

共有仮想環境のための高信頼マルチキャスト方式

南端 邦彦、佐藤 文明、水野 忠則
静岡大学

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。順序制御されたデータを信頼性をも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なう方式を提案した。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している。

キーワード：共有仮想環境、高信頼マルチキャスト

A Reliable Multicast for Shared Virtual Environment

Kunihiko Minamihata, Fumiaki Sato, Tadanori Mizuno
Shizuoka University

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. The purpose of this method is to improve response time, reliability, and number of messages compared with previous methods.

keywords: shared virtual environment, reliable multicast

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な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の利用が目されるようになった。MBone (virtual Mul-

ticast Backbone On the interNEt)[1]などを用いて高信頼マルチキャストを行なうプロトコルは多数、提案・研究されているが、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。本篇では以下に本研究での基盤システムについて説明する。そして、従来の方式を紹介し、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・検討を行なう。

2 共有仮想環境のシステムモデル

2.1 システム構成

インターネット環境での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)。

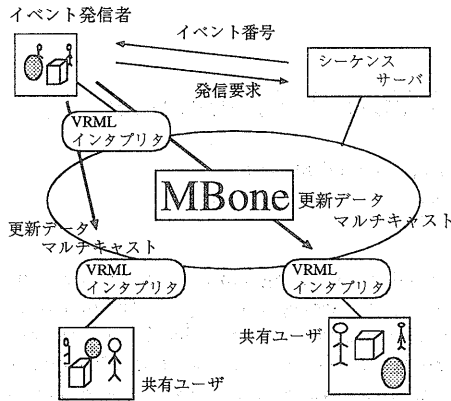


図1: 仮想空間の共有

仮想環境を共有するすべてのユーザは初期状態をWWWサーバからVRML[2]ファイルとして受信し、受信後、相互通信を開始する。

仮想環境は、イベント実行者が空間の差分データを他のすべてのユーザに送信し正しい順序で実行されることにより更新する。

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、更新データに順序制御されたイベント番号を付加して送信する。

このように、仮想空間が複数のユーザによって共有される。

2.2 Mbone

MBone(virtual Multicast Backbone On the interNEt)はインターネット上に張り巡らされたマルチキャスト通信網である。リアルタイムのインタラクティブなマルチメディアをインターネット上で配信および、アクセスできるようにする技術である。

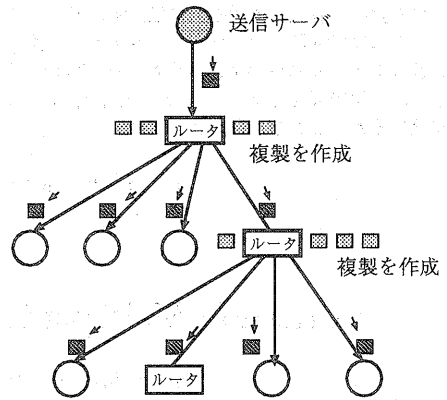


図2: マルチキャスト(Mbone)

MBone(図2)では、多数の特定の端末に対しデータ配信する際、サーバにかかる負荷の軽減とグループに到達するまでのトラフィックの改善がなされている。

従来のマルチキャストのようにサーバで情報パケットの複製を作成せず、ただ1つのパケットをグループに対して送信する。マルチキャスト対応ルータが複製を作成し、グループに属するクライアントに配信する。

IPマルチキャストでの配送は、オーバーヘッドの少ないUDPでコネクションレスで行なわれる。

2.2.1 共有仮想環境の特徴

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、ユーザは出来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。

3 高信頼マルチキャスト

3.1 従来 방식

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、代表的なリアルタイム・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)[3]はAckとNackを組み合わせてもちいることで高い信頼性を保証する。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、バッファクリアのタイミングの把握を行なう。

