

アドホックネットワークにおける更新ログ配布方式の性能評価について

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 1-5

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

あらまし これまでに筆者らは、アドホックネットワークにおいて、移動体による複製への暫定的なアクセスの有効性を効率的に確認するため、各移動体が更新ログを管理して、必要に応じて他の移動体に更新ログを配布する方式を提案した。本稿では、提案方式の有効性を検証するために行ったシミュレーション実験について説明する。シミュレーション結果から、データ更新が頻繁に発生する環境では、トポロジ変更時に更新ログを配布する ULD-C 方式が有効であることを確認した。一方、データ更新があまり発生しない環境では、データアクセス時にオリジナルをもつ移動体から更新ログを配布する ULD-DA+方式が有効であることを確認した。

キーワード アドホックネットワーク、複製配置、データ更新、データ利用性、移動体計算環境

On Performance Evaluation of Update Log Dissemination Methods in Ad Hoc Networks

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

[†] Dept. of Multimedia Eng., Graduate School of Information Science and Technology, Osaka University
1-5 Yamadaoka, Suita-shi, Osaka, 565-0871 Japan

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

Abstract We have proposed update log dissemination methods in which each mobile host manages update logs of data items in an ad hoc network and transmits them as needed. These methods enable mobile hosts to efficiently check the validity of tentative accesses to replicas. In this paper, we present simulation results to evaluate the performance of our proposed methods and verify their effectiveness. The results show that when data items are frequently updated, the ULD-C method, which disseminates update logs in the entire network on topology change, works well. On the other hand, when data items are not frequently updated, the ULD-DA+ method, in which a mobile host holding the original data item disseminates update logs on data access, works well.

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

1. はじめに

近年、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、データの利用率が低下してしまう。例えば図 1 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。この問題を解決するため、移動体に他の移動体が保持するオリジナルデータの複製を配置することが有効である。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、災害時の救助活動やセンサネットワークにおけるデータ共有が

あり、移動体が他の移動体のもつデータにアクセスするものも多い。したがって、アドホックネットワークでは、データの利用率を向上させるための効果的な複製配置が重要な課題となる [5]~[7]。これまでに筆者らは、文献 [1] において、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、周期的（再配置周期ごと）に複製配置を決定する方式を提案した。

ここで実環境では、一般的にデータ更新が発生するため、移動体がオリジナルとバージョンの異なる古い複製にアクセスする可能性がある。古い複製へのアクセスはロールバック処理を発生させるため、消費電力が重要な問題となる移動体計算環境では好ましくない場合が多い。そこで筆者らは、文献 [2] において、古い複製へのアクセス回数を削減するため、各移動体が無効化情報を放送して、古い複製を無効化する方式を提案した。

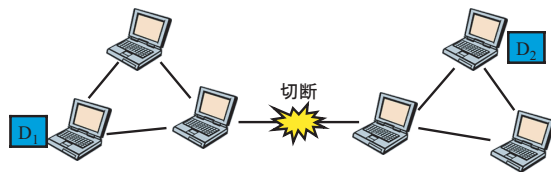


図 1 ネットワークの分断
Fig. 1 Network division.

さらに文献 [3] では、データの利用率を向上させるため、移動体が更新データを配布して、古い複製を更新する方式を提案した。

データ更新の発生する環境におけるもう一つの問題として、オリジナルをもつ移動体と接続していない移動体は、自身のもつ複製がオリジナルと同じバージョンであるかを確認できない。したがって、複製へのアクセスは暫定的に行い、後に、移動体がアクセスしたデータの更新ログを参照してアクセスの有効性を確認する必要がある。そこで文献 [4] では、複製へのアクセスの有効性を効率的に確認できるように、各移動体がデータの更新ログを管理して、必要に応じて他の移動体に配布する方式を提案した。しかし、文献 [4] では方式の提案のみを行っており、提案方式の性能評価や有効性の検証は行っていなかった。本稿では、筆者らが提案した更新ログ配布方式の性能をシミュレーション実験により評価し、その有効性を検証する。

以下では、2. で想定環境について述べる。3. で複製の管理方法について述べた後、4. で更新ログの配布方式について説明する。5. で提案方式のシミュレーション実験の結果を示す。最後に 6. で結論と今後の課題について述べる。

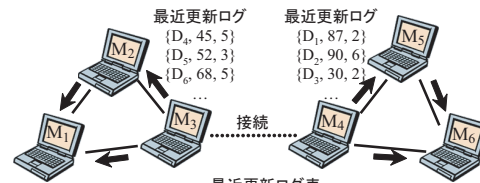
2. 想定環境

本稿では、データ更新の発生するアドホックネットワークにおいて、移動体が他の移動体のもつデータの複製を作成する環境を想定する。文献 [4] で提案した更新ログ配布方式は、複製配置方式とは独立に動作するため、特に配置方式を限定しない。

移動体のデータへのアクセス要求は、移動体がオリジナルにアクセスした場合、またはオリジナルと同じタイムスタンプ（バージョン）の複製にアクセスした場合に成功（有効）とみなし、オリジナルと異なるタイムスタンプの古い複製にアクセスした場合は失敗（無効）とみなす。アクセス要求は、自身もしく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をもつ場合、即座に成功となる。ここで、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。自身もしく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をもたず、複製のみをもつ場合、暫定的に複製にアクセスする。その後、移動体がアクセスしたデータの更新ログを参照して、複製へのアクセスの有効性を確認する。そのアクセスが無効だった場合は、必要に応じてロールバック処理を行う。一方、自身もしく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、アクセス要求は即座に失敗となる。

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 個のデータ（識別子： D_1, D_2, \dots, D_n ）が存在し、各々



		最近更新ログ表						
		D_1	D_2	D_3	D_4	D_5	D_6	...
M_1		87, 2	90, 6	30, 2	23, 3	10, 1	20, 2	...
M_2		87, 2	90, 6	30, 2	23, 3	10, 1	20, 2	...
比較	M_3	87, 2	90, 6	30, 2	23, 3	10, 1	20, 2	...
	M_4	50, 1	65, 4	20, 1	45, 5	52, 3	68, 5	...
M_5		50, 1	65, 4	20, 1	45, 5	52, 3	68, 5	...
M_6		50, 1	65, 4	20, 1	45, 5	52, 3	68, 5	...

