

# アドホックネットワークにおけるクォラムシステムを用いた一貫性管理のためのトラフィック削減について

澤井 陽平<sup>†1</sup> 篠原 昌子<sup>†2</sup> 神崎 映光<sup>†2</sup>  
原 隆浩<sup>†2</sup> 西尾 章治郎<sup>†2</sup>

アドホックネットワークでは、データの複製を作成することでデータの利用率が向上する。ここで、データ更新の発生により、複製間の一貫性が損なわれる可能性があるため、クォラムシステムを用いた一貫性管理が有効となる。しかし、従来手法では、クォラム内の全移動体が保持するデータを更新する必要があるため、トラフィックが増加してしまう。そこで本論文では、データ更新時の操作に要するトラフィックの削減を目的として、ポインタ情報を用いて複製間の一貫性を管理する方式を提案する。提案方式では、データの更新時に、構成したクォラム内の一部の移動体が保持するデータのみを更新することで、トラフィックを削減する。また、更新データを持つ移動体に関する情報を、クォラム内の全移動体に送信することで、複製間の厳密な一貫性を維持する。

## On Traffic Reduction for Consistency Management Based on Quorum Systems in Ad Hoc Networks

YOHEI SAWAI,<sup>†1</sup> MASAKO SHINOHARA,<sup>†2</sup> AKIMITSU KANZAKI,<sup>†2</sup>  
TAKAHIRO HARA<sup>†2</sup> and SHOJIRO NISHIO<sup>†2</sup>

Data replication is effective for improving data availability in ad hoc networks. In an environment where data updates occur, replicas of a data item may be inconsistent. To solve this problem, quorum based consistency management is a promising approach. However, since a mobile host needs to update all replicas in a quorum, traffic for updating replicas increases. In this paper we propose a quorum based consistency management method which can reduce communication traffic. In our proposed method, when a mobile host updates replicas, it updates part of replicas in a quorum to reduce traffic. In addition, to maintain strict consistency among replicas, all mobile hosts in the quorum are notified the information of mobile hosts whose replicas were updated.

### 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とめない、ルータの機能を持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注目が高まっている。アドホックネットワークでは、移動体どうしでデータを共有し、互いの持つデータにアクセスすることが多い。しかし、移動体の移動によりネットワークが分断された場合、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。たとえば、図1の中央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、オリジナルデータを持つ移動体以外の移動体に、データの複製を配置することが有効である<sup>2),5),14)</sup>。図1の例では、ネットワークの切断前にデータ  $D_1$  と  $D_2$  の複製をそれぞれ右側と左側の3台の移動体のいずれかに配置することで、切断後もすべての移動体が両方のデータにアクセスできる。

ここで、実環境では、データ更新が発生することが一般的である。アドホックネットワークでは、移動体の移動によりネットワークが頻りに分断されるため、ネットワーク内に存在するすべての複製につねにアクセスできるとは限らない。そのため、すべての複製をつねに最新にすることは現実的に不可能であり、移動

†1 株式会社野村総合研究所  
Nomura Research Institute, Ltd.

†2 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  
Department of Multimedia Engineering, Graduate  
School of Information Science and Technology, Osaka  
University

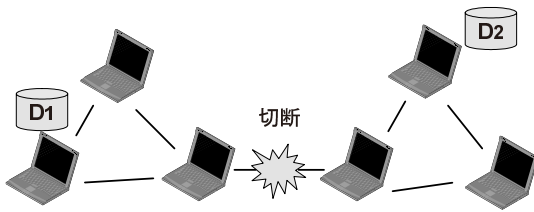


図1 ネットワークの分断  
Fig.1 Network partition.

体が古い複製にアクセスしてしまう可能性がある。このような環境では、動的クオラムシステムを用いた複製間の一貫性管理が有効であり、これまでにいくつかの方式が提案されている<sup>1),3),4),7),9)-11),13)</sup>。本論文では、複製間の厳密な一貫性とは、読み出し操作を実行した移動体が、最新のデータ（最近に書き込みが行われたデータ）に必ずアクセスできることを指すものと定義する。これは、従来の分散データベースにおける書き込み操作どうしの衝突を考慮した、一般的な一貫性と同一の定義である。

クオラムシステムを用いた一貫性管理方式では、複数の移動体がクオラムと呼ばれるグループを構成する。このとき、任意の書き込み操作クオラムと読み出し操作クオラムが、共通集合を持つように設定する。移動体がデータ更新（書き込み）を行う場合、1つの書き込み操作クオラムを構成する全移動体の持つデータ（複製）に、書き込みを行う。一方、移動体がデータアクセス（読み出し）を行う場合、1つの読み出し操作クオラムを構成する全移動体の持つデータ（複製）を参照する。これにより、ネットワークが分断された場合においても、クオラムを構成するすべての移動体（プロキシ）が相互接続していれば、書き込み（読み出し）操作を実行できる。また、書き込み操作クオラムと読み出し操作クオラムは共通集合を持つため、すべての読み出し操作クオラムには、最新のデータを持つ移動体が必ず1つ以上存在する。したがって、参照したデータの中で、最も新しいデータを読み出すことで、複製間の厳密な一貫性を保持できる。

ここで、クオラムを構成する移動体数が増えると、データの書き込みや読み出しに要するトラフィックも増加する。そこで、これまでに筆者らは、文献12)において、少数の移動体でクオラムを構成することで、厳密な一貫性を保持しつつ、トラフィックを抑制するCQ (Crisscross Quorum) 方式を提案した。しかし、CQ方式を用いたとしても、各移動体は、書き込み操作を書き込み操作クオラム内の全プロキシに

実行するため、依然としてデータの書き込みによるトラフィックが大きいという問題があった。そこで本論文では、書き込み操作に要するトラフィックをさらに削減するため、ポインタ情報を用いて複製間の一貫性を管理するCQ/P (CQ with Pointer) 方式を提案する。CQ/P方式では、書き込み操作を、構成した書き込み操作クオラム内の一部のプロキシのみ実行する。また、書き込み操作を行ったプロキシに関する情報（ポインタ情報）を、クオラム内の全プロキシに送信する。これにより、複製間の厳密な一貫性を保持しつつ、書き込み操作によるトラフィックを削減する。

以下では、2章で関連研究について紹介し、3章で本論文の想定環境について述べる。4章で文献12)で提案したCQ方式について述べる。5章で提案方式であるCQ/P方式について述べ、6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。最後に7章で、本論文のまとめと今後の課題について述べる。

## 2. 関連研究

これまでに、アドホックネットワークにおいて、クオラムシステムを用いて複製間の一貫性を管理する方式が数多く提案されている<sup>1),3),4),7),9)-11),13)</sup>。

文献3)では、サーバの役割を担う一定数の移動体に複製を配置し、それらがクオラムを構成することで、複製間の一貫性を厳密に保持する方式が提案されている。しかし、文献3)では、具体的なクオラム構成法に関する議論がなされていない。文献7)では、様々な一貫性レベルを定義し、クオラムシステムを用いた実現方式について考察がなされている。その中で、厳密な一貫性を保持する方式(GC方式)では、移動体の移動可能領域を複数の部分領域に分割し、各部分領域を管理する移動体間で一貫性を管理する。このとき、書き込み、および読み出し操作のクオラムを構成する移動体数の和を、部分領域の総数より大きくすることで、複製間の一貫性を厳密に保持する。この方式は、複製間の一貫性を厳密に保持する点で、本研究のアプローチと類似している。しかし、クオラムを構成する移動体数が大きい場合、通信オーバーヘッドによるトラフィックが増加してしまう。

