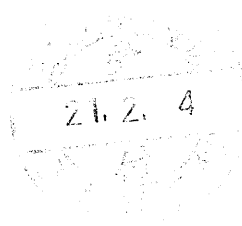




修士論文

P2P ネットワークにおける n 対 n での
データ交換について



平成 20 年度修了
三重大学大学院 工学研究科
博士前期課程 情報工学専攻

小林 裕司

目次

はじめに	1
第1章 ビデオ会議システム	2
1.1 n対nでのデータ交換	2
1.2 H.323について	3
1.2.1 集中型	3
1.2.2 非集中型	5
1.2.3 ハイブリッド型	5
1.3 XCAST6	5
第2章 マルチキャスト	6
2.1 クライアント・サーバモデル	6
2.2 IP マルチキャスト	7
2.3 XCAST	8
2.4 アプリケーション層マルチキャスト	10
第3章 提案手法	11
3.1 従来手法の問題点	11
3.2 P2P ネットワークを用いた手法	12
3.2.1 論理ネットワークポロジ	12
3.2.2 動作	12
3.2.3 動作例	14
3.2.4 性能	16
3.2.5 ノード数が2のべき乗以外の場合	17
3.3 マルチキャスト木を用いた手法	19
第4章 評価実験	21
4.1 実験環境	21
4.1.1 ネットワークモデル	21

4.1.2 シミュレーション動作	23
4.1.3 実験方法	23
4.2 ノード数変化における実験	24
4.2.1 実験方法	24
4.2.2 実験結果 (上り 1Mbps, 下り 10Mbps)	24
4.2.3 実験結果 (上り 2Mbps, 下り 10Mbps)	25
4.2.4 実験結果 (上り 5Mbps, 下り 40Mbps)	25
4.2.5 実験結果 (上り 100Mbps, 下り 100Mbps)	26
4.3 上り帯域幅変化における実験	26
4.3.1 実験方法	26
4.3.2 実験結果 (12 ノード, 下り 10Mbps)	27
4.3.3 実験結果 (24 ノード, 下り 10Mbps)	27
4.4 データ量変化における実験	28
4.4.1 実験方法	28
4.4.2 実験結果: データ量が 20Byte, 1Mbit の場合	28
4.5 考察	29
おわりに	30
謝辞	31
参考文献	32

はじめに

近年、コンピュータのハードウェアの発達、ネットワークの広帯域化を背景としてインターネットを用いたマルチメディア配信が活発に行われるようになり、ビデオ会議システムや IP 電話、オンラインゲーム等の遠隔地にいる者同士がコミュニケーションをはかったり、協調して作業を行うといったシステムの需要が高まってきている。これらのシステムは各参加者がそれぞれ作り出すビデオフレームデータや音声データ、ゲームコマンド等を全参加者で共有して利用する必要があるが、従来、その共有手法モデルとして用いられてきたクライアント・サーバモデルでは、サーバに負荷が集中し大規模な参加者数でのシステムに適していない。大規模な参加者でシステムを稼働させるためには、サーバの拡充を行う必要があり、管理コストが高いという問題がある。そのため、IP マルチキャストや XCAST が提案されている。しかし、この手法もインターネット上で利用するためには未だ問題があり、実用化にはいたっていない。

本研究では、クライアント役端末、サーバ役端末といった端末の種類を分ける事なく、全端末が対等な役割を行う P2P ネットワークを構築する事により、負荷が集中するのを避け、参加者数が大規模になっても耐えうる手法を提案している。すなわち、参加者ノードに ID 番号 ($0 \sim n-1; n$ はノード数) を割り振り、自身と ID 空間上で 2 のべき乗離れたノードとの接続を確立した P2P 型ネットワークを構築する。その後、各参加ノードは自身のデータを作成/更新した後、2 のべき乗離れたノードへ順にその時点で自身が得ているデータ全てを送信するという動作を並列に行うことにより、高速な n 対 n でのデータ交換を実現する。また、ノード数が 2 のべき乗でない場合のために、1 台のノードが 2 ノード分の通信を代行する事によって 2 のべき乗に足りない分を補完する代行ルーティング法と、2 のべき乗上のノードでネットワークを生成した後、そこに端数ノードを接続するハイブリッド法を提案する。評価基準として、ビデオ会議システムへの応用を想定し、各端末が持つ 1 つのデータを、全参加者が全端末分集めるのにかかる時間を挙げる。

本論文は全 4 章から構成される。第 1 章では、研究背景であるビデオ会議システムの基本的な仕組みや、現状で定められている H.323 という規格、そして既存ソフトウェアである XCAST6 について述べる。第 2 章では、特定グループにデータを配信するマルチキャスト技術について、そこに使われている手法を説明し、ビデオ会議への応用についての問題点について述べる。第 3 章では、既存の n 対 n でのデータ交換の手法とその問題点について述べ、本研究で提案する手法について、その動作や従来の手法との性能比較について述べる。第 4 章では、実インターネットを模倣したシミュレーションにて行った実験について、シミュレータの構成や、実験環境や実験方法、そして実験結果と考察を述べる。

第1章

ビデオ会議システム

インターネット回線の広帯域化とともに、動画配信やビデオ会議といった新しいアプリケーションが多数生み出された。本章では、ビデオ会議システムにおいて発生する n 対 n でのデータ交換について述べる。次に、現状で製品化までされている H.323 規格について述べる。最後に、マルチキャスト技術を用いた XCast6 について述べる。

1.1 n 対 n でのデータ交換



図1 ビデオ会議システム

ビデオ会議システムでは、まず図1のように各々の参加者がカメラやマイクを用いてビデオフレームデータや音声データを入力する。そのデータをネットワークを用いて全参加者が全参加者分のデータを取得し、各参加者はディスプレイに自身以外の人物の動画像を表示する。

データ交換後は図2のようになる。この n 対 n でのデータ交換について、ネットワークの帯域負荷が著しいため、様々な研究がされている。[1-4]

また、 n 対 n でのデータ交換を利用するアプリケーションとして他に

- 会議、ミーティング系



図2 n対nでのデータ交換後

- デザイン (複数の担当者による検討, 評価, 論議)
- 設計 (複数の担当者による検討, 評価, 論議)
- 芸術系 (造形, 美術, 音楽など)
- 体育 (実技指導など), 教育
- エンターテイメント
 - オンラインゲーム
 - イベント

などへの応用が想定されており, 特にオンラインゲームへの研究も盛んに行われている [5, 6].

1.2 H.323 について

H.323 規格 [7-9] は, 図 3 のような構成をしており, インターネットなどの IP ベースのネットワーク上で音声と映像とデータの通信を行うための基盤を提供している. マルチポイントビデオ会議の機能も有しており, 集中型と非集中型の概念を用いている.

