

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

南 端 邦 彦[†] 佐 藤 文 明^{††} 水 野 忠 則^{††}

共有仮想環境とは、遠隔地に分散した複数の計算機間で仮想的な三次元空間を共有するシステムである。仮想空間内でやりとりされるデータは順序性・信頼性が重要になっている。そのため順序制御されたデータを信頼性を持たせて複数の計算機間で効率的に配送する通信プロトコルの研究・開発が必要となっている。本稿では、インターネット環境で分散空間を共有するシステムの構築を目標とする。MBoneによるIPマルチキャストを利用した分散共有モデルにおいて、順序性・信頼性が重要である共有データを効率的に複数の計算機に配送する手段として、再送管理を分散して行う方式を提案した。また、従来の方式と本方式の比較シミュレーションを行い、これを評価した。本方式では従来の方式よりも応答時間、信頼性、メッセージ数を改善することを目的とする。

A Method of Reliable Multicast for Shared Virtual Environment and Its Evaluation

KUNIHICO MINAMIHATA,[†] FUMIAKI SATO^{††} and TADANORI MIZUNO^{††}

Shared Virtual Environment is the system which shares virtual space among two or more remote computers. As for the data which is exchanged in the virtual space, the message ordering and the reliability become important. The study and the development of the communication protocol which delivers sequential data efficiently with high reliability among several computers are wanted. The goal of this paper is building the system which shares virtual environment in the internet environment. Since it's important to deliver data with the right message ordering and high reliability among several computers efficiently, we propose a method of distributed retransmission management. We compared this method with previous method by computer simulation. The purpose of this method is to improve response time, reliability, and number of messages compared with previous methods.

1. はじめに

共有仮想環境とは、遠隔地に分散した複数の計算機間で仮想的な三次元空間を共有するシステムである。仮想環境に存在するユーザは、それぞれ仮想環境の複製を所有しそれら複製の一貫性管理を行うことで空間の共有を実現する。

近年、多数のユーザが仮想環境を共有するための通信媒体としてインターネットが利用され、またインターネット環境において共有ユーザに対して効率的にデータを配送する手段として、インターネットマルチキャストが注目されるようになった。マルチキャストはコネクションレス配送であるため、データの信頼性は保証されない。データに信頼性を持たせるためにMBone

(virtual Multicast Backbone On the interNET)¹⁾などを用いて高信頼マルチキャストを行うプロトコルは多数、提案・研究されている^{2)~4)}。仮想環境の更新データは、信頼性、順序性が保証されている必要があり、インターネット内に散在する共有ユーザから頻繁に発生する。従来の高信頼マルチキャストでそのようなデータを扱うとさまざまな問題が生じる。更新データを効率良く配送し、仮想環境を共有するユーザがイベントを正しい順序でロスなく実行できるようなプロトコルが必要であると考えられる。

本稿ではまず従来の共有仮想環境モデルを紹介し、本研究での共有仮想環境の基盤システムについて説明する。次に従来の高信頼マルチキャストを紹介する。そして仮想環境内で送受信されるマルチキャストデータの信頼性を分散して管理する方式を提案し詳細について述べる。また、従来の方式と本方式において比較シミュレーションを行い、結果について考察する。

[†] 静岡大学大学院理工学研究科
Graduate School of Science and Technology, Shizuoka University

^{††} 静岡大学情報学部
Faculty of Information, Shizuoka University

2. 共有仮想環境

2.1 従来の研究

2.1.1 SPLINE

SPLINE⁵⁾は三菱電気で研究開発されている分散仮想環境基盤ソフトウェアである。リアルタイムで三次元グラフィックスと音声を扱うことができ、環境内にいるユーザ同士で音声を使った対話が可能となっている。SPLINEは、完全分散管理をとっており、スケーラビリティにおいて優れている。そして、ネットワークとアプリケーションレベルでオープンなインタフェースを提供しており拡張性にも優れている。

2.1.2 DIVE

DIVE (Distributed Interactive Virtual Environment)⁶⁾はサーバレス方式のシステムであり、仮想環境に参加するユーザはそれぞれ環境の複製を所持し、それらの一貫性を保つことで仮想環境を共有する。DIVEでは一貫性保証プロトコルと状態転送プロトコルが用いられ、その上に三次元オブジェクトに対するさまざまな操作を実装している。一貫性保証プロトコルにより仮想環境の厳密な一貫性管理を行う。また状態転送プロトコルによって途中参加のユーザに現在の仮想環境の状態を転送することができる。このシステムは当初 LAN と高性能なワークステーション上で稼働することを前提としていたため、インターネットのように帯域が狭く遅延の大きいネットワークではスケーラビリティに問題があり、最大参加者は 10 人で限界である。