一方、文献1),4),9)-11),13)では、データの利用率を向上させるために、クオラムを構成する全移動体が相互接続していない場合でも、書き込み（読み出し）操作を実行する楽観的なアプローチに基づく方式を提案している。文献9)では、クオラムを構成する際、相互接続していなかった移動体を記録し、記

録された移動体が含まれるクォラムを一定時間選択しないことで、クォラムの構成が失敗する回数を削減する方式を提案している．文献 11) では、確率的なクォラムシステムとゴシップ型のメッセージ制御を用いることで、データの利用性を向上させる方式を提案している．また、文献 10) では、書き込み操作を東西方向、読み出し操作を南北方向にそれぞれ伝播させることで、できる限り最近に更新されたデータを読み出す方式を提案している．これらの方式は、複製間の一貫性を厳密に保持しないため、移動体がつねに最新のデータを読み出せるとは限らない．

また、文献 6) では、アドホックネットワークにおいて、データ更新を考慮した複製配置方式が提案されている．また、文献 8) では、更新データを配布する方式が提案されている．これらの方式は、更新発生後の古い複製を無条件に無効と判断することで、ネットワーク内に存在する複製間の一貫性を厳密に保持する．しかし、移動体が最新のデータにアクセスできることを保証しないため、複製間の厳密な一貫性を保持できない．また、暫定的な古い複製へのアクセスを認めるため、無駄なデータアクセスやロールバック処理によるシステム性能の低下が考えられる．

### 3. 想定環境

本論文では、各移動体が、自身の記憶領域にデータの複製を配置し、自身または相互接続している移動体の持つデータにアクセスするアドホックネットワークを想定する．ここで、相互接続とは、1 ホップ以上の無線リンクで接続し、互いに通信可能な状態を指す．

各移動体がネットワーク内に存在するすべての複製間の一貫性を厳密に管理するためには、複製を保持しているすべての移動体と相互接続している必要がある．しかし、アドホックネットワークでは、移動体の移動にともなう無線リンクの切断によってネットワークの分断が発生するため、複製を持つすべての移動体にアクセスできない可能性がある．そのため、文献 7), 12) と同様に、動的クォラムに基づくクォラムシステムを用いて、複製間の一貫性を管理する．また、移動体の移動可能領域を複数の部分領域に分割し、その部分領域（以下では単に「領域」と称する）を担当する移動体間で一貫性管理を行うものとする．

想定環境の詳細を以下に示す．

- (1) 移動体として、ピアとプロキシの 2 種類が存在する．
- (2) ピアとプロキシは、GPS などにより自身の現在位置を把握し、領域内を移動する．各プロキシ

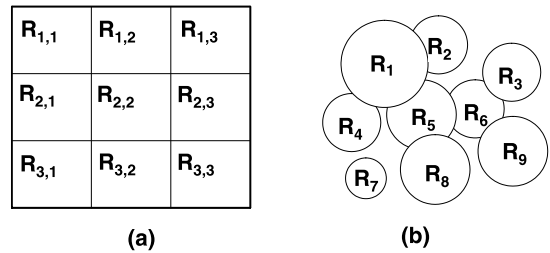


図 2 部分領域の構成例  
Fig. 2 Examples of regions.

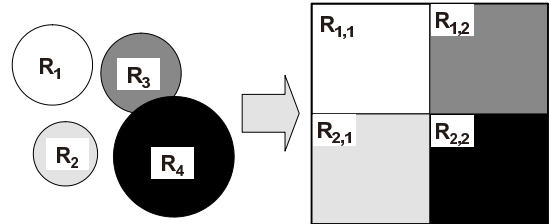


図 3 領域の変換  
Fig. 3 Transformation of regions.

シは、特定の領域を担当し、自身の担当領域の内部のみを移動する．

- (3) 各プロキシが担当する領域の構成法には、特に制限を設けない．たとえば、図 2 (a) のような共通部分および隙間のない方形領域でもよいし、図 2 (b) のような共通部分や隙間のある円形領域でもよい．ここで、図 2 (b) のような場合を含め、任意の領域の構成法に対して、適当なラベル付けを行うことにより、仮想的に図 2 (a) のような  $l$  行  $\times$   $m$  列の領域と見なすことが可能である (図 3)．そこで本論文では、図 2 (a) のような領域の構成を用いて、議論を進める．
- (4) すべてのピアとプロキシは、各領域の識別子と、移動可能領域内における地理的位置、およびその領域を担当しているプロキシを把握している．
- (5) ピアおよびプロキシは、各データに対し、不定期に書き込みを行う．
- (6) 各ピアは、自身および他のプロキシが持つデータに対して、データの読み出しを行う．
- (7) 各プロキシの記憶領域に制限はなく、ネットワーク内に存在するすべてのデータの複製を配置できる．この想定は実環境では必ずしも妥当ではないが、文献 7), 9), 11) にならい、簡単化のために行っている．また、各移動体の位置情報や作業の進捗情報などのサイズの小さなデータを共有する場合などでは、この想定が妥当となる．
- (8) クォラムは、複数のプロキシで構成される．

$i$  行  $j$  列の領域  $R_{i,j}$  を担当するプロキシ  $r_{i,j}$  が書き込み操作および読み出し操作に用いるクオーラムを、それぞれ  $QW_{i,j}, QR_{i,j}$  と表記する。

- (9) 各ピアがデータの書き込み（読み出し）を行う場合、自身が所属する領域を担当するプロキシに書き込み（読み出し）操作依頼を送信する。各依頼を受信したプロキシは、書き込み（読み出し）操作のクオーラムを構成し、クオーラム内のプロキシに書き込み（読み出し）操作要求を送信する。

ここで、上記の想定環境に合致する応用例として、緊急災害時の救助活動における作業の効率化があげられる。この場合、作業の開始前に全災害地域を複数の作業領域に分割し、各作業グループのリーダーがプロキシとして分割領域を担当する。ここで、リーダー間で共有情報の一貫性を管理することで、リーダーは自身のグループ内の作業状況を把握し、領域内のピアに対して効率的に作業を指示できる。

#### 4. CQ (Crisscross Quorum) 方式

筆者らが文献 [12] において提案した CQ 方式では、書き込み（読み出し）操作クオーラムを、自身の担当領域を含む横 1 行（縦 1 列）の領域を管理するプロキシで構成することで、書き込み操作クオーラムと読み出し操作クオーラムの間で共通集合が存在することを保証する。たとえば、図 4 のように、書き込み操作および読み出し操作のクオーラムを、それぞれ領域  $\{R_{3,1}, R_{3,2}, R_{3,3}, R_{3,4}\}, \{R_{1,2}, R_{2,2}, R_{3,2}, R_{4,2}\}$  を担当するプロキシで構成することで、領域  $R_{3,2}$  を担当するプロキシが共通部分となり、複製間の厳密な一貫性が保持できる。ここで、

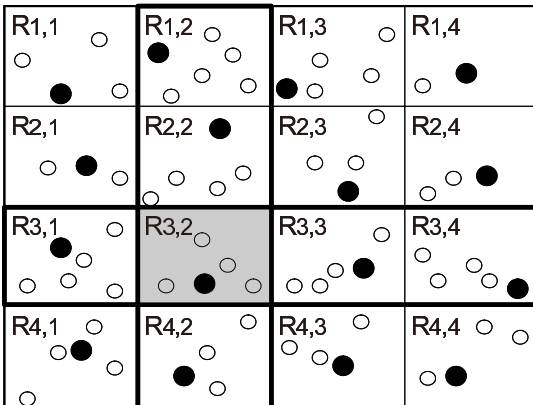


図 4 クオーラムの構成  
Fig. 4 Quorum construction.

