

# 高信頼マルチキャストにおける再送木構築方式の提案

森田 悟史<sup>†</sup> 古川 雄宣<sup>††</sup>  
山田 和宏<sup>††</sup> 佐藤 文明<sup>††</sup>

近年、コンピュータネットワークの高速化により分散アプリケーションの使用が一般的になってきた。分散アプリケーションの開発において、高信頼マルチキャストは効果的かつ必要不可欠なインフラとなっている。様々なアプリケーションや規模に応じてすでに多くの高信頼マルチキャストが開発されている。高信頼マルチキャストにおける重要な課題は、効率的なパケットの再送処理である。本稿では木構造を再送に利用する高信頼マルチキャスト方式に注目する。この方式では、再送の負荷を分散させるため均等かつ、小さくまとまった木を構築することを目標としている。そして、この木の規模と負荷の分散がどの程度なされているかを調べるために、木構造への参加脱退処理といった構築処理のシミュレーションを行った。木を構築する別の方式でのシミュレーションを行い、提案方式と比較した結果、木の規模が小さく再送の負担が均等に分散する木構造を構築することができることが分かった。さらに、予期せぬ断線が発生しうる不安定なネットワークにも対応できるようにバックアップリンクを保持する方式を提案する。この方式でシミュレーションを行った結果、通常的方式に比べ、メンバ間の接続が突然切断される場合において有効であることが分かった。

## A Tree Configuration Algorithm for Reliable Multicast Protocol

SATOSHI MORITA,<sup>†</sup> TAKANORI HURUKAWA,<sup>††</sup> KAZUHIRO YAMADA<sup>††</sup>  
and FUMIAKI SATO<sup>††</sup>

In development of distributed applications, Reliable Multicast Protocols are the important infrastructure. For different applications and scales of the systems, many Reliable Multicast Protocols have been developed. The main theme of the Reliable Multicast is the efficient retransmission mechanism. In this paper, we propose a new tree configuration algorithm for the Reliable Multicast, which has tree based retransmission mechanisms. This algorithm aims to configure a balanced and small tree, for distributed retransmission. And we propose Backup-Link algorithm for network what occur suddenly disconnection between multicast members. Simulation of joining and leaving tree was performed for evaluation of the scale of the tree and balance of the retransmission load. From the comparison with the other method, the tree of our approach was smaller and more balanced. And, simulation of Backup-Link algorithm shows the effectiveness for that network what often cause disconnection.

### 1. はじめに

近年、コンピュータネットワークの高速化により分散アプリケーションの使用が一般的になってきた。分散アプリケーションの例としては、共有仮想環境<sup>1),2)</sup>、電子会議システム<sup>3)</sup>、分散共有メモリ<sup>4)</sup>、コンピュータ支援共同作業システム<sup>5)</sup>などがある。分散アプリケーションの設計において、高信頼マルチキャストは効果的かつ必要不可欠なネットワーク基盤となっている。また、参加しているユーザが増加しても少人数の場合

と同様のサービスを提供できる性能が求められる。高信頼マルチキャストについてはこれまで多くの研究が世界中の研究機関においてなされており、現在様々な全順序マルチキャスト方式が提案されている<sup>6),7),11)</sup>。高信頼マルチキャスト方式においてパケットの欠落に対応する再送処理はシステム全体の性能において少なからず影響を与える。そのため、システム全体の性能を落とさない効率的なパケットの再送処理を行う工夫が必要である。

本研究では、メンバが再送木を構築して、その構造に従って再送処理を行う方式に注目する。この方式では、送信者に再送要求が集中するのを避けられるのでスケーラビリティが高い。しかし木構造のメンバ管理など、複雑な処理が必要となる。

<sup>†</sup> 静岡大学大学院情報学研究科

Graduate School of Information, Shizuoka University

<sup>††</sup> 静岡大学情報学部

Faculty of Information, Shizuoka University

本稿では、マルチキャストグループに参加するメンバが動的に変化する木構造を構築し、それに従って再送を行い、パケットを補完する方式を提案する。この方式では、各メンバは構築する木の規模を小さくし、再送の負荷を分散するためのデータを保持する。自身が再送管理するメンバ数をリンク数とし、木において隣接するメンバまでの通信遅延を距離値としてこれをノード深度とし、自身から木の末端までのノード深度を最大ノード深度として、これらのデータを木に参加するときや脱退するときを使うことで木は小さく均等に構築される。小さく均等な木構造によりグループ全体で再送すべきパケットを効率的に補うことができる。

さらに、この通常の方式に加え、メンバ間のリンクが突然切断される場合に対応するためのバックアップ方式、データ更新頻度設定方式、参加要求パケットの送信範囲設定方式という拡張を提案する。

再送木の規模と負荷の分散がどの程度なされているかを調べるために、木構造への参加脱退といった構築処理のシミュレーションを行った。木を構築する別の方式でのシミュレーションを行い提案方式と比較した結果、木の規模が小さく再送の負荷が均等に分散する木構造を構築できると分かった。また、この提案方式と、各バージョンの方式とをシミュレーションで評価し比較した。その結果、バックアップリンク方式ではメンバ間のリンクが突然切断されるような不安定なネットワークにおいても木が分断されにくく、データ更新頻度設定方式、参加要求パケットの送信範囲設定方式においてはネットワーク資源をより消費しない再送木の構築が行われていることが分かった。

以下、2章において従来研究について述べる。3章においては提案方式における参加脱退プロトコルを詳細に説明する。4章において提案方式の拡張を述べる。5章では今回行ったシミュレーションの結果とそれにおける評価を述べ、6章においては提案方式と提案方式の各拡張方式との比較についてシミュレーションを行い、その結果と評価を述べる。7章ではシミュレーションにおける結果の考察を述べ、最後に8章で本稿のまとめを述べる。