2.1.3 Community Place

Community Place⁷⁾は SONY で研究開発されている共有仮想環境システムである。Community Place は共有仮想社会 (Virtual Society) をインターネット上に構築するためのソフトウェア群であり、VRML ブラウザである CP ブラウザ、サーバとなる CP ビューロ、そして仮想空間内のオブジェクトとそれに関連付けられたアプリケーションである CP アプリケーションオブジェクトの 3 つからなる。Community Place では厳密な一貫性保証は行わずに高いスケーラビリティを実現している。

2.2 共有仮想環境モデル

インターネット環境での共有仮想環境のモデルとして次のようなシステム構成を提案する (図 1)。我々は上記にあげた方式の中間のアプローチをとっている。厳密な一貫性保証を行いながらスケーラビリティについても改善する。

インターネット上の MBone に共有ユーザ、VRML

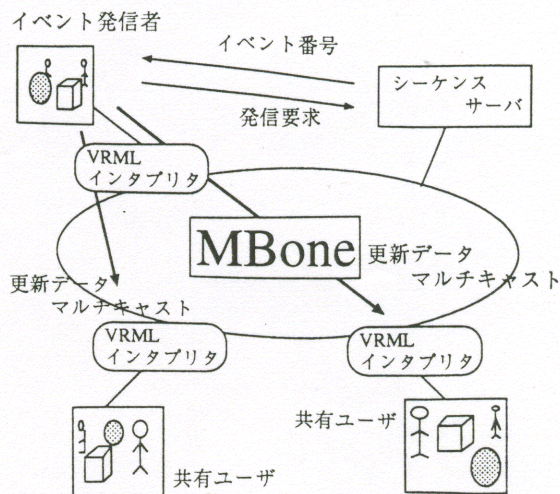


図 1 仮想空間の共有

Fig. 1 Sharing virtual environment.

サーバ、シーケンスサーバを配置する。

共有ユーザはそれぞれ仮想環境の複製を保持しており、すべてのユーザにおいて同様に更新される。

また VRML サーバは仮想環境の初期状態を持ち、シーケンスサーバはイベント発行権の要求に対して一連の順序番号を返す。

2.3 仮想環境の共有

仮想環境を共有するすべてのユーザは VRML⁸⁾で記述された初期状態を VRML サーバから受信し、受信後、相互通信を開始する。

仮想環境はユーザがオブジェクトの作成・移動・削除などのイベントを発生させた際に更新される。イベント実行者は空間の差分データを他のすべてのユーザに送信し、受け取ったユーザはそれを実行し仮想環境を更新する。

しかし、複数のユーザからの更新データを同時に受け取った際にそれらを実行する順序を間違えるとそれぞれの仮想環境が異なったものになってしまう恐れがある。そのためイベント発生者は、発生するイベントの順序性を保証するためにシーケンスサーバを用い、シーケンスサーバから受け取った番号を更新データに付加して送信する。そして受信者は順序制御されたイベントを正しい順序で実行する。

このように仮想環境の更新がすべてのユーザにおいて同様に行われ、仮想環境が複数のユーザによって共有される。

2.4 共有仮想環境の特徴

仮想環境を共有する際に使用するデータはサイズも小さく、インターネットに分散して存在する多数の共有ユーザから頻繁に発生する。またデータは順序制御されており、スムーズに空間を更新していくためには、

ユーザはできる限り正しい順序でパケットを受け取らなくてはならない。マルチキャストはUDPを利用するコネクションレス配送なので信頼性が保証されない。データの信頼性を保証するプロトコルが必要である。

3. 高信頼マルチキャスト

3.1 従来の方式

研究段階のものも含めると、十数種以上ある。プロトコルによってリアルタイム性、信頼性のレベル、適用ノード数、対象ネットワークの特性などさまざまであるが、代表的なりライアブル・マルチキャストのなかでリアルタイム性の高いプロトコルの概要を以下に3つ示す。

3.1.1 RMP

RMP (Reliable Multicast Protocol)²⁾は Ack と Nack を組み合わせて用いることで高い信頼性を保証する (図 2)。ネットワーク内で仮想的なトークンリングを形成し、トークンを回しながらパケットロスの検出、負荷バランスの均等化、バッファクリアのタイミングの把握を行う。RMPでは複数の送信者が出したマルチキャストパケットを受信側で正しい順番で受け取ることができる。トークンサイトはイベントの順序決定権を持っている。トークンサイトに到着したイベントはタイムスタンプが割り振られ、その内容を Ack としてマルチキャストする。Ack を受け取ったサイトはその順序でイベントを実行し、タイムスタンプ上のイベントが欠落している場合は Nack を上流のトークンサイトに出す。