図 4 において、黒丸および白丸は、それぞれ各領域を担当するプロキシとその他のピアを表す。

また、CQ 方式では、ネットワークの接続状況に応じて、クオーラムを動的に再構成することで、特定の横 1 行（縦 1 列）の領域を担当する全プロキシと接続していない場合でも、書き込み（読み出し）操作の失敗を防ぐ。具体的には、書き込み操作クオーラムは、それまでに書き込み操作クオーラムとして選択されていない別の横 1 行の領域を管理するプロキシで再構成する。一方、読み出し操作クオーラムは、相互接続していないプロキシのみを、そのプロキシと同じ行の領域を管理するプロキシに交換して再構成する。CQ 方式では、書き込み操作クオーラムを、横 1 行の領域を担当する全プロキシで構成するため、読み出し操作クオーラムを、各行から 1 つずつ移動体を選択して、それらの移動体で構成することで、複製間の厳密な一貫性を保持できる。そのため、読み出し操作クオーラムは、書き込み操作クオーラムよりも柔軟に構成できる。

#### 5. CQ/P (CQ with Pointer) 方式

CQ/P 方式では、CQ 方式と同様に、書き込み（読み出し）操作クオーラムを、自身の担当領域を含む横 1 行（縦 1 列）の領域を管理するプロキシで構成する。ただし、書き込み操作に要するトラフィックを削減するため、書き込み操作は、自身およびクオーラム内の一部のプロキシにのみ実行し、書き込み操作が実行されたプロキシの情報（ポインタ情報）を書き込み操作クオーラム内の全プロキシに送信する。一方、読み出し操作を実行する場合、読み出し操作クオーラム内の全プロキシからポインタ情報を収集する。読み出し操作クオーラムには、最新のポインタ情報を持つプロキシが必ず 1 つ以上存在するため、最新のデータを持つプロキシを把握できる。

以下では、複製間の一貫性を管理する具体的な手順について述べる。

##### 5.1 書き込み（読み出し）操作

各プロキシは、各データに対して、タイムスタンプ（そのデータが最近更新された時間）とポインタ情報（書き込み操作が実行されたプロキシの識別子）を管理する。また、書き込み（読み出し）操作を実行する際、表 1 に示すメッセージを用いる。ここで、表 1 中の“依頼ピア ID”は書き込み（読み出し）操作を依頼するピアの識別子、“要求プロキシ ID”は書き込み（読み出し）操作を要求するプロキシの識別子、“受領プロキシ ID”は書き込み（読み出し）操作を受領す

表 1 CQ/P 方式で用いられるメッセージ  
Table 1 Messages in CQ/P method.

メッセージ名	要素
操作依頼	依頼ピア ID, データ ID, フラグ, 要求プロキシ ID
依頼受領	依頼ピア ID, 要求プロキシ ID
クォーラム構成	要求プロキシ ID, データ ID, フラグ, プロキシ ID リスト
クォーラム構成可能 (書き込み)	要求プロキシ ID, データ ID, 受領プロキシ ID
クォーラム構成可能 (読み出し)	要求プロキシ ID, データ ID, TS, 受領プロキシ ID, ポインタ情報リスト
書き込み操作	要求プロキシ ID, データ ID, TS, データ, プロキシ ID リスト, ポインタ情報リスト
情報更新	要求プロキシ ID, データ ID, TS, プロキシ ID リスト, ポインタ情報リスト
読み出し操作	要求プロキシ ID, データ ID, 受領プロキシ ID
完了 (書き込み)	要求プロキシ ID, データ ID, 受領プロキシ ID
完了 (読み出し)	要求プロキシ ID, データ ID, データ, 受領プロキシ ID

るプロキシの識別子, “データ ID” は書き込み (読み出し) 操作が実行されるデータの識別子, “プロキシ ID リスト” はクォーラムを構成するプロキシの識別子, “ポインタ情報リスト” は書き込み操作が行われたプロキシの識別子, “TS” はタイムスタンプを表す. また, フラグは, メッセージが書き込み操作か読み出し操作かを表す. なお, 表 1 において, 太字で示されたメッセージおよび要素は, CQ/P 方式においてのみ用いられ, それ以外は, CQ 方式と CQ/P 方式の両方式において用いられる.

以下では, 領域  $R_{I,J}$  に存在するピアが, あるデータに対して書き込みおよび読み出し操作を実行する動作について, それぞれ説明する.

#### [ 書き込み操作 ]

ピアは, まず自身が存在する領域  $R_{I,J}$  を管理するプロキシ  $r_{I,J}$  に対して, 操作依頼メッセージを送信する. 操作依頼メッセージを受信した  $r_{I,J}$  は, 依頼受領メッセージをピアに返信する. ピアは, 一定時間内に  $r_{I,J}$  からの返信がない場合,  $r_{I,J}$  と相互接続していないと判断し, 別のプロキシに操作依頼メッセージを送信する. ピアは,  $R_{I,J}$  に近い領域から順に, プロキシが見つかるまで, この操作を繰り返す. 1 つのプロキシも見つからなかった場合, 書き込み操作は失敗となる.

操作依頼メッセージを受信したプロキシ  $r_{I,J}$  は, CQ 方式におけるクォーラム構成法に基づいて, 書き込み操作クォーラム  $QW_{I,J}$  を構成し, 構成したクォーラム内の全プロキシに, クォーラム構成メッセージを送信する. クォーラム構成メッセージを受信したプロキシは, 要求元のプロキシ  $r_{I,J}$  に, クォーラム構成可能メッセージを返信する. プロキシ  $r_{I,J}$  は, クォーラム内の全プロキシからクォーラム構成可能メッセー

ジを受信すると, クォーラムを構成するプロキシのうち, 自身からのホップ数が小さい  $X$  ( $1 \leq X \leq M|M$  はクォーラムを構成するプロキシの数) 個のプロキシに書き込み操作メッセージを, それ以外のプロキシに情報更新メッセージを送信する. これらのメッセージにおいて, タイムスタンプはメッセージの送信時刻に設定する. また, ポインタ情報リストには, 書き込み操作を実行する  $X$  個のプロキシの識別子を格納する. 書き込み操作メッセージを受信したプロキシは, 自身の持つデータに対して書き込みを行い, そのデータのタイムスタンプ, およびポインタ情報を更新する. 一方, 情報更新メッセージを受信したプロキシは, メッセージに含まれるデータ ID に対応するデータのタイムスタンプ, およびポインタ情報を更新する. なお, 全プロキシからクォーラム構成可能メッセージを受信できなかった場合, CQ 方式におけるクォーラム再構成法に基づいてクォーラムを再構成し, 同様の操作を行う. これらの操作がすべて失敗し, プロキシがそれ以上クォーラムを再構成できなかった場合, 書き込み操作の失敗をピアに通知する.

図 5 に,  $X = 1$  の場合に,  $r_{2,2}$  が, 時刻 80 において 3 行目の領域を担当するプロキシ  $\{r_{3,1}, r_{3,2}, r_{3,3}, r_{3,4}\}$  で書き込み操作クォーラムを構成し, データ  $D_1$  の書き込み操作を実行する手順を示す. 表 2 (a) は, プロキシが管理する情報を示し, 各プロキシは, 自身の識別子に対応した行の情報を保持している. また, 表中の要素は左側の値がデータのタイムスタンプ, 右側がポインタ情報リストを示す. たとえば, プロキシ  $r_{1,1}$  は, データ  $D_1$  のうち時刻 70 に更新されたものを  $r_{1,1}$  自身が保持しており, データ  $D_2$  のうち時刻 30 に更新されたものをプロキシ  $r_{1,2}$  が保持していることを把握している. 書き込み操作クォーラム

表 2 プロキシが管理する情報

Table 2 Information managed by proxies.