1.2.1 集中型

集中型マルチポイント会議では, 図 4 のようなシステム構成となり, C/S 型のトポロジでのビデオ会議システムとなる. この構成は, マルチポイント会議を提供できるようにするための MCU (多地点接続装置) が必要である. 典型的な集中型のマルチキャストをサポートする MCU は, MC(マルチポイントコントローラ)と, 音声, 映像, データのそれぞれまたは組み合わせの

コラボレーションT.120	コンファレンスコントロール &コールシグナリング	オーディオ	ビデオ
TAVC T.126 T.127	H.245&H.225	G.711	H.261
T.124 T.122 T.125		G.728 G.722 G.729 G.723	H.263
X.224		RTP RTPC (H.223)	RTP RTPC (H.225)
ネットワーク：マルチキャスト，ポートアサインメンツ，etc			
TCP		UDP	
IP			
LAN			

図3 H.323の構成

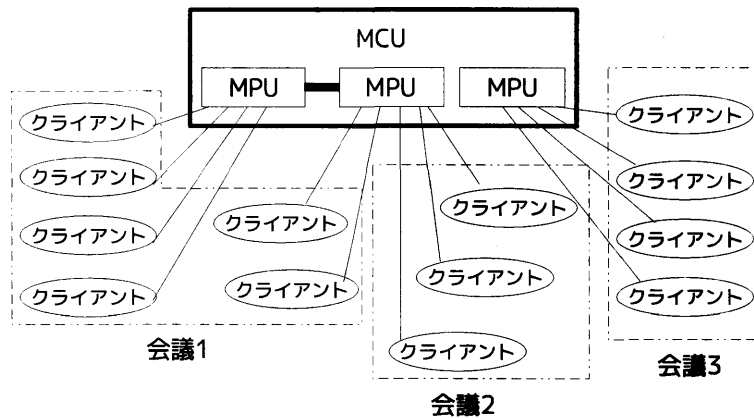


図4 集中型のシステム構成

MPU(マルチポイントプロセッサ装置)とで構成される。

全てのクライアントは、音声、映像、データ及び制御のストリームを、ポイント・ツー・ポイントの方式で、MCUに向けて送出する。MCは、個々のターミナルに対する機能も定義するH.245制御機能を用いて会議を集中的に管理する。

MPUは、マルチポイント会議で典型的に行われる音声ミキシング、データ分配、映像のスイッチングとミキシング処理を行い、結果のストリームを参加しているターミナルに送る。MPUは、異なるコードやビットレート間の変換を行ったり、マルチキャストによって処理した映像を分配する。

これをハードウェアを用いて実装したシステムに東芝のStrata™ VCSシステム [10]がある。

1.2.2 非集中型

非集中型マルチポイント会議では、マルチキャストの技術を利用する。参加している H.323 ターミナルは、MCU にデータを送らずに、他の参加しているターミナルに音声及び映像をマルチキャストする。ここでのマルチキャストには IP マルチキャストを用いる。

受信するターミナルは、送り込まれる複数の音声と映像のストリームを処理する責任がある。ターミナルは、H.245 コントロール・チャンネルを用いて、同時にデコードできる映像と音声のストリームの数を MC に知らせる。ある一つのターミナルが同時に処理できる能力の数が、会議においてマルチキャストされる映像または音声の数に制限を与えない。MP は、非集中型会議においても、映像を選択し、音声をミキシングすることが出来る。

1.2.3 ハイブリッド型

ハイブリッド・マルチ会議では、集中型と非集中型の機能の組み合わせを用いる。H.245 シグナルと、映像または音声ストリームが、MCU に向けたポイント・ツー・ポイントメッセージによって処理される。残りの信号（音声または映像）は、マルチキャストによって参加している H.323 のターミナルに向けて伝送される。

1.3 XCAST6

XCAST6[11] は XCAST というマルチキャスト技術を用いて実現する IPv6 用のビデオ会議システムである。XCAST については次章で記述するが、IP ヘッダのオプションを利用して対応ルータでパケットの複製を行う事により、IP マルチキャスト技術の問題点であったグループ管理などの煩雑さを解決し、送信コストの削減も可能としている。しかしながら対応ルータの普及が必要であること、IPv4 ではヘッダの容量から実装が出来ていない等の問題点がある。

第2章

マルチキャスト

マルチキャストとは、ネットワーク上の複数のホストに同時に同じデータを送信することであり、1対nでのデータ配信を効率良く行う技術を指す。マルチキャストを用いる事で、動画像などの大容量マルチメディアデータを効率的に配信できる。

本研究ではビデオ会議システムなどを想定したn対nでのデータ交換を目的としているが、前章のH.323の非集中型ではIPマルチキャストが使用されており、XCAST6ではXCASTが使用されている等、ビデオ会議システム等の複数参加者グループによるマルチメディアを用いたコミュニケーションにはマルチキャストは必須の技術となっている。

本章では、マルチキャストの手法ではないが、複数のホストに同時に同じデータを送る技術として最も広く用いられているクライアント・サーバモデルについて述べ、次に、マルチキャストの手法であるIPマルチキャストとXCASTの概要とそれらの問題点について述べる。

2.1 クライアント・サーバモデル

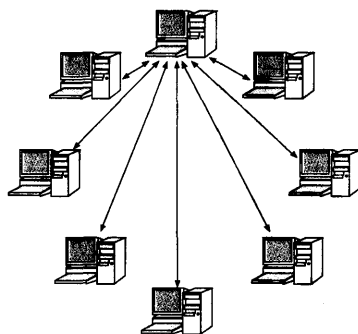


図5 クライアント・サーバモデル

クライアント・サーバモデルは、最も一般的なインターネットの利用形態である。図5のように、1台のサーバに複数台のクライアントが接続し、各クライアントの要求に応じてサーバからクライアントへユニキャストを使用してデータを送信する。

サーバが複数のクライアントから要求を受けた場合、同時に複数のクライアントへのデータ送信が生じるため、サーバの上り帯域や処理への負荷が大きい。例えば、図6のようにデータ配信サー

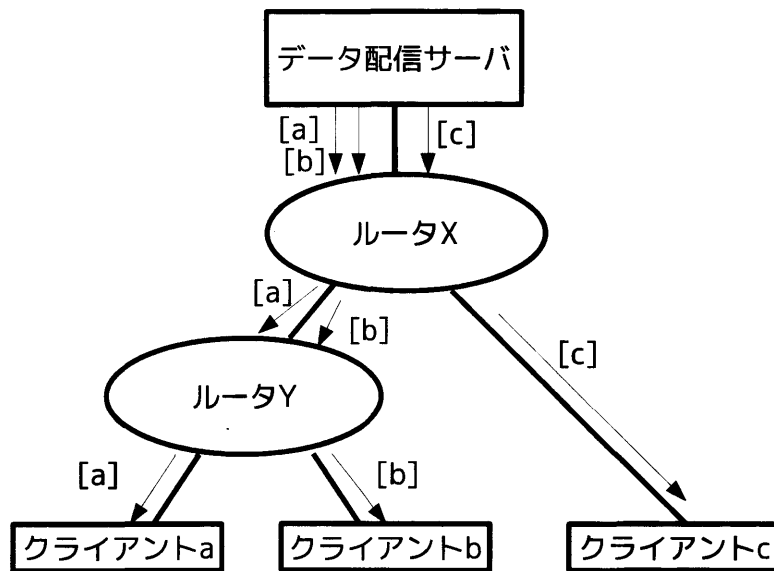


図6 クライアント・サーバモデルでのデータ配信

サーバがクライアント a,b,c に送信する場合、サーバとルータ X を結ぶパスには 3 回同じデータが流れ、ルータ X とルータ Y を結ぶパスには 2 回同じデータが流れる。このように、ユニキャストを用いるためにネットワークに冗長な伝送が生じるため、サーバやサーバ付近のルータには広帯域なネットワーク伝送幅や高い処理能力を求められる。また、例えそれらが満たされていても、クライアントの増大に対してスケラブルに処理出来ないという問題がある。