3.1.2 RMTP

RMTP (Reliable Multicast Transport Protocol)³⁾では Ack と Nack を用いて高い信頼性を保証している (図 3)。リージョンと呼ばれる互助単位を基にしたツリー構造を構築する。各リージョンには DR (Designated Reciever) と呼ばれるリージョン・マネージャを配置する。DR はリージョンのメンバであり、1 レベル上のリージョンのメンバでもある。送信者は通常どおりメッセージをマルチキャストする。メンバは受信結果に応じて Ack または Nack を DR に返す。すべての子からの Ack を受け取るとデータを消去し DR に Ack を出す。Nack が来た場合、データが存在すれば再送を行い、自分が保持しないメッセージについてはさらにその親リージョンに Nack を発行し該当データを手に入れる。

3.1.3 SRM

SRM (Scalable Reliable Multicast)⁴⁾は雲型のアーキテクチャをとるプロトコルである (図 4)。ベスト・

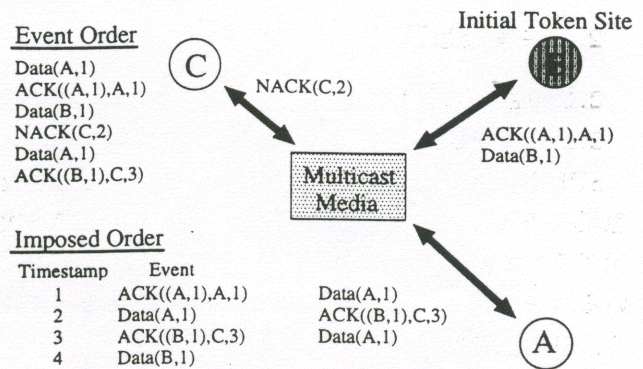


図 2 Reliable Multicast Protocol
Fig. 2 Reliable Multicast Protocol.

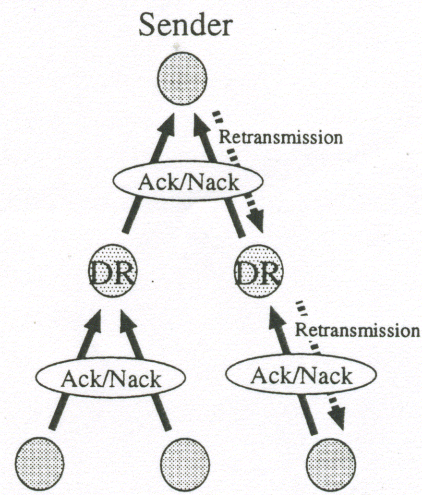


図 3 Reliable Multicast Transport Protocol
Fig. 3 Reliable Multicast Transport Protocol.

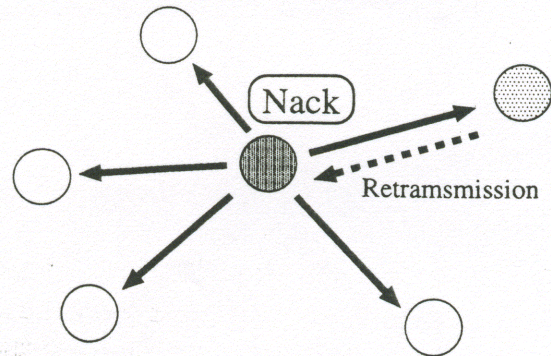


図 4 Scalable Reliable Multicast
Fig. 4 Scalable Reliable Multicast.

エフォート型の通信をベースとする。適用ノード数を損なわない範囲で、信頼性を確保する。Ack は使わず Nack のみ用いる。受信者はメッセージが届かないと、Nack をグループにマルチキャストする。Nack を受け取り、メッセージを正しく受け取れた近所の受信者がマルチキャストする。上流でパケットロスが起こった際に受け取れなかった膨大な数のサイトから Nack が発生するのは好ましくない。SRM はタイマを用い、パケットロスの起こった付近のサイトからのみ Nack

表1 高信頼マルチキャストの比較
Table 1 Reliable multicast protocols.