3.1.2 RMTP

RMTP (Reliable Multicast Transport Protocol)[5]ではAckとNackを用いて高い信頼性を保証している。リージョンと呼ばれる互助単位を元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ーであり、1レベル上のリージョンのメンバーでもある。メンバーは受信結果に応じてAckまたはNackをDRに返す。DRは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を発行する。

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。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のみもちいる。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。

3.2 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なっているため、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な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出来ず、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もたせて効率的にデータ配送をおこなう通信プロトコルが必要である。

3.3 提案方式

3.3.1 設計方針

前述した問題点から、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な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、順番通りに実行される。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。

3.3.2 ネットワーク構成

ネットワーク形態としてツリートポロジータを取り、それぞれのサイトは自サイトと物理的に隣接したサイトで互助リージョン MAR (Mutual Aid Region) を構成する。またそのサイトは隣接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。

サイト 2 はサイト 1、サイト 4、サイト 5 を互助メンバ (R2) としてもち、サイト 4、サイト 5 もまた互助メンバ (R4, R5) としてサイト 2 をもつ (図 3)。

Ack/Nack とともにそれら互助メンバにのみ出され、それらを受け取ったサイトがバッファ管理・再送管理をする。

サイトは互助メンバがどこまでイベントを受け取れたかの情報をテーブル (図 4) に保持し、隣接サイトからの Ack によりバッファをタイミング良く消去する。

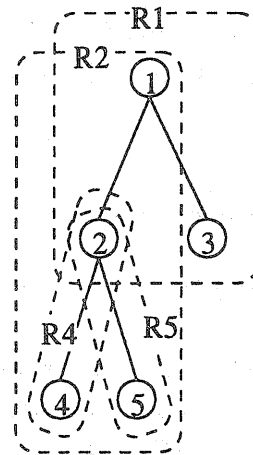


図 3: 互助メンバの構成

3.3.3 メッセージ形式

本方式では Ack/Nack 共に用い、高い信頼性を保証する。

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) をのせ、メンバに対してマルチキャストする。

Nack は、再送要求のイベント番号をのせ、メンバにマルチキャストする。

Nack を受け取ったサイトは、Nack を出したサイト以降にマルチキャストする。

3.3.4 再送バッファ管理

すべてのサイトは再送用バッファをもち再送に備える。Ack を受け取ったサイトは、テーブルを更新する。テーブル内の最小番号までが消去可能となり、テーブルを更新するごとにバッファをクリアしていく。

3.3.5 応答メッセージの発生タイミング

マルチキャストはコネクションレスのデータ配送であるため、信頼性は保証されない。そのため、イベント受信者はどこかでイベントが落ちたのか、また、そのイベントはまだ発生して

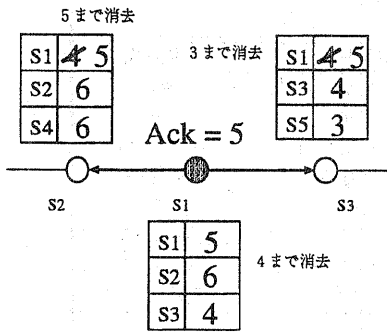


図 4: イベント到着状況テーブル

いないのか判断することが出来ない。また、イベントは頻繁に発生するのですべてのイベントについての Ack を返すのは Implosion を引き起こす可能性がある。

そのため Ack/Nack を発生させるタイミングを考える必要がある。Ack は通進路を占有しない程度で互助メンバのバッファをできるかぎりクリアできるような間隔で発生させる。Nack の発生タイミングについては以下に示す。

- 次に到着する予定のイベントについての Nack

イベント n が到着すると `follow_timer` (およびそのイベント到着間隔) をセット・スタート。到着すればリセット・スタート。タイムアウトでイベント $n+1$ についての Nack を発生させる。タイマーの値を 2 倍してセット・スタート。これを繰り返す。なにかイベントが到着した時点でタイマーを元の値に戻す。

- 抜け落ちたイベントについての Nack

イベント n が到着し、続いてイベント m ($n+2$ 以降) が到着した時点で、`back_timer` (最大通信遅延) をセット・スタート。抜け落ちたパケットすべてが到着すればリセット。タイムアウトで抜け落ちたパケットすべてについての Nack を発生。イベント $m+1$ が到