## 2. 高信頼マルチキャスト通信と従来の木構築方式

本章では高信頼マルチキャストの概要とシミュレーションで比較する従来の木構築方式について述べる。

### 2.1 高信頼マルチキャスト

マルチキャストはマルチキャストグループに参加しているすべてのメンバに対してパケットを送信する通

信形態であり、パケットの複製は送信者ではなくルータによって行われるためネットワーク資源の浪費を抑えることができる<sup>8)~10)</sup>。

マルチキャストにおけるデータ配信はコネクションレス型通信であるIP、またはUDPを通じて行われるため、信頼性があまり高くない。高信頼マルチキャストでは参加するすべてのメンバがマルチキャストパケットを受信する必要がある。そのため、欠落したパケットを補完するためにパケットの欠落を検知する方法や再送を行う方法などを決定しなくてはならない。

再送処理には送信者が再送を行う場合と送信者以外が再送を行う場合がある。送信者が再送を行う場合、シンプルではあるが送信者に再送処理のための負荷が集中し、また送信者付近のネットワークにも負荷をかける。受信者からのNackのタイミングをずらすなどの改善策も存在する。

送信者以外が再送を行う場合では、雲形や木形、リング形に受信者を組織化する方法があり、それぞれ特徴は変わってくる。雲形はNackが集中することがなく再送の負荷が分散されスケラビリティに優れているが、再送の最終責任者が特定されず、メンバ情報を管理しようとする複雑になるといった欠点がある。木形では木に沿ってNackを送信し再送を要求する。受信者は送信元に直接コンタクトすることはなく、大規模な受信者グループにも対応できる。リング形ではトークンを回しデータ伝送および応答を同期化し、高パフォーマンスという利点がある。しかし、アルゴリズムの複雑さとプロトコルステータスの多さのため実装は困難である。また、各メンバがメンバシップ情報を維持しなければならずスケラビリティは比較的限制される。

マルチキャストを利用するアプリケーションには様々な種類がある。それぞれ、リアルタイム性や通信の信頼性やスケラビリティなど、マルチキャストに要求する条件が異なる。

これらの問題をすべて1つのプロトコルで解決するのは難しい。しかし、それぞれのマルチキャストアプリケーションが必ずしもすべての条件を要求するわけではない。そのため、それぞれの要件に特化した多数のプロトコルが提案、研究されている<sup>11),13)~15)</sup>。

### 2.2 従来の木構築方式

ここでは提案方式と比較するマルチキャスト配送木構築方式について述べる<sup>16)</sup>。

この方式はメンバを木構造で管理し、グループ通信パケットを木構造に従って配信するマルチキャスト方式である(図1)。これにより、送信者は木構造上で隣

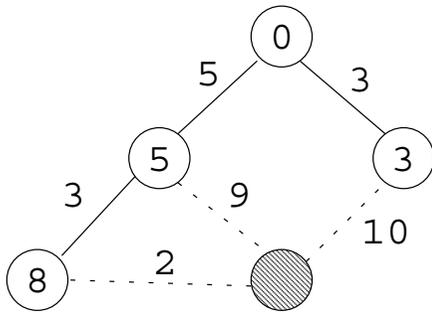


図1 従来方式における参加処理  
Fig. 1 Join process of the previous research.

接するメンバにパケットを送るだけでメンバ全員へパケットを送信できるようになるので、送信処理が軽量化される。また再送処理がメンバ全員へ分散されるため、ネットワークへの負荷も軽減される。メンバを管理する木構造はサーバを root とする 2 分木構造で管理する。ユーザは不規則に参加、脱退を行うので、メンバの木構造を動的に変更できるメンバ管理が必要となる。この方式の参加、脱退処理とグループ通信処理について説明する。

### 2.2.1 各メンバが保持するデータ

システムに参加しているメンバはメンバ管理に用いるデータとして、木構造上で隣接するメンバのアドレスと、自分から root までの距離を表した距離値を保持している。距離値は後述するメンバの参加処理で利用する。

### 2.2.2 木構造への参加処理

新しいクライアントが参加するとき、まずサーバへのユニキャストでシステムへの参加要求を出し、その応答としてメンバのリストを受け取る。次に新規クライアントはメンバのリストを元にメンバ全員にユニキャストによって、木構造への参加要求を出す。子ノードを追加できるメンバは自分が保持している距離値を付加した参加許可のパケットを新規クライアントに送る。この間、新規クライアントは送信を出してから参加許可パケットが帰ってくるまでの時間を各参加許可パケットに対して計測する。参加許可を受け取った新規クライアントは計測した応答時間と、参加許可パケットに付加されてきた距離値を合計し、その値が最も小さくなるメンバを親ノードとして選択する。またその値は新規のクライアント自身が保持する距離値となる。新規のクライアントは親ノードと選択したメンバに参加パケットを、それ以外の参加許可パケットを送ってきたメンバには不参加パケットを送信する。そしてサーバからアプリケーションの現状のデータ

を受け取る。このようにして参加処理は終了する。

### 2.2.3 木構造からの脱退処理

システムから脱退するとき、子ノードを持つメンバが脱退する場合がある。その場合、親を失ったメンバはもう一度グループ通信への参加処理を行うことでメンバリストの木を保つ。

### 2.2.4 グループ通信処理

メンバ全体へのグループ通信処理を行うとき、送信者は自分に隣接しているメンバへユニキャストで送信する。それを受信したメンバはパケットを複製し、さらに自分に隣接しているメンバへ送信する。これを繰り返すことでメンバ全体にグループパケットが送信される。

この方式は、構成する 2 分木のノードの距離を距離値として利用することで木のルートから末端までの距離を小さくすることができる。この方式は 2 分木だけでなく N 分木にも応用が可能である。しかし N 分木でこの方式を利用すると新たな問題が発生する。新規のクライアントをグループ通信に参加させるとき、ルートまでの距離だけを親ノード選択の判断基準としているため、すでに多数の子ノードを持つメンバを親ノードとして選択してしまい、1 人のメンバだけに負荷が偏ることがある。また、root ノードが木の中心にあるとは限らないため、必ずしもコンパクトな木を構成できない可能性がある。