2.2 IP マルチキャスト

IP マルチキャストとは、ネットワーク層においてパケットを複製し、特定のマルチキャストアドレスを用いて特定グループにパケットを伝送する技術である。

ネットワーク層にてルータは複数の参加者を受け付ける。図 7 のようにルータはマルチキャストパケットを自身が管理する参加者やルータの全てに対して複製して送信する。

各ルータの参加者が分岐する地点でマルチキャスト木を構築するため、ネットワークを流れるトラフィックに一切の無駄を発生させない。

特定のマルチキャストアドレスを用いる事から、全ての経路途中のルータは IP マルチキャストの機能を有している必要がある。

現在ほとんどのルータに実装されてはいるが、インタードメインルーティングの困難さやアクセス制御やグループ管理の困難さといった問題から、企業内 LAN での用途には使えるが、家庭等の一般的なインターネット環境では使えない。

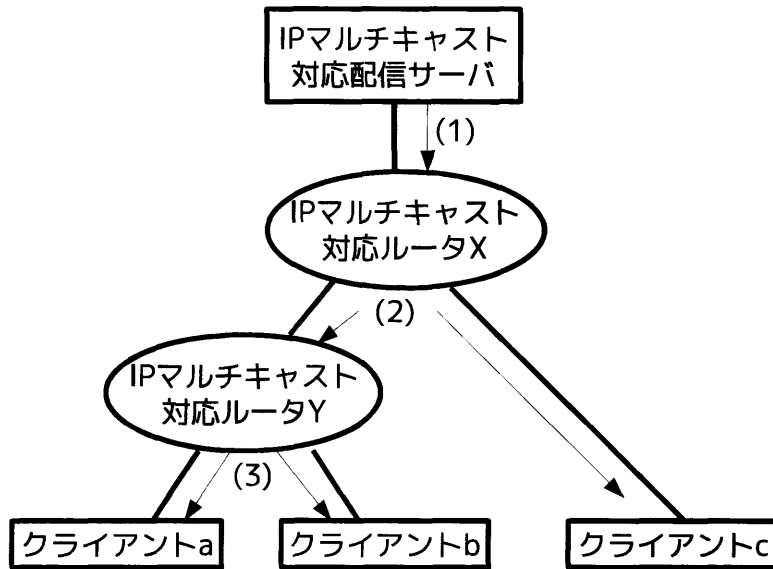


図7 IP マルチキャストによるパケットの流れ

2.3 XCAST

XCAST(eXplicit Multi-Unicast) とは、IP ヘッダに宛先リストを保持し、図 8 のように対応ルータがパケットを複製しながら、リストの受信者全員に配送する方法である。

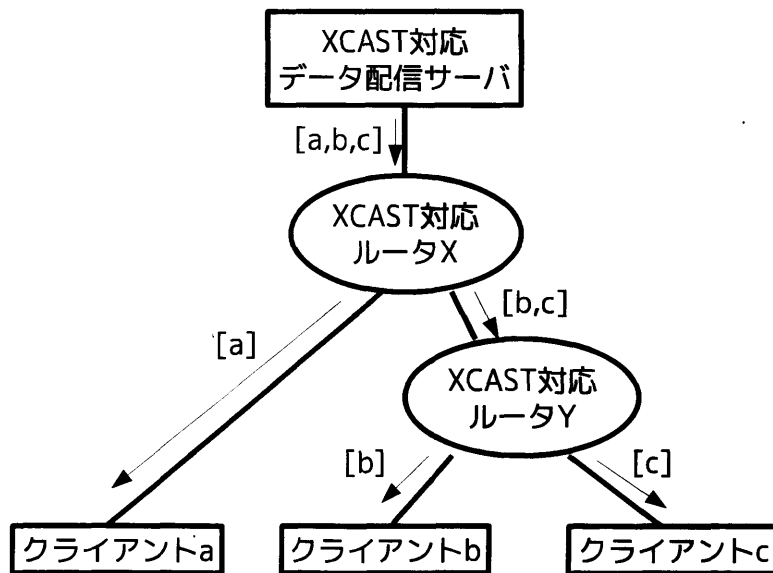


図8 XCAST パケットの複製

宛先欄には宛先リストから任意に取り出した1つのIPアドレスを保持する経路上のXCAST対応ルータでは宛先リスト全てに経路を算出し、経路が同じ宛先をまとめて新たにXCASTパケットを生成する。よって図8において宛先a,b,cを受信したルータXは宛先aについては宛先がaのパケットをクライアントaに送信し、宛先b,cについては、宛先リストにb,cを入れたXCASTパケットを新たに生成してルータYに送信する。経路上のXCAST非対応ルータでは宛先欄に入ったIPアドレスを用いて通常のIPパケットとして処理する。IPマルチキャストは経路中の全てのルータが対応していないと使えないが、XCASTは宛先リストをIPヘッダのオプション領域に持ち、宛先IPアドレスにはそのうちの一つを選択して付けるため、XCASTパケットはユニキャストにしか対応していないルータでも通過出来る。この時、XCAST非対応ルータは宛先のほうに送信するが、その先のXCAST対応ルータが改めて分岐させる。

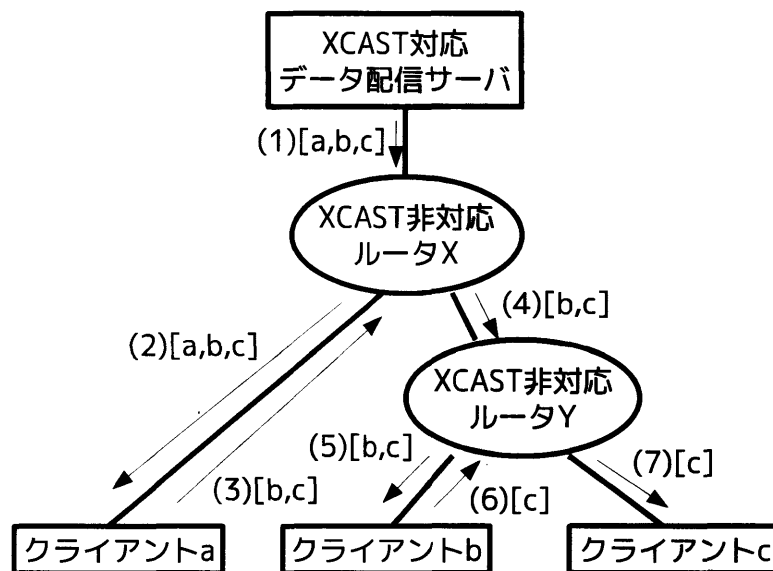


図9 非対応ルータによるXCASTパケットの流れ

経路上のルータの全てがXCAST対応ルータならば、この方式でIPマルチキャストと同様のパケット伝送効率が見込めるが、逆に経路上のルータの全てがXCAST非対応ルータならば、図9のように、クライアント間でパケットをユニキャストによってパケットリレー方式で伝送するのと変わらず、データ配信サーバから各クライアントへユニキャストを複数生成して伝送するよりも効率が落ちる場合がある。