名称	RMP	RMTP	SRM	本方式
開発元	NASA	Bell la.	S.Floyd	著者
トポロジ	TR	木	雲	木
ノード数	少	中	大	中
グループ	多対多	少対多	少対多	多対多
信頼性	高	高	中	高
Ack	MC	DR	×	MAR
Nack	MC	DR	MC	MAR
再送元	TS	DR	付近	MAM

DR: Designated Receiver, TS: Token Site,
TR: Token Ring

が発生するよう工夫している。

3.2 高信頼マルチキャストの比較

前述した方式と今回提案した方式の比較表を示す(表1)。

本方式についての詳細は次章で説明する。

3.3 問題点

これらのプロトコルを仮想環境の共有に適用するとさまざまな問題が生じる。RMPはトークンの持ち回りでパケットロスの検出を行っているため、通信遅延などにより適用ノード数は少なくなってしまう。また、AckおよびNackはマルチキャストされるため、ネットワークにメッセージが氾濫してしまう。このような理由から、インターネットのような環境で適用するのは難しい。RMTPはバルクデータ転送に適したプロトコルであり、送信データに対するツリーの構成に手間がかかる。頻繁に起こるデータそれぞれについてツリー構築を行うのは効率的でない。SRMについては適用ノード数は多いがAckを用いないためバッファ管理ができず、完全な信頼性は保証されない。仮想環境ではすべての更新データについて信頼性が保証されないと、一貫性が保証されない。

多数のサイトに対して信頼性を持たせて効率的にデータ配送を行う通信プロトコルが必要である。

4. 提案方式

4.1 設計方針

高信頼マルチキャストプロトコルは、研究段階のものも含めると十数種以上ある。プロトコルによってリアルタイム性、信頼性のレベル、適用ノード数、対象ネットワークの特性など対象はさまざまである。

仮想空間を共有する際に使用するデータはサイズも小さく、インターネットに分散して存在する多数の共有ユーザから頻繁に発生する。またデータは順序制御されており、スムーズに空間を更新していくためには、ユーザはできる限り正しい順序でパケットを受け取ら

なくてはならない。

これらの問題点から、仮想環境における高信頼マルチキャスト方式を以下のような目標に沿って設計していく。

- TCP/IP 程度の信頼性

パケットロスが起こると空間が正常に更新されない。そのため再送管理を行うことで信頼性を保証する。

- 送信データの効率的な再送

イベントは順序制御され、順番どおりに実行される。再送が遅れると再送パケットが到着するまでイベントが実行できなくなるため、素早い再送が必要である。

- データ送信者の負荷の分散

イベント発生がかたよるとAckの管理や再送用バッファ管理などの負荷がかかってしまう。

- Ack/Nack Implosionの軽減

ネットワークにAck/Nackが氾濫すると通信路が飽和状態になり、多くのメッセージが失われてしまう。

これらの目標をふまえ、本稿では分散再送管理方式を提案する。

4.2 分散再送管理

本方式は分散再送管理を行うプロトコルである。分散再送管理とはデータの再送管理を分散して行う方式である。

再送管理を分散させることは次のような利点がある。再送管理を分散化することで送信者にかかる負荷を軽減する。また再送バッファを分散させることにより送信者のバッファ量も減少する。そして送信者へのAck、Nackメッセージの集中を回避することができる。

本方式では仮想環境に参加しているサイトを互助リージョンと呼ばれる単位で区切り、その中で再送管理を行う。

4.3 互助リージョン

ネットワーク形態としてツリートポロジをとる。それぞれのサイトはネットワーク上で近接したサイトとともに互助リージョンMAR (Mutual Aid Region)を構成する。リージョンを構成するサイトは管理サイトと呼び、その他のサイトは管理サイトが構成するリージョンの互助メンバMAM (Mutual Aid Member)となる。互助メンバは管理サイトからTTLを絞ったパケットを送信し検出する。またMARはパケットロスが起きたとき再送が途中で途切れないように、それぞれ不足なく重なり合っている必要がある。

R3の管理サイトであるサイト3はサイト1、サイ

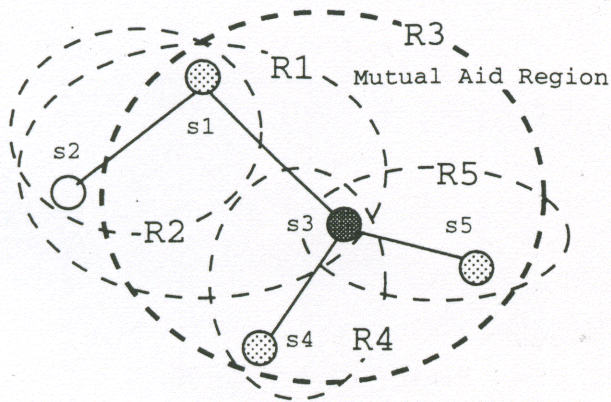


図5 互助リージョン
Fig. 5 Mutual Aid Region.