(a) 更新前				(b) 更新後			
	$D_1$	$D_2$	...		$D_1$	$D_2$	...
$r_{1,1}$	70, $r_{1,1}$	30, $r_{1,2}$		$r_{1,1}$	70, $r_{1,1}$	30, $r_{1,2}$	
$r_{1,2}$	70, $r_{1,1}$	30, $r_{1,2}$		$r_{1,2}$	70, $r_{1,1}$	30, $r_{1,2}$	
$r_{1,3}$	70, $r_{1,1}$	30, $r_{1,2}$		$r_{1,3}$	70, $r_{1,1}$	30, $r_{1,2}$	
$r_{1,4}$	70, $r_{1,1}$	30, $r_{1,2}$		$r_{1,4}$	70, $r_{1,1}$	30, $r_{1,2}$	
$r_{2,1}$	50, $r_{2,4}$	40, $r_{2,3}$		$r_{2,1}$	50, $r_{2,4}$	40, $r_{2,3}$	
$r_{2,2}$	50, $r_{2,4}$	40, $r_{2,3}$		$r_{2,2}$	<b>80,<math>r_{3,2}</math></b>	40, $r_{2,3}$	
$r_{2,3}$	50, $r_{2,4}$	40, $r_{2,3}$		$r_{2,3}$	50, $r_{2,4}$	40, $r_{2,3}$	
$r_{2,4}$	50, $r_{2,4}$	40, $r_{2,3}$		$r_{2,4}$	50, $r_{2,4}$	40, $r_{2,3}$	
$r_{3,1}$	40, $r_{3,2}$	50, $r_{3,3}$		$r_{3,1}$	<b>80,<math>r_{3,2}</math></b>	50, $r_{3,3}$	
$r_{3,2}$	40, $r_{3,2}$	50, $r_{3,3}$		$r_{3,2}$	<b>80,<math>r_{3,2}</math></b>	50, $r_{3,3}$	
$r_{3,3}$	40, $r_{3,2}$	50, $r_{3,3}$		$r_{3,3}$	<b>80,<math>r_{3,2}</math></b>	50, $r_{3,3}$	
$r_{3,4}$	40, $r_{3,2}$	50, $r_{3,3}$		$r_{3,4}$	<b>80,<math>r_{3,2}</math></b>	50, $r_{3,3}$	
$r_{4,1}$	20, $r_{4,1}$	60, $r_{4,2}$		$r_{4,1}$	20, $r_{4,1}$	60, $r_{4,2}$	
$r_{4,2}$	20, $r_{4,1}$	60, $r_{4,2}$		$r_{4,2}$	20, $r_{4,1}$	60, $r_{4,2}$	
$r_{4,3}$	20, $r_{4,1}$	60, $r_{4,2}$		$r_{4,3}$	20, $r_{4,1}$	60, $r_{4,2}$	
$r_{4,4}$	20, $r_{4,1}$	60, $r_{4,2}$		$r_{4,4}$	20, $r_{4,1}$	60, $r_{4,2}$	

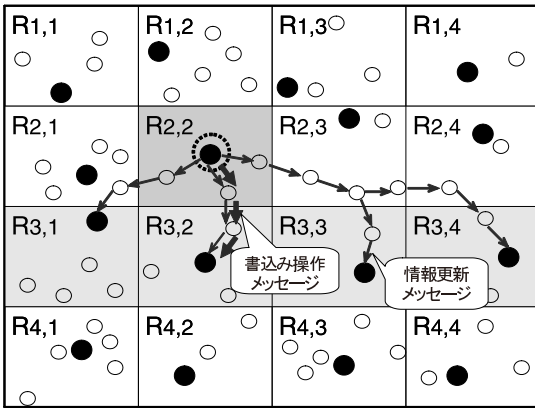


図 5 書き込み操作の例

Fig. 5 Example of executing the write operation.

内の全プロキシからクォーラム構成可能メッセージを受信した  $r_{2,2}$  は、自身からのホップ数が最も小さいクォーラム（ここでは  $r_{3,1}$  および  $r_{3,2}$ ）のうち1つ（ここでは  $r_{3,2}$ ）を選択し、書き込み操作メッセージを送信する。また、クォーラム内のプロキシで書き込み操作を実行しないプロキシ  $\{r_{3,1}, r_{3,3}, r_{3,4}\}$  に、書き込み操作を実行したプロキシ  $r_{3,2}$  の識別子を含む情報更新メッセージを送信する。図5では、太い矢印は書き込み操作メッセージ、細い矢印は情報更新メッセージを示す。各プロキシは、書き込み操作メッセージ、または情報更新メッセージに従い、自身の管理する情報を更新する。書き込み操作完了後に各プロキシが管理する情報を、表2(b)に示す。

ここで、要求元プロキシは最新のデータを必ず取得しているため、自身が書き込み操作クォーラムに属

していない場合でも、自身の持つデータに対して書き込み操作を行うことができる。たとえば図5において、要求元プロキシ  $r_{2,2}$  は、 $r_{3,2}$  に書き込み操作メッセージを送信すると同時に、自身の持つデータに対して書き込み操作を行うことができる。この場合、ポインタ情報リストに要求元プロキシの識別子を含めることで、読み出し操作の際に最新のデータを読み出せる可能性が高くなる。しかし、書き込み操作に用いるメッセージのサイズが増加するため、書き込み操作に要するトラフィックが若干増加してしまう。この影響は、書き込み操作の頻度や、書き込むデータのサイズによって変動するため、CQ/P方式では、要求元プロキシが書き込み操作クォーラムに属していない場合に、自身に書き込み操作を行うかどうかを選択可能とする。

[読み出し操作]

ピアは、書き込み操作と同様の手順で、操作依頼メッセージを送信する。

操作依頼メッセージを受信したプロキシ  $r_{I,J}$  は、CQ方式におけるクォーラム構成法に基づいて、読み出し操作クォーラム  $QR_{I,J}$  を構成し、構成したクォーラム内の全プロキシに、クォーラム構成メッセージを送信する。このメッセージを受信したプロキシは、要求元のプロキシ  $r_{I,J}$  に、クォーラム構成可能メッセージを返信する。このメッセージのTSとポインタ情報リストには、クォーラム構成メッセージに含まれるデータIDに対するタイムスタンプとポインタ情報が含まれる。プロキシ  $r_{I,J}$  は、クォーラム内の全プロキシからクォーラム構成可能メッセージを受信すると、これらのメッセージと自身が管理する情報から、

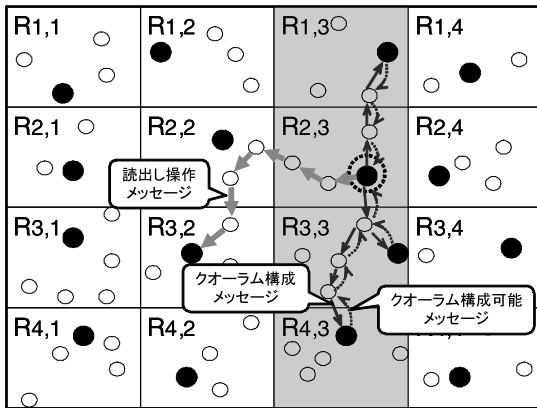


図 6 読み出し操作の例

Fig. 6 Example of executing the read operation.

