

適応的全順序マルチキャストの拡張

細 谷 篤[†] 佐 藤 文 明^{††} 水 野 忠 則^{††}

分散仮想環境や電子会議システム、分散共有メモリなどの分散アプリケーションにおいて、全順序グループ通信サービスは必要不可欠なインフラとなっている。なかでも全順序マルチキャストについてはこれまでに多くの研究が世界中の研究機関においてなされてきた。通常分散アプリケーションにおいては多数のメンバが参加することが想定され、またそのメンバは広範囲にわたり散在していると考えられるので、システムには規模拡張性と全順序性の保証が要求される。本稿では、全順序性を保証しながらも従来方式よりも規模拡張性の優れた全順序マルチキャストプロトコルの設計を目標とする。その方法として従来方式であるシーケンスサーバ方式と適応的全順序マルチキャストプロトコルを統合した新方式を提案する。本方式では従来方式で指摘されていたシーケンスサーバへの負荷集中、ダミーメッセージの過剰などの問題点を改善することを目的としている。

Extension of Adaptive Totally Ordered Multicast Protocol

ATSUSHI HOSOYA,[†] FUMIAKI SATO^{††} and TADANORI MIZUNO^{††}

In the distributed applications, such as distributed virtual environment, teleconference system and distributed shared memory, Totally Ordered Group Communication Service (TO-GCS) is the essential infrastructure. Especially, totally ordered multicast has been researched all over the world. In general, since it is assumed that many members take part in the group communication and members are scattered in large area, the system is required to guarantee scalability and totally ordering. In this paper, our goal is to design the totally ordered multicast protocol which is better than existing method in terms of scalability. To realize this method, we propose the new method which integrates sequence server method with adaptive totally ordered multicast protocol, guaranteeing totally ordering. The purpose of this method is to improve the problem on load centralization and too many dummy messages.

1. はじめに

分散アプリケーションの設計において、全順序グループ通信サービス (TO-GCS: Totally Ordered Group Communication Service) は必要不可欠なインフラとなっている。分散アプリケーションの例としては、共有仮想環境^{1)~3)}、電子会議システム^{4),5)}、分散共有メモリ、コンピュータ支援協同作業 (CSCW: Computer Supported Cooperative Work) アプリケーションなどがある。

分散アプリケーションにおいては1つのオブジェクトや共有資源に対して複数のユーザがアクセスすることが想定されるので、その更新データは、関係のあるすべてのメンバに知らされなければならない。この通信を1対1通信であるユニキャストで行うとネット

ワークに多大な負荷がかかってしまう。そこで、複数のユーザに対して同じデータを効率的に送信するマルチキャスト技術^{6)~8)}が開発された。マルチキャスト通信はユニキャスト通信と異なり1対多で行われる通信の手法である。マルチキャスト通信ではパケットは受信者に到着するまでにその途中のルータによって複製されるため、ネットワーク資源の浪費を防ぐことができる。

分散アプリケーション上ではメンバ間の順序一貫性を保証するために、更新イベントはすべてのメンバのアプリケーションに同じ順序で配送される必要がある。順序一貫性を保証するマルチキャストのことを全順序マルチキャストという。

全順序マルチキャストにおける従来研究では順序付けにトークンを用いて優先度を付ける方式や、シーケンスサーバを用いてメッセージに順序番号を付ける方式⁹⁾などが提案されてきた。しかしトークンを用いる方式ではメッセージを発生できるメンバが1つだけに限定されてしまうのでイベント発生に対する即応性に

[†] 静岡大学大学院情報学研究科

Graduate School of Information, Shizuoka University

^{††} 静岡大学情報学部

Faculty of Information, Shizuoka University

明らかな限界がある。またシーケンスサーバを用いる方式ではシーケンスサーバに負荷が集中するため、規模拡張性が低いという問題がある。

そこでメッセージの優先度を広い範囲で割り当て、サーバを用いず順序付けを行う適応的全順序マルチキャストプロトコル¹⁰⁾が提案された。後に詳しく述べるが、この方式ではダミーメッセージという空のデータが頻繁に使用される。ダミーメッセージはデータとしては意味を持たないメッセージであるのでその数はできる限り少ない方が好ましいが、この方式では多くのダミーメッセージが発生してしまうという問題がある。

そこで本稿ではこれらの問題を解決するために適応的全順序マルチキャストプロトコルとシーケンスサーバ方式を統合した新方式を提案する。また新方式についてシミュレーション、評価を行い、従来方式で指摘されていた問題点が改善されていることを証明する。

2. 全順序マルチキャスト

2.1 マルチキャスト

分散アプリケーション上では、イベントの更新情報は複数のプロセスに知らされる必要がある。このとき1対1通信であるユニキャストを用いて各プロセスに更新情報を送信するのは非効率的であり、1つのプロセスから複数のプロセスに対してメッセージを送信するマルチキャストの方が適している。マルチキャスト通信ではメッセージの複製は送信者ではなくルータによって行われるのでネットワーク資源の浪費を抑えることができる。