## 3. 提案方式

前述した従来方式では、N 分木でメンバ管理を行うと、1 人のメンバにシステム全体の負荷が偏ることがあるという問題が残されていた。

これを解決する今回提案する方式を説明する。

### 3.1 目的

メンバは、小さく均等な木構造を構築することを目的とする。木が小さくまとまっていると再送要求である Nack がすべてのメンバに短時間で行き渡り、再送が効率良く行われる。均等な木構造を作ることにより、1 つのメンバに処理が集中することを避ける。各メンバの処理が分散されることでシステム全体の性能に影響を与えることを防ぎ、メンバ数が増大してもメンバ数が少数のときとほぼ同等のサービスを提供することができる。

木を構築するために必要な情報は各メンバに分散させて管理するので、各自が保有するデータの独立度が高い。そのため、木に新しくメンバが参加したり、途中でメンバが脱退したりしても、それらの処理がアプリケーション全体に大きな影響を与えることはない。

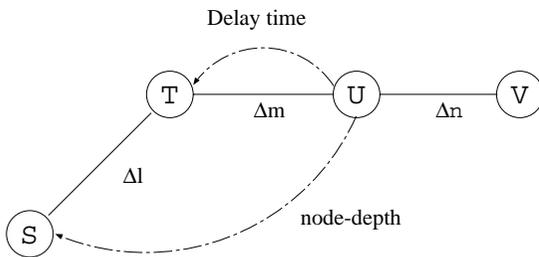


図2 ノード深度  
Fig. 2 Node depth.

よって、木構造の操作が容易となる。

本稿では参加する時点と脱退する時点に注目し、負荷が分散され、木が小さくまとまるような参加、脱退プロトコルを提案する。

### 3.2 参加, 脱退処理

前述したような特徴を持つ木構造を構成するための参加, 脱退方式を説明する。

#### 3.2.1 メンバが保持するデータ

システムに参加しているメンバは、そのメンバに何人のメンバがリンクを張っているかを示すリンク数という値を持つ。また、メンバに隣接するメンバごとの通信遅延をノード間の距離値として保存する。それらの値とは別に、各ノードから見た各リンク方向の木の大きさに対応して与えられるノード深度という値がある。

ノード深度とは、そのノードから先の最も遠い末端のメンバまでの距離を表す。メンバはノードから先の木構造を完全に知ることはできないが、ノード深度によってどれだけの広がりを持っているのかおおよそ知ることができる。木構造では、1人のメンバからのびるリンクが複数存在する場合がある。それらのノードが持つノード深度の中で最も大きい値を最大ノード深度と呼ぶ。図2では4人のメンバS, T, U, Vがいる。それぞれのノードの距離値は $\Delta l, \Delta m, \Delta n$  ( $\Delta l > \Delta m > \Delta n$ )である。Uにつながる2本のリンク方向のノード深度は $\Delta l + \Delta m$ と $\Delta n$ であり、最大ノード深度は $\Delta l + \Delta m$ となる。

これらの値は参加, 脱退処理で利用される。

#### 3.2.2 システムへの参加処理

新規に参加するメンバはシステムがグループ通信に利用しているマルチキャストアドレスを知っており、メンバ全員に参加要求をマルチキャストする。この参加要求を受信したメンバは、自身が持っているリンク数と最大ノード深度の情報を返す。また、新規メンバは要求を送信した時刻を記録し、各メンバからの応答が返ってくるまでの時間を計測し、これを通信遅延と

して保存する。新規メンバはリンク数, 最大ノード深度, 通信遅延の合計が最も小さくなるメンバを自分の隣接ノードとして選択する。そして、再送管理の関係を結ぶためのノード接続要求 packets を選択したメンバに送信する。

#### 3.2.3 脱退処理

メンバが木構造から脱退する場合、木構造が切断されてしまうことがあるので、同時に木構造の再構築をしなくてはならない。また、新たに構築した再送管理関係で再送すべき packets の同期をとる必要がある。

それらの処理について説明する。木構造から脱退するメンバは、自分にリンクを張っているメンバの中から、リンク数と通信遅延の合計が最小の値を示すメンバを選択して、脱退後の再接続先メンバとして選択する。

次に脱退 packets を隣接しているノードに対して送信する。脱退 packets には先ほど選択したメンバのアドレスが付加されている。脱退 packets を受け取ったメンバは、付加されてきたアドレスと自身のアドレスと比較する。自身と同じ場合、そのメンバは待機する。異なるならば、そのアドレスを持つメンバが新しい隣接メンバになる。新しい隣接メンバに再参加処理 packets を送信し、再送されるべき packets の同期をとって再送管理の関係を新しい隣接メンバに切り替える。再参加処理が終わったメンバは、脱退するメンバに対して脱退許可 packets を送信する。脱退するメンバに隣接していたメンバ全員から脱退許可 packets を受け取った時点で、木構造からの脱退処理が終了する。

#### 3.2.4 ノード深度の変更処理

以上のようにして参加, 脱退をする場合、木の形が変わることがある。それにともないノード深度の変更が加わる場合がある。新規参加処理の場合、新規メンバまでの距離値 $\Delta t$ がそのままノード深度 $\Delta Q$ になる。再参加処理の場合、新規隣接メンバから送られてきた最大ノード深度 $\Delta Q$ と距離値 $\Delta t$ を加えた値がノード深度 $\Delta R$  ( $\Delta R = \Delta Q + \Delta t$ )になる。新しくできたノードのノード深度の大きさによって、既存の隣接メンバの中にノード深度を変更する必要があるメンバが現れることがある。その場合、ノード深度の変更が必要と思われるメンバに対して新しくできたノードのノード深度 $\Delta R$ を、ノード深度変更 packets として送信する。ノード深度変更 packets を受け取ったメンバは、送信元のメンバへの距離値 $\Delta s$ と送られてきたノード深度 $\Delta R$ を加算し、新たなノード深度 $\Delta S = \Delta R + \Delta s$ として設定する。この処理を繰り返すことで、変更の必要があるメンバの持つノード深