技術の総称としてXCAST、IPv4での実装がXCAST4、IPv6での実装がXCAST6という。しかし、実際にはIPヘッダのオプション領域の不足等からIPv4用の実装であるXCAST4は実装されておらず、存在しない。ドラフトにはXCAST4の仕様も書かれているが、仕様レベルのみの話

である。

次節で述べるアプリケーション層マルチキャストとは違い、経路途中のネットワーク層でパケットの複製を行っているので、ネットワークの通信トラフィックの減少にとっても効果的である。しかしながら、対応ルータが十分に無ければ、結局は末端の受信者ホストが新たな宛先へ向けてパケットの複製を行わなければならないので、手法の効果は少なくなる。また、オプション領域の不足から、IPv4での実装例は無い。その点では、汎用性の面でもアプリケーション層マルチキャストの方が優れている。

2.4 アプリケーション層マルチキャスト

アプリケーション層マルチキャストとは、アプリケーション層でパケットの複製を行い効率的にデータ配信を行う技術である。アプリケーション層での論理リンクでマルチキャストツリーを構築して、根から葉へ向かって節でデータを複製しながら配信していくツリー型である OMNI[12] や、バッファ、ビットマップといった概念を用いて複数の隣接ノードへデータ配信要求をし、同時に複数ノードからデータ受信をしていくデータ駆動型である CoolStreaming[13] 等がある。前述したように、IP マルチキャストは経路上に対応したルータを必要とし、XCAST は経路上に非対応のルータが含まれていると最適な性能が発揮できず、場合によってはクライアント・サーバモデルよりも性能が落ちてしまう。よって本研究で想定するインターネットを用いた家庭内の個人で使用するビデオ会議システムにはアプリケーション層での論理ネットワークによる配信技術が必須となる。しかしながら既存研究では1つの配信ノードからグループ全員への配信をいかに効率良く行うかについての研究が盛んであるが、ビデオ会議システムのように n 対 n でのデータ交換という見識からの研究はほとんどされていない。

第3章

提案手法

前章では、ビデオ会議システムを実現するための重要な技術であるマルチキャストについて述べた。既存技術の IP マルチキャストや XCAST にはネットワーク上の機器が対応する必要があり、本研究の想定する家庭内の個人によるビデオ会議システム等のアプリケーションは実現できない。また、アプリケーション層マルチキャストにおいては、既存研究では1つの配信ノードからグループ全員への配信をいかに効率良く行うかについての研究が盛んであるが、ビデオ会議システムのように n 対 n でのデータ交換という見識からの研究はほとんどされていない。

本章では一般的にビデオ会議システムをインターネット上で行う際の従来手法の問題点を明確にし、その改善案として新たな手法を述べる。まず P2P ネットワークを用いた手法について述べ、次にマルチキャスト木を利用した手法について述べる。

3.1 従来手法の問題点

本研究では一般的な家庭のインターネット環境にいる参加者同士でのビデオ会議コミュニケーションを想定している。従来手法では主に、クライアント・サーバ型でのシステムを用いてきた。これはクライアント・サーバ型の論理ネットワークを参加者同士で構築し、中央のサーバ役ノードへ周りのクライアント役ノードからビデオフレームデータを送信し (図 10(a))、その後、中央のサーバ役ノードから周りのクライアント役ノードへ集まったデータを送信していく (図 10(b)) というものである。つまり、第一章で述べた H.323 の集中型における MCU を参加者の中の代表者がアプリケーション層で担っているといえる。

この方式では一度のデータ交換について、クライアント役ノードはサーバ役ノードへ自身の入力値データを送り、その後サーバ役ノードから自身以外の端末の入力値データを受け取る。よって参加端末数を n とした時、クライアント役ノードのデータ送受信回数は $O(1)$ であり、データ送受信量は $O(n)$ である。しかしサーバ役ノードはクライアント役ノードの全てからデータを受け取り、クライアント役ノードへそのノード以外のデータを送信するため、データ送受信回数は $O(n)$ となり、データ送受信量は $O(n^2)$ となるため、送受信に対する負荷がサーバ役ノードに集中しているという問題点がある。

会議の平等性を確保するためには、データの交換成立時刻をなるべくユーザ間でそろえる必要がある。よって、一番コストの大きい参加者に合わせて行わ

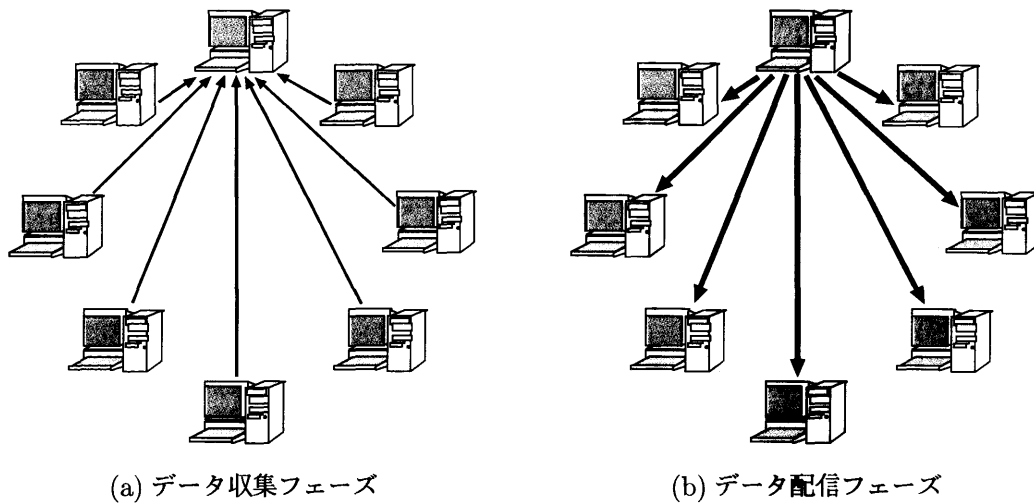


図 10 従来手法

れるので、負荷が集中する事は全体のパフォーマンス低下につながる。

以上の事から、この手法では大人数同時参加の実現は難しい。

3.2 P2P ネットワークを用いた手法

本節で述べる手法では、従来の手法においての問題点となったサーバ役端末に負荷が集中する事を解決する手法について述べる。まず論理ネットワークのトポロジについて述べ、その動作アルゴリズムについて述べる。また、これらから推測される性能評価も行う。

3.2.1 論理ネットワークトポロジ

提案手法では図 11 のような P2P 型ネットワークをユーザ同士で構築する。図中の矢印はその方向へデータ送信が出来る状態に接続が確立している事を示している。この論理ネットワークの構築により負荷が参加者全員に分散され、多数のユーザが参加してもシステムに支障の無い通信時間を実現する。

具体的なアルゴリズムとして、参加ノード数が $n=2^m$ の時、各ノードは $0 \sim n-1$ のユニークな ID 番号を持ち、自身と ID 空間上で 2 のべき乗離れたノードと接続を確立している。