図 2 接続時の最近更新ログのフラッディング
Fig. 2 Flooding of latest update logs on connection.

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。

- 各データは、そのオリジナルをもつ移動体によって更新される。更新発生後の古い複製は無効なものとなる。

3. 複製の管理

本稿では、移動体が古い複製にアクセスした場合、必要に応じてロールバック処理が発生するものと想定する。したがって、複製へのアクセスの有効性を確認する方法とは別に、複製のバージョンをできる限り最新のものに保つ管理方法が必要になる。そこで本稿では、文献 [2] の提案方式を参考にして、以下の管理方法を用いる。

各移動体は、各データの最近の更新ログを記録する最近更新ログ表を保持する。この表は、データ識別子、タイムスタンプ、および更新回数を含む。タイムスタンプは、データの更新時刻を示す。更新回数は、データ識別子で示すデータを過去何回更新したかを示し、初期値は 0 で、更新する度にインクリメントされる。まず、移動体がデータを更新すると、更新したデータの識別子、タイムスタンプ、および更新回数を含む最近更新ログをフラッディングする。これを受信した移動体は、最近更新ログ表にその情報を記録する。さらに、その古い複製を保持する場合は、複製を無効化する。

また、新しく接続した二つの移動体は最近更新ログ表を比較して、お互いの古い情報を更新する。接続した当事者は、最近更新ログ表の更新部分を、自身にもともと相互接続している移動体にフラッディングする。最近更新ログを受信した移動体は、自身の最近更新ログ表の更新と古い複製の無効化を行う。図 2 は、接続した M_3 と M_4 が最近更新ログをフラッディングする動作を示す。表中の要素は最近更新ログを示し、左がタイムスタンプ、右が更新回数を示す。

この方法により、相互接続している移動体間で同じ最近更新ログ表と同じタイムスタンプの複製を保持できる。つまり、移動体はどの移動体のもつデータ（複製）にアクセスしても、同じタイムスタンプのものとなる。このことを考慮して本稿では、以下のアクセス方法を用いる。表 1 は、移動体がデータアクセスで用いるパケットを示す。表中の ID は識別子、フラグはそのデータがオリジナルまたは複製かを表すものである。

- (1) アクセス要求する移動体は、自身がアクセス対象のオリジナルをもつならば、即座にアクセスする。そうでない場合、

表 1 データアクセスで用いるパケット

Table 1 Packets for data access.

パケット名	要素
データ問合せ	送信元 ID, データ ID
データ問合せ返信	送信元 ID, 宛て先 ID, データ ID, フラグ
データ要求	送信元 ID, 宛て先 ID, データ ID

相互接続している移動体にデータ問合せパケットをフラッディングする。

(2) 問合せパケットを受信した移動体は、要求されたデータのオリジナルまたは複製をもつ場合、要求してきた移動体にデータ問合せ返信パケットを送信する。

(3) アクセス要求した移動体は、一定の待ち時間の間に返信してきた移動体の中で、自身から最小ホップ数で到達できる移動体にデータ要求パケットを送信する。なお、一つの移動体からも返信がなかった場合、その要求は失敗となる。

4. 更新ログ配布方式

本章では、文献 [4] で提案した更新ログ配布方式について説明する。

4.1 アクセスログの管理

複製に暫定的にアクセスした移動体は、後でその更新ログを参照して、アクセスの有効性を確認する。具体的には、移動体が複製にアクセスした時点で、その複製がオリジナルと同じタイムスタンプであったかを確認する。そのため、複製にアクセスした移動体は、アクセスログをアクセス履歴表に記録する。この表は、データ識別子と、その複製へのアクセスログ(アクセス時刻, アクセスした複製のタイムスタンプ, その更新回数)のリストを含む。ここで、移動体が更新履歴を長時間参照できず、有効性を確認できない場合、アクセス履歴表のサイズが大きくなる。また、アプリケーションによっては、移動体が複製へのアクセスの有効性を確認するまでに、それほど長い時間は待てないものもある。例えば、災害時の救助活動において、移動体(救助隊員)が他の移動体の作業状況データにアクセスして、自身の行動を決定する場合を考える。このとき、複製へのアクセスの有効性を確認するまでに長い時間がかかり、その結果が無効だった場合、救助隊員が長時間行った作業が無駄になってしまう。そこで、アクセス履歴表には、過去 TO_a 時間(アクセスタイムアウト時間)以内のアクセスのログだけを保持する。アクセスしてから TO_a 時間経過しても有効性を確認できない場合は、アクセスが無効となり、そのログをアクセス履歴表から削除する。

4.2 オリジナルをもつ移動体による更新ログの配布

移動体は、自身のもつオリジナルを更新すると、更新ログを更新履歴表に記録する。更新履歴表は、オリジナルの識別子と更新ログ(タイムスタンプ, 更新回数)のリストを含む。ここで、各移動体は TO_a 時間経過したアクセスのログを削除するため、更新ログを保持し続けても、アクセスの有効性の確認に用いられなくなる。そこで更新履歴表には、過去 TO_u ($\gg TO_a$) 時間(更新ログタイムアウト時間)以内に更新されたデータのログだけを保持する。

ここで、複製にアクセスした移動体がオリジナルをもつ移動体から更新ログを得るには、オリジナルをもつ移動体との接

表 2 ULD-DA(+) 方式で用いるパケット

Table 2 Packets for ULD-DA(+) method.