度を変更する。

#### 4. 提案方式における各拡張方式

高信頼マルチキャストを利用するアプリケーションには様々な種類がある。それぞれ、リアルタイム性や通信の信頼性やスケーラビリティなど、マルチキャストに要求する条件が異なる。これらの問題をすべて1つのプロトコルで解決するのは難しい。しかし、それぞれのマルチキャストアプリケーションが必ずしもすべての条件を要求するわけではない。そのため、それぞれの要件を満たすよう提案方式にバックアップリンク方式、送信範囲設定方式、データ更新頻度設定方式を加えた。

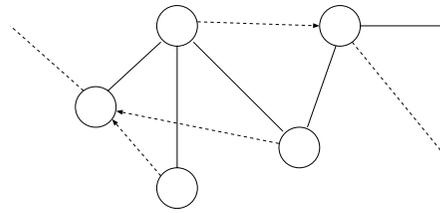
以下に各方式について説明する。

##### 4.1 バックアップリンク方式

メンバ間のリンクが突然切断される場合、再送木が分断され、再送が正しく行われないことがあり、大変致命的である。そこで本来のリンクとは別にバックアップリンクを保持する方式を提案する(図3)。リンクが突然切断される場合には、即座にバックアップリンクを利用し、再送木が分断されるのを防ぐ。

バックアップリンク先は通常のリンク先と同様の方法で選択される。つまり、新規メンバは再送木における最適なメンバと通常のリンクを保持し、次に最適なメンバとバックアップリンクを保持する。

バックアップリンク方式では、バックアップリンク先も通常のリンク先と同様にパケットの再送管理を行うが、実際に再送は行わない。つまり、受信したパケットの Ack はバックアップリンク先と通常のリンク先に送信するが、再送要求は通常のリンク先にだけ送信する。また、通常のリンク先では双方向の再送管理を行うが、バックアップリンク先では片方向のみの再送管理となる。これにより通常のリンクが切断された場合、または通常のリンク先のノードが故障した場合、バックアップリンク先が再送管理を行っているため、バックアップリンクを通常のリンクに切り替えることにより従来より確実に再送が行われる。従来の方式に比べ、再送管理コストは多いが、突然のリンク切断が存在しても再送の信頼性は非常に高い。再送木に参加するメンバを  $N$  とすると、通常の方式でのリンク数は  $N - 1$  となる。再送コストは通常のリンクでは双方向なので  $2(N - 1)$  とすると、バックアップリンク方式では、片方向のバックアップリンクが加わるのでは  $2(N - 1) + N$  となる。再送管理のコストはおおよそ1.5倍かかるというてよい。突然のリンク切断が一定の頻度で発生しても、従来方式に比べ再送木がどの程



— Normal Link

----- Backup Link

図3 バックアップリンク方式

Fig. 3 Backup-link method.

度稼働し続けるかシミュレーションで評価する。

また再接続にかかるコストはバックアップリンク先に送信するパケットとその応答パケットとなる。つまりバックアップリンク先とのパケットの往復時間だけで済む。通常の提案方式では、まず再接続先を決定するために、切断されたノードは、切断されたリンク先ノードにおけるいくつかの隣接ノードに再接続要求パケットを送信する。受信したノードは応答を返し、これにより評価式に基づき再接続先を決定する。次に決定した再接続先とパケットをやりとりしリンクを復旧する。つまり2つの往復時間がかかる。

##### 4.2 参加要求の範囲設定

従来方式では参加処理を行う際に、再送木を構築しているすべてのメンバに参加要求を送信する。しかし、メンバ数が増加するに従って参加要求に回答するパケットの数も増大し、ネットワーク全体に多大な負荷を与える。よって、新規参加者は一定の範囲にのみ参加要求を送信する。実際には参加要求パケットの TTL を制限する、TTL とはパケットの生存可能時間を表している。実際には時間ではなくホップカウントをもって TTL 値とし、ルータを1つ通過すごとに値を1つ減らし、0になるとそのパケットを破棄する。または参加要求パケットに送信時間を付与し一定時間を過ぎて受信した場合は応答しないという方法が考えられる。この場合は、各ノードにおける時計がある程度一致していることを前提としなければならない。

TTL で制限する場合は、新規参加者の周りにおけるノードの密度が分からないため1から開始し、徐々に TTL を増加させていくことにより大量の応答パケットを防ぐことができると考えられる。

参加要求の範囲を制限した場合、従来に比べて再送木がどの程度劣化するかシミュレーションで評価する。

##### 4.3 データ更新の頻度調整

データの更新は、ノード深度が変更されるとき、つまり木全体においてメンバが参加・脱退するたびに行

われる。メンバ数が増大した場合、データ更新におけるネットワークの負荷はシステム全体に少なからず影響を与える。そこで木構造が1回変化することに更新するのではなく、一定回数ごとにデータ更新を行う。しかし、データ更新を控えることにより木構造のパラメータが最新の情報にはならず、再送木が最適に構築されとは限らない。

データ更新頻度の削減がどのような影響を再送木に与えるかシミュレーションを行い従来方式と比較する。

### 5. シミュレーションによる評価

従来方式のシミュレーションを行い、参加、脱退、定常運用時において提案方式との比較を行う。

また、バックアップリンク方式とデータ更新における通信遅延を考慮した方式とでシミュレーションを行い通常の提案方式と比較した。

シミュレーションは独自のものでC言語により作成した。

#### 5.1 基本方式の評価

従来方式と提案方式との比較は、参加脱退時と定常運用において最大ノード深度とリンク数という基準でシミュレーションを行った。最大ノード深度により木の広がりが分かる。木が広がっていると端から端までの遅延が大きくなることから、より小さいほうが良好な木といえる。リンク数は各ノードにつながっている隣接ノードの数であり、すべてのノードのうち最大のリンク数を比較した。最大のリンク数が少ないほうが負荷が分散されておりバランスのとれた木だといえる。

##### 5.1.1 参加脱退時における比較

6000のホストを配置した10000 \* 10000の座標を使い、木への参加、脱退を行って再送木がどのような変化をするのかを比較、評価する。このとき、ホストの配置、参加する順番、脱退する順番はすべて同じである。また、参加、脱退によってネットワーク構成が変化したときに、最大ノード深度の変更がメンバ全員に伝播するための十分な時間間隔があるものとする。

