ一度だけ辿れるものとする。例えば、図 2 の M_3 は、隣接リスト表から隣接する M_1 と $\{M_3 \rightarrow M_1 \rightarrow M_2 \rightarrow M_3\}$ の閉路を形成することがわかる。隣接移動体との閉路の確認は、隣接リスト表を管理するための部分更新パケットや一括更新パケットの受信時に行われる。以下に、 M_i が、隣接する M_j との閉路を検出できない場合に行われる複製配置の変更の手順について説明する。

(1) M_i は、 M_j に複製配置変更メッセージを送信する。これを受信した M_j は、 M_i に自身のもつ保持データ表を送信する。

(2) M_i は、隣接リスト表を用いて、 M_j とは別の隣接移動体との閉路 (配置変更グループ) を検出する。その際、複数の閉路が検出できた場合は、閉路を構成する移動体数が多いものを配置変更グループとする。一方、閉路を一つも検出できなかった場合は、 M_i のみを配置変更グループとする。複製配置の変更は、配置変更グループに属する移動体に対して行われる。

(3) 各移動体のデータ領域には制限があるため、 M_i が M_j に要求する全データを配置変更グループに配置できるとは限らない。そのため、以下の手順で M_i が M_j に要求するデータを決定する。

i) 配置変更グループのもつデータと M_j からの保持データ表に含まれるデータに関して、配置変更グループに含まれる各移動体のアクセス頻度の和を計算し、その値を各データに対するグループのアクセス頻度とする。

ii) 配置変更グループが配置可能な複製の最大数を C_g とする。このとき、配置変更グループのもつオリジナルデータを除き、グループのアクセス頻度の和が高い上位 C_g 個までの

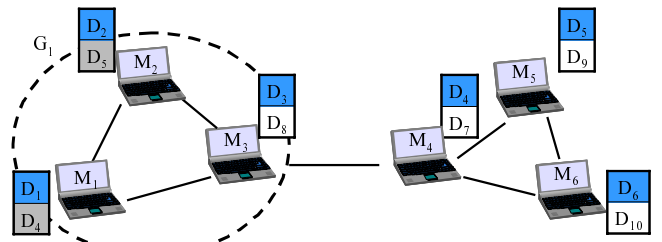


図 8 複製配置の変更
 Fig. 8 Changes of replica allocation ($N = 2$).

表 3 データのアクセス頻度
 Table 3 Access frequencies to data items.

データ	移動体						グループ
	M_1	M_2	M_3	M_4	M_5	M_6	
D_1	0.65	0.25	0.17	0.22	0.31	0.24	1.07
D_2	0.44	0.62	0.41	0.4	0.42	0.46	1.47
D_3	0.35	0.44	0.5	0.25	0.45	0.37	1.29
D_4	0.41	0.25	0.2	0.6	0.09	0.1	0.86
D_5	0.51	0.41	0.43	0.38	0.71	0.2	1.35
D_6	0.13	0.12	0.1	0.15	0.2	0.62	0.35
D_7	0.06	0.05	0.11	0.03	0.4	0.32	0.22
D_8	0.22	0.33	0.21	0.23	0.21	0.23	0.76
D_9	0.18	0.16	0.19	0.17	0.24	0.21	0.53
D_{10}	0.09	0.08	0.06	0.11	0.12	0.09	0.23

データの中で、 M_j からの保持データ表に含まれるものを要求する。

その後、 M_j は、 M_i に要求されたデータを N ホップ内の移動体に要求し、 M_i にこれらのデータを転送する。

(4) 要求したデータを受信した M_i は、配置変更グループに属する移動体の中からその複製の配置先を決定する。まず、データ領域に空きのある移動体が存在する場合、その移動体に配置する。そうでない場合、配置変更グループのもつデータにおいて、重複するものをもつ移動体が存在するならば、重複するものを破棄した後で、要求したデータの複製を配置する。その際、重複する中にオリジナルが存在する場合は複製をもつ移動体、複製のみが存在する場合は要求したデータのアクセス頻度が最も高い移動体の配置を変更する。一方、重複するものをもつ移動体が存在しない場合、グループのアクセス頻度が最小のデータの複製をもつ移動体に対して、その複製を破棄させ、要求したデータの複製を配置させる。

