

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

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

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、移動体に他の移動体のもつオリジナルデータの複製を配置することが有効である。ここで、データ更新の発生する環境では、移動体のもつ複製がオリジナルとバージョンが異なる古いものになる可能性がある。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。そのため、複製へのアクセスは暫定的に行い、後ほど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。そこで本稿では、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更新ログを配布する方式を提案する。

On Dissemination of Data Update Logs in Ad Hoc Networks

Hideki HAYASHI Takahiro HARA 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

In ad hoc networks, it is effective that each mobile host allocates replicas of original data items held by others. In an environment where data items are updated, mobile hosts hold old replicas with different versions of the originals. If the mobile hosts do not connect with mobile hosts holding the originals, they cannot check whether their holding replicas are up-to-date. Therefore, it is effective that they tentatively accesses the replicas and afterward check the validity of the accesses by referring update logs of their originals. In this paper, we propose update log dissemination methods in which each mobile host manages update logs of data items and transmits them as needed. In our proposed methods, mobile hosts can efficiently check the validity of accesses to replicas.

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等のデータ共有があり、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において、データの利用率を向上させるための効果的な複製配置は重要である。

これまでに筆者らは、文献 [4] 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、自身の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 [5] では、文献 [4] の方式を拡張して、データ間に相関性が存在する環境において、効果的な複製配置方式を提案した。

一方、実環境では、データ更新が発生することが一般的である。この場合、移動体がオリジナルとバージョンが異なる古い複製にアクセスする可能性

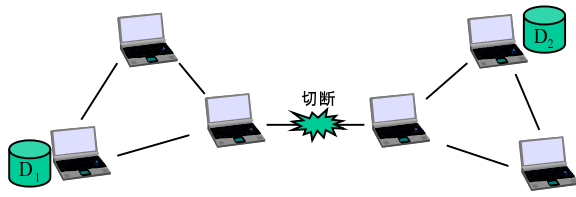


図 1: ネットワークの分断

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 [6] において、オリジナルをもつ移動体がデータを更新する環境を想定して、古い複製へのアクセス回数を削減するために、各移動体が無効化情報を放送する方式を提案した。また文献 [7] 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する方式を提案した。

データ更新の発生する環境では、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、後ほど、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。文献 [6, 7] では、オリジナルをもつ移動体が更新ログを記録し、複製に暫定的にアクセスした移動体がオリジナルをもつ移動体との接続を検出したときに、アクセスの有効性を確認するものと想定していた。しかし、複製へのアクセスの有効性を確認する具体的な方法については、特に規定していなかった。また、アプリケーションによっては、移動体が複製への暫定的なアクセスをできるだけ早く確認したいものもある。例えば、災害時の救助活動では、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が考えられる。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を必要として、ようやく確認できた結果が無効だった場合、その救助隊員が長時間行った作業が無駄になってしまう。このとき、複製へのアクセスが無効であることを早く確認できたならば、無駄となる作業量を少なくできる。

そこで本稿では、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更新ログを配布することで、複製へのアクセスの有効性を効率的に確認する方式をいくつか提案する。

以下では、2章で関連研究について紹介し、本研究との比較を行う。3章で想定環境について述べる。4章で複製の管理方法について述べ、5章で更新ログの配布方式について説明する。最後に6章で結論と今後の課題について述べる。

2 関連研究

固定ネットワーク上にサーバが存在するモバイル環境において、移動体への複製配置に関する研究がいくつか行われている。文献 [1] では、無線通信コストの削減のため、サーバが周期的に無効化情報を放送することで、移動体のもつ古い複製を効率的に無効化する方式を提案している。この方式では、移動体が自身の複製にアクセスする場合、次の放送周期まで待ち、サーバのもつデータとアクセスする複製が同じバージョンならば、複製にアクセスする。したがって、移動体がデータを要求してから複製にアクセスするまでに遅延（アクセス遅延）が生じる。文献 [2] では、アクセス遅延の削減のため、無効化情報の放送間隔の間に、サーバが前回の無効化情報の放送時から更新したデータの情報を短い周期で放送する方式を提案している。文献 [1, 2] では、移動体が複製の更新の有無を確認してから、複製にアクセスする点で、本研究と異なる。本研究では、移動体が複製に暫定的にアクセスし、後で更新ログを参照してアクセスの有効性を確認する。一方、文献 [3] では、トランザクション処理のスループット向上のため、2層複製法を提案している。この方法では、ネットワークに接続していない移動体が自身のもつ複製に暫定的なトランザクションを実行して、ネットワークに再接続後、主コピーに正式なトランザクションを実行する。この研究は、移動体が主コピーをもつ固定ネットワーク上の端末と接続した場合にだけ、複製への操作の有効性を確認する点で、本研究と異なる。