ト 4, サイト 5 を互助メンバとして持ち, 管理サイト 4, 管理サイト 5 もまた互助メンバとしてサイト 3 を持つ (図 5).

4.4 再送アルゴリズム

すべてのサイトは再送のためのバッファを持ち再送に備える. 到着したイベントは実行済みであるか実行済みでないかにかかわらずバッファに格納される. しかしバッファには限りがあるのでバッファをタイミング良く消去しなければならない. そのためすべてのサイトはバッファ管理を行うため, MAM の情報と自サイトを含めた各々がどれだけイベントを実行できているかの情報を保持するテーブルを持つ (図 6).

本方式は Ack/Nack メッセージとともに用い, 高い信頼性を保証する.

そして, Ack/Nack とともに MAM にのみ出され, それらを受け取ったサイトがバッファ管理・再送管理を行う.

4.4.1 バッファ管理

再送用バッファ管理は Ack メッセージのやりとりによって行われる (図 7). Ack には自サイトの情報と現在実行済みのイベント番号 (連番で到着している番号) の情報が含まれている.

Ack は一定間隔で MAM に出される. Ack の発生間隔はイベントの発生間隔と同様が望ましい. しかしイベントはすべてのサイトから不規則に出されるためそれを予測するのは難しい. そこで到着履歴から推測するか, または MAM の再送バッファサイズにより適当な値を出す. Ack を受け取ったサイトはテーブル内の Ack を出したサイトの到着済イベント番号を更新する. そしてすべての MAM についてのテーブルの到着済イベント番号を参照し, 最小のイベント番号までの再送バッファを消去する.

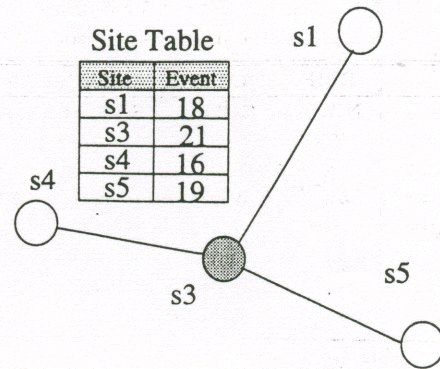


図6 サイトテーブル
Fig. 6 Site table.

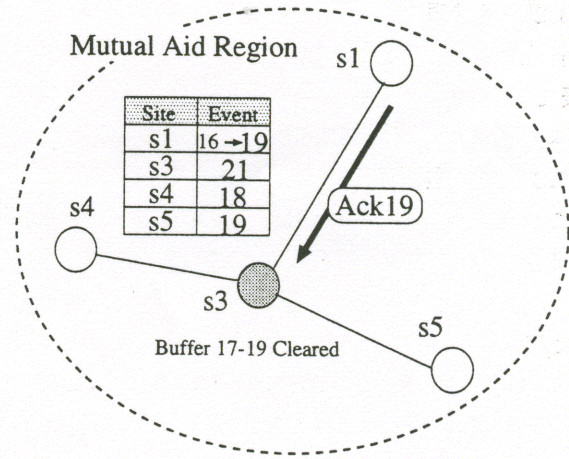


図7 バッファ管理
Fig. 7 Event buffer management.

4.4.2 再送管理

イベントの再送管理は Nack メッセージのやりとりによって行われる (図 8). Nack には再送要求のイベント番号の情報が含まれている.

Nack はパケットロスが検出されたときに MAM に出される. ロスの検出のためにタイムカウンタを用いる. タイマはイベントが順不同で到着した場合と最新のイベントが到着した場合にセットされる. イベントが順不同で到着した場合は抜け落ちたパケットがすべて到着すればタイマをクリアする. それまでのすべてのイベントが到着している場合は次のイベントが到着すればタイマをクリアする. タイムアウトが起こると必要なイベント番号を Nack に乗せ, MAM にマルチキャストする. タイマの間隔はできるだけ短い方がよいが, ネットワークを混雑させないように調整する必要がある.