階層的なノード配置を行うTiersモデルの利用も考えられるが、本来再送ノードは受信者ノード数より少なく、その配置もまばらであると考えられるため、今回はランダムなホスト配置としている。また、評価式により多数のノードとつながっているホストは選択されにくくなっているため、ノードとつながることのできる数の制限は設けていない。

参加脱退時における従来方式のN分木では、Nを2から100まで変化させてシミュレーションした結果、N = 50とした場合に最も小さく負荷分散が行われる

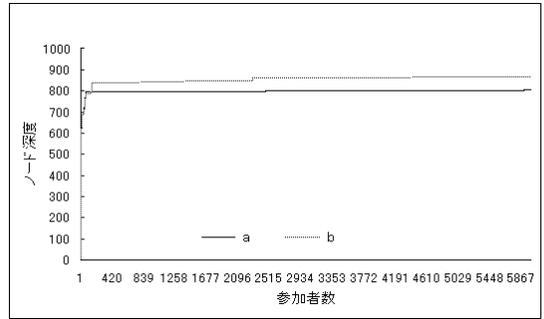


図4 参加時における最大ノード深度の推移  
Fig. 4 Max-node-depth for join process.

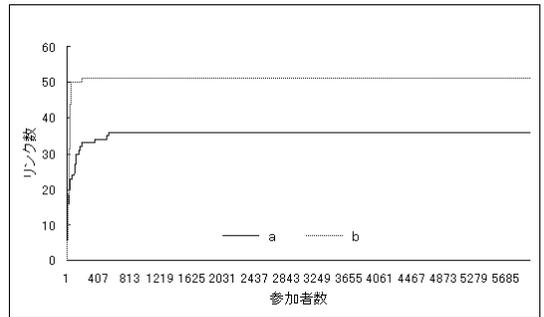


図5 参加時における最大負荷数の推移  
Fig. 5 Max-wegiht for join process.

表1 参加時における通算の最大値  
Table 1 Maxvalue of join process.

	a	b
ノード深度	803.7	870.35
リンク数	32	51

木構造を構築できることが明らかになった。この場合のシミュレーションの結果と提案方式の参加、脱退シミュレーションを比較する。

まず、参加のシミュレーションについて比較する(図4, 図5, 表1)。aが提案方式を使った場合の結果で、bが従来方式を使用した場合の結果である。

以上の結果から、提案方式のほうが小さくて負荷分散ができる木構造を構築していることが明らかになった。

次に、脱退のシミュレーションについて比較する(図6, 図7, 表2)。aが提案方式を使った場合の結果で、bが従来方式を使用した場合の結果である。

以上の結果から、提案方式の方が小さくて負荷分散ができる木構造を維持しながら脱退処理を行えることが明らかになった。

##### 5.1.2 定常運用時についての比較

ここでは提案方式と従来方式で定常運用を行った

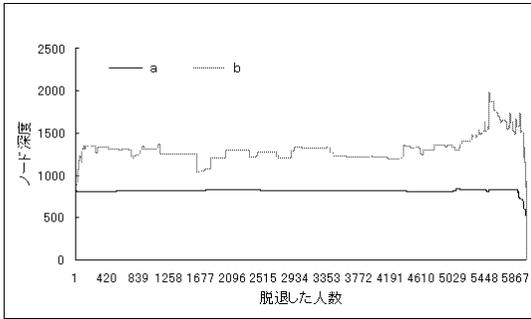


図 6 脱退時における最大ノード深度の推移  
Fig. 6 Max-node-depth for leave process.

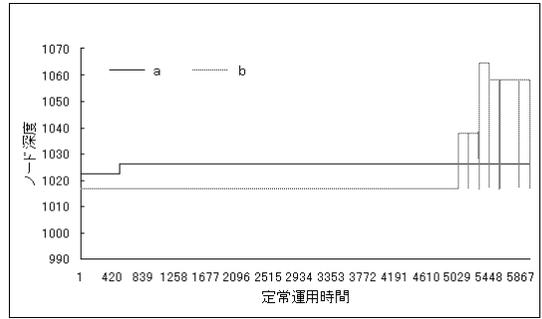


図 8 定常運用における最大ノード深度の推移  
Fig. 8 Max-node-depth for long-term operation.

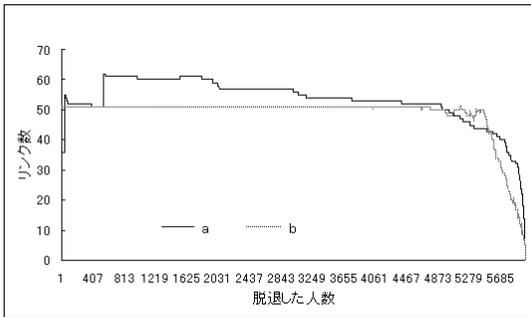


図 7 脱退時における最大負荷数の推移  
Fig. 7 Max-weight for leave process.

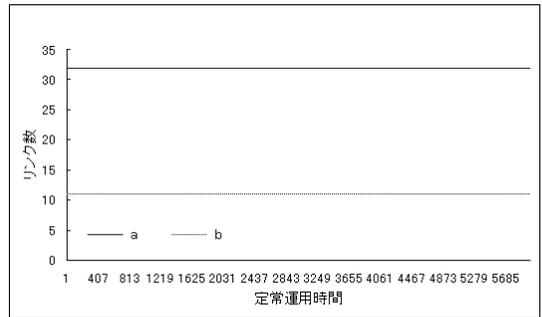


図 9 定常運用における最大負荷数の推移  
Fig. 9 Max-weight for long-term operation.