対象とするデータの最新のタイムスタンプを割り出す。プロキシ  $r_{I,J}$  は、最新のタイムスタンプを持つポインタ情報リストに含まれる（最新のデータを持つ）プロキシに、読み出し操作メッセージを送信し、データを読み出す。なお、ポインタ情報リストに複数のプロキシが含まれる場合、自身の担当する領域から近い順に、読み出し操作を実行するプロキシを決定する。一方、全プロキシからクォーラム構成可能メッセージを受信できなかった場合、CQ方式におけるクォーラム再構成法に基づいてクォーラムを再構成し、同様の操作を行う。これらの操作がすべて失敗し、プロキシがそれ以上クォーラムを再構成できなかった場合、読み出し操作の失敗をピアに通知する。

図 6 に、プロキシ  $r_{2,3}$  が、時刻 90 において、3 列目の領域を担当するプロキシ  $\{r_{1,3}, r_{2,3}, r_{3,3}, r_{4,3}\}$  で読み出し操作クォーラムを構成し、データ  $D_1$  の読み出し操作を実行する手順を示す。ここで、各プロキシは、表 2 (b) に示す情報を管理しているものとする。クォーラム構成可能メッセージをプロキシ  $\{r_{1,3}, r_{3,3}, r_{4,3}\}$  から受信したプロキシ  $r_{2,3}$  は、これらのメッセージに含まれる TS とポインタ情報、および自身が管理する情報を解析する。たとえば、プロキシ  $r_{1,3}$  からのクォーラム構成可能メッセージに含まれる情報（図 2 (b) の 4 行 2 列目）から、データ  $D_1$  のうち時刻 70 において更新されたものをプロキシ  $r_{1,1}$  が持っていることを把握する。同様に、プロキシ  $r_{2,3}$  は、プロキシ  $\{r_{3,3}, r_{4,3}\}$  からのメッセージに含まれる情報（それぞれ表 2 (b) の 12 行 2 列目、16 行 2 列目）および自身が管理する情報（表 2 (b) の 8 行 2 列目）を調べ、そのうちプロキシ  $r_{3,3}$  からのメッセージに含まれる情報から、最新のタイムスタンプが 80 であり、そのデータを持つプロキシが  $r_{3,2}$  であることを把握する。

プロキシ  $r_{2,3}$  は、最新のデータを持つプロキシ  $r_{3,2}$  に読み出し操作メッセージを送信し、データ  $D_1$  を読み出す。図 6 では、太い矢印、細い矢印、および点線の矢印は、それぞれ読み出し操作メッセージ、クォーラム構成メッセージ、およびクォーラム構成可能メッセージを示す。

## 5.2 考察

CQ/P方式では、プロキシは、書き込み操作クォーラム内の一部のプロキシのみに書き込み操作を実行し、それ以外のプロキシには更新データそのものを含まない情報更新メッセージを送信する。これにより、更新データそのものを含むメッセージをクォーラム内の全プロキシに送信する CQ方式と比較して、書き込み操作に要するトラフィックを削減できる。このとき、ポインタ情報は書き込み操作クォーラム内の全プロキシに送信される。そのため、読み出し操作を実行する場合、読み出し操作クォーラム内の全プロキシからポインタ情報を収集することで、最新のデータを持つプロキシを必ず把握でき、CQ方式と同様に複製間の一貫性を厳密に保持できる。しかし、CQ/P方式では、書き込み操作の際に更新データそのものを送信するプロキシ数が減少するため、CQ方式と比較して最新のデータを持つプロキシが少なくなる。そのため、CQ方式と異なり、読み出し操作クォーラムが構成できた（読み出し操作クォーラム内の全プロキシと相互接続している）場合でも、最新の複製を読み出せるとは限らず、データの利用率が低下する可能性がある。

また、CQ/P方式では、クォーラム内の  $X$  個のプロキシに書き込み操作を実行するため、 $X$  の大小によって、書き込み操作に要するトラフィックとデータの利用率の間にトレードオフが発生する。たとえば、 $X$  を小さく設定した場合、書き込み操作に要するトラフィックは大幅に削減できるものの、最新の複製を持つプロキシが少なくなるため、データの利用率が低下する。一方、 $X$  を大きく設定した場合、最新の複製を持つプロキシと相互接続しやすくなるため、データの利用率は維持できる反面、CQ方式に対するトラフィックの減少量が小さくなる。

## 6. 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。

### 6.1 シミュレーションモデル

シミュレーション実験では、 $L[m] \times L[m]$  の 2 次元平面上を、それぞれが  $L/R[m] \times L/R[m]$  の  $R \times R$

表 3 メッセージのサイズ  
Table 3 Message size.

メッセージ名	サイズ
クォーラム構成	3+(プロキシ ID リストの要素数)
書き込み操作のクォーラム構成可能	3
読み出し操作のクォーラム構成可能 (CQ)	4
読み出し操作のクォーラム構成可能 (CQ/P)	4+(ポインタ情報リストの要素数)
書き込み操作 (CQ)	10,000+3+(プロキシ ID リストの要素数)
書き込み操作 (CQ/P)	10,000+3+(プロキシ ID リストの要素数) +(ポインタ情報リストの要素数)
情報更新 (CQ/P)	3+(プロキシ ID リストの要素数) +(ポインタ情報リストの要素数)
読み出し操作	3
書き込み操作の完了	3
読み出し操作の完了	10,000+3

表 4 パラメータ設定

Table 4 Parameter configuration.

パラメータ	値
$L$	500 (200 ~ 600)
$R$	6 (4 ~ 10)
$X$	1 (1 ~ R)

領域に分割した。移動体数を 200 とし、そのうち  $R^2$  個の移動体をプロキシ、その他の移動体をピアとした。各移動体は、すべての方向に等確率に、0 から 10 [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。各移動体の無線通信範囲は半径 50 [m] の円とし、各移動体の読み出し操作と書き込み操作の頻度は、ともに 0.08 [1/秒] とした。すなわち、各移動体の読み出し操作と書き込み操作の発行割合は 1 対 1 とした。また、本実験では、筆者らが文献 [12] で提案した CQ 方式を比較対象とした。

各メッセージに含まれるデータ以外の項目の情報量を 1 とし、データのサイズを 10,000 とした。このときの各メッセージのサイズを表 3 に示す。また、シミュレーション実験で用いたパラメータを表 4 に示す。各パラメータは、基本的には定数値をとるが、いくつかのパラメータは、シミュレーション実験において表中の括弧内の範囲で変化させた。

CQ/P 方式において、書き込み操作時に要求元プロキシが書き込み操作用クォーラムに属していない場合、自身の持つデータに書き込み操作を行い、自身の識別子をポインタ情報リストに含めて送信するものとした。

以上の環境において、各移動体の初期位置をランダムに決定し、10,000 [秒] を経過させたときの、以下の評価値について調べた。

- 書き込み (読み出し) 成功率  
シミュレーション時間内に発生した書き込み (読み出し) 操作要求の総数に対する、書き込み (読

み出し) 成功回数の割合。

- 制御トラヒック

シミュレーション時間内に発生した、表 3 中のメッセージの転送ホップ数に、メッセージのデータを除くサイズを掛けたものの総和。ただし、操作依頼、および依頼受領メッセージについては、すべての方式で等しいため制御トラヒックには含まれない。