図 8 に、 $N = 2$ の場合において、図 6 の M_3 が隣接する M_4 との閉路を検出できず、配置変更グループ G_1 に属する移動体 (M_1, M_2, M_3) の複製配置を変更した様子を示す。各移動体 (M_1, \dots, M_6) と配置変更グループ G_1 の各データ (D_1, \dots, D_{10}) へのアクセス頻度は表 3 で表されるものとする。図中の薄い灰色の四角は、提案方式により変更された複製を表す。

4. 性能評価

本章では、提案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。

4.1 シミュレーション環境

1000[m] × 1000[m] の 2 次元平面内に、70 個の移動体 ($M = M_1, \dots, M_{70}$) が存在するものとした。各移動体は 2

表 4 パケットのサイズ

Table 4 Packet size.

パケット名	サイズ
隣接リスト表 部分更新	4
一括更新	2+ (隣接リストの要素数)
保持データ表 部分更新	4
一括更新	2+ (データ ID リストの要素数)
データ	250,000

次元平面内からランダムに目的地を決定し、0 から v (最大移動速度) [m/秒] の範囲でランダムに決定した速度で移動するものとした。移動体が目的地に到達すると、平均 10 単位時間の様分布に基づく間隔で一時停止した後、再び次の目的地を決定し、移動するものとした。なお、ここでの 1 単位時間は、1[秒]とした。各移動体の無線通信範囲は、半径 100[m] の円とした。

ネットワーク内には、70 種類のデータ ($D = D_1, \dots, D_{70}$) が存在し、 D_j は M_j ($j = 1, \dots, 70$) にオリジナルデータとして保持されるものとした。各移動体の各データ D_j へのアクセス頻度は等しく、 $p_j = 0.1 \times (1 + 0.001j)$ とした。初期位置として各移動体をランダムに配置し、シミュレーションの単位時間ごとに、各データのアクセス頻度に基づいてアクセス要求を発行させた。また、更新パケットに含まれる項目一つ分をデータ量の単位とし、各データ D_j のサイズを全て等しく 250,000 としたときの各パケット、およびデータのサイズを表 4 に示す。この場合、各パケットに含まれる項目の一つを 4 [バイト] と仮定すると、データのサイズは 1 [メガバイト] となる。各移動体は自身のデータ領域に最大 C 個までデータの複製を作成できるものとした。

シミュレーション実験では、文献 [1] で提案した三つの複製配置方式を比較対象とした。各方式の概要は、次の通りである。なお、これらの方式における再配置周期は 100 単位時間とした。SAF 方式：各移動体において、自身のデータ領域の許す限り、アクセス頻度の高いデータから順に複製を配置する。DAFN 方式：再配置周期ごとに SAF 方式を用いて各移動体に複製を暫定的に配置した後、隣接する移動体間で複製の重複を解消する。