2.2 全順序性

全順序性とはマルチキャストメッセージを受信したメンバすべてにおいて、メッセージが同じ順序でアプリケーションに配送されることをいう。

分散アプリケーションでマルチキャストが使用される際、全順序性が重要な課題となる。もしも全順序性が保証されず、あいまいな順序で更新情報のメッセージがアプリケーションに配送された場合、メンバごとにデータの更新状況に違いが出てしまい、マルチキャストグループ間でデータの一貫性がとれなくなってしまう。つまり全順序性の保証は分散アプリケーションの設計において最も基本的な要求であるといえる。

2.3 従来研究

本節では全順序マルチキャストの従来研究としてシーケンスサーバ方式と適応的全順序マルチキャストプロトコルについて述べ、その問題点を指摘する。

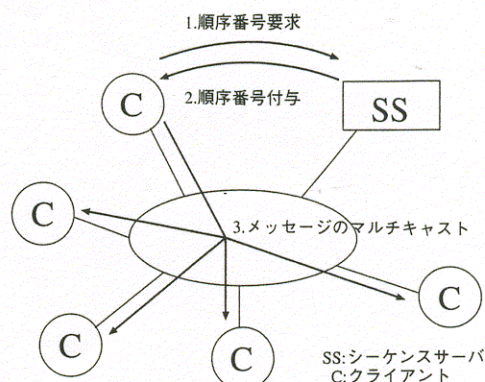


図1 シーケンスサーバ方式
Fig. 1 Sequence server method.

2.4.1 シーケンスサーバ方式

シーケンスサーバ方式では、マルチキャストグループにシーケンスサーバを1つ設置し、このシーケンスサーバが順序管理を行うという方式である(図1)。

メッセージをマルチキャストしようとするメンバはまずシーケンスサーバに順序番号を要求し、シーケンスサーバはその要求に対して順序番号を与える。メンバはその順序番号をメッセージに添付しマルチキャストする。メッセージを受信したメンバはこの順序番号の順序どおりにアプリケーションに配送することで全順序性が保証される。

2.4.2 適応的全順序プロトコル

適応的全順序プロトコル(ATOP: Adaptive Total Ordering Protocol)は各メンバに送信頻度に応じた優先度を広い範囲で割り当て、受信側ではその優先度と各クライアントの持つ乱数生成器によってメッセージの順序付けを行う方式である。すべてのクライアントは同一の乱数生成器を持っていることを前提として全順序性を保証する。またこの優先度の割当てを動的に行うことにより配送をスムーズに行うことを目的とした方式である。

各クライアントのATOPはAP(Adaptive Policy)モジュールとATOM(Adaptive Total Ordering Mechanism)モジュールからなる(図2)。

(1) APモジュール

APモジュールは受信したメッセージとその送信元IDをチェックして、その送信頻度に応じて重みを割り当てる。重みはすべてのクライアントの和が1となるようにし、送信頻度が大きいクライアントは大きな重みが割り当てられる。さらに分布と呼ばれる重みとクライアントを対応付けたリストを作成し、定期的にATOMモジュールに送信する。分布には一連の識別子が付けられる。

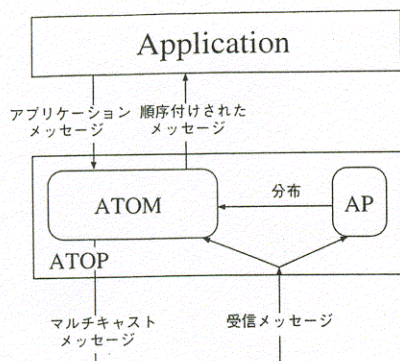


図2 APモジュールとATOPモジュール
Fig. 2 AP module and ATOP module.

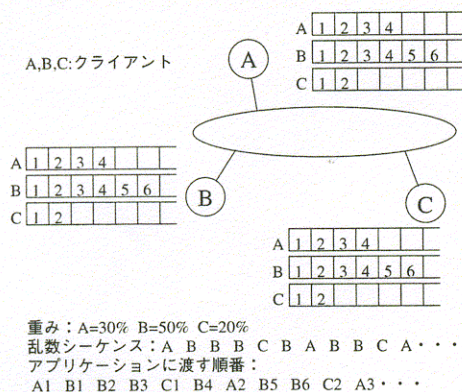


図3 適応的全順序プロトコルにおける順序保証
Fig. 3 Ordering guarantee of ATOP.