- データ転送によるトラヒック

シミュレーション時間内に発生した、データが含まれるメッセージの転送に要する通信ホップ数にデータのサイズのみを掛けたものの総和。

## 6.2 $L$ の影響

まず、移動領域の大きさ  $L$  を変化させたときのシミュレーション結果を、図 7、図 8、および図 9 に示す。これらの図において、横軸は  $L$  を表し、縦軸はそれぞれ、書き込み (読み出し) 成功率、制御トラヒック、データ転送によるトラヒックを表す。

図 7 の結果より、 $L$  の増加にともなって、いずれの方式も書き込み (読み出し) 成功率が減少することが分かる。これは、 $L$  が大きくなると、移動体どうしが接続する機会が減少し、クォーラムの構成に失敗しやすくなるためである。また、CQ/P 方式において、書き込み成功率が、読み出し成功率よりも高いことが分かる。これは、CQ/P 方式では、自身と書き込み操作クォーラム内の一部のプロキシにのみ書き込み操作を実行するため、最新の複製を持つプロキシが少なく、読み出し操作において最新の複製を読み出せる機会が減少するためである。一方、CQ/P 方式の書き込み成功率は CQ 方式と等しくなることが分かる。これは、両方式において、書き込み操作クォーラムを構成するプロキシ数が等しいためである。

図 8 の結果より、いずれの方式も、 $L$  の増加にと



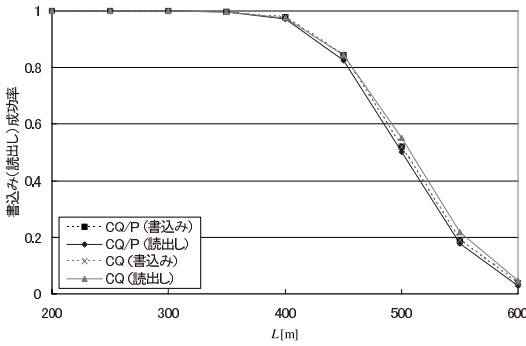


図 7  $L$  と書き込み (読み出し) 成功率  
Fig. 7  $L$  and success ratio.

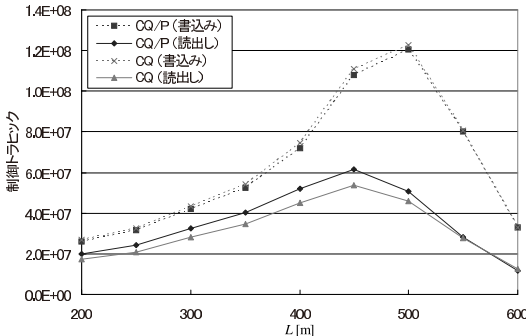


図 8  $L$  と制御トラフィック  
Fig. 8  $L$  and control traffic.

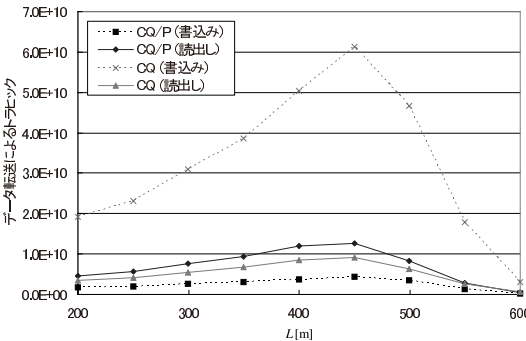


図 9  $L$  とデータ転送によるトラフィック  
Fig. 9  $L$  and data traffic.

もなって、制御トラフィックが増加することが分かる。これは、 $L$  が大きくなると、クォーラムを構成するプロキシ間のホップ数が増加するためである。しかし、 $L$  がさらに大きくなると、移動体どうしが接続する機会が減少し、パケットを転送する移動体が少なくなるため、両方式における制御トラフィックが減少する。また、CQ/P 方式の読み出し操作における制御トラフィックは、CQ 方式より大きいことが分かる。これは、CQ/P 方式では、読み出し操作におけるクォーラム構成可能メッセージにポインタ情報リストが含まれてい

るためである。また、CQ/P 方式の書き込み操作における制御トラフィックは、CQ 方式より若干少ないことが分かる。CQ/P 方式では、情報更新メッセージを書き込み操作クォーラム内の全プロキシに送信するため、CQ 方式より制御トラフィックが増加する。その一方で、CQ/P 方式では、自身の領域から近い領域を担当するプロキシから順に  $X$  個のプロキシに書き込み操作を実行するため、すべてのプロキシに書き込み操作を実行する CQ 方式よりも書き込み操作メッセージによる制御トラフィックは小さい。本シミュレーション実験におけるパラメータ設定 ( $X = 2$ ) では、情報更新メッセージによる制御トラフィックの増加分より、書き込み操作メッセージによる制御トラフィックの減少分が大きくなり、その結果、CQ/P 方式における書き込み操作による制御トラフィックが、CQ 方式より小さくなったと考えられる。

図 9 の結果より、 $L$  の増加にともなって、いずれの方式も、データ転送によるトラフィックが増加することが分かる。これは、 $L$  が大きくなると、クォーラムを構成するプロキシ間のホップ数が増加するためである。しかし、 $L$  がさらに大きくなると、移動体どうしが接続する機会が減少し、パケットを転送する移動体が少なくなるため、両方式におけるデータ転送によるトラフィックが減少する。また、CQ/P 方式の書き込み操作におけるデータ転送によるトラフィックが、CQ 方式より大幅に小さくなることが分かる。これは、CQ/P 方式において、プロキシが、書き込み操作クォーラム内の一部のプロキシにのみ書き込み操作を実行するためである。一方、CQ/P 方式は、読み出し操作におけるデータ転送によるトラフィックが、CQ 方式より若干大きいことが分かる。これは、CQ/P 方式では、書き込み操作を書き込み操作クォーラム内の一部のプロキシにしか実行しないため、読み出し操作クォーラム内のプロキシが最新の複製を保持しない可能性があり、その結果、読み出し操作を実行するプロキシから最新の複製を持つプロキシまでのホップ数が、CQ 方式より大きくなる可能性があるためである。

### 6.3 $R$ の影響

次に、領域分割数  $R$  を変化させたときのシミュレーション結果を、図 10、図 11、および図 12 に示す。これらの図において、横軸は  $R$  を表し、縦軸はそれぞれ、書き込み (読み出し) 成功率、制御トラフィック、データ転送によるトラフィックを表す。

図 10 の結果より、 $R$  の増加にともなって、いずれの方式も書き込み (読み出し) 成功率は増加することが分かる。これは、 $R$  が大きくなると、領域が細分化

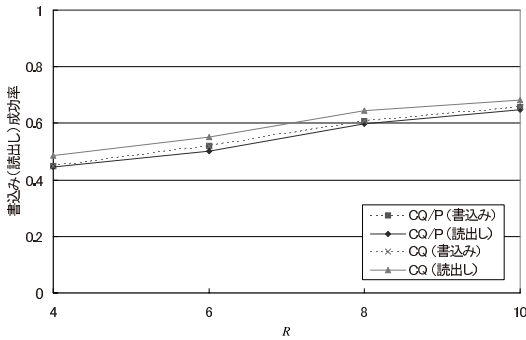


図 10  $R$  と書き込み (読み出し) 成功率  
Fig. 10  $R$  and success ratio.