DCG 方式：再配置周期ごとに安定度の高い移動体のグループ (二連結成分) を作成し、グループ内で複製を共有する。

以上のシミュレーション環境において、100,000 単位時間を経過させたときの下記の評価値について調べた。

- アクセス成功率

シミュレーション時間内に発生したアクセス要求の総数に対するアクセス成功回数の割合。

- 複製配置の変更に要するトラヒック

シミュレーション時間内に発生した複製配置の変更に要するデータ転送に要した通信ホップ数に、データサイズを掛けたものの総和。

- トポロジ情報の管理に要するトラヒック

シミュレーション時間内に発生した隣接リスト表の管理に用いる部分更新と一括更新パケットの送信に要した通信ホップ数に、各パケットのサイズを掛けたものの総和。

- 保持データ情報の管理に要するトラヒック

シミュレーション時間内に発生した保持データ表の管理に用

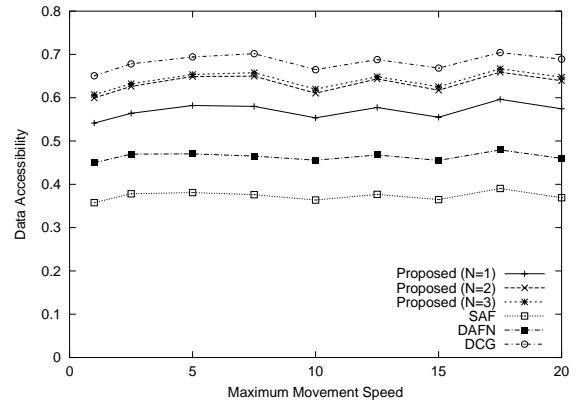


図 9 最大移動速度 v とアクセス成功率

Fig. 9 Maximum movement speed v and data accessibility

いる部分更新と一括更新パケットの送信に要した通信ホップ数に、各パケットのサイズを掛けたものの総和。

4.2 最大移動速度 v の影響

まず、最大移動速度 v を変化させたときの提案方式の性能を調べた。その結果を、図 9、図 10、図 11、図 12 に示す。これらの図において、横軸は最大移動速度 v を示す。縦軸は、図 9 ではアクセス成功率、図 10 では複製配置の変更に要するトラヒック、図 11 ではトポロジ情報の管理に要するトラヒック、図 12 では保持データ情報の管理に要するトラヒックを示す。グラフ中の 'Proposed' は本稿の提案方式を示し、シミュレーション実験では $N = 1, 2, 3$ の場合について調べた。

図 9 の結果から、提案方式は、SAF 方式と DAFN 方式よりアクセス成功率が高いことがわかる。これは、提案方式がグループ内 (閉路) で複製を共有するため、各移動体が多種類のデータにアクセスできるからである。一方、提案方式は、DCG 方式よりアクセス成功率が低いことがわかる。これは、DCG 方式では、ネットワーク全体のトポロジを考慮して作成したより大きなグループ内で多種類の複製を共有するためである。これに対し、提案方式は、ネットワークの分断前に、仮に分断した場合の二つの部分ネットワーク内でアクセス頻度の高いデータにアクセスできるように、重複をもたせて複製配置するため、分断前の一つのネットワーク内で共有するデータの種数が少なくなる。また、提案方式において、 N を大きくした場合、アクセス成功率が高くなることがわかる。これは、 N を大きくすると、ネットワークの分断をより正確に検出できるだけでなく、多くの移動体の保持データ情報を利用して、効果的に複製配置できるからである。また、最大移動速度はアクセス成功率にほぼ影響していないことがわかる。

図 10 の結果から、最大移動速度が大きくなると、提案方式の複製配置の変更に要するトラヒックが大きくなること分かる。これは、移動体同士の接続や切断回数が多くなるため、複製配置の変更が多く発生するからである。また、提案方式の N が大きくなると、複製配置の変更に要するトラヒックが大きくなること分かる。これは、複製配置を変更する際に、より広範囲の移動体間でデータ転送が行われるからである。

図 11 と図 12 の結果から、最大移動速度が大きくなると、提案方式のトポロジ情報と保持データ情報の管理に要するトラヒックが大きくなること分かる。これは、移動体同士の接続

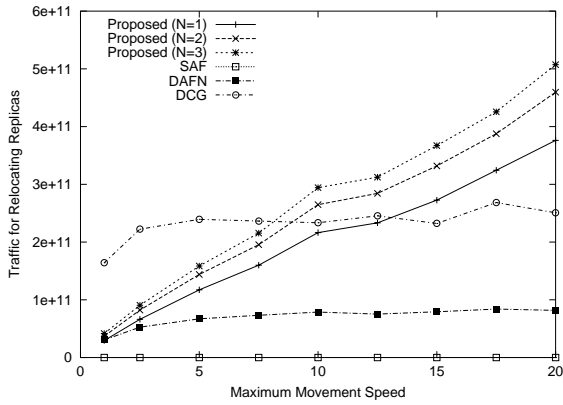


図 10 最大移動速度 v と複製配置の変更に要するトラフィック
Fig.10 Maximum movement speed v and traffic for relocating replicas.