着しても継続。イベント $m+2$ 以降が到着した場合、別の同じタイマーをセット・スタート。

3.3.6 考察

パケットロスの検出、再送手続き、バッファクリアを互助リージョン単位で行なうため、応答メッセージが他のリージョンに流れず、メッセージの集中が回避できる。

また、Ack と Nack を効率的に用いることで、再送要求に対する迅速な対応や必要な再送用パケットの保持、不必要な再送用パケットの破棄が可能になっている。

しかし、互助メンバに到着しなかったパケット以降のパケットも再送の可能性があるととして、保持してしまう。個々のパケットについての Ack の管理も必要ではないかと考えられる。

4 他のマルチキャスト方式との比較

先に述べた高信頼マルチキャストと本方式において、トポロジ、適用ノード数、メッセージの種類、再送元などについて比較を表にまとめた (表 1)。

RMP と本方式は多対多通信であり、DBMS (DataBase Management System) の複製などに適している。残りの 2 つは少対多通信であり、RMTP はバルクデータ転送、SRM はビデオデータの配送などに適している。

本方式は適用ノード数は中程度である。Ack/Nack とともに互助リージョンにのみ出される。

本方式は、SRM と比べると適用ノード数は少ないが、信頼性は高くなっている。また、すべての方式に比べて、本方式はメッセージの流出を最小限にとどめ、かつ再送を出来る限り近いサイトから行なっている。

表 1: 高信頼マルチキャストの比較

名称	RMP	RMTP	SRM	本方式
開発元	NASA 他	Bell 研究所	Sally Floyd 氏 他	著者
トポロジ	トークンリング	木	雲	木
ノード数	少 (~ 100)	中 (~ 1000)	大 (~ 数千)	中 (~ 1000)
グループ	多対多	少対多	少対多	多対多
信頼性	高	高	中	高
Ack	○マルチキャスト	○DR	×	○互助リージョン
Nack	○マルチキャスト	○DR	○マルチキャスト	○互助リージョン
再送元	トークンサイト	DR	付近	互助メンバ

DR: Designated Receiver

5 まとめ

共有仮想環境において、仮想環境を共有するすべてのユーザは、サーバであり、クライアントである。すべてのユーザは同様にネットワーク内でデータを送受信することが可能でなくてはならない。そのため通信は1対nではなく、n対nを目標とすべきである。そして、インターネットのような、通信遅延が大きく信頼性の低いネットワークにおいて、順序性と信頼性を保証するような通信プロトコルの必要性から本方式を提案した。

本方式は迅速な再送を第一に考え、そのためにすべてのサイトは最大限のログを残しておく。万一パケットロスが起こっても一番近いサイトからの再送を期待できる。分散仮想環境を効率的に更新する(ユーザ間で滑らかなインタラクションを行なう)という面で効果が発揮できると思われる。

ネットワーク障害などによりパケットロスが頻繁に起こった場合、バッファサイズが大きくなってしまふといった問題が生じる。その対策として障害時の互助リージョンの再構成などを今後考えなくてはならない。

参考文献

- [1] Vinay Kumer 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年12月

- [2] 山本 精一著, VRML2.0 パーフェクトガイド, 技術評論社, 1996年11月
- [3] Todd Montgomery, John R. Callahan, Brian Whetten, Fault Recovery in the Reliable Multicast Protocol, NASA/West Virginia University Software IV&V Facility, Nov 3, 1995
- [4] Sally Floyd, Van Jacobson, Steven McCanne, A Reliable Multicast Framework for light-weight Sessions and Application Level Fraing, IEEE/ACM Transactions on Networking, Nov., 1995
- [5] Armstrong, S., Freier, A., and Marzullo, K., Multicast Transport Protocol, Request for Comments(RFC)1301, Feb., 1992