なお、ノード数が 2 のべき乗でない場合についてはこの論理ネットワークを構成できないが、後に述べる手法にて解決している。

3.2.2 動作

各端末の動作について述べる。各端末はそれぞれ以下の動作を並列に実行し、データ交換を実現する。

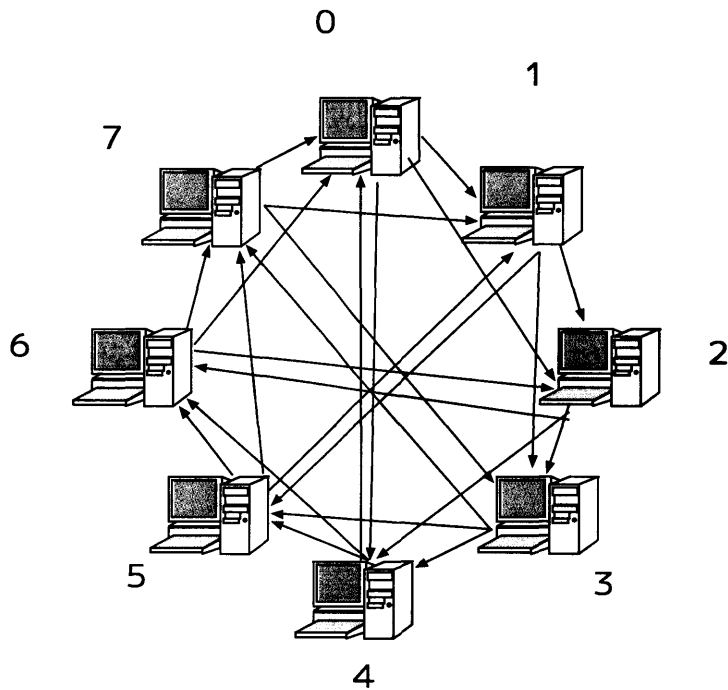


図 11 提案手法のネットワーク

1. 自身のデータを作成 or 更新する
2. i を $m-1$ から 0 まで以下の処理を行う
 - (a) 自身が得ているデータ全てを ID 空間上で 2^i 前方の端末に送信する
 - (b) ID 空間上で 2^i 後方の端末からデータを受信する
3. 外部プログラムに揃ったデータを渡し、1へ

ただし、2 のべき乗以外のノード数の場合は、適当な端末が 2 ノード分の通信を代行する事とする。また、外部プログラムとはデータ交換を利用するアプリケーションの事であり、ビデオ会議システム等を指す。

この方式では、全ての端末は上記アルゴリズムより送受信回数が $O(\log n)$ となる。また、データ送受信量においても動作アルゴリズムの 2 において 1 回目はデータ 1 個、2 回目はデータ 2 個、 m 回目はデータ 2^{m-1} 個の送受信を行っているので、合計は $1 + 2 + \dots + 2^{m-1} (= \frac{n}{2}) = n - 1$ となり、 $O(n)$ を実現している。表 1 に比較表を記す。

表1 従来手法との比較

	従来手法 (クライアント役端末)	従来手法 (サーバ役端末)	提案手法
送受信回数	$O(1)$	$O(n)$	$O(\log n)$
送信データ量	$O(1)$	$O(n^2)$	$O(n)$
受信データ量	$O(n)$	$O(n)$	$O(n)$

3.2.3 動作例

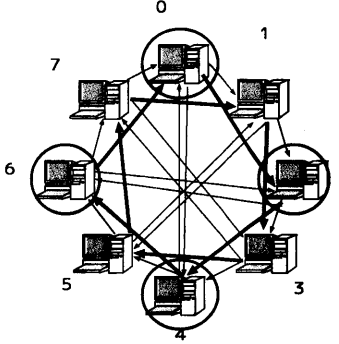
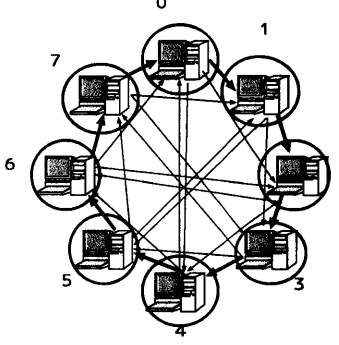
以下では、前述した提案手法による動作例を述べる。例として参加端末数 $n = 8 (= 2^3)$ の場合を挙げる。

ページ左部の動作ではその時点で行っている動作を述べる。

ページ中央部のデータ [0] を保持する端末ではその時点で ID 番号 0 の端末が生成したデータを保持している端末に丸を付けていき、さらにその時点でデータが通る論理線を太く描く。ID 番号 0 の端末が全ての端末のデータを揃えた時、その他の ID 番号を持つ端末も同様にデータが揃うという事に注意していただきたい。

ページ右部の端末\保持データでは各端末がその時点で保持するデータに丸を付けていく。

動作	データ [0] を保持する端末	端末\保持データ																																																																																	
<p>1. 自身のデータを作成 or 更新する</p>		<table border="1"> <thead> <tr> <th></th> <th>0</th> <th>1</th> <th>2</th> <th>3</th> <th>4</th> <th>5</th> <th>6</th> <th>7</th> </tr> </thead> <tbody> <tr> <th>0</th> <td>○</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <th>1</th> <td></td> <td>○</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <th>2</th> <td></td> <td></td> <td>○</td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <th>3</th> <td></td> <td></td> <td></td> <td>○</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <th>4</th> <td></td> <td></td> <td></td> <td></td> <td>○</td> <td></td> <td></td> <td></td> </tr> <tr> <th>5</th> <td></td> <td></td> <td></td> <td></td> <td></td> <td>○</td> <td></td> <td></td> </tr> <tr> <th>6</th> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td>○</td> <td></td> </tr> <tr> <th>7</th> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td>○</td> </tr> </tbody> </table>		0	1	2	3	4	5	6	7	0	○								1		○							2			○						3				○					4					○				5						○			6							○		7								○
	0	1	2	3	4	5	6	7																																																																											
0	○																																																																																		
1		○																																																																																	
2			○																																																																																
3				○																																																																															
4					○																																																																														
5						○																																																																													
6							○																																																																												
7								○																																																																											
<p>2.(a) 自身が得ているデータ全てをID空間上で2^2前方の端末に送信する</p> <p>2.(b) ID空間上で2^2後方の端末からデータを受信する</p>		<table border="1"> <thead> <tr> <th></th> <th>0</th> <th>1</th> <th>2</th> <th>3</th> <th>4</th> <th>5</th> <th>6</th> <th>7</th> </tr> </thead> <tbody> <tr> <th>0</th> <td>○</td> <td></td> <td></td> <td></td> <td>○</td> <td></td> <td></td> <td></td> </tr> <tr> <th>1</th> <td></td> <td>○</td> <td></td> <td></td> <td></td> <td>○</td> <td></td> <td></td> </tr> <tr> <th>2</th> <td></td> <td></td> <td>○</td> <td></td> <td></td> <td></td> <td>○</td> <td></td> </tr> <tr> <th>3</th> <td></td> <td></td> <td></td> <td>○</td> <td></td> <td></td> <td></td> <td>○</td> </tr> <tr> <th>4</th> <td>○</td> <td></td> <td></td> <td></td> <td>○</td> <td></td> <td></td> <td></td> </tr> <tr> <th>5</th> <td></td> <td>○</td> <td></td> <td></td> <td></td> <td>○</td> <td></td> <td></td> </tr> <tr> <th>6</th> <td></td> <td></td> <td>○</td> <td></td> <td></td> <td></td> <td>○</td> <td></td> </tr> <tr> <th>7</th> <td></td> <td></td> <td></td> <td>○</td> <td></td> <td></td> <td></td> <td>○</td> </tr> </tbody> </table>		0	1	2	3	4	5	6	7	0	○				○				1		○				○			2			○				○		3				○				○	4	○				○				5		○				○			6			○				○		7				○				○
	0	1	2	3	4	5	6	7																																																																											
0	○				○																																																																														
1		○				○																																																																													
2			○				○																																																																												
3				○				○																																																																											
4	○				○																																																																														
5		○				○																																																																													
6			○				○																																																																												
7				○				○																																																																											