Nack を受け取ったサイトは再送バッファよりその番号のイベントを探す. 合致したイベントが再送バッファに存在すれば MAM に対して再送イベントをマルチキャストする. 存在しなければ何もしない. こ

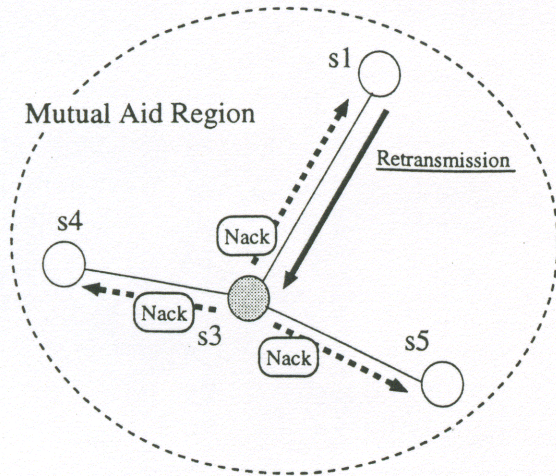


図8 再送管理

Fig. 8 Retransmission management.

では MAR 外にパケットが伝播しないように TTL を絞ってマルチキャストする。ユニキャストを行わないのは複数の MAM にイベントが届いていない場合に再送が効率良く行えるためである。

5. シミュレーション

共有仮想環境においてはユーザのスムーズなインタラクションを行うために、再送を含めたイベントの迅速な配送が望まれる。そこで、本方式と RMP (Reliable Multicast Protocol)²⁾について比較シミュレーションを行った。平均到着時間、再送バッファサイズ、実行待ちバッファサイズについて測定した。平均到着時間はイベントが発生してから再送も含めてすべてのサイトに到着した時間の平均をとったものである。再送バッファサイズはすべての再送バッファの平均をとった。実行待ちバッファは到着はしているが実行されていないイベントを格納するバッファである。これについてもすべてのサイトの平均をとった。

シミュレーション条件は、イベント発生率を 1/サイト数、通信遅延を 10ms/1 リンク、パケットロス率を 5%・20% に設定した。結果を図 9, 10, 11 に示す。

グラフ 1 から、同じパケットロス率において本方式のほうのイベント到着時間が短いことが分かる。本方式は定期的に互助メンバのイベント到着状況や自サイトのログのチェックを行うためパケットロスの素早い検出が可能となっている。再送も含めてイベントが発生後、素早くサイトに到着しており仮想空間がスムーズに更新できるということが分かる。

また、本方式はパケットロス率を大きくした場合においても性能があまり低下していない。RMP ではパケットのロス検出を Ack のマルチキャストによって行っている。そのため Ack が頻繁に落ちるような環境

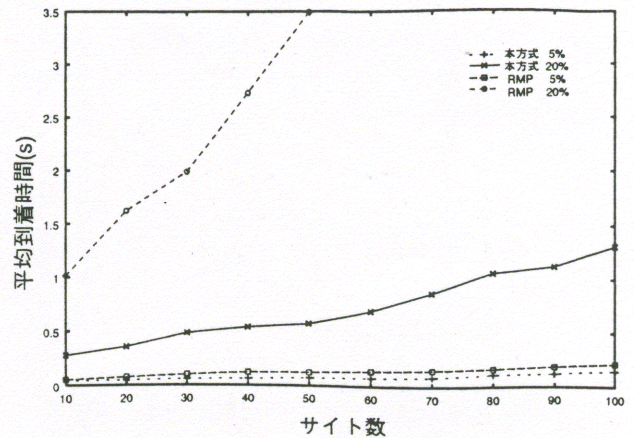


図9 平均到着時間

Fig. 9 Reach time average.

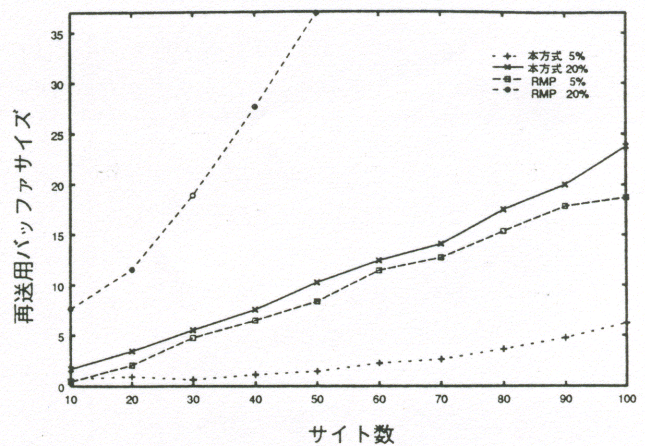


図10 再送用バッファサイズ

Fig. 10 Event buffer size for retransmission.