表 2 脱退時における通算の最大値  
Table 2 Maxvalue of leave process.

	a	b
ノード深度	835.4	1981.1
リンク数	62	51

表 3 定常運用における通算の最大値  
Table 3 Maxvalue of long-term operation.

	a	b
ノード深度	1025.8	1064.4
定常運用前	1022.4	1017.0
リンク数	32	11

場合、構築する木構造にどのような違いが現れているか比較する。定常運用のシミュレーションは 6000 のホストを配置した 10000 \* 10000 の座標を使い、一定期間参加と脱退を繰り返させ、木構造の変化を計測する。

定常運用における従来方式の N 分木では、N を 2 から 100 まで変化させてシミュレーションした結果、N = 10 とした場合に最も小さく負荷分散が行われる木構造を維持できることが明らかになった。この場合のシミュレーションの結果と提案方式の定常運用シミュレーションを比較する(図 8, 図 9, 表 3)。

以上の結果から、負荷分散の点では子ノードの上限が 10 で制限されている従来方式のほうが優れているといえる。しかし定常運用を長く続けていると木構造が広がってきていることが分かる。よって提案方式を使用したほうが、小さく負荷分散ができる木構造を維持しながら定常運用ができるといえる。

また従来方式において経過時間が 5000 を超えたあたりでノード深度が振動しているが、これは中間ノードの脱退による一時的なノード深度の減少であると考えられる。

### 5.1.3 データ更新における通信遅延の考慮

データ更新における遅延を考慮した場合の比較は、リンク数と最大ノード深度の平均のデータを採取した。ともに小さい値のほうが負荷が集中されることがなく、端から端までの遅延の少ない木が構築され、良好な木といえる。

50, 100, 200 とメンバの数を変化させて配置した 1000\*1000 の座標を使い再送木を構築させ、遅延を考慮した場合としていない場合でメンバの参加頻度を変化させてシミュレーションを行った。遅延は距離 100 を平均 0.1 秒とする(図 10, 図 11, 図 12)。ネットワークの遅延を考慮すると、木の更新情報は古くな

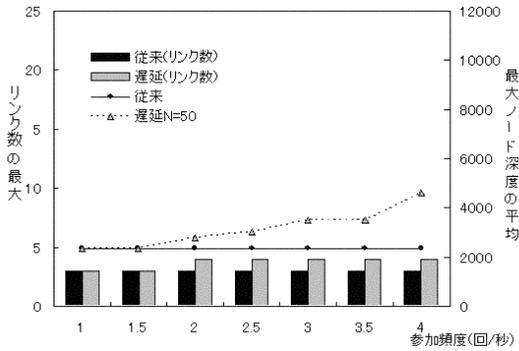


図 10 参加頻度に対する最大ノード深度とリンク数の変化 : 50 members.  
Fig. 10 Max-depth and weight for join frequency: 50 members.

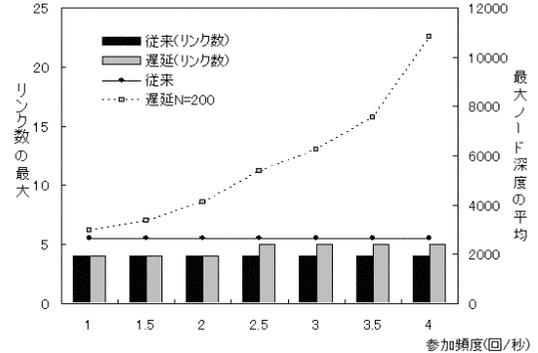


図 12 参加頻度に対する最大ノード深度とリンク数の変化 : 200 members.  
Fig. 12 Max-depth and weight for join frequency: 200 members.

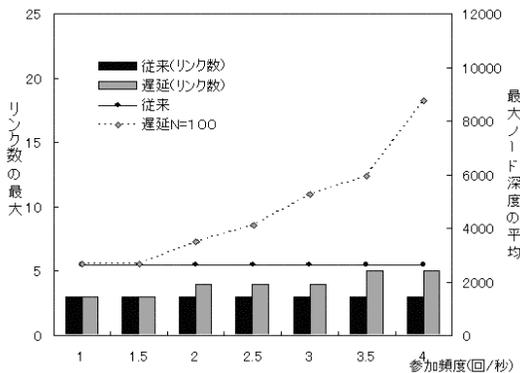


図 11 参加頻度に対する最大ノード深度とリンク数の変化 : 100 members.  
Fig. 11 Max-depth and weight for join frequency: 100 members.

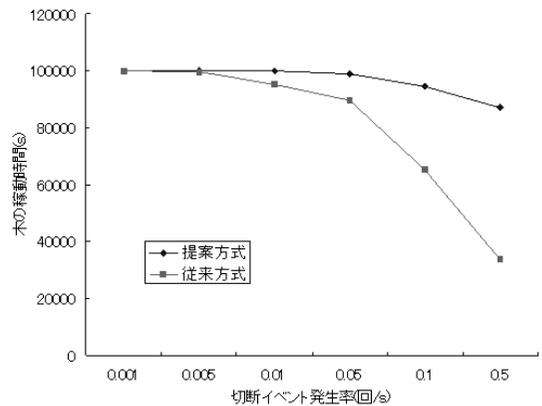


図 13 木の稼働時間の比較

Fig. 13 Operating time of the retransmission tree.

り、木は劣化するはずである。どの程度劣化するかを調べる。

ネットワークの遅延を考慮した結果、参加頻度が高くなるにつれ、古い情報に基づく木の構築が行われるため遅延を考慮しない評価に比べて劣化した木が構築されることが分かる。また、メンバの数が増加するほど遅延の影響が大きくなっていることが分かる。

## 5.2 各拡張方式の評価

以上のシミュレーションによって評価された提案方式を通常方式とし、各拡張の方式と比較した。

拡張方式との比較において、バックアップリンク方式との比較は、切断イベント発生率を増加させた場合の木の稼働時間のデータを採用した。切断イベント発生率は、ネットワークの不安定さを表しており、発生率が高いほど不安定なネットワークといえる。木の稼働時間は、再送木が分断しておらず機能する時間のことであり、この時間が長いほど安定した再送が行われるといえる。つまり、シミュレーション全体の時間が

ら、木が分断されてから再接続されるまでの時間を除いた時間が、木の稼働時間となる。

参加要求の送信範囲を設定した場合とデータ更新費率を設定した場合には、データ更新における遅延を考慮した場合と同様に、リンク数と最大ノード深度の平均のデータを採用した。