<p>3.(a) 自身が得ているデータをID空間上で2^1前方の端末に送信する</p> <p>3.(b) ID空間上で2^1後方の端末からデータを受信する</p>		<table border="1"> <thead> <tr> <th></th> <th>0</th> <th>1</th> <th>2</th> <th>3</th> <th>4</th> <th>5</th> <th>6</th> <th>7</th> </tr> </thead> <tbody> <tr> <th>0</th> <td>○</td> <td></td> <td>○</td> <td></td> <td>○</td> <td></td> <td>○</td> <td></td> </tr> <tr> <th>1</th> <td></td> <td>○</td> <td></td> <td>○</td> <td></td> <td>○</td> <td></td> <td>○</td> </tr> <tr> <th>2</th> <td>○</td> <td></td> <td>○</td> <td></td> <td>○</td> <td></td> <td>○</td> <td></td> </tr> <tr> <th>3</th> <td></td> <td>○</td> <td></td> <td>○</td> <td></td> <td>○</td> <td></td> <td>○</td> </tr> <tr> <th>4</th> <td>○</td> <td></td> <td>○</td> <td></td> <td>○</td> <td></td> <td>○</td> <td></td> </tr> <tr> <th>5</th> <td></td> <td>○</td> <td></td> <td>○</td> <td></td> <td>○</td> <td></td> <td>○</td> </tr> <tr> <th>6</th> <td>○</td> <td></td> <td>○</td> <td></td> <td>○</td> <td></td> <td>○</td> <td></td> </tr> <tr> <th>7</th> <td></td> <td>○</td> <td></td> <td>○</td> <td></td> <td>○</td> <td></td> <td>○</td> </tr> </tbody> </table>		0	1	2	3	4	5	6	7	0	○		○		○		○		1		○		○		○		○	2	○		○		○		○		3		○		○		○		○	4	○		○		○		○		5		○		○		○		○	6	○		○		○		○		7		○		○		○		○
	0	1	2	3	4	5	6	7																																																																											
0	○		○		○		○																																																																												
1		○		○		○		○																																																																											
2	○		○		○		○																																																																												
3		○		○		○		○																																																																											
4	○		○		○		○																																																																												
5		○		○		○		○																																																																											
6	○		○		○		○																																																																												
7		○		○		○		○																																																																											
<p>4.(a) 自身が得ているデータをID空間上で2^0前方の端末に送信する</p> <p>4.(b) ID空間上で2^0後方の端末からデータを受信する</p>		<table border="1"> <thead> <tr> <th></th> <th>0</th> <th>1</th> <th>2</th> <th>3</th> <th>4</th> <th>5</th> <th>6</th> <th>7</th> </tr> </thead> <tbody> <tr> <th>0</th> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <th>1</th> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <th>2</th> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <th>3</th> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <th>4</th> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <th>5</th> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <th>6</th> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <th>7</th> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> </tr> </tbody> </table>		0	1	2	3	4	5	6	7	0	○	○	○	○	○	○	○	○	1	○	○	○	○	○	○	○	○	2	○	○	○	○	○	○	○	○	3	○	○	○	○	○	○	○	○	4	○	○	○	○	○	○	○	○	5	○	○	○	○	○	○	○	○	6	○	○	○	○	○	○	○	○	7	○	○	○	○	○	○	○	○
	0	1	2	3	4	5	6	7																																																																											
0	○	○	○	○	○	○	○	○																																																																											
1	○	○	○	○	○	○	○	○																																																																											
2	○	○	○	○	○	○	○	○																																																																											
3	○	○	○	○	○	○	○	○																																																																											
4	○	○	○	○	○	○	○	○																																																																											
5	○	○	○	○	○	○	○	○																																																																											
6	○	○	○	○	○	○	○	○																																																																											
7	○	○	○	○	○	○	○	○																																																																											

以上のような動作で、全ての端末が全ての端末のデータをそろえる事ができる。また、通信回数は $O(\log n)$ を実現している。データ [0] を保持する端末から分かるように、ID 番号が 0 の端末のデータが他の全ての端末に届いている。これは、端末-保持データからも分かるように、ID 番号が 0 以外の端末についても同様の事が言え、全ての端末のデータが全ての端末に届いている。

3.2.4 性能

ビデオ会議システムでは、フレームデータのようにサイズの大きなデータを頻繁に送りあう事となる。よって、データ交換にかかる所要時間は送受信量が主な基準となると予測される。また、平等な会議のためには参加者間で遅延の差がなるべく生じないように同期を取る必要があり、全ての

端末がデータ交換を完了するまで処理は停止する事となる。つまり、負荷が特定の端末に集中している場合、その端末が性能の基準となる。

従来手法では、負荷がサーバ役端末に集中している。サーバ役端末は全てのクライアント端末 ($n-1$ 台) からのデータを受信し、その後全てのクライアント端末 ($n-1$ 台) へ $n-1$ 個のデータを送信する。よってデータ送信量は $O(n^2)$ 、データ受信量は $O(n)$ となり、これがデータ交換所用時間の値となる。

提案方式では、前述したアルゴリズムや動作例から分かるように、全ての端末においてデータ送受信量は $O(n)$ となり、これがデータ交換所要時間の値となる。

しかし、端末数が 2 のべき乗でない場合、後述する代行ルーティング法かハイブリッド法のいずれかの方式での動作を行う必要がある。この際の性能についてはそれぞれの手法とともに次節で述べる。

3.2.5 ノード数が 2 のべき乗以外の場合

前節までで提案手法である P2P ルーティング法についてトポロジや動作について、ノード数が 2 のべき乗である事を前提として概要を述べた。では 2 のべき乗以外の場合はどうするかという問題について、本研究では以下の 2 つの方法を提案し、次章の実験にて評価している。本節では提案している方式を記述する。

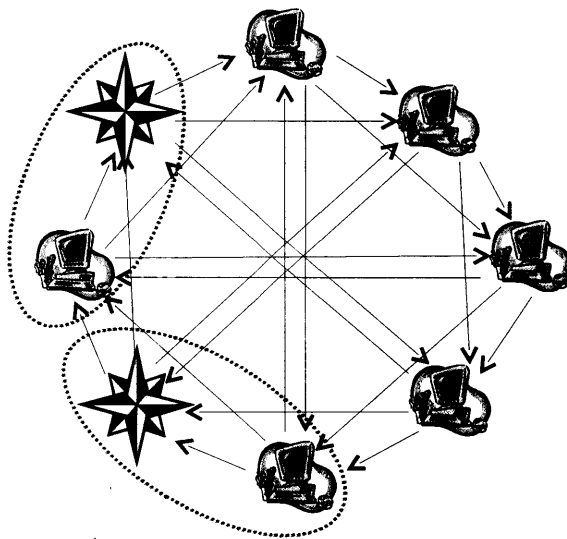


図 12 代行ルーティング法

■代行ルーティング法 代行ルーティング法では図 12 のように 2 のべき乗に足りない部分に対し、1 台のホストが 2 ノード分の通信を代行する事によって論理的に 2 のべき乗のノード数とし