アドホックネットワークの研究分野でも、データ更新の発生する環境を想定した複製配置に関する研究がいくつか行われている。文献 [8, 9] では、サーバの役割を担う移動体に複製を配置し、クォーラムシステムに基づいて複製の一貫性管理を行う方式を提案している。また文献 [10] では、移動体が位置情報を用いて更新と問合せによるトラフィックを削減しながら、複製の一貫性を保つ方式を提案している。これらの研究は、移動体が複製へのアクセスの有効

性を確認する方法について検討していない点で、本研究と異なる。

3 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オリジナルの複製を作成する環境を想定する。各移動体は何らかの方式を用いて、複製として配置するデータの種類を決定する。本稿で提案する更新ログの配布方式は、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、即座に成功とみなす。なお本稿では、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、複製に暫定的にアクセスする。複製へのアクセスの有効性は、後ほど、移動体がアクセスしたデータの更新ログを参照して確認する。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みなす。

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となる。

4 複製の管理

アドホックネットワークにおいて、データ更新が発生する環境では、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。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計算環境では好ましくない。したがって、移動体が複製へのアクセスの有効性を確認する方法とは別に、複製の管理方法が必要になる。そこで本稿では、文献 [6] の提案方式を参考にして、以下の管理方法を用いるものと想定する。

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。タイムスタンプとは、データの更新時刻を指す。また更新回数とは、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、この値は初期状態では 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シングする。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。そのデータの古い複製を保持する場合は、複製を無効化する。

また、ネットワーク内で移動体同士の新たな接続が起こった場合、接続した二つの移動体が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シングする。最近更新ログを受信した移動体は、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シ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を保持できる。それに伴い、相互接続している移動体同士は、同じタイムスタンプの複製を保持できる。したがって、移動体はアクセス要求をする場合、どの移動体をもつデータ（複製）にアクセスしても、同じタイムスタンプ（バージョン）のものとなる。本稿では、この特徴を考慮した以下のアクセス方法を用いる。表 1 は、移動体がデータへのアクセス要求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している。

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

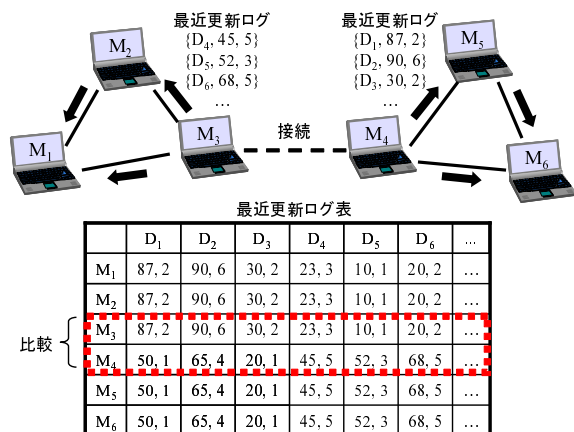


図 2: 接続した移動体による最近更新ログのフラッシング

表 1: データへのアクセス要求で用いるパケット

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

セスして、処理を終了する。そうでない場合、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シングする。

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にデータ要求パケットを送信する。複数の移動体が返信してきた場合には、自身から最もホップ数の近い移動体に要求する。また、一つの移動体からも返信がなかった（相互接続している移動体が要求したデータ（複製）をもっていなかった）場合、その要求は失敗となり、処理を終了する。