パケット名	要素
更新ログ要求	送信元 ID, 宛て先 ID, データ ID, TS リスト
TS 問合せ	送信元 ID, データ ID, TTL
TS 問合せ返信	送信元 ID, 宛て先 ID, データ ID, TS リスト

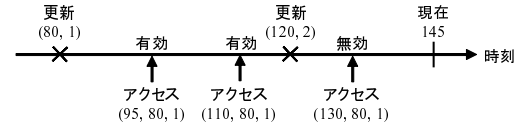


図 3 複製へのアクセスの有効性の確認

Fig. 3 Checking validity of accesses to replicas.

続を検出する必要がある。提案方式では、この検出に移動体のデータアクセスの動作を利用する。この過程では、要求されたデータのオリジナルをもつ移動体は、要求を発行した移動体にオリジナルの保持を示すフラグを含むパケットを返信する。これにより、要求を発行した移動体は、オリジナルをもつ移動体との接続を検出できる。

以下では、データアクセスの過程で、オリジナルをもつ移動体が更新ログを配布する ULD-DA (Update Log Dissemination on Data Access) 方式について述べる。次に、オリジナルをもつ移動体から更新ログを受信した移動体が、周囲の移動体に更新ログを再配布する ULD-DA+方式について述べる。表 2 に、これらの方式で用いるパケットを示す。表中の TS はタイムスタンプ, TTL (Time To Live) はパケットの送信範囲を示す。

4.2.1 ULD-DA 方式

ULD-DA 方式では、オリジナルをもつ移動体がアクセス要求した移動体に更新ログを配布する。このとき、オリジナルをもつ移動体が、更新ログをすべて配布すると、トラフィックが非常に大きくなる。ここで、複製へのアクセスの有効性は、移動体がアクセスした複製のタイムスタンプの次のタイムスタンプがわかれば確認できる点に着目する。図 3 は、データの更新ログと、移動体のアクセスログを数直線で表したものである。数直線上の \times 印は更新の発生を示し、括弧内の情報は更新ログであり、左がタイムスタンプ, 右が更新回数を示す。矢印は複製へのアクセスの発生を示し、括弧内の情報はアクセスログであり、左からアクセス時刻, タイムスタンプ, 更新回数を示す。このとき、タイムスタンプ 80 の複製へのアクセスの有効性は、タイムスタンプ 120 の更新ログにより確認できる。例えば、この複製への時刻 95 と 110 のアクセスは有効であるが、時刻 130 のアクセスは無効となる。したがって、この方式では、オリジナルをもつ移動体は、アクセス要求してきた移動体が暫定的にアクセスした複製のタイムスタンプの情報を受信し、その複製の次の更新に対する更新ログを配布する。以下に、移動体 M_i が、データ D_j のオリジナルをもつ移動体 M_j との接続を検出したときの動作を示す。

(1) M_i は、 D_j のアクセスログを保持している場合、 M_j に更新ログ要求パケットを送信する。このパケットの送信元 ID は M_i , 宛て先 ID は M_j , データ ID は D_j , TS リストは M_i が過去 TO_a 時間以内にアクセスした D_j (複製) のタイムスタンプのリストを含む。

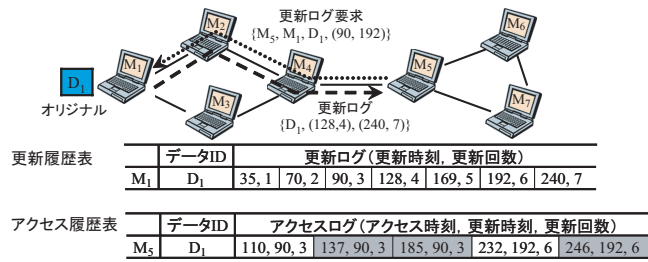


図 4 ULD-DA 方式における更新ログの配布

Fig. 4 Dissemination of update logs in ULD-DA method.

(2) M_j は、自身の更新履歴表から、受信したパケット内の各タイムスタンプ以降の次の更新に該当する更新ログを、 M_i に配布する。

(3) M_i は、受信した更新ログにより、 D_j の複製へのアクセスの有効性を確認する。確認できたアクセスのログは、アクセス履歴表から削除する。

図 4 は、移動体 M_5 が D_1 のオリジナルをもつ移動体 M_1 に更新ログを要求した際の動作を示す。パケット内の情報は、表 2 におけるパケットの要素の順番に従い括弧内に示す。アクセスの有効性を確認した結果、アクセスログの白色が有効だったもの、灰色が無効だったものを示す。

4.2.2 ULD-DA+ 方式

ULD-DA 方式では、アクセス要求した移動体のみが複製へのアクセスの有効性を確認できる。そのため、アクセス要求しない移動体は、複製へのアクセスの有効性を長時間確認できない可能性がある。そこで ULD-DA+方式では、アクセス要求した移動体がオリジナルをもつ移動体から更新ログを受信した後、自身から $N_{DA}(\geq 1)$ ホップ内の移動体に更新ログを再配布する。以下に、移動体 M_i が、データ D_j のオリジナルをもつ移動体 M_j との接続を検出したときの動作を示す。

(1) M_i は、自身から N_{DA} ホップ内の移動体がアクセスした D_j の複製のタイムスタンプを調べるため、TS 問合せパケットを N_{DA} ホップの範囲でフラッディングする。このパケットの送信元 ID は M_i 、データ ID は D_j 、TTL は N_{DA} となる。