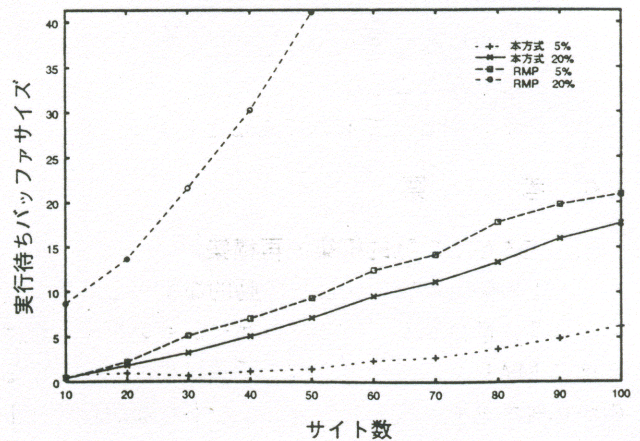


図11 実行待ちバッファサイズ

Fig. 11 Event buffer size waiting for execution.

では性能が極端に低下する。しかし、本方式では MAR 単位の再送管理を行っており、Ack/Nack とともに隣接サイトにしか出されないためメッセージのロスが起こりにくい。このことから、本方式が遅延変動の大きいインターネット環境において有効であるといえる。

図 10 より、本方式のほうが RMP よりも再送用バッ

ファが少なく済むことが分かる。再送用バッファが適当なタイミングでクリアされないとバッファ溢れが起こり、要求されたイベントが再送できずに信頼性が損なわれてしまう。本方式では MAM に対する Ack を頻繁に発生させることにより、MAM の再送用バッファクリアを促している。これにより素早いタイミングでのバッファクリアが可能となっている。それに対して RMP はトークンの持ち回りでバッファクリアのタイミングを測っているためサイト数が増加するとそれだけバッファが溜まってしまふ。これにより本方式はスケラビリティにおいても優れていることが分かる。

図 11 においても本方式において実行待ちバッファサイズについて改善が見られた。実行待ちバッファサイズが小さいということはイベントが溜まることなくスムーズに実行できているということである。RMP ではイベントの順序決定がトークンを持っているサイトで行われる。その他のサイトはトークンサイトからの Ack が到着しないことにはイベントの順序が決定できない。そのためすでに到着しているイベントが実行できずに溜まってしまふ。本方式では順序制御はシーケンスサーバで行われるため、順序番号が付加されたイベントは正しい順序であれば到着後すぐさま実行される。また本方式ではイベントが順不同で到着した場合、タイマにより Nack を発生させる。このように素早い迅速な再送が可能となっており、仮想環境が止まることなくスムーズに更新される。

平均到着時間、再送用バッファサイズ、実行待ちバッファサイズのすべてにおいて本方式において改善が見られた。これらのことから、本方式が共有仮想環境の通信プロトコルとして有効であると考えられる。

6. 考 察

6.1 MAR の動的構築・再構築

MAR の構築する手法として動的なものや静的なものがある。静的構築ではアプリケーションを立ち上げる前に MAR を設定する。この手法は構成を管理するのが単純であり、実装しやすい。しかしながらサイトに障害が起こったり通信に障害が起こったりすると、アプリケーションが停止してしまふ。一方、動的構築の場合では必要に応じて MAR を構築・再構築する。この手法は複雑であり、実装するのは困難であるが、アプリケーションを停止させることなくシステムを拡張し再構築することが可能である。我々は以下に MAR の動的構築法を述べる。

6.1.1 MAR の構成と MAR への参加

共有仮想環境ではある 1 つの IP マルチキャストア

ドレスを用いることを提唱している。まず最初に、状態の仲介をする VRML サーバとシーケンスサーバを立ち上げる。今もしあるサイトがクライアントアプリケーションを立ち上げたとする。アプリケーションは自分が仮想環境のどこに位置するかを問い合わせるためにマルチキャストを行う。一致するサーバと管理サイトはアプリケーションに返答する。

仮想環境に参加しようとしているサイトはこれらのサーバから情報を受け取り、状態の同期をとる。同時に、サイトは参加要求を最も迅速に対応可能である管理サイトに送信する。参加要求メッセージはサイトが受け取った最新のシーケンス番号を含んでいる。参加要求を受け取った管理サイトはそのシーケンス番号をみて再送バッファをチェックする。もしそのシーケンス番号が再送バッファの最小のシーケンス番号よりも小さい場合、その最小の番号はサイトに返され、バッファのクリアはしばらくの間抑えられる。もしサイトが管理サイトの最小番号まで受け取れば、番号を付加したメッセージを待ち、管理サイトに再び参加要求を出す。