5 更新ログの配布

複製に暫定的にアクセスした移動体は、そのデータの更新ログを後に参照して、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、移動体は複製にアクセスしたとき、その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（複製にアクセスした時刻、アクセスした複製のタイムスタンプ、アクセスした複製の更新回数）のリストを含む。ここで、移動体が有効性を確認できないアクセスのログを保持し続ける場合、アクセス履歴表のサイズが大きくなる。また、先述のように、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。そこで本稿では、アクセス履歴表に保持するログは、過去 TO_a 時間以内のアクセスに関するものに限定する。移動体が複製にアクセスしてから TO_a 時間経過すると、そのアクセスは無効なもののみなし、アクセス履歴表からそのログを削除する。

本章の以下では、複製へのアクセスの有効性を効率的に確認するために、各移動体がデータの更新ログを管理し、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する。まず、文献 [3] を参考にして、各移動体が自身のもつオリジナルの更新ログだけを管理して、更新ログを配布する方式を提案する。次に、アドホックネットワークではオリジナルをもつ移動体と必ず接続できるとは限らないことを考慮して、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、更新ログを再配布する方式を提案する。

5.1 オリジナルをもつ移動体による更新ログの配布

本節では、各移動体が自身のもつオリジナルの更新ログのみを管理し、更新ログを配布する方式を提案する。

まず、移動体は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。ここで移動体は、複製にアクセスしてから、その有効性を確認するまでに最大 TO_a 時間しか待てないため、時間の経過と共に古いアクセスログが削除される。したがって、移動体がデータの更新ログを保持し続けても、古い更新ログは、複製の有効性を確認するために使用されなくなる。そこで、移動体が更新履歴表に保持するログは、過去 $TO_u (\gg TO_a)$ 時間以内に更新されたもの

表 2: ULD-DA(+) 方式で用いるパケット

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

に限定する。

複製にアクセスした移動体は、オリジナルをもつ移動体から更新ログを得る。そのため、オリジナルをもつ移動体との接続を検出する必要がある。本稿では、この検出に移動体のデータアクセスの動作を利用する。この動作の過程では、要求されたデータのオリジナルをもつ移動体は、要求した移動体にオリジナルをもつことを示すフラグを立てた返信パケットを送信する。したがって、要求した移動体がこのパケットを受信すると、自身がオリジナルをもつ移動体と接続していることを検出できる。

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) 方式について説明する。さらに、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配布する ULD-DA+ 方式について説明する。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ホップ数で表したものである。

5.1.1 ULD-DA 方式

ULD-DA 方式では、オリジナルをもつ移動体が複製にアクセスした移動体に更新ログを配布する。ここで、オリジナルをもつ移動体が、その更新ログを全て配布すると、トラヒックが非常に大きくなる。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。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×印が更新の発生を示し、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。一方、矢印が移動体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、左から順にアクセス時刻、タイムスタンプ、更新回

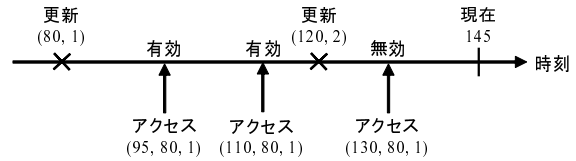


図 3: 複製へのアクセスの有効性の確認

数を示す。この場合、タイムスタンプ 120 の更新ログにより、タイムスタンプ 80 の複製へのアクセスの有効性を確認できる。例えば、この複製への時刻 95 および 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。そのため、ULD-DA 方式では、オリジナルをもつ移動体は、アクセス要求し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け取り、その複製に対する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) のタイムスタンプリストを含む。
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 の複製への暫定的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 4 は、移動体 M_5 が D_1 のオリジナルをもつ移動体 M_1 との接続を検出したとき、 M_1 が M_5 に更新ログを配布する動作を示す。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。また、アクセスの有効性を確認した結果、アクセスログの白色部分が有効だったもの、灰色部分が無効だったものを示す。