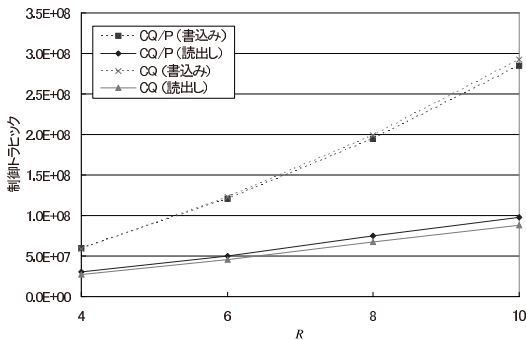


図 11  $R$  と制御トラフィック  
Fig. 11  $R$  and control traffic.

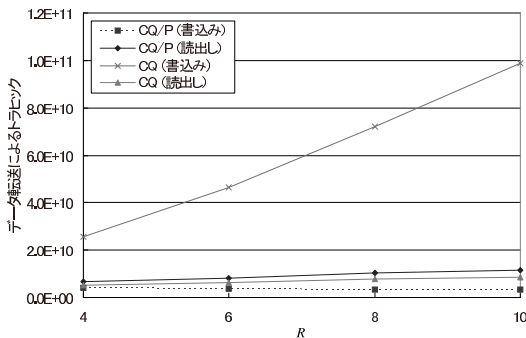


図 12  $R$  とデータ転送によるトラフィック  
Fig. 12  $R$  and data traffic.

されるため、各領域の広さがプロキシの無線通信範囲に対して小さくなり、プロキシ間の接続性が向上するためである。また、CQ/P方式は、CQ方式に比べて、読み出し成功率が低く、書き込み成功率がほぼ等しいことが分かる。これは、図7の結果における考察と同様である。

図11の結果より、 $R$ の増加にともなって、いずれの方式も制御トラフィックが増加することが分かる。これは、 $R$ が大きくなると、クォーラムを構成するプロキシが多くなり、それらの中で交換される制御メッ

セージ数が増加するためである。また、両方式において、書き込み操作における制御トラフィックの増加率が、読み出し操作より大きいことが分かる。これは、読み出し操作のクォーラムを再構成する際、相互接続していないプロキシのみを変更することでクォーラムを再構成するのに対して、書き込み操作のクォーラムを再構成する際、異なる行を用いてクォーラムを再構成するためである。

図12の結果より、 $R$ の増加にともなって、両方式において、読み出し操作におけるデータ転送によるトラフィックが若干増加することが分かる。これは、図10の結果における考察と同様に、 $R$ の増加にともない読み出し成功率が向上するためである。また、CQ方式において、書き込み操作におけるデータ転送によるトラフィックが増加することが分かる。これは、図10の結果における考察と同様である。しかし、CQ/P方式では、 $R$ が増加しても、書き込み操作におけるデータ転送によるトラフィックは増加しないことが分かる。これは、CQ/P方式では、クォーラムを構成するプロキシの数にかかわらず、一定数のプロキシに書き込み操作を実行するためである。

#### 6.4 $X$ の影響

最後に、CQ/P方式において、データを書き込むプロキシの数 $X$ を変化させたときのシミュレーション結果を、図13、図14、および図15に示す。これらの図において、横軸は $X$ を表し、縦軸はそれぞれ、書き込み (読み出し) 成功率、制御トラフィック、データ転送によるトラフィックを表す。

図13の結果より、 $X$ の増加にともなって、読み出し成功率が増加することが分かる。これは、 $X$ の増加にともなって、書き込み操作を実行するプロキシ数が増加し、読み出し操作において最新の複製を読み出せる可能性が高くなるためである。また、 $X$ にかかわらず、書き込み操作の成功率がつねに等しいことが分かる。これは、プロキシが書き込み操作を実行する際、書き込み操作クォーラム内の全プロキシと相互接続している場合のみ、書き込み操作を実行するためである。

図14の結果より、 $X$ の増加にともなって、書き込み操作における制御トラフィックが増加することが分かる。これは、 $X$ が増加することで、ポイント情報に含めるプロキシ数が増加し、情報更新メッセージのサイズが大きくなるためである。また、 $X$ の増加にともなって、読み出し操作における制御トラフィックが若干減少することが分かる。これは、 $X$ が増加することで、より多くのプロキシが最新の複製を持つように

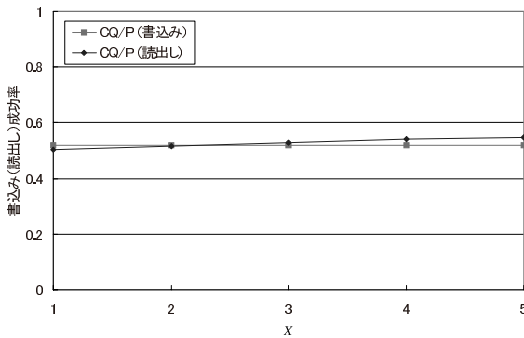


図 13 X と書き込み (読み出し) 成功率  
Fig. 13 X and success ratio.

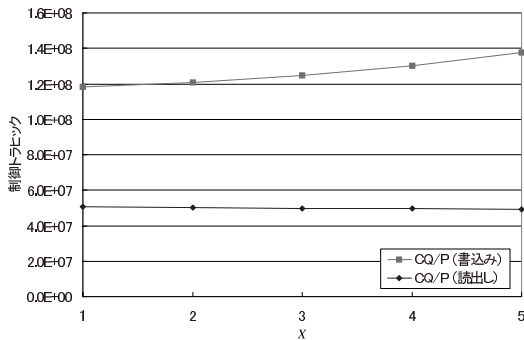


図 14 X と制御トラフィック  
Fig. 14 X and control traffic.

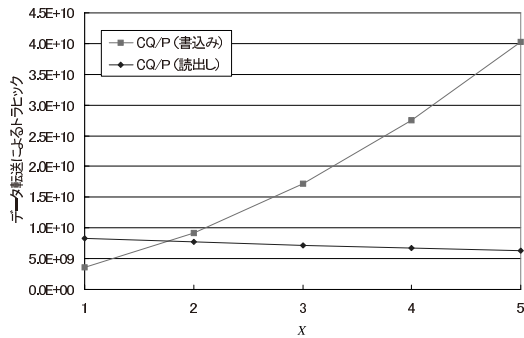


図 15 X とデータ転送によるトラフィック  
Fig. 15 X and data traffic.

なり、読み出し操作に失敗する可能性が低くなることで、クォーラムを再構成する回数が少なくなるためである。

図 15 の結果より、 $X$  の増加にともなって、書き込み操作におけるデータ転送によるトラフィックが増加することが分かる。これは、 $X$  の増加にともなって、書き込み操作を行うプロキシ数が増加するためである。一方、読み出し操作におけるデータ転送によるトラフィックは、 $X$  の増加にともなって減少することが分かる。これは、 $X$  の増加にともなって、より多くのプロ

キシが最新の複製を持ち、より近い領域に存在するプロキシから読み出し操作を実行できるためである。

### 6.5 評価結果のまとめ

シミュレーション実験の結果より、CQ/P 方式によって、CQ 方式における書き込み操作成功率を維持したまま、データ転送によるトラフィックを大幅に削減できることが分かる。

しかし、読み出し操作に関しては、CQ 方式と比較して、CQ/P 方式における成功率が低下し、制御トラフィックやデータ転送によるトラフィックが増加することが分かる。そのため、CQ/P 方式は、読み出し操作の発生頻度が書き込み操作と比較して大きい環境では、その性能を発揮できない可能性がある。一方、6.4 節の結果より、 $X$  を増加させることで、CQ/P 方式における読み出し操作の成功率が増加し、またトラフィックが若干減少することが分かる。したがって、書き込み操作に対して、読み出し操作の発生頻度が大きい環境では、 $X$  を大きく設定することが有効であるものと考えられる。