て運用する方式である。1台のホストが2ノード分の通信を代行するので、代行ルーティング法と呼ぶ。

代行ルーティング法を採用した場合、代行通信プロセスは、ネットワーク全体のプロセス数が2のべき乗となるように生成されるので、参加者数が $2^m + 1 \sim 2^{m+1}$ の間において、同じ性能を出す。また、代行プロセスを生成した端末は自身の通信と代行通信により、データ送受信量は $O(n)$ となるが、実際には2倍の送受信量を必要とし、これが所要時間となる。

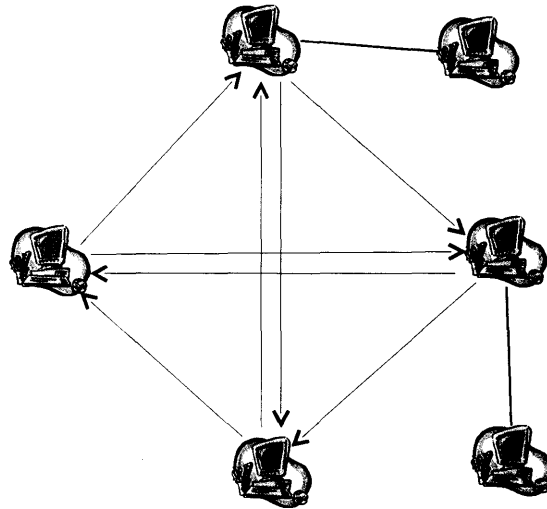


図 13 ハイブリッド法

■ハイブリッド法 ハイブリッド法では図 13 のように 2 のべき乗で提案方式のトポロジを構成し、端数とそのトポロジのノードに接続する。その際の動作は、まず端数となったノードからそのノードに接続されたノードにデータを送信する。そしてべき乗のトポロジを構成するノードは前述した動作でデータ交換をする。その時、端数が接続されているノードは自信のデータと端数者から受け取ったデータをセットにして扱う。べき乗のトポロジでのデータ交換が終了した後に、端数者が接続されたノードは、その端数者に揃ったデータを送信する。この方式は端数者と、端数者が接続されたノードの間でクライアント・サーバ方式の関係を取るなのでハイブリッド法と呼ぶ。

ハイブリッド法を採用した場合、データ送受信量は $O(n)$ となるが、端数者が接続されたノードはデータ送信量がべき乗のトポロジを構成する他のノードに比べて2倍となる。また、データ受信のタイミングについて端数者とそれ以外で差が出る。

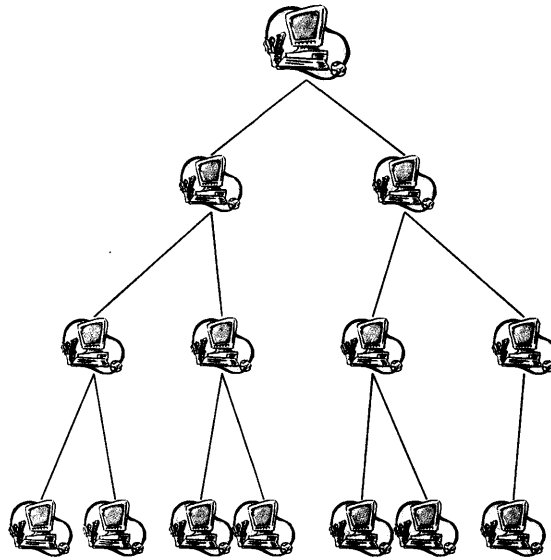


図 14 マルチキャスト木手法のトポロジ

3.3 マルチキャスト木を用いた手法

参加者同士で図 14 共有の完全二分木を作って利用する。根ノードは 2 台の子ノードと論理リンクを持ち、節ノードは 1 台の親ノードと論理リンクを持ち、2 台以下の子ノードとの論理リンクを持つ。葉ノードは 1 台の親ノードを持つ。

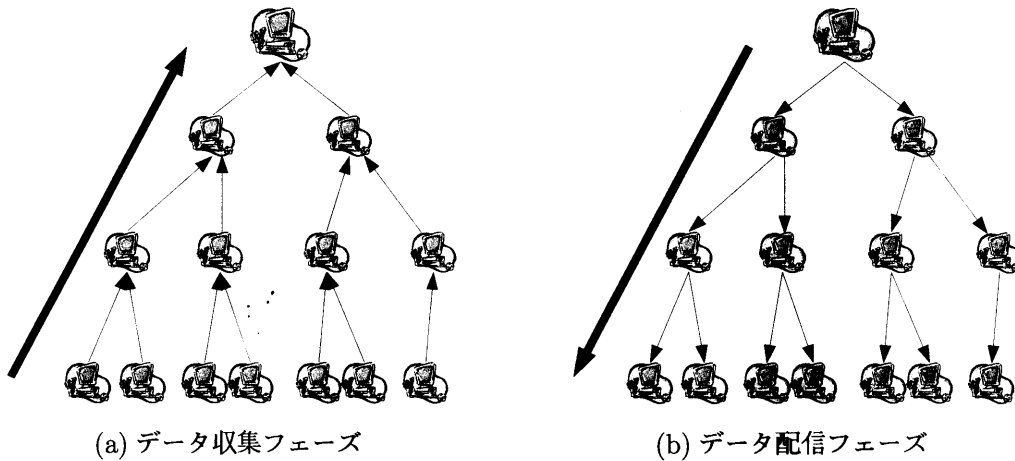


図 15 マルチキャスト木手法でのデータ交換

まず、図 15-(a) のように、まず根へ向かってデータを集める。葉ノードは自身のデータを自身の親ノードへ送信する。節ノードは子ノードから受信したデータと、自身のデータとを合わせて自

身の親ノードへ送信する。根ノードは子ノードからデータを受信する。この手順により、根ノードには全てのノードのデータが集まる。

次に、図 15-(b) のように、葉に向かってデータを配る。根ノード、節ノードは、自身の子ノードが持っていないデータを送信していく。

完全二分木でマルチキャストツリーを構築するので、各々のノードのデータ送信量は最悪でも参加者数の 2 倍のデータで済み、 $O(n)$ となる。

しかしながら、根ノードから末端ノードまでに $O(\log n)$ のホップ数を要する。

シミュレータ実験において実ネットワークのバックボーンモデルは図 16 を用いる。これは中心 3 台を IX ルータと見立て 1Gbps で相互接続し、それぞれに 3 台ずつ一次プロバイダと見立てたルータを 100Mbps で接続し、さらにそれぞれに二次プロバイダと見立てたルータを 100Mbps で接続したものである。

実ネットワークのバックボーンモデルの構築には文献 [14] による Power-Law 特性に考慮しており、さらに文献 [15] に記述されている 4 つの特徴を用いてネットワークモデルの評価を行っている。以下に用いられる特徴について記述する。

Node num

ネットワークにおけるノード数。実ネットワークのモデルならば、ルータの数に対応する。

Average degree

各ノードの次数の平均である。ノード数を n 、辺の数を m としたときの $2m/n$ であり、葉では次数が 1 となる。実ネットワークのモデルに適用する場合は、各ルータに接続されたパスの数の平均である。