5.1.2 ULD-DA+ 方式

ULD-DA 方式では、アクセス要求した移動体だけが、複製へのアクセスの有効性を確認できる。そ

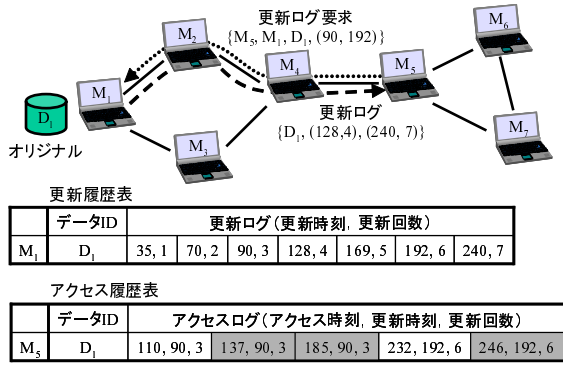


図 4: ULD-DA 方式における更新ログの配布

のため、あまりアクセス要求しないデータに関しては、オリジナルをもつ移動体との接続をチェックできず、複製へのアクセスの有効性を確認するまでに長い時間を要する場合がある。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、 $N (\geq 1)$ ホップ内の移動体に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、自身から N 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 ホップの範囲でフラディングする。このパケットの送信元 ID は M_i 、データ ID は D_j 、TTL は N 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として M_j に送信する。
4. M_j は、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求めて、 M_i に配布する。

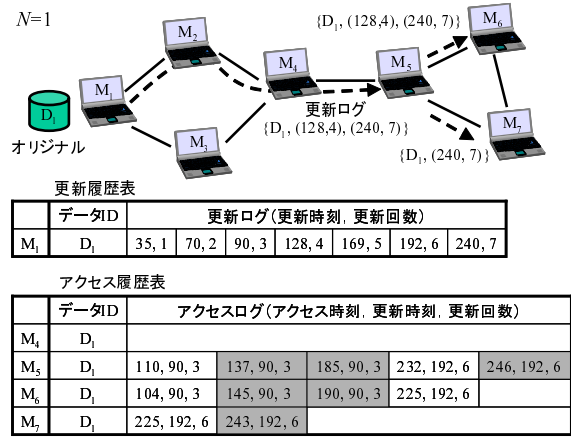


図 5: ULD-DA+ 方式における更新ログの配布

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。その後、自身から N ホップ内の移動体にアクセスの有効性を確認するため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 は、 M_1 から更新ログを受信した M_5 が、1 ホップ内で接続している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部分は有効だったもの、灰色部分は無効だったものを示す。

これにより、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。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。

5.2 その他の移動体による更新ログの配布

各移動体が自身のもつオリジナルの更新ログだけを管理する場合、複製にアクセスした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、アクセスの有効性を確認できる。アドホックネットワークでは、移動体が、特定の移動体に必ず接続できる保証はない。そこで本節では、各移動体が自身のもつオリジナルの更新ログだけではなく、他の移動体のもつオリジナルの更新ログも管理し、新しく

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) 方式を提案する。なお、前節と同様に、更新履歴表に保持するログは過去 TO_u 時間以内に更新されたものとする。

5.2.1 ULD-C 方式

まず、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、4章の複製管理の方法により、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て、お互いが保持していない更新ログを、接続した当事者にもともと相互接続していた移動体に配布する。ここで、ネットワーク全体に更新ログを配布することは、トラヒックが非常に大きくなるため、移動体の消費電力に厳しい制約のある環境に適さない。そこで、適用する環境に応じて更新ログの配布範囲 N を設定して、接続した当事者から $N(\geq 1)$ ホップの範囲で更新ログを配布する。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、 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履歴を比較する。
 - 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 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 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。

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に、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、新たに接続した M_6 と M_7 が D_1 の更新履歴を比較し、もともと相互接続していた移動体に更新ログを 2 ホップの範囲で配布する様子を示す。

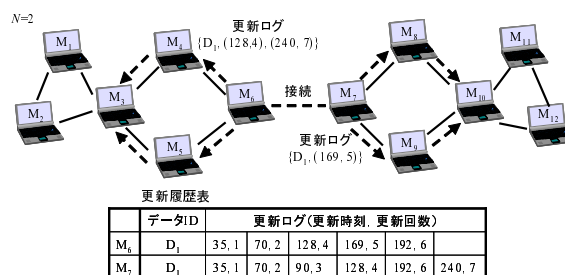


図 6: ULD-C 方式における更新ログの配布

ULD-C 方式では、ULD-DA(+) 方式と比較して、更新ログがネットワーク全体に速く伝播するため、移動体が複製にアクセスしてから短い時間で、その有効性を確認できるようになる。それに伴い、 TO_u 時間経過したために無効になる複製へのアクセス数が少なくなる。しかし、接続した移動体間での更新履歴表の比較と、接続した移動体から N ホップ内の移動体への更新ログの配布により、トラヒックが大きくなる。そこで、ULD-C 方式の適用環境に応じて、更新ログの配布によるトラヒックを抑制する方法について考察する。