(2) TS 問合せパケットを受信した移動体は、 D_j の複製にアクセスしていた場合、 M_i に TS 問合せ返信パケットを送信する。このパケットの送信元 ID は自身の識別子、宛て先 ID は M_i 、データ ID は D_j 、TS リストはアクセスした複製のタイムスタンプのリストとなる。その後、問合せパケットの TTL を一つ減らして、隣接移動体に放送する。この操作は、TTL が 0 になるまで繰り返される。

(3) M_i は、自身がアクセスした D_j の複製のタイムスタンプも含めて、受信した返信パケット内の TS リストの情報をまとめて、更新ログ要求パケットを M_j に送信する。

(4) M_j は、要求パケット内の各タイムスタンプ以降の次の更新に該当する更新ログを求めて、 M_i に配布する。

(5) M_i は、受信した更新ログを用いて、 D_j の複製へのアクセスの有効性を確認し、そのログをアクセス履歴表から削除する。その後、自身から N_{DA} ホップ内の各移動体にアクセスの有効性の確認に必要な更新ログを配布する。

(6) M_i からパケットを受信した移動体は、 M_i と同様に複製へのアクセスの有効性を確認する。

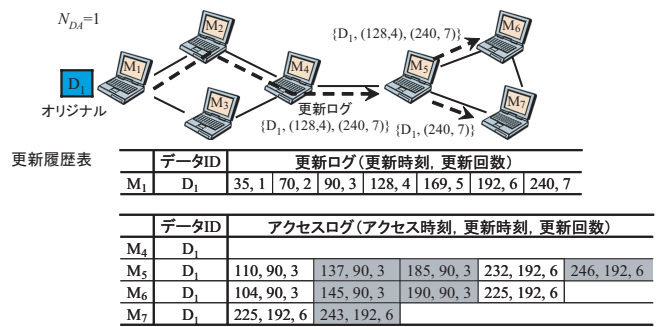


図 5 ULD-DA+方式における更新ログの配布

Fig. 5 Dissemination of update logs in ULD-DA+ method.

図 5 は、 M_1 から更新ログを受信した M_5 が、1 ホップ内の移動体に更新ログを配布する動作を示す。アクセスの有効性を確認した結果、アクセスログの白色が有効だったもの、灰色が無効だったものを示す。

ULD-DA+方式は、ULD-DA 方式と比べて、各移動体が複製へのアクセスの有効性を確認するまでの時間が短縮されるものと考えられる。しかし、アクセス要求した移動体が周囲の移動体に更新ログを配布するため、トラヒックが大きくなる。

4.3 その他の移動体による更新ログの配布

ULD-DA(+方式)では、移動体がオリジナルをもつ移動体に接続した場合にのみ、複製へのアクセスの有効性を確認できる。しかし、アドホックネットワークでは、移動体が特定の移動体に接続できる保証はない。そこで本節では、ULD-DA(+方式)に加え、各移動体が他の移動体のもつオリジナルの更新ログも管理し、接続した移動体同士で更新ログを配布する ULD-C (Update Log Dissemination on Connection) 方式について説明する。各データの更新履歴表に保持する更新ログは、前節と同様に過去 TO_u 時間以内に更新されたものとする。

4.3.1 ULD-C 方式

移動体が自身のもつオリジナルを更新するとき、そのログを更新履歴表に記録する。また、3. における複製の管理方法により、移動体が最近更新ログを受信したときに、その情報を更新履歴表に記録する。さらに、移動体同士が新しく接続(隣接)したとき、両者の更新履歴表を比較し、お互いのもたない更新ログをそれぞれがもともと相互接続していた移動体に配布する。このとき、ネットワーク全体に更新ログを配布すると、トラヒックが大きくなるため、移動体の消費電力に厳しい制約のある環境では適さない。そこで、適用環境に応じて更新ログの配布範囲 N_C を設定し、接続した移動体から $N_C(\geq 1)$ ホップ内の移動体に配布する。以下に、二つの移動体 M_i と M_j ($i < j$) が接続した場合の手順を示す。

(1) M_i は、 M_j に自身の更新履歴表を送信する。

(2) M_j は、 M_i と M_j の更新履歴表の各データの更新ログを比較する。

- M_i がもち、 M_j がもたなかった更新ログに関しては、 M_j の更新履歴表に新しく追加した後、 M_j に N_C ホップ以内でもともと相互接続していた移動体に配布する。

- M_j がもち、 M_i がもたない更新ログに関しては、 M_i にその更新ログを配布する。 M_i は更新履歴表に受信した更新ログを新しく追加した後、 M_i に N_C ホップ以内でもともと相互

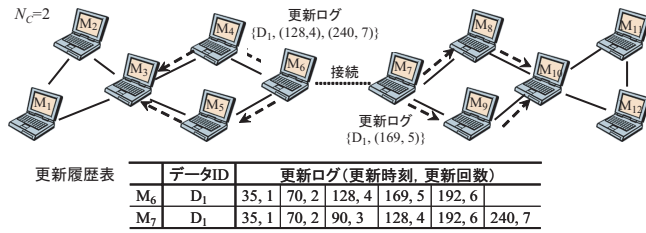


図 6 ULD-C 方式における更新ログの配布

Fig. 6 Dissemination of update logs in ULD-C method.