(2) ATOM モジュール

ATOM モジュールはメッセージの送信の際に、自らの持つ最新の分布の識別子とクライアント ID、自分のメッセージについての順序番号をメッセージに添付して送信する。受信側の ATOM モジュールでは受信メッセージに添付情報と乱数生成器によってメッセージの順序付けを行い、アプリケーションに配送する。

順序保証

各クライアントは配送の際、分布を基にクライアント ID を要素とする列である乱数シーケンスを乱数生成器により発生する。乱数シーケンスは分布での重みが大きなクライアントほど優先的に登場するように生成される。各クライアントは乱数シーケンスの順にクライアントを選択し、そのクライアントより受信したメッセージを順序番号どおりに配送する。乱数シーケンスはすべてのクライアントで同一のものが生成されることを前提としているので、全順序性は保証される。

図3において分布の重みが $A = 30\%$, $B = 50\%$, $C = 20\%$ であるとする。この分布と乱数生成器によって乱数シーケンスが A, B, B, B, C, B, A, B, B, C, A と生成されたとすると、アプリケーションへ配送されるメッセージの順番は A1, B1, B2, B3, C1, B4, A2,

B5, B6, C2, A3 と決まることになる。これはすべてのクライアントで同じ順序で配送されることになる。

ダミーメッセージ

一定の頻度でメッセージを送信していたクライアントがある時点から突然多くのメッセージを送信し始めたり、あるいは突然沈黙してしまうなど、クライアントの送信頻度が急激に変化したとき、各クライアントでは配送すべきメッセージが受信バッファに存在せず、またなかなか到着しないために配送がブロックされるという問題が発生する。このような状況を回避するために各クライアントはネットワークを監視し、最新の分布中の自クライアントの重みとその時点での送信頻度の割合に隔たりがある場合、分布が更新されるまでの間送信すべきメッセージが存在しないときでもダミーメッセージと呼ばれる空のデータを送信することで送信頻度のバランスを調整する。

2.4.3 従来方式の問題点

シーケンスサーバ方式は順序管理を行うシーケンスサーバに負荷が集中してしまうという問題がある。シーケンスサーバに大きな負荷がかかると、処理効率が低下し遅延を引き起こす。そのためシステムが正常に動作するためにシーケンスサーバの処理能力に合わせたサイズにクライアント数が制限されることになるので、規模拡張性が低いといえる。

ATOP では配送のブロックを回避するためにダミーメッセージを用いているが、重みの割当てをクライアントごとに行っているため、あるクライアントの送信頻度の変化に他のクライアントが敏感に反応しダミーメッセージを送信してしまう。この結果、ダミーメッセージの数が増大しやすいという問題がある。

3. 適応的全順序マルチキャストの拡張

全順序マルチキャストにおける従来方式では前章で述べたような問題点がある。そこで、従来方式であるシーケンスサーバ方式と適応的全順序プロトコル (ATOP) を統合した拡張適応的全順序マルチキャストプロトコル (拡張 ATOP) を提案する。

3.1 システム構成

本方式では、マルチキャストグループ内のすべてのクライアントをいくつかのドメインに分ける (図4)。また、すべてのクライアントは同一の乱数生成器を持っていることを前提としている。

各ドメインにはシーケンスサーバを1つずつ設置し、ドメイン内の順序管理はそれぞれのシーケンスサーバが行う。またドメイン間の順序付けには ATOP のアルゴリズムを適用する。ATOP では送信頻度に応じた

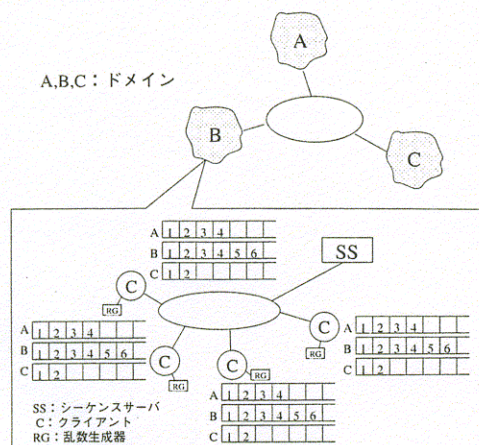


図4 拡張 ATOP のシステム構成

Fig. 4 System structure of extended ATOP.

重み付けをクライアント単位で行っていたが、本方式ではドメイン単位で重み付けを行う。つまりドメイン内全体のメッセージの送信頻度に応じて各ドメインに重みを割り当て、分布を生成するというのである。

ドメイン内の順序はシーケンスサーバが与える順序番号によって保証される。ドメイン間の順序は各クライアントが同一の乱数生成器を持ち、同一の乱数シーケンスを発生することで保証される。したがって、マルチキャストグループ全体の順序が保証されていることになる。

本方式には従来方式と比較して以下のような利点、特徴がある。

(1) シーケンスサーバの負荷軽減

シーケンスサーバ方式ではシーケンスサーバに負荷が集中していたが、本方式ではドメインごとに1つずつ、全体としては複数のシーケンスサーバを設置することでシーケンスサーバの負荷を分散、軽減することができる。

(2) ダミーメッセージの抑制

ATOP ではダミーメッセージの数が多くなってしまうという問題があったが、本方式では送信頻度による重みの管理をドメイン単位で行うことにより、ダミーメッセージを抑制することができる。

あるクライアントが突然沈黙し、しばらくの間メッセージをまったく送信しなかったとしてもドメイン内の他のクライアントはメッセージを送信し続けると考えられるからドメイン全体として沈黙することはありえない。通常ドメイン内では送信頻度が大きくなっているクライアントや小さくなっているクライアントが混在していると考えるのが自然であり、ドメイン全体の送信頻度はそれほど大きな変動を起こさない。送信頻度の変動が小さければ結果的にダミーメッセージの