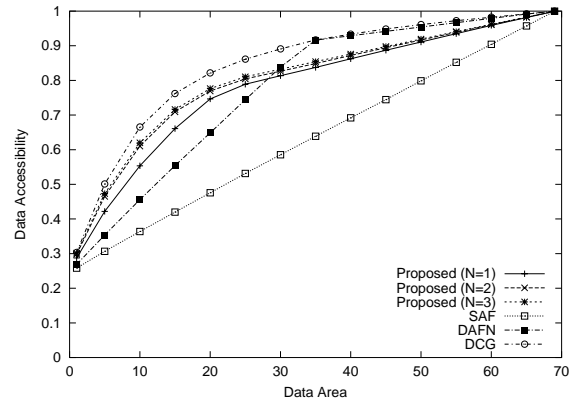


図 13 データ領域 C とアクセス成功率
Fig.13 Data area C and data accessibility.

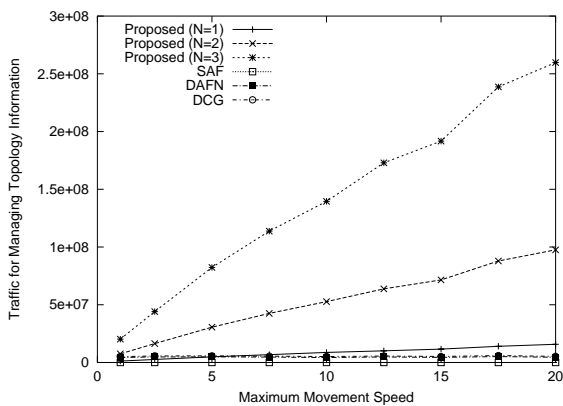


図 11 最大移動速度 v とトポロジ情報の管理に要するトラフィック
Fig.11 Maximum movement speed v and traffic for managing topology information.

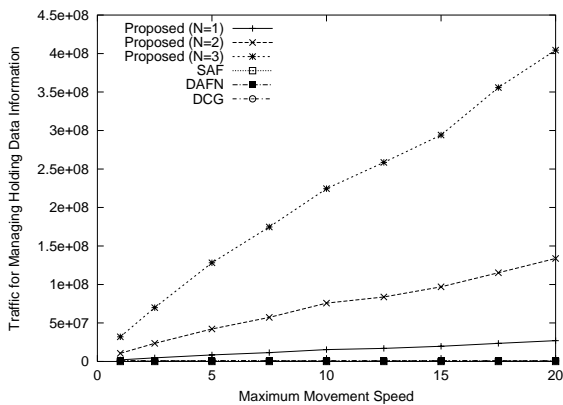


図 12 最大移動速度 v と保持データ情報の管理に要するトラフィック
Fig.12 Maximum movement speed v and traffic for managing holding data information.

や切断が頻繁に発生し、各情報を管理するための部分更新と一括更新パケットが頻繁に送信されるからである。また、提案方式の N が大きくなると、両トラフィックが大きくなる。これは、提案方式の性質上、自明の結果になっている。

以上の結果より、提案方式は、移動体が低速で移動し、移動体間の接続や切断があまり発生しない環境において、小さなトラフィックで DCG 方式に次ぐアクセス成功率を示すため、特に

有効である。また、提案方式では、複製配置の変更に要するトラフィックが、トポロジ情報や保持データ情報の管理に要するトラフィックよりも非常に大きなものとなる。なお、DAFN 方式と DCG 方式に要する各トラフィックは再配置周期に大きく依存するため、提案方式との比較において、この結果があらゆる環境において必ずしも妥当であるとは限らない。

4.3 データ領域 C の影響

次に、データ領域 C を変化させたときの提案方式の性能を調べた。その結果を、図 13、図 14、図 15、図 16 に示す。これらの図において、横軸はデータ領域 C を示す。縦軸は、図 13 ではアクセス成功率、図 14 では複製配置の変更に要するトラフィック、図 15 ではトポロジ情報の管理に要するトラフィック、図 16 では保持データ情報の管理に要するトラフィックを示す。