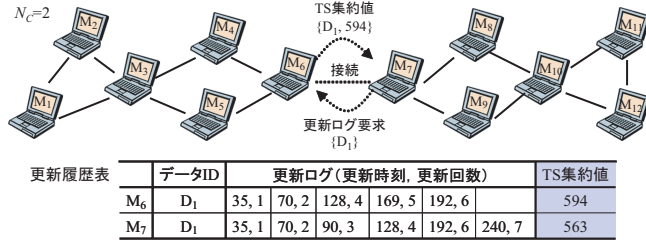


図 7 更新ログの要求

Fig. 7 Request for update logs.

接続していた移動体に配布する。

更新ログを受信した移動体は、自身の更新履歴表を更新した後、複製へのアクセスの有効性を確認する。確認できたアクセスログは、アクセス履歴表から削除する。

図 6 は、接続した M_6 と M_7 が D_1 の更新履歴表を比較し、それぞれが更新ログを 2 ホップの範囲で配布する様子を示す。

ULD-C 方式では、ULD-DA(+) 方式と比べて、更新ログがネットワーク全体に早く伝播するため、複製へのアクセスの有効性を多く確認できると考えられる。しかし、移動体が接続する度に更新履歴表を送信するため、トラフィックが大きくなる。

4.3.2 ULD-C 方式のトラフィックの抑制

各移動体がデータを頻繁に更新する環境では、更新履歴表のサイズが非常に大きくなる。このとき、接続した移動体間で更新履歴表そのものを送信すると、トラフィックが非常に大きくなる。そこで本稿では、このトラフィックを抑制するため、ULD-C 方式の拡張を行う。具体的には、接続した移動体間で各データの更新ログリストに含まれる更新ログのタイムスタンプの和(タイムスタンプ集約値)を比較して、その値が異なるデータの更新ログのみを配布する。以下に、二つの移動体 M_i と M_j ($i < j$) が接続した場合の手順を示す。

(1) M_i は、各データごとのタイムスタンプ集約値を計算し、その情報を M_j に送信する。

(2) M_j も、各データごとのタイムスタンプ集約値を計算し、 M_i から受信した情報と比較して、その値が異なるデータの更新ログのみを M_i に要求する。

(3) M_i は M_j から要求された更新ログを送信し、ULD-C 方式の手順 2 と同様の操作を行う。

図 7 は、 M_6 と M_7 がデータ D_1 のタイムスタンプ集約値を比較して、 M_7 が M_6 に D_1 の更新ログを要求する動作を示す。

この方法により、接続した移動体間で同じ更新ログをもつデータに関しては、タイムスタンプ集約値のみを送信するだけで良いため、無駄なトラフィックを抑制できる。実際は異なる更新ログリストが偶然同じ集約値となる場合もあるが、確率的に

表 3 パラメータ設定

Table 3 Parameter configuration.

パラメータ	値
U	400 (100~4,000)
TO_a	1,500 (100~3,000)
N_{DA}	2
N_C	2

は非常に低いと考えられる。

5. 性能評価

本章では、更新ログ配布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

500[m] × 500[m] の 2 次元平面上に、40 個の移動体 ($M = M_1, \dots, M_{40}$) が存在し、各移動体は 2 次元平面内からランダムに目的地を決定して、0 から 1[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、0 から 1000[秒] の範囲でランダムに決定した時間だけ停止し、停止時間が経過すると、次の目的地を決定して再び移動するものとした。移動体の無線通信範囲は、半径 70[m] とした。

ネットワーク内には、40 種類のデータ ($D = D_1, \dots, D_{40}$) が存在し、 M_j ($j = 1, \dots, 40$) が D_j をオリジナルデータとして保持するものとした。各移動体 M_i の各データ D_j へのアクセス頻度は、 $p_{ij} = 0.005 \times (1 + 0.0001j)$ [回/秒] とした。データのサイズは移動体のデータ領域に比べて十分に小さいものとし、移動体はアクセスしたデータの複製をすべて自身のデータ領域に作成できるものとした。

各移動体は、自身のもつオリジナルデータを平均 U [秒] (平均更新間隔) の指数分布に基づいた間隔で更新するものとした。また各移動体は、自身の更新履歴表に過去 10,000[秒] 以内に更新されたデータの更新ログを記録するものとした。

表 3 に、本実験で用いたパラメータを示す。各パラメータは基本的には定数値をとるが、そのパラメータの影響を調べる際には括弧内の範囲で変化させた。以上のシミュレーション環境において、各移動体の初期位置をランダムに決定し、200,000[秒] 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。

- 古い複製にアクセスした割合

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、古い複製へのアクセス回数の総数の割合。古い複製へのアクセス回数とは、移動体が更新ログを参照して、複製へのアクセスの有効性を確認した結果、失敗であった回数を指す。

- 複製へのアクセスがタイムアウトした割合

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、複製にアクセスしてから TO_a [秒] 経過しても有効性を確認できなかった回数の割合。

- 複製へのアクセスの判断に要する時間

移動体が複製に暫定的にアクセスしてから、更新ログを参照して有効性を確認するまでの時間、または複製へのアクセスがタイムアウトするまでの時間の平均値。

- 更新ログの配布によるトラヒック

シミュレーション時間内に発生した、更新ログと制御パケットの送信に要する通信ホップ数に、それぞれのサイズを掛けたものの総和。

5.2 平均更新間隔の影響

まず、平均更新間隔 U 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 8, 図 9, 図 10, 図 11, 図 12 に示す。各図の横軸は平均更新間隔を示す。縦軸は、図 8 ではアクセス成功率、図 9 では古い複製にアクセスした割合、図 10 では複製へのアクセスがタイムアウトした割合、図 11 では複製へのアクセスの判断に要する時間、図 12 では更新ログの配布によるトラヒックを示す。図中の凡例の ULD-C 方式の括弧内において、'DA(+)' は ULD-DA(+)' 方式を併用していることを示し、'Agg'(Aggregation) はトラヒック抑制のためにタイムスタンプ集約値を用いていることを示す。