Diameter

任意のノードから他の全てのノードへの最短パスを構築した際の直径である。評価では、ネットワーク内で最も大きい直径の値を参照している。

Number of biconnected components

biconnected components とは、どの 2 つの辺も共有のシンプルなサイクルに含まれる辺の最大セットである。この数値は、グラフの接続性 (connectedness) や冗長性 (redundancy) の度合を計るものである。biconnected components の数を評価値としており、この数値がノード数に近くなるほど、ネットワークに冗長性が無い事を示している。

これらの特徴を用いられた理由は 2 つある。グラフのいくつかの局面から概要を測れるようにし、比較的評価を簡単にするためである。これらは独立してはなく、特にノード次数の平均が増加するのに関連して、直径の減少と、少ない bicomponents が見られる。

文献 [15] で記されている広域ネットワークモデル (CERFnet, Sesquinet) とシミュレータのバックボーンに用いたモデルに対してこれらの評価基準を使って評価したものを表 2 に載せる。

表から分かるように、シミュレータに使用したモデルは、実際の広域ネットワークである CERFnet と近い性質を持っており、このモデルを用いたシミュレータ実験により実ネットワークでの挙動を模倣できると考えられる。

表2 ネットワークモデルの性質

Network	Nodes	Avg Deg(*1)	Diam(*2)	Bicomp(*3)
CERFnet(regional networks)	63	2.1	5	59
Sesquinet(regional networks)	94	2.1	9	77
自作シミュレータ	63	2.0	5	60

(*1)Avg Deg : 各ノードに接続されたパス数の平均

(*2)Diam : 最大直径ホップ数

(*3)Bicomp : グラフ中の Biconnected components の数

4.1.2 シミュレーション動作

本実験のシミュレータでは、逐次的な処理によりパケットの伝送をネットワーク層のレベルでシミュレートしている。UDP 通信を想定しており、トランスポート層については何も処理しない。アプリケーション層において各手法の動作を行い、実際にデータが交換されるまでの時間を計測する。

ネットワーク層では、ルータはパケットを一旦蓄積してから宛先 IP アドレスからルーティングテーブルを参照して隣接するルータへ転送する。ルーティングアルゴリズムには RIP を採用しており、回線の速度に関係無くホップ数によって次に送る隣接ルータを決定している。ルータからルータへの転送はデータサイズ/帯域の遅延を加えるとともに、ルーティング遅延として 2ms の固定値の遅延を与える。パケットの宛先が自身宛であった場合は、アプリケーション層で各手法の処理を実行する。

4.1.3 実験方法

実験では、まずインターネットのバックボーンの二次プロバイダに見立てたルータをランダムに決め、各手法を実行するノードを接続する。それを任意のノード数だけ行い、各手法におけるデータ交換をシミュレータの時間で 3600 秒になるまで実行する。データ交換が何回行われたかを計測し、最後に 3600 秒/データ交換成立回数の計算によって、データ交換 1 回当たりにかかった時間を求める。

以降では、

- ノード数変化における実験
- 上り幅帯域変化における実験
- 交換するデータ量を変えた実験

の3種類の実験を行い、その結果と考察を述べる。

4.2 ノード数変化における実験

ノード数の変化に対して、各手法におけるデータ交換に必要な時間にどのような変化があるかを実験を行った。

4.2.1 実験方法

各手法において、各ノードがビデオフレームデータを想定した1KBのデータを生成し、それをそれぞれの手法を用いて全ノードが共有するまでにかかった時間を測定している。C/S routingは従来手法におけるクライアントサーバ型、P2P routingは提案手法において代行ルーティング法を採用した場合、Hybrid routingは提案手法においてハイブリッド法を採用した場合、MCastTree routingは完全二分木のマルチキャスト木を利用したアルゴリズムの場合の結果を表している。

4.2.2 実験結果 (上り 1Mbps, 下り 10Mbps)

ADSL回線の実効速度を想定した速度で実験を行った。結果を図17, 18に載せる。

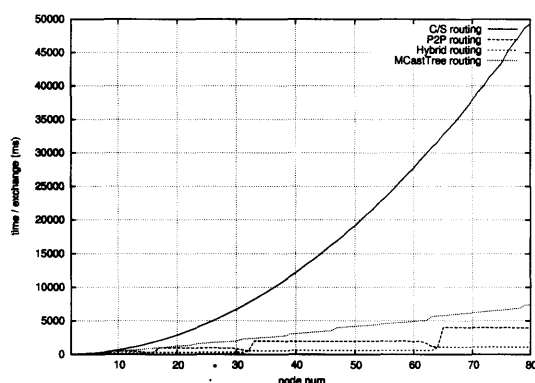
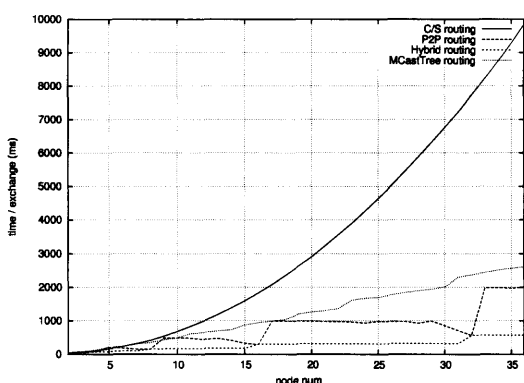


図17 実験結果:上り 1M, 下り 10M, 2~36 ノード 図18 実験結果:上り 1M, 下り 10M, 2~80 ノード

従来のC/S型の手法ではノード数に対するデータ交換時間は $O(n^2)$ で増大してしまっており、ビデオ会議システムでの参加者数増大に耐え得ない。提案手法のマルチキャスト木手法では $O(n)$ となっており、従来手法に比べ高いスケーラビリティを示す。提案手法のP2P型ルーティングの代行ルーティング法では、手法の特徴である2の冪乗+1ノードでの論理ネットワークの階層レベルの増大に合わせてデータ交換にかかる時間が増大しており段差の軌跡を描いている。段差の頂点を結ぶと $O(n)$ であり、従来手法よりも高いスケーラビリティを示している事が分かる。提案手法のP2P型ルーティングのハイブリッド法では $O(n)$ の軌跡を描いており、手法の中で一番良い結果となっている。代行ルーティング法でのオーバーヘッドがうまく取り除かれて高速化につながった

と思われる。以上の事から提案手法は従来に比べオーダの次元を削減するほどの高いスケーラビリティを持っており、さらに P2P ルーティングのハイブリッド法が一番良いスループットを出す事が分かった。

4.2.3 実験結果 (上り 2Mbps , 下り 10Mbps)

ADSL 回線の比較的良好な状況での実効速度を想定した速度で実験を行った。結果を図 19, 20 に載せる。

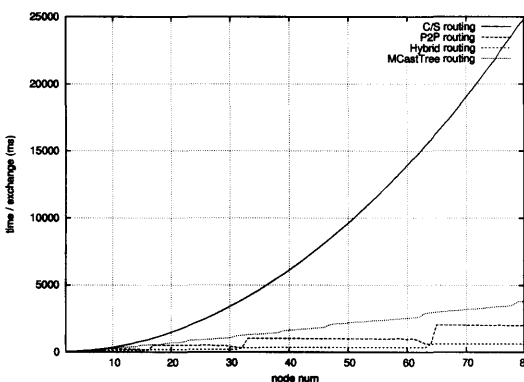