図 13 の結果から、データ領域が大きくなると、提案方式のアクセス成功率は最初 DAFN 方式より高くなるが、ある値を超えると、DAFN 方式より低くなるのがわかる。データ領域が小さい場合、提案方式はグループ内で複製を共有するため、各移動体が多種類のデータにアクセスできる。一方、データ領域が大きい場合、DAFN 方式では隣接する二つの移動体間ではほぼ全てのデータを保持できるが、提案方式では重複をもたせて複製配置するため、提案方式は DAFN 方式よりアクセス成功率が低くなる。また、提案方式の N が大きくなると、アクセス成功率が高くなるのがわかる。これは、図 9 の結果における考察と同様である。

図 14 の結果から、データ領域が大きくなると、提案方式の複製配置の変更に要するトラフィックが最初は高くなるが、ある値を超えると減少することがわかる。データ領域が小さい場合、各移動体が配置を変更する複製の数が少なくなるため、トラフィックがあまり生じない。一方、データ領域が大きい場合、各移動体がほぼ全種類のデータを保持でき、複製配置の変更自体があまり発生しなくなり、トラフィックがあまり生じない。また、提案方式は、複製配置の変更に要するトラフィックが最大値となるデータ領域の値が、DAFN 方式や DCG 方式と比べて、小さくなる。これは、提案方式が、DAFN 方式と DCG 方式に比べて、データ領域が小さい場合でも、移動体間で多種類の複製を共有するため、複製配置の変更時に転送する複製の数が多くなるからである。

図 15 の結果から、全ての方式において、トポロジ情報の管

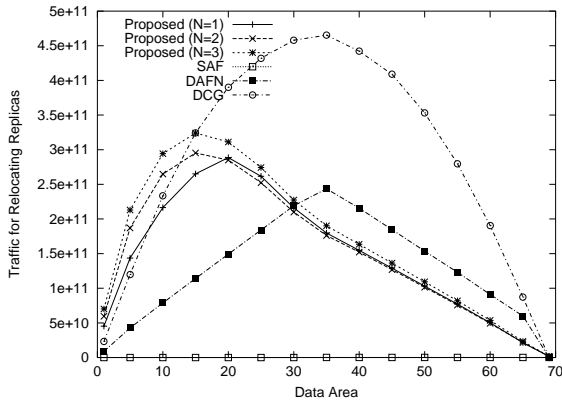


図 14 データ領域 C と複製配置の変更によるトラフィック
Fig. 14 Data area C and traffic for relocating replicas.

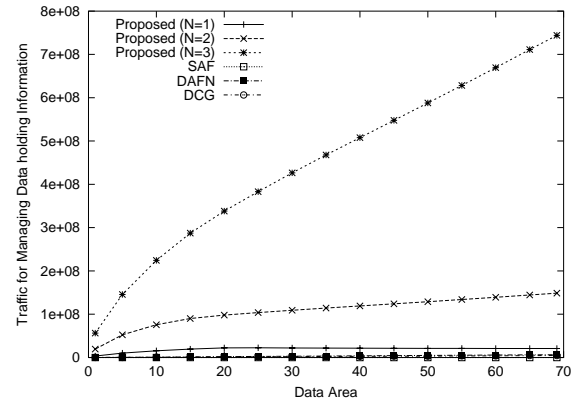


図 16 データ領域 C と保持データ情報の管理によるトラフィック
Fig. 16 Data area C and traffic for managing holding data information.

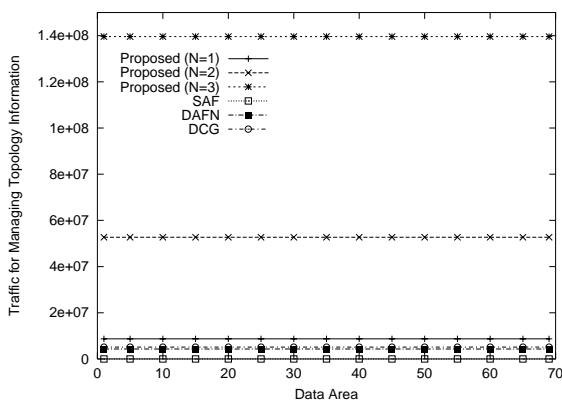


図 15 データ領域 C とトポロジ情報の管理によるトラフィック
Fig. 15 Data area C and traffic for managing topology information.