図 8 の結果から、平均更新間隔が大きくなると、各移動体のもつ複製が有効な時間が長くなるため、いずれの方式もアクセス成功率が高くなることわかる。各方式を比較すると、ULD-DA+方式と ULD-C 方式を併用する場合 (ULD-C(DA+) と ULD-C(DA+, Agg)) のアクセス成功率が最も高くなる。これは、移動体がデータアクセスしたときと、二つの移動体が接続したときの両方において更新ログを配布できるため、各移動体が複製へのアクセスの有効性を多く確認できるからである。また、ULD-DA 方式を用いる場合 (ULD-DA, ULD-C(DA) および ULD-C(DA, Agg)) と ULD-DA+方式を用いる場合 (ULD-DA+, ULD-C(DA+) および ULD-C(DA+, Agg)) を比較すると、後者のアクセス成功率の方が高く、平均更新間隔が大きくなるとその特性が顕著に現れている。最新の複製 (オリジナルと同じタイムスタンプのもの) へのアクセスは、オリジナルをもつ移動体に最新のタイムスタンプを問合せたときにだけ確認できる。ここで、ULD-DA+方式はオリジナルをもつ移動体からの最新のタイムスタンプ情報をより多くの移動体に配布できるため、最新の複製へのアクセスの有効性を多く確認できている。さらに、ULD-DA(+)' 方式と ULD-C 方式を併用する場合において、タイムスタンプ集約値を用いる場合と用いない場合を比較すると、同じ結果になっている (図 9~ 図 11, 図 13~ 図 16 も同様)。これは、接続した移動体間の更新履歴表の送信方法において、タイムスタンプ集約値を用いるか否かだけの違いで、その他の動作は同じだからである。またこの結果から、本実験では、異なる更新ログリストの集約値が偶然等しくなる場合が生じていないことがわかる。

図 9 の結果から、平均更新間隔が大きくなると、いずれの方式も古い複製にアクセスした割合が低くなることわかる。各方式を比較すると、ULD-C 方式を用いる場合に古い複製にアクセスした割合が高くなる。ULD-C 方式では、更新ログがネットワーク全体に伝播されるため、オリジナルをもつ移動体から離れた移動体でも複製へのアクセスの有効性を確認できる。しかし、このような移動体のもつ複製は古いものである可能性が高いため、古い複製にアクセスした割合が高くなる。

図 10 の結果から、平均更新間隔が大きくなると、いずれの方式も複製へのアクセスがタイムアウトした割合が高くなる傾向にある。平均更新間隔が大きくなると、移動体が最新更新口

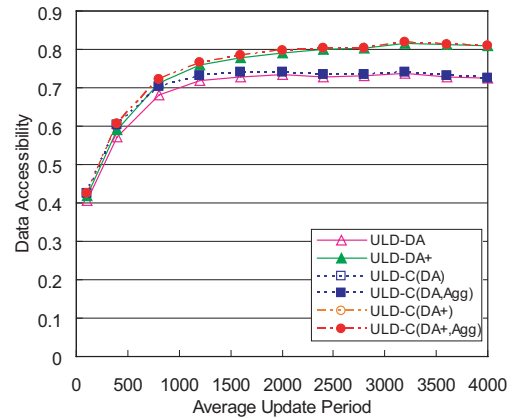


図 8 平均更新間隔とアクセス成功率

Fig. 8 Average update period and data accessibility.

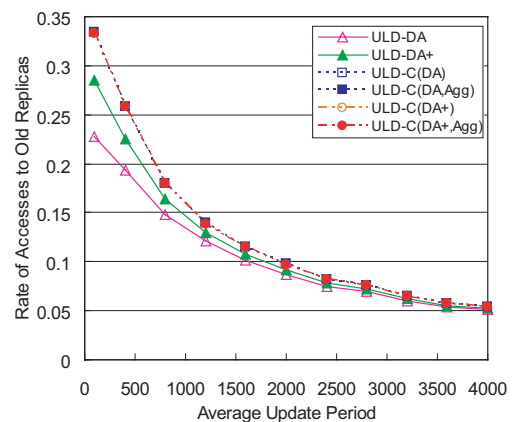


図 9 平均更新間隔と古い複製にアクセスした割合

Fig. 9 Average update period and rate of accesses to old replicas.

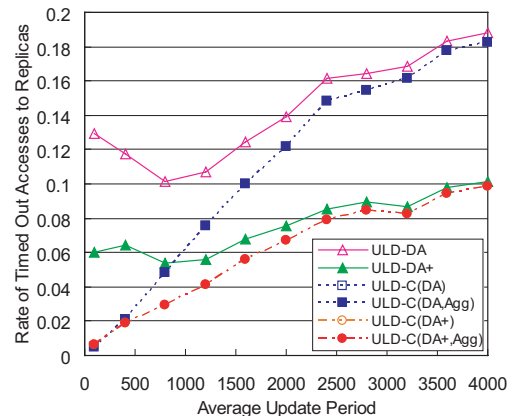


図 10 平均更新間隔と複製へのアクセスがタイムアウトした割合

Fig. 10 Average update period and rate of timed out accesses to replicas.