### 5.2.1 バックアップリンク方式

100のメンバを配置した1000\*1000の座標を使い再送木を構築させ、突然のリンク切断として、あるノード間の接続を切断させるイベントを一定の間隔で発生させた。シミュレーションは100,000秒行い、木の稼働時間を、従来方式とバックアップリンク方式とで比較した(図13)。バックアップリンク方式ではバックアップリンクコストとして1.5倍のコストがかかるが、木の稼働時間において、そのコストに見合う時間大きければよい。

以上の結果からリンク間の突然の切断が多いほど

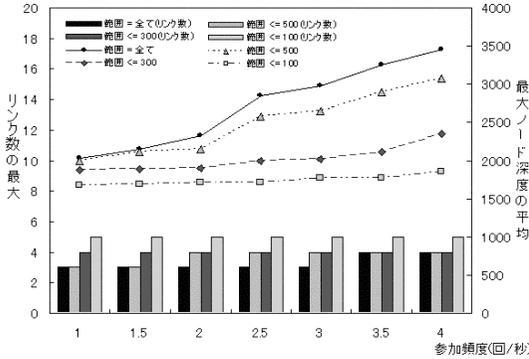


図 14 参加要求の送信範囲を設定した場合との比較

Fig. 14 Max-depth and weight for join frequency: Transmission range of join request is limited.

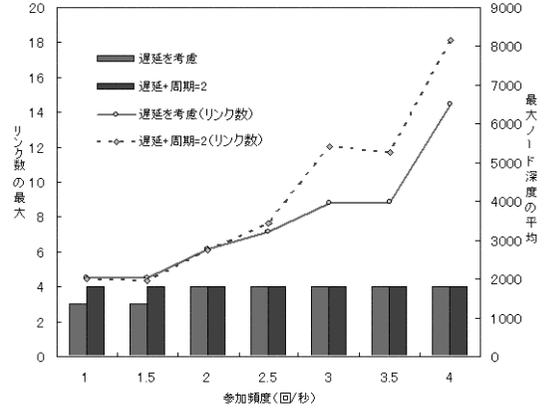


図 15 データ更新頻度を設定した場合との比較

Fig. 15 Max-depth and weight for join frequency: Update frequency of node depth is limited.

バックアップリンク方式と従来の方式の差が出るのが分かる。発生率が高くなるとバックアップリンク方式における木の稼働時間は従来の2倍以上となり、これは1.5倍のコストの価値はあるといえる。

5.2.2 参加要求の送信範囲の限定

50のメンバを配置した1000 \* 1000の座標を使い再送木を構築させ、遅延を考慮した場合と、それに加え参加要求の送信範囲を設定した場合とでシミュレーションを行った。送信範囲を距離500, 300, 100以内と設定しない場合の4つで参加頻度を変化させた(図14)。メンバ数を50としたのは、木の構築具合が遅延を考慮しない場合とあまり変わらないからである。送信範囲を制限した場合、参加要求パケットと応答パケットは大幅に削減される。しかし、全体のメンバから最適なメンバを選択しないことになるため、再送木はある程度劣化するはずである。削減されたコストにおいて、どの程度木が劣化するかを比較する。

送信範囲が最も狭い100以内ではリンク数が全体的に多いが、木全体は大きく広がらないことが分かる。近くのメンバにしか接続しないことで木は小さくなるが負荷が集中するノードが生じてしまうことがある。送信範囲を限定することは、そこに存在するノード数を限定することであり、100以内ということは、全体の約30分の1程度のノードに参加要求を送っていることになる。しかし、木の劣化はパケットの削減率ほど大きくなく、有効な手法と考えられる。

5.2.3 データ更新頻度の設定

上記と同様の環境で、遅延を考慮した場合と、それに加えデータ更新頻度を設定した場合とでシミュレーションを行った。データ更新頻度は通常の場合は木が毎回変化することに更新しているが、設定した場合は2回に1回の頻度で更新している(図15)。データ

更新パケットを2分の1に削減できているが、このコスト削減のためにどの程度木が劣化するか比較する。

更新頻度が変化してもリンク数の最大値はあまり変わらないことが分かる。また木全体のバランスは更新頻度を2回に1回としたほうが広がっていることがわかる。コストは半分に削減されているが、木はそれほど劣化しておらず有効な手法と考えられる。

5.3 結果の考察

従来の木構築方式について、定常運用のシミュレーションを行い、提案方式との比較を行った。

従来方式はN分木を利用し、新しい参加者は、rootまでの通信遅延が最も小さくなるように隣接メンバを選択する。N=50が参加脱退時における適切な値であり、このときのシミュレーションの結果と提案方式の結果とを比較し、提案方式の方が優れていることが示された。

定常運用の場合、小さくて負荷分散ができる木構造を維持するのは、N=10の場合が適切である。このときのシミュレーションと、提案方式のシミュレーションとを比較した。その結果、提案方式は負荷を分散する木を維持するという点で従来方式よりも劣っているが、定常運用を長く続けていると従来方式の木構造が徐々に広がり始めることが明らかになった。提案方式では木構造が広がるということではなく、小さくて負荷の分散ができる木構造を維持しながら定常運用を行うことができるということが明らかになった。遅延を考慮すると遅延を考慮しない評価より再送木は劣化し大きく広がることが分かった。メンバの数が少ない場合はあまり変化はないが、メンバが増加するほどその差は顕著に現れることが分かった。

さらに、提案方式にバックアップリンク方式を追加

した場合と、遅延を考慮して参加要求の送信範囲を設定した場合、遅延を考慮してデータ更新頻度を設定した場合とで再送木の構築シミュレーションを行い、通常の提案方式と比較した。

バックアップリンク方式ではメンバ間のリンクが突然切断されることが多い環境では非常に有効であるといえる。しかし、通常の安定したネットワークでは、逆にバックアップリンクを保持するコストがかかり有効とはいえない。ネットワークの状態によってどちらの方式を使用するかを決定する必要がある。