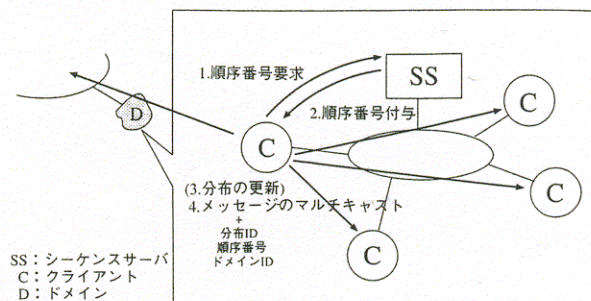


図5 メッセージの送信

Fig. 5 Sending messages.

抑制につながる。またドメイン中のクライアント数が多ければ多いほど変動の傾向の偏りがなくなり、その変動は緩和されやすくなると考えられる。

(3) 重み管理の容易化

ATOP ではクライアント単位で行われていた重みの管理を、ドメイン単位で行うことにより重みを割り当てる対象が減ることになるので、その管理は容易になる。

3.2 全順序性の保証

3.2.1 メッセージの送信

イベントを発生したクライアントは、まずシーケンスサーバに順序番号を要求する。シーケンスサーバは要求に対し、自らの持つ分布と順序番号をクライアントに返信する。順序番号はドメイン内で唯一のものであり、ドメイン内でのメッセージの順序は保証される。シーケンスサーバからの応答を受信したクライアントはサーバより送信された分布の ID と自分の持つ分布の ID とを比較し、自分の持つ分布の方が古いものであったらサーバより送信された分布に更新する。そしてメッセージに分布の ID と順序番号、ドメイン ID を添付しマルチキャストする (図5)。

3.2.2 分布生成と更新

分布は定期的に更新され、単調増加的な ID 番号が付けられる。分布の生成、送信はシーケンスサーバ群のうちの1つが行うことでその一貫性を保証する。

分布はそれが生成された時点でのドメイン群の送信頻度の重みを与えるリストである。メッセージの配送順序は重みに依存して決定されるため、分布はつねにその時点でのグループ内の送信頻度状況を反映したものではなくてはならず、その更新が滞ってはならない。

マルチキャストグループ内の送信頻度状況を反映させた配送が行われるためには、メッセージの送信者はつねに最新の分布の ID 番号を付けてメッセージをマルチキャストする必要がある。順序番号が最新のメッセージは最新の分布 ID を付けて送信されなければならない。つまりドメイン内のメッセージに関して順序

番号の増加に対し、分布 ID が古くなるようなことはあってはならない。同一ドメイン内での順序番号と分布 ID の一貫性を保つために、分布を生成するサーバは新しい分布を生成したとき、まず他のドメインのシーケンスサーバに分布を送信する。そして分布を受信した各ドメインのシーケンスサーバは自分のドメインのクライアントに対して分布を送信する。このように新しい分布を配布することで順序番号の新しいメッセージには、それ以前のメッセージよりも古い分布 ID は付かないことが保証される。

順序分布と送信分布

クライアントにはさまざまな分布 ID が付けられたメッセージが送信されてくる。各クライアントはメッセージを配送する際にはそれぞれの分布に応じた乱数シーケンスを発生させてドメインを選択、配送を行う必要がある。つまり配送に備えて複数の分布を保持していなければならないということになる。

先に述べたように同一ドメイン内の順序番号と分布 ID は一貫性が保証されている。つまり、あるドメインからのメッセージを順序番号どおりに配送していく過程で新しい分布 ID が付けられたメッセージが到着したならば、そのドメインからはその分布 ID よりも古い分布 ID のメッセージが到着することはないことが分かる。すべてのドメインより新しい分布 ID のメッセージが到着したら、それ以前の分布は必要ないことになり破棄することが可能になる。

そこで各クライアントは分布ごとに乱数生成器から乱数シーケンスを発生させ、その分布 ID が付けられたメッセージを次々に配送していくようにする。選択されたドメインのメッセージの分布 ID が現在配送に使用している分布 ID よりも新しいものであった場合、そのドメインの順番はスキップされる（乱数シーケンスの次のドメインを選択して配送する）。すべてのドメインから新しい分布 ID が付けられたメッセージを受信したら、その分布で配送すべきメッセージは存在しないことになるのでその分布を破棄し、配送用の分布を更新する。そして更新された配送用の分布と乱数生成器により乱数シーケンスを発生させ、引き続き配送を行う。この配送用の分布を、そのクライアントが順序付けを行っている分布という意味で順序分布と呼ぶことにする。またクライアントの持つ最新の分布のことを送信時にメッセージに付けられる分布という意味で送信分布と呼ぶことにする。以降、クライアントにおける分布は順序分布、送信分布と区別して表記する。