グを送信して複製を無効化する機会が減るため、ネットワーク全体に複製が多く残る。したがって、移動体が複製にアクセスする回数が増えるに伴い、タイムアウトする回数も増えてしまう。また、ULD-DA(+)' 方式と ULD-C 方式を併用する場合のグラフの傾きが大きく、ULD-DA(+)' 方式を用いる場合の結果に近づくことがわかる。平均更新間隔が大きくなると、各移動体が最新の複製にアクセスする機会が増える。最新の複製へのアクセスの有効性は、オリジナルをもつ移動体に問合せたとき

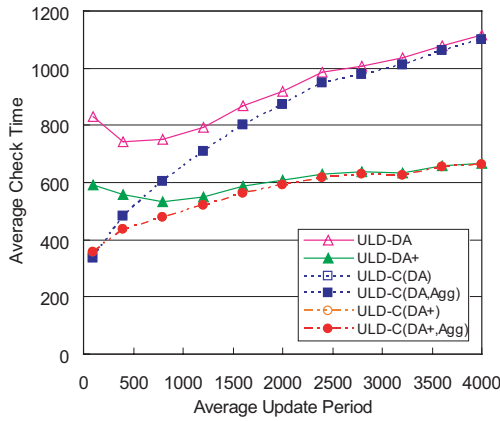


図 11 平均更新間隔と複製へのアクセスの判断に要する時間
Fig. 11 Average update period and average check time.

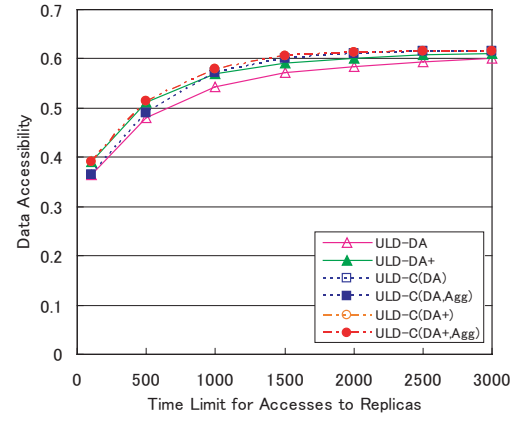


図 13 アクセスタイムアウト時間とアクセス成功率
Fig. 13 Time limit for accesses to replicas and data accessibility.

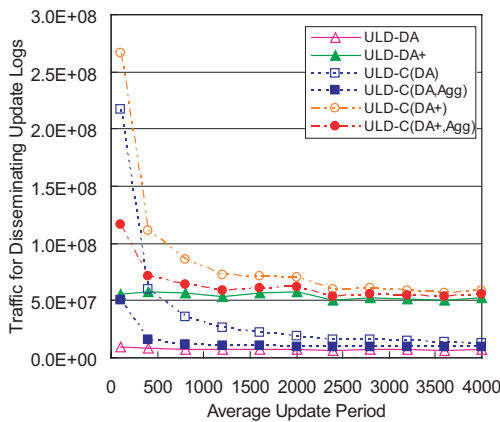


図 12 平均更新間隔と更新ログの配布によるトラフィック
Fig. 12 Average update period and traffic for disseminating update logs.

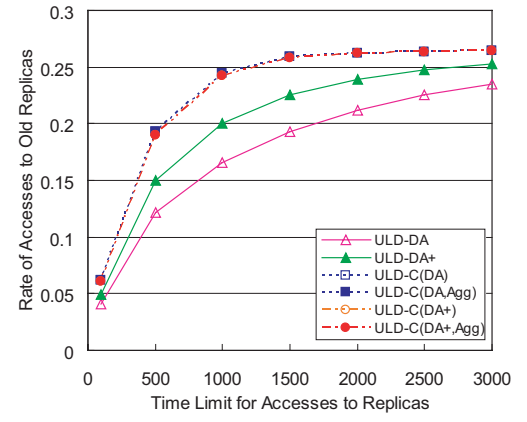


図 14 アクセスタイムアウト時間と古い複製にアクセスした割合
Fig. 14 Time limit for accesses to replicas and rate of accesses to old replicas.

にだけ確認できるため、ULD-DA(+)方式の影響が強くなる。
図 11 の結果から、平均更新間隔が大きくなると、いずれの方式も複製へのアクセスの判断に要する時間が長くなるのがわかる。これは、図 10 の考察と同様の理由によるものである。各方式を比較すると、ULD-C 方式と ULD-DA+方式を併用した場合において、複製へのアクセスの判断に要する時間が最も短くなるのがわかる。これは、他の場合に比べて、更新ログを配布できる機会が多いからである。

図 12 の結果から、平均更新間隔が大きくなると、ULD-C 方式を用いる場合では、各移動体のもつ更新履歴表のサイズが大幅に小さくなるため、更新ログの配布によるトラフィックが大幅に小さくなる。また、ULD-DA(+)方式と ULD-C 方式を併用して、タイムスタンプ集約値を用いる場合と用いない場合を比較すると、前者の方が大幅に小さいことがわかる。これは、タイムスタンプ集約値を用いることで、接続した移動体間における更新履歴表の送信トラフィックを効果的に抑制できていることを示す。

5.3 アクセスタイムアウト時間の影響

次に、アクセスタイムアウト時間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、図 17 に示す。各図の横軸はアクセスタイムアウト時間を示す。縦軸は、図 13 ではアクセス成功率、図 14 では古い複製にアクセ

スした割合、図 15 では複製へのアクセスがタイムアウトした割合、図 16 では複製へのアクセスの判断に要する時間、図 17 では更新ログの配布によるトラフィックを示す。

図 13 の結果から、アクセスタイムアウト時間が大きくなると、いずれの方式もアクセス成功率が高くなるのがわかる。これは、各移動体が他の移動体から更新ログを受信できる機会が増え、複製へのアクセスの有効性を多く確認できるからである。各方式を比較すると、ULD-DA+方式と ULD-C 方式を併用した場合にアクセス成功率が最も高くなる。これは、図 8 の結果における考察と同様である。また、アクセスタイムアウト時間の増加に対して ULD-DA+方式のみを用いた場合と ULD-DA 方式と ULD-C 方式を併用した場合を比較すると、最初は前者の方が高いが、その後、後者の方が高くなる。これは、ULD-C 方式により多くの移動体が、複製へのアクセスをタイムアウトするまでに更新ログを受信できていることを示している。