参加要求の送信範囲を設定した場合では、設定したほうが再送木が広がらずコンパクトに構築されることが分かった。メンバの負荷は少ししか変わらず、ネットワーク資源の消費を軽減することからも参加要求の送信範囲を設定することは非常に有効であるといえる。

データ更新頻度を設定した場合では、参加頻度が大きくなると通常の場合と比べ再送木が広がること分かった。参加頻度がそれほど大きくない場合では、どちらの場合でもあまり変わらないので、通信コストを軽減しているデータ更新頻度を設定する場合のほうが非常に有効であるといえる。

## 6. おわりに

本稿では再送処理を動的な木構造で管理されたメンバに分担させる、高信頼マルチキャストにおける再送木構築方式について提案を行い、木構造を構築するための参加、脱退プロトコルについてシミュレーションを行った。またこの提案方式を通常方式として、バックアップリンク方式を追加した場合、通信遅延を考慮した場合、通信遅延を考慮し参加要求の送信範囲を設定した場合、通信遅延を考慮しデータ更新頻度を設定した場合とでシミュレーションを行い評価した。

本方式では再送処理の負荷を分散させることに重点を置いた。そのためにメンバを動的な木構造で分散管理するようにした。また、再送処理の迅速な完了のために、木構造の規模ができる限り小さくなるように、参加、脱退プロトコルを設計した。木構造に参加しているメンバの位置を表すパラメータとしてノード深度を加えることによって、木の規模が小さく、再送の負荷が均等に分散する木構造を構築することができた。

バックアップリンク方式ではメンバ間の突然のリンク切断がよく発生するネットワークの場合に通常の提案方式に比べ非常に安定した再送が行えることが分かった。通信遅延を考慮すると遅延を考慮しない提案方式と比べ再送木が劣化し広がることが分かった。参加要求の送信範囲を設定することで、再送木はよりコ

ンパクトに構築され、かつ通信コストを軽減できることが分かった。データ更新頻度を設定した場合、参加頻度がそれほど多くない状況では通常の方式とあまり変わらずネットワーク資源の消費を軽減でき、非常に有効であることが分かった。

今後の課題としては参加要求の送信範囲設定とデータ更新頻度の設定を組み合わせ、最適な再送木を構築しつつどの程度まで通信コストを減少できるか検討する必要がある。また、バックアップリンク方式において、多数のノードがいる場合、バックアップリンクの設定基準を考える必要がある。これは、評価式において1番目をメインとし2番目をバックアップリンクとしているが、基準が同じであることにより、この2つの経路が同じである可能性が高いと考えられるためである。つまり、メインのリンクが故障した場合、同様にバックアップリンクも故障する可能性が高いと考えられる。最後に本方式を実装し評価することがあげられる。

## 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*, pp.18-25 (1997).
- 3) 山田善大, 池谷利明, 峰野博史, 太田 賢, 水野忠則: モバイル電子会議システム PARCAE の提案, 情報処理学会全国大会講演論文集, pp.3-541-3-542 (1997.3).
- 4) Fekete, A., Lynch, N. and Shvartsman, A.: Specifying and Using a Partitionable Group Communication Service, *16th Annual ACM Symposium on Principles of Distributed Computing* (1997.8).
- 5) Rodden, T.: A survey of CSCW systems, *Interacting with Computers*, pp.319-353 (1991.3).
- 6) Sato, F., Minamihata, K., Fukuoka, H. and Mizuno, T.: A Reliable Multicast Protocol with Total Ordering for Distributed Virtual Environment, *Proc. IEEE ICDCS2000 Workshop on Group Communication Systems* (2000).
- 7) Montgomery, T., Callahan, J.R. and Whetten, B.: Fault Recovery in the Reliable Multicast Protocol, *NASA/West Virginia University Software, IV and V Facility* (1995.11).
- 8) Deering, S.: Host Extensions for IP Multicasting, Indianapolis, IN, New Riders (1996).
- 9) Eriksson, H.: Mbone: The multicast back-

- bone, *Comm. ACM*, Vol.37, No.8, pp.54-60 (1994).
- 10) Macedonia, M.R. and Brutzman, D.P.: Mbone Provides Audio and Video Across the Internet, *IEEE Computer*, Vol.27, No.4, pp.30-36 (1994).
- 11) Gemmell, J.: Scalable Reliable Multicast Using Erasure Correcting Resends, Technical Report, MSR-TR-97-20, Microsoft Research, Redmond, WA (1997.6).
- 12) Floyd, S., Jacobson, V. and McCanne, S.: A Reliable Multicast Framework for light-weight Sessions and Application Level Fraing, *IEEE/ACM Trans. Networking* (1995).
- 13) Lin, J-C. and Paul, S.: RMTP: A Reliable Multicast Transport Protocol, *IEEE INFO-COM 1996*, pp.1414-24 (1996).
- 14) Paul, S., et al.: Reliable Multicast Transport Protocol (RMTP), *IEEE JSAC*, Vol.15, No.3 (1997).
- 15) Whetten, B. and Taskale, G.: An Overview of Reliable Multicast Transport Protocol II, *IEEE Network* (2000.2).
- 16) Vuong, S., Lo, A., Yung, A., Yi, Z. and Sato, F.: Multicasting in Distributed Virtual Environment, *6th Internatonal conference on distributed multimedia systems* (1999.7).

(平成 14 年 7 月 5 日受付)

(平成 14 年 12 月 3 日採録)



森田 悟史 (正会員)

昭和 53 年生。平成 13 年静岡大学情報学部情報科学科卒業。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、アドホックネットワークに興味

を持つ。

古川 雄宣 (正会員) 昭和 54 年生。平成 14 年静岡大学情報学部情報科学科卒業。同年京セラコミュニケーション (株) 入社、現在に至る。IP マルチキャストに興味を持つ。

山田 和宏 (正会員) 昭和 54 年生。平成 14 年静岡大学情報学部情報科学科卒業。同年リコー (株) 入社、現在に至る。IP マルチキャストに興味を持つ。



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。