【移動体のアクセス特性に局所性のある環境】

実環境では、移動体のアクセス特性が局所性をもつ場合が多い。例えば、災害時の救助活動において、地理的に活動領域をいくつかの領域に分割し、各領域にいくつかの移動体（救助隊員）を担当させる場合を考える。このとき、移動体は、自身と同じ領域を担当する移動体の作業状況データに頻繁にアクセスするものと考えられる。つまり、近くの移動体同士は類似したアクセス特性をもつため、同じ種類のデータの複製にアクセスしている可能性が高い。このような場合、ULD-C 方式の N の値を小さく設定して、更新ログの配布範囲を狭くしても、多くの複製へのアクセスの有効性を確認できるものと考えられる。

【グループ特性のある環境】

実環境では、移動体がグループを形成して、協調作業する場合も多い。同じグループに属する移動体同士は近い距離を保ちながら移動することになり、接続し易い。一方、異なるグループに属する移動体同士は、分断されている可能性が高い。したがって、グループ特性のある環境では、異なるグループに属する移動体同士が接続（隣接）した場合に、更新履

履歴表の比較と更新ログの配布を行った方が有効である。ここで、異なるグループに属する移動体同士が隣接する場合、前回隣接してから長い時間が経過しているものと考えられる。そこで、お互いが隣接していなかった時間（非隣接時間）が閾値を超えた場合に、更新履歴表の比較と更新ログの配布を行うことで、トラヒックを削減できる。また、新しく隣接した移動体同士の非隣接時間が長い場合、更新履歴表において、片方が保持し、他方が保持しない更新ログが多くなるものと考えられる。この場合、隣接した当事者から広い範囲で更新ログを配布すると、広い範囲に存在する多くの移動体の複製へのアクセスを確認できる。したがって、非隣接時間に単調増加する関数を定義し、非隣接時間が長いほど、広範囲に更新ログを配布することが有効である。

6 まとめ

本稿では、複製へのアクセスの有効性を効率的に確認するため、移動体がデータの更新ログを管理し、必要に応じて他の移動体に更新ログを配布する方式について議論した。まず、各移動体が自身のもつオリジナルの更新ログを管理する環境において、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 方式を提案した。さらに、この動作の後で、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 ULD-DA+ 方式について説明した。また、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する環境において、更新ログを配布する ULD-C 方式を提案した。ULD-C 方式では、接続した二つの移動体が更新履歴表を比較し、お互いが保持していない更新ログを、接続した移動体から N ホップの範囲で配布する。さらに、ULD-C 方式の適用環境に応じて、複製へのアクセスの有効性を多く確認しながら、トラヒックを抑制する方法について検討した。今後は、移動体の移動やデータアクセスの様々な特性を考慮して、本稿で提案した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 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。

参考文献

- [1] D. Barbara and T. Imielinski, “Sleepers and workaholics: caching strategies for mobile environments,” *Proc. ACM SIGMOD’94*, pp. 1–12 (1994).
- [2] G. Cao, “A scalable low-latency cache invalidation strategy for mobile,” *IEEE Transactions Knowledge and Data Engineering*, vol. 15, no. 5, pp. 1251–1265 (2003).
- [3] J. Gray, P. Helland, P.E. O’Neil, and D. Shasha, “The dangers of replication and a solution,” *ACM SIGMOD’96*, pp. 173–182 (2003).
- [4] 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).
- [5] T. Hara, N. Murakami, and S. Nishio, “Replica allocation for correlated data items in ad-hoc sensor networks,” *ACM SIGMOD Record*, vol. 33, no. 1, pp. 38–43 (2004).
- [6] 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).
- [7] 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).
- [8] 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).
- [9] 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).
- [10] J.B. Tchakarov, and N.H. Vaidya, “Efficient content location in wireless ad hoc networks,” *Proc. IEEE Int’l Conf. on Mobile Data Management (MDM’04)*, pp. 74–85 (2004).