図 14 の結果から、アクセスタイムアウト時間が大きくなると、いずれの方式も古い複製にアクセスした割合が高くなるのがわかる。これは、図 13 の結果における考察と同様である。また、各方式を比較すると、ULD-C 方式を用いる場合に、古い複製にアクセスした割合が高くなっていく。これは、図 9 の結果における考察と同様である。

図 15 の結果から、アクセスタイムアウト時間が大きくなると、いずれの方式も複製へのアクセスがタイムアウトした割合

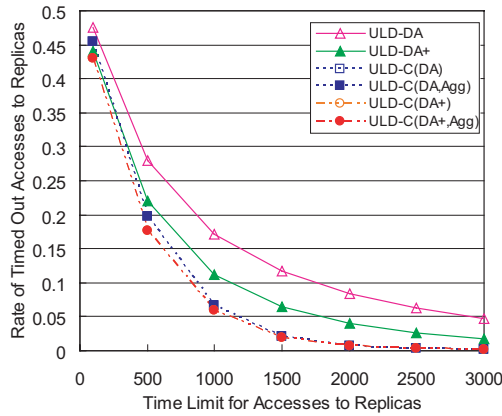


図 15 アクセスタイムアウト時間と複製へのアクセスがタイムアウトした割合

Fig. 15 Time limit for accesses to replicas and rate of timed out accesses to replicas.

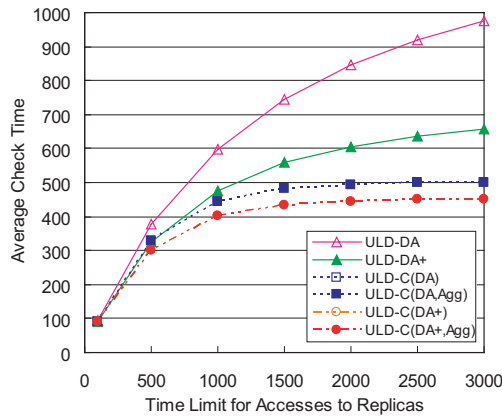


図 16 アクセスタイムアウト時間と複製へのアクセスの判断に要する時間

Fig. 16 Time limit for accesses to replicas and average check time.

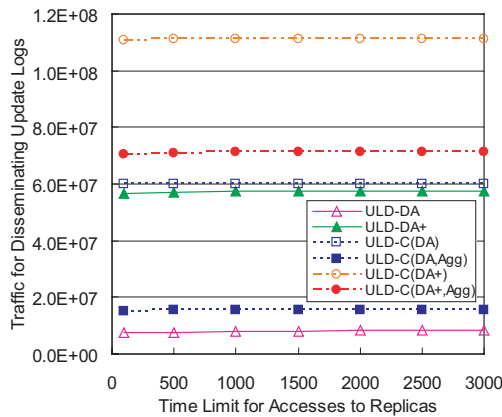


図 17 アクセスタイムアウト時間と更新ログの配布によるトラフィック

Fig. 17 Time limit for accesses to replicas and traffic for disseminating update logs.

が低くなる。これは、図 13 の結果における考察と同様である。特に、ULD-C 方式を用いる場合は、複製へのアクセスがタイムアウトした割合が 0 に近づいている。

図 16 の結果から、アクセスタイムアウト時間が大きくなると、いずれの場合も複製へのアクセスの判断に要する時間が大きくなる。これは、図 13 の結果における考察と同様である。

図 17 の結果から、アクセスタイムアウト時間の変化に対して、いずれの方式も更新ログの配布によるトラフィックがほぼ一定であることがわかる。アクセスタイムアウト時間が大きくなると、移動体が保持するアクセスログの数が多くなるが、ULD-DA(+) 方式では、移動体があるタイムスタンプの複製に何回アクセスしたとしても、オリジナルをもつ移動体にそのタイムスタンプの次のタイムスタンプだけを問合せのため、更新ログの配布によるトラフィックにあまり影響しない。また、ULD-DA(+) 方式と ULD-C 方式を併用した場合において、タイムスタンプ集約値を用いる場合と用いない場合を比較すると、図 12 の結果と同様に、前者の方が大幅に小さいことがわかる。

6. まとめ

本稿では、文献 [4] で提案した更新ログ配布方式の性能を、シミュレーション実験により評価した。シミュレーション結果から、データ更新が頻繁に発生する環境では、ネットワーク全体に更新ログを配布できる ULD-C 方式が有効であることを確認した。一方、データ更新があまり発生しない環境では、各移動体が最新の複製にアクセスする機会が増えるため、オリジナルをもつ移動体からの更新ログを多くの移動体に配布する ULD-DA+ 方式が有効であることを確認した。さらに、ULD-C 方式においてタイムスタンプ集約値を用いることで、更新ログによるトラフィックを抑制できることを確認した。

今後は、移動体の移動やアクセス特性を考慮して、ULD-C 方式のトラフィックを抑制する方法を検討する予定である。

謝辞 本研究の一部は、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」、文部科学省若手研究 (A)(16680005)、および (財) 立石科学技術振興財団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] 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).
- [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).
- [4] 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の更新ログの配布について,” 情報処理学会データベースシステム研究会報告 (2005-DBS-135), vol. 2005, no. 6, pp. 103–110 (2005).
- [5] 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).
- [6] 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).
- [7] 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).