以上より、CQ/P 方式は、書き込み操作が頻繁に発生する環境において、特に有効な方式であるといえる。このような環境としては、3 章で述べた緊急災害時における救助活動があげられる。この利用形態では、各作業員の作業状況や周辺状況は時々刻々と変化するため、データへの書き込みが頻繁に発生するものと考えられる。また、作業員間で互いのデータを参照する場合、更新されていないデータにアクセスすると、作業に大きな支障をきたすものと考えられるため、複製間の一貫性を厳密に管理することが必要となる。

## 7. おわりに

本論文では、書き込み操作に要するトラフィックを削減するため、ポイント情報を用いて複製間の一貫性を管理する CQ/P 方式を提案した。CQ/P 方式では、筆者らがこれまでに提案した CQ 方式に基づいてクォーラムを構成する。また、データの書き込み操作を一部のプロキシにのみ実行することで、書き込み操作に要するトラフィックを削減する。さらに、ポイント情報を書き込み操作用クォーラム内の全プロキシに送信することで、読み出し操作時に最新の複製を持つプロキシを把握でき、複製間の一貫性を厳密に保持する。

性能評価のために行ったシミュレーション実験の結果から、CQ/P 方式は、CQ 方式と比較して、データ転送によるトラフィックを大幅に削減することを確認した。この結果より、CQ/P 方式は、データのサイズが大きい環境に適しているものと考えられる。

本論文では、複製間の一貫性を厳密に保持することを目的とした。しかし、実環境でのアプリケーションによっては、ユーザが必ずしも最新のデータを読み出す必要がなく、多少古いデータを読み出しても、そのユーザにとってある程度の有効性を持つ情報が得られる場合もある。そこで、今後は、このようなアプリケーションを想定し、複製間の一貫性を緩めることで、データの利用率をさらに向上させる方式について検討する必要がある。

謝辞 本研究の一部は、文部科学省科学研究費補助金・基盤研究(A)(17200006)、および財団法人コミュニケーション基金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) Bhattacharya, S.: Randomized location service in mobile ad hoc networks, *Proc. ACM Int'l Workshop on Modeling Analysis and Simulation of Wireless and Mobile System (MSWiM'03)*, pp.66–73 (2003).
- 2) Cao, G.: Supporting cooperative caching in ad hoc networks, *Proc. IEEE Infocom 2004* (2004).
- 3) Haas, Z.J. and Liang, B.: Ad hoc mobility management with uniform quorum systems, *Proc. IEEE/ACM Trans. Networking*, Vol.7, No.2, pp.228–240 (1999).
- 4) Haas, Z.J. and Liang, B.: Ad hoc mobility management with randomized database groups, *Proc. IEEE ICC'99*, Vol.3, pp.1756–1762 (1999).
- 5) 原 隆浩：アドホックネットワークにおけるデータ利用率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).
- 6) 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).
- 7) Hara, T. and Madria, S.K.: Consistency management among replicas in peer-to-peer mobile ad hoc networks, *Proc. Symposium on Reliable Distributed Systems (SRDS'05)*, pp.3–12 (2005).
- 8) 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).
- 9) Karumanchi, G., Muralidharan, S. and Prakash, R.: Information dissemination in partitionable mobile ad hoc networks, *Proc. Symposium on Reliable Distributed Systems (SRDS'99)*, pp.4–13 (1999).
- 10) Liu, D., Stojmenovic, I. and Jia, X.: A scalable quorum based location service in ad hoc and sensor networks, *Proc. IEEE Int'l Conf. on Mobile Ad hoc and Sensor Systems (MASS'06)*, pp.489–492 (2006).
- 11) Luo, J., Hubaux, J.P. and Eugster, P.: PAN: Providing reliable storage in mobile ad hoc networks with probabilistic quorum systems, *Proc. ACM MobiHoc'03*, pp.1–12 (2003).
- 12) 澤井陽平，篠原昌子，神崎映光，原 隆浩，西尾章治郎：アドホックネットワークにおけるクオラムシステムを用いた一貫性管理方式，情報処理学会論文誌，Vol.48, No.2, pp.551–561 (2007).
- 13) Tchakarov, J.B. and Vaidya, N.H.: Efficient content location in wireless ad hoc networks, *Proc. Int'l Conf. on Mobile Data Management (MDM'04)*, pp.74–85 (2004).
- 14) Wang, K. and Li, B.: Efficient and guaranteed service coverage in partitionable mobile ad-hoc networks, *Proc. IEEE INFOCOM'02* (2002).

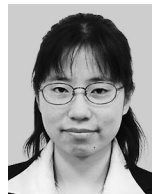
(平成 19 年 5 月 2 日受付)

(平成 19 年 11 月 6 日採録)



澤井 陽平

2005 年京都工芸繊維大学工芸学部電子情報工学科卒業。2007 年大阪大学大学院情報科学研究科博士前期課程修了。現在、株式会社野村総合研究所所属。モバイル環境におけるデータの一貫性管理に興味を持つ。



篠原 昌子 (学生会員)

2004 年大阪大学工学部電子情報エネルギー工学科卒業。2006 年同大学院情報科学研究科博士前期課程修了。現在、同大学院情報科学研究科博士後期課程在学中。モバイル環境における消費電力を考慮したデータ管理に興味を持つ。日本学術振興会特別研究員。日本データベース学会の学生会員。



神崎 映光 (正会員)

2002年大阪大学工学部情報システム工学科卒業。2004年同大学院情報科学研究科博士前期課程修了。2005年同大学院情報科学研究科博士後期課程中退後、同大学院情報科学研究科マルチメディア工学専攻特任助手を経て、2007年より同助教となり、現在に至る。博士(情報科学)。移動体ネットワーク、通信プロトコル、分散処理に興味を持つ。IEEE, 電子情報通信学会, 日本データベース学会の各会員。



原 隆浩 (正会員)

1995年大阪大学工学部情報システム工学科卒業。1997年同大学院工学研究科博士前期課程修了。同年同大学院工学研究科博士後期課程中退後、同大学院工学研究科情報システム工学専攻助手、2002年同大学院情報科学研究科マルチメディア工学専攻助手、2004年より同大学院情報科学研究科マルチメディア工学専攻准教授となり、現在に至る。工学博士。1996年本学会山下記念研究賞受賞。2000年電気通信普及財団テレコムシステム技術賞受賞。2003年本学会研究開発奨励賞受賞。データベースシステム、分散処理に興味を持つ。IEEE, ACM, 電子情報通信学会, 日本データベース学会の各会員。



西尾章治郎 (フェロー)

1975年京都大学工学部数理工学科卒業。1980年同大学院工学研究科博士後期課程修了。工学博士。京都大学工学部助手, 大阪大学基礎工学部および情報処理教育センター助教授, 大阪大学大学院工学研究科情報システム工学専攻教授を経て、2002年より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教授となり、現在に至る。2000年より大阪大学サイバーメディアセンター長, 2003年より大阪大学大学院情報科学研究科長, その後2007年より大阪大学理事・副学長に就任。この間, カナダ・ウォータールー大学, ビクトリア大学客員。データベース, マルチメディアシステムの研究に従事。現在, Data & Knowledge Engineering等の論文誌編集委員。本会理事を歴任。電子情報通信学会フェローを含め, ACM, IEEE等8学会の各会員。