あるクライアントにおいて、順序分布は送信分布よりも新しいことはなく、順序分布と送信分布の間のす

べての分布を配送のために保持しておく必要がある。

3.2.3 メッセージの受信と配送

クライアントはさまざまなドメインからメッセージを受信する。受信されたメッセージは、メッセージに付けられたドメイン ID から、送信ドメインごとのバッファに仕分けされ溜められる。クライアントは順序分布と乱数生成器により乱数シーケンスを発生、ドメインを選択する。そして選択されたドメインのバッファからメッセージを順序番号どおりに配送する。もし選択されたドメインのバッファにメッセージが存在しない、あるいは配送すべき順序番号のメッセージが到着していないなどの場合は配送はそこでブロックされ、配送すべきメッセージが到着するまで配送はいつさい行われぬ。また選択されたドメインのメッセージが順序分布の ID 番号より新しい分布の ID 番号を持っていたら、現在の順序分布で配送されるべきメッセージはそのドメインからは今後届かないということなので配送はスキップされ、引き続き他のドメインのメッセージを対象に配送が行われる。すべてのドメインから順序分布より新しい分布で送信されたメッセージを受信したら、順序分布を更新し、続けて配送を行う。

3.3 ダミーメッセージ

本方式においても ATOP と同様にダミーメッセージが必要となる。

図 6 のような例を考える。分布が更新されその分布による各ドメインの重みはそれぞれ $A = 20\%$ 、 $B = 30\%$ 、 $C = 50\%$ であった。しかしここでドメイン A の送信頻度が急激に大きくなったとすると、各クライアントはドメイン A から多くのメッセージを受信し、ドメイン A のバッファにメッセージが溜まっていく。しかし分布におけるドメイン A の重みは 20% であるので乱数シーケンスによってドメイン A が選択され

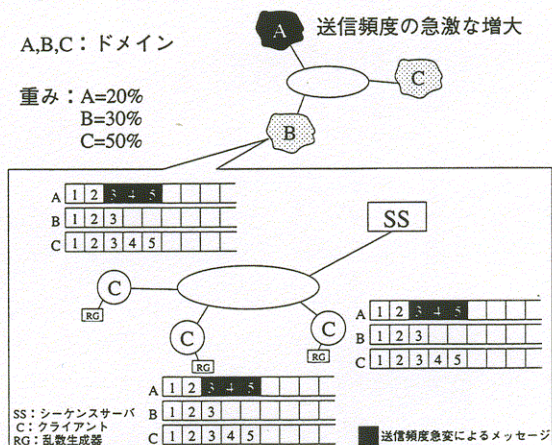


図 6 送信頻度の急激な増大

Fig. 6 Sharp increase of transmission rates.

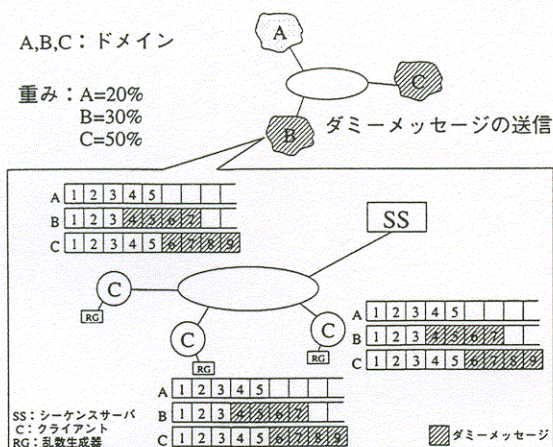


図7 ダミーメッセージによる送信頻度バランスの調整

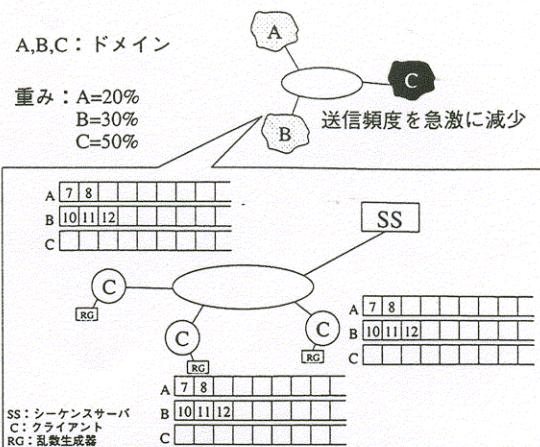
Fig. 7 Adjustment of transmission rates balance with dummy messages.

るのは、およそ5回に1回程度である。次に分布が更新されるまではこの割合で配送され続けるので、ドメイン A が送信頻度を保ったままメッセージを送り続けるとクライアントの受信バッファにはドメイン A からのメッセージが溜まっていく一方になってしまう。クライアントのバッファサイズは有限であるのでこの状況は好ましくない。

そこで各ドメインのシーケンスサーバはネットワークを監視し、その時点での送信頻度の状況が分布の重み付けに近いものであるかどうかを調べるようにする。自分のドメインの送信頻度の割合が分布の重みに届いていない場合、適当な数のダミーメッセージをマルチキャストし、送信頻度のバランスが分布のものに近づくように調整する(図7)。

逆の場合も考える。上の例と同じように分布が更新され各ドメインの重みはそれぞれ A = 20%, B = 30%, C = 50%であった。一定時間後、ドメイン C の送信頻度が急激に小さくなったとすると、その後各クライアントはドメイン C からのメッセージをそれまでの頻度のように受信しない。分布におけるドメイン C の重みは 50%であるから、およそ2回に1回は乱数シーケンスによって配送するドメインに選択されることになるが、ドメイン C からはなかなかメッセージが送信されてこないため、ドメイン C が選択されてもバッファが空であることが多くなり、たびたび配送がブロックされることになる(図8)。

この場合は、ドメイン C 自体のシーケンスサーバがドメイン C の送信頻度の割合が分布の重みに届いていないことを感知し、分布の重みである 50% の送信頻度を保つようにダミーメッセージをマルチキャストすることで各クライアントにおける配送がブロックされることを回避する(図9)。



メッセージをいくつか配送後、ドメイン C からはメッセージがあまり届かなくなり、バッファは空になる。ここでドメイン C が配送ドメインに選択されると配送はブロックされる。

図8 送信頻度の急激な減少

Fig. 8 Sharp decrease of transmission rates.

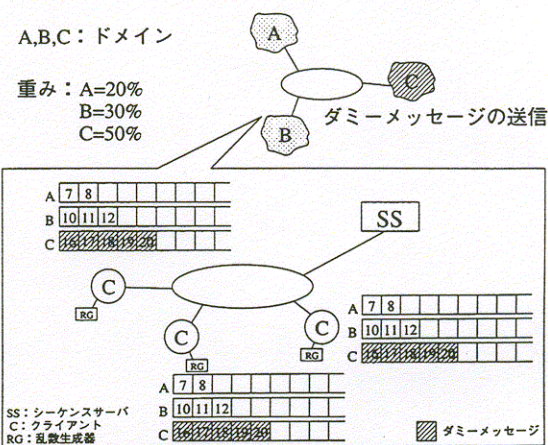


図9 ダミーメッセージによる配送ブロックの回避

Fig. 9 Avoidance of delivery blocking with dummy messages.

4. シミュレーション

本章では提案モデルについてシミュレーションを行い評価する。パラメータの変化により擬似的にシーケンスサーバ方式、ATOP と本方式との比較を行い、シーケンスサーバの負荷がどれだけ軽減されているか、またダミーメッセージがどれだけ抑制できているかについて評価する。

4.1 シミュレーションモデル

今回はダミーメッセージ数とメッセージの平均到着時間についてシミュレーションを行った。両者において共通のパラメータを以下のように設定した。

- メッセージ平均発生間隔 0.5 s/クライアント
- 分布更新間隔: 5 s
- ドメイン内通信の平均通信遅延: 100 ms
- ドメイン外通信の平均通信遅延: 300 ms
- 全クライアント数: 100

- ドメイン数：1, 2, 5, 10, 20, 100
- 送信頻度急変イベント頻度：なし, 1回/15s, 1回/7.5s

ドメイン内のクライアント数はすべてのドメインで同数であるとした。全クライアント数が100であるのでドメイン数1, 2, 5, 10, 20, 100に対し、各ドメインのクライアント数は100, 20, 10, 5, 2, 1である。

ドメイン数1, クライアント数100のときはシーケンスサーバ1台なので擬似的にシーケンスサーバ方式と同じモデルになる。またドメイン数100, クライアント数1のときはメッセージ送信の際にシーケンスサーバを介するものの、結果的にクライアント単位で重みを割り当てることになるので擬似的にATOPと同じモデルとなる。

送信頻度急変イベントとは人為的に送信頻度の変動を起こすイベントで、今回のシミュレーションにおいては全体の8%のクライアントをランダムに抽出し、そのクライアントの送信頻度を10秒間だけ3倍にする。このイベントの回数が多いほど頻繁に送信頻度の変動が起きているということである。

また今回のシミュレーションは、パケットの欠落、マルチキャストグループに対するメンバの参加、脱退はないものとして行っている。

ダミーメッセージ数は1秒間あたりにシステム全体で発生したダミーメッセージ数をデータとしてとっている。このシミュレーションではシーケンスサーバの処理時間は1msで一定としている。

平均到着時間はクライアントがメッセージを発生してから、受信クライアントがメッセージを受信するまでの平均時間である。これにはシーケンスサーバへの順序番号を要求、その応答も含まれる。このシミュレーションはシーケンスサーバの処理時間を4ms, 10ms, 25msと変化させて行っている。

4.2 結果

4.2.1 ダミーメッセージ数

ダミーメッセージ数は送信頻度の変動の大きさに比例して多くなると考えられる。ドメインあたりのクライアント数が多いほどその変動は緩和されやすくなることはすでに3章で述べた。擬似的にATOPと同一のシステム構成になっているドメイン数100, クライアント数1の組合せでは非常に多くのダミーメッセージが発生することが予想され、徐々にドメイン数とクライアント数の割合を逆転させていくことで、ダミーメッセージ数がどれだけ抑制できるのかを検証する。

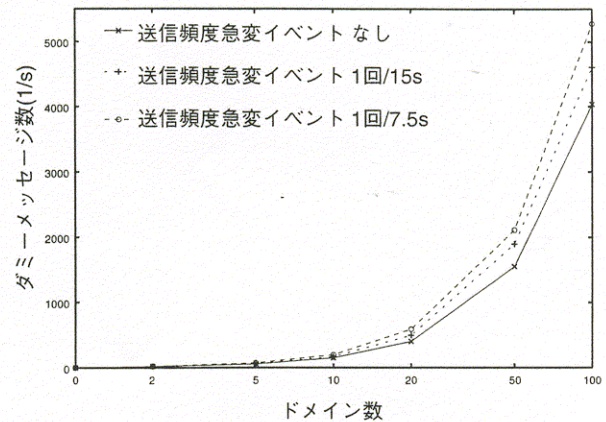


図10 ダミーメッセージ数

Fig. 10 Number of dummy messages.

図10のグラフにおいて、左端の点が擬似的なシーケンスサーバ方式、右端の点が擬似的なATOPである。

シーケンスサーバ方式はダミーメッセージを必要としないシステムなので、ダミーメッセージはいっさい発生しない。これはグラフではドメイン数1の点に相当し、ダミーメッセージが発生していないことが見てとれる。逆にドメイン数100の点においては莫大な数のダミーメッセージが発生しているのが分かる。ドメイン数5, 10, 20の各点でのダミーメッセージ数はドメイン数100のときと比較すると格段に少ない発生量でその数はドメイン数が増えるごとに増加していることが分かる。

またドメイン数1の場合を除き、いずれの場合も送信頻度の変動イベントの回数が多いほどダミーメッセージの数も増加しているのが分かる。

4.2.2 平均到着時間

メッセージ発生から到着までの時間はできる限り早いことが望ましい。ネットワークの平均遅延時間は変動しないので平均到着時間はシーケンスサーバからの応答の速さに依存する。シーケンスサーバにその処理能力以上の要求が集中すると処理待ち時間が長くなり、応答も遅くなる。ここではシーケンスサーバの処理時間をいくつかとり、それぞれの場合において処理限界がどこにありどこで最大のパフォーマンスを出せるのかを検証し、同時にどれほどシーケンスサーバの負荷が分散、軽減されているのかを検証する。

シーケンスサーバの処理時間が速い(4ms)のときは、ドメイン数をいくつにとってもおよそ一定の到着時間である。処理時間を10ms, 25msと遅くしていくとドメインあたりのクライアント数が少ない場合は一定の到着時間であるが、クライアント数が増えていくにつれ到着時間がとたんに大きくなる。このグラフからは処理時間10msのときドメイン数5, つまりド

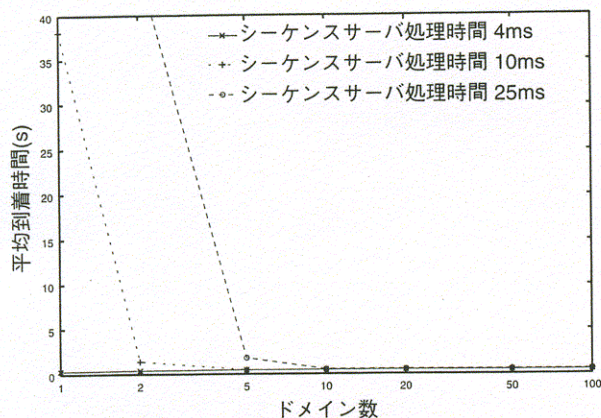


図 11 平均到着時間

Fig. 11 Reach time average.

メインあたりのクライアント数が 20, 程度がサーバの処理の限界であり, 同様に処理時間 25 ms のときはドメイン数 2, つまりドメインあたりのクライアント数が 50 程度が限界であることが分かる (図 11)。

4.3 考 察

ダミーメッセージに関しては, 擬似的に ATOP と同一のシステム構成であったドメイン数 100 の点で莫大な数のダミーメッセージを計測したが, ドメイン数を減らしていくことでダミーメッセージ数は大きく抑制することができている。つまりダミーメッセージの抑制という目的はシミュレーションによって達成が確かめられたことになる。

平均到着時間については, 今回は採取したデータの関係上グラフが粗くなってしまったが, シーケンスサーバの処理能力の違いによってクライアント数の限界がおおよそ読み取れる。また同一の処理能力のグラフにおいてドメイン数 1, すなわちクライアント数 100 のときは非常に遅かった到着時間もドメイン数を増加させ, クライアント数を減らすことですぐに一定のパフォーマンスに戻ることが分かった。つまり本方式のようにドメイン化を導入することでサーバの負荷が軽減できていることが分かる。

この 2 つのグラフを見比べるとドメイン数を大きくしていくとダミーメッセージの数が増え, 逆にドメイン数を減らしていくとサーバへの負荷が大きくなりメッセージの到着時間に大きく影響を及ぼすことが分かる。両方の妥協点をとることがシステムを設計するうえで重要なポイントとなる。

たとえば今回のシミュレーションの結果では, もしサーバの処理時間が 25 ms だったとするとその処理能力の限界はクライアント数では 10 前後であるので 1 ドメインあたりのクライアント数はその程度の数しか参加できないようにするべきである。

5. おわりに

本稿では, 分散アプリケーションにおいて利用するための新しい全順序マルチキャストプロトコルを提案し, その性能について評価した。

従来の全順序マルチキャストプロトコルでは規模拡張性が低い, ネットワーク資源の浪費などの問題があった。そこで本研究では新方式として拡張適応的全順序マルチキャストプロトコル (拡張 ATOP) を提案した。

拡張 ATOP は従来方式であるシーケンスサーバ方式と適応的全順序マルチキャストプロトコル (ATOP) を統合したものである。本方式はシーケンスサーバ方式の主な問題点であったシーケンスサーバへの負荷の集中の軽減と, ATOP の問題点であったダミーメッセージの発生を抑制を目的としている。そこで上記の要求が満たされているか調べるために本方式についてシミュレーションを行い, 評価した。その結果, シーケンスサーバの負荷軽減, ダミーメッセージの抑制に成功したことが証明された。

今後の課題としては, サーバの処理能力とネットワーク構成によるプロトコルの性能への影響をより詳しく調査し最適なパラメータを導き出し, そのうえで他のマルチキャスト方式との比較を行って本方式が規模拡張性の面で優れていることをシミュレーションによって明らかにすることがあげられる。また, 今回はマルチキャストグループメンバの参加と脱退, またネットワーク通信時のパケットロスについては考慮しなかった。今後は以上のような条件を加えてプロトコルの性能評価を行うことが課題である。

参 考 文 献

- 1) Macedonia, M.R., et al.: NPSNET: A Network Software Architecture for Large-Scale Virtual Environments, *Presence*, Vol.3, No.4 (1994).
- 2) Waters, R.C. and Barrus, J.B.: The rise of shared virtual environments, *IEEE Spectrum* (1997).
- 3) Hagsand, O.: Interactive Multi-User VEs in the DIVE System, *IEEE Multimedia Magazine*, Vol.3, No.1 (1996).
- 4) 山田善大, 池谷利明, 峰野博史, 太田 賢, 水野忠則: モーバイル電子会議システム PARCAE におけるサーバの実現, 第 54 回情報処理学会全国大会論文集, pp.3-541-3-542 (1997).
- 5) 山田善大, 池谷利明, 峰野博史, 太田 賢, 水野忠則: モーバイル電子会議システム PARCAE の

提案, 第 54 回情報処理学会全国大会講演論文集, pp.3-543-3-544 (1997).

- 6) Deering, S.: *Host Extensions for IP Multicasting*, Indianapolis, IN: New Riders (1996).
- 7) Kumar, V.: *Mbone: Interactive Multimedia on the Internet*, Indianapolis, IN: New Riders (1996).
- 8) Armstrong, S., Freier, A. and Marzullo, K.: *Multicast Transport Protocol, Request for Comments (RFC) 1301* (1992).
- 9) 南端邦彦, 佐藤文明, 水野忠則: 共有仮想環境のための高信頼マルチキャスト方式の提案と評価, 情報処理学会論文誌, Vol.41, No.2, pp.254-261 (2000).
- 10) Chockler, G.V., Huleihwl, N. and Dolev, D.: *An Adaptive Totally Ordered Multicast Protocol that Tolerates Partitions*, *The 17th ACM Annual Symposium on Principles of Distributed Computing* (1998).

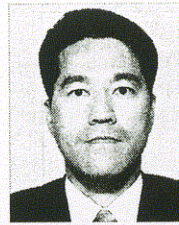
(平成 12 年 5 月 19 日受付)

(平成 12 年 10 月 6 日採録)



細谷 篤

昭和 51 年生. 平成 12 年静岡大学工学部知能情報工学科卒業. 同年静岡大学大学院情報学研究科情報学専攻入学, 現在に至る. IP マルチキャスト, 分散システムに興味を持つ.



佐藤 文明 (正会員)

昭和 37 年生. 昭和 61 年東北大学大学院工学研究科電気及通信工学専攻博士前期課程修了. 同年三菱電機(株)入社. 通信ソフトウェアの研究開発に従事. 平成 7 年 1 月より静岡大学工学部助教授, 現在, 静岡大学情報学部助教授. 工学博士. 通信ソフトウェア, 形式言語, 分散処理システムに関する研究に興味を持つ. 電子情報通信学会, IEEE Computer Society 各会員.



水野 忠則 (正会員)

昭和 20 年生. 昭和 43 年名古屋工業大学経営工学科卒業. 同年三菱電機(株)入社. 平成 5 年静岡大学工学部情報知識工学科教授, 現在, 情報学部情報科学科教授. 工学博士. 情報ネットワーク, プロトコル工学, モバイルコンピューティングに関する研究に従事. 著書としては「プロトコル言語」(カットシステム), 「分散システム入門」(近代科学社)等がある. 電子情報通信学会, IEEE, ACM 各会員.