理に要するトラフィックは、データ領域の大きさに影響されず、一定であることがわかる。提案方式では、トポロジ情報の管理パケットは移動体同士の接続や切断が発生したときのみ送信される。一方、DAFN 方式と DCG 方式では再配置周期のときのみ送信される。

図 16 の結果から、データ領域が大きくなると、提案方式の保持データ情報の管理に要するトラフィックが大きくなることがわかる。これは、各移動体のもつデータの種別が多くなるため、保持データ情報の一括更新パケットのサイズが大きくなるからである。

5. まとめ

本稿では、アドホックネットワークにおいて、トポロジ変化に適応した複製配置方式を提案した。この方式では、各移動体が $N(\geq 1)$ ホップ内の移動体のトポロジ情報と保持データ情報を管理することで、ネットワークの分断前に局所的な複製配置の変更を行う。

シミュレーション実験の結果より、提案方式は、移動体が低速で移動し、移動体間の接続や切断があまり発生しない環境において、小さなトラフィックで、高いアクセス成功率を示すため、特に有効であることを確認した。また、各移動体のデータ領域が小さい環境でも、アクセス成功率を高めることを確認した。さらに、トポロジ情報と保持データ情報を管理する移動体の範

囲を広くすると、アクセス成功率を高める反面、トラフィックが大きくなることを確認した。

今後は、様々な移動体の移動モデルやデータへのアクセスモデルを用いた性能評価を行い、提案方式が有効となる環境について検討する予定である。また、筆者らは文献 [4], [6] で、データ更新を考慮した複製管理方法を提案している。そこで、データ更新の発生する環境で、文献 [4], [6] の提案方式も併用する環境を想定し、本稿での提案方式の性能評価を行う予定である。

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」、および文部科学省 21 世紀 COE プログラム「ネットワーク共生環境を築く情報技術の創出」の研究助成によるものである。ここに記して謝意を表す。

文 献

- [1] 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.
- [2] 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.
- [3] 林 秀樹, 原 隆浩, 西尾章治郎, “アドホックネットワークにおけるデータ更新間隔を考慮した複製配置,” 情報処理学会論文誌, Vol. 44, No. 12, pp. 2898–2911, 2003.
- [4] 林 秀樹, 原 隆浩, 西尾章治郎, “不定期データ更新が発生するアドホックネットワークにおけるキャッシュ無効化方式,” 情報処理学会論文誌, Vol. 44, No. 12, pp. 3139–3151, 2003.
- [5] 林 秀樹, 原 隆浩, 西尾章治郎, “アドホックネットワークにおけるトポロジ変化に適応した動的複製配置について,” 情報処理学会データベースシステム研究会報告 (2004-DBS-132), Vol. 2004, No. 3, pp. 1–8, 2004.
- [6] 林 秀樹, 原 隆浩, 西尾章治郎, “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更新データ配布方式,” 電子情報通信学会和文論文誌 D-I, Vol. J87-D-I, No. 2, pp. 188–201, 2004.
- [7] G. Karumanchi, S. Muralidharan, and R. Prakash, “Information dissemination in partitionable mobile ad hoc networks,” Proc. Symposium on Reliable Distributed Systems (SRDS'99), pp. 4–13, 1999.
- [8] J. Luo, J.P. Hubaux, and P. Eugster, “PAN: providing reliable storage in mobile ad hoc networks with probabilistic quorum systems,” Proc. ACM MobiHoc'03, pp. 1–12, 2003.
- [9] K. Wang and B. Li, “Efficient and guaranteed service coverage in partitionable mobile ad-hoc networks,” Proc. IEEE Infocom'02, Vol. 2, pp. 1089–1098, 2002.