管理サイトが受理できる参加要求を受け取ると、サイトテーブルに新しいエントリが作られ、そこにシーケンス番号が書かれる。それからその受理はサイトに伝えられる。その後、再送バッファはサイトテーブルに従ってクリアされる。

6.1.2 MAR の変更

サイトが仮想環境の別の場所に移動すると、古い場所の情報は意味がなくなる。そして現在いる場所の新しい情報が要求される。これを MAR 間の移動という。仮想環境の新しい場所の情報は前述の手続きで得ることができる。MAR から脱退するためにサイトは管理サイトに単純に脱退要求を送信する。管理サイトはサイトテーブルから一致するエントリを削除し、サイトに Ack を返す。

もしサイトが仮想環境の同じ場所で MAR 間を移動する場合（たとえば現在の管理サイトに障害が起こった場合に起こる）も、基本的には上記の手続きを用いることができる。

また自分が MAR の管理者であり MEM の管理を行っている場合、その MEM の管理を他の管理サイトに委譲しなければならない。

6.2 ネットワーク障害、分断

いくつかの通信路でネットワーク障害が起こった場合、管理サイトの再送バッファはオーバフローを起こすかもしれない。なぜなら管理サイトは他のサイトから Ack を受け取れないからである。そのため管理サイ

トは長い間管理サイトに Ack を送信していないサイトを削除する。もしネットワークの分断が起こった場合、シーケンスサーバのある側が正しい区画となる。この問題解決策として、シーケンス番号を生成する機能を有する区画を正しい区画としたり、両方正しいとし並行に実行したりするような手法が考えられる。

7. おわりに

本稿ではインターネット上で仮想環境を共有するためのモデルを想定し、共有仮想環境における通信手段として、IP マルチキャストに基づいた分散再送管理方式を提案した。また、従来の方式と本方式をシミュレーションで比較した結果、再送効率や再送用バッファサイズまた実行待ちバッファサイズについて改善がみられた。これにより本方式が共有仮想環境におけるマルチキャストプロトコルとして有効であることが証明された。

今後の課題として、ツリーの自動構築や最適な応答メッセージの発生タイミングの検討があげられる。また近年携帯端末が普及し、不安定な条件で接続するユーザが増えることが予想される。このように接続が不安定なシステムに再送を任せることはできない。そのため不安定なサイトは読み取り専用サイトとしてリージョンの管理サイトに接続し、障害が起こった際にイベントが消滅しないようにしなければならない。また接続からの復旧は管理サイトの最新スナップショットを受信することで対応する。すべてのユーザが自由に共有仮想環境に出入りできるようなシステムを作成しなければならない。

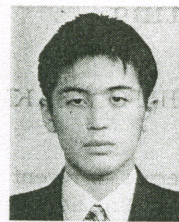
参考文献

- 1) Kumer, V. (著), 楠本博之 (監訳): インターネットマルチキャスト MBone, インプレス (1996).
- 2) Montgomery, T., Callahan, J.R. and Whetten, B.: *Fault Recovery in the Reliable Multicast Protocol*, NASA/West Virginia University Software IV&V Facility (1995).
- 3) Armstrong, S., Freier, A. and Marzullo, K.: *Multicast Transport Protocol*, RFC 1301 (1992).
- 4) Floyd, S., Jacobson, V. and McCanne, S.: *A Reliable Multicast Framework for lightweight Sessions and Application Level Fraing*, *IEEE/ACM Trans. Networking* (1995).

- 5) Waters, R.C. and Barrus, J.B.: *The rise of shared virtual environments*, *IEEE Spectrum*, pp.18-25 (1997).
- 6) Hagsand, O.: *Interactive Multiuser VEs in the DIVE System*, *IEEE Multimedia Magazine*, Vol.3, No.1 (1996).
- 7) Honda, Y., et al.: *Extending WWW to Support Multiuser Interactive 3D Environment*, *ACM VRML '95* (1995).
- 8) Hartman, J. and Wernecke, J.: *The VRML2.0 Handbook*, Addison-Wesley Developers Press (1996).

(平成 11 年 4 月 30 日受付)

(平成 11 年 12 月 2 日採録)



南端 邦彦 (学生会員)

昭和 50 年生。平成 10 年静岡大学工学部情報知識工学科卒業。同年静岡大学大学院理工学研究科計算機工学専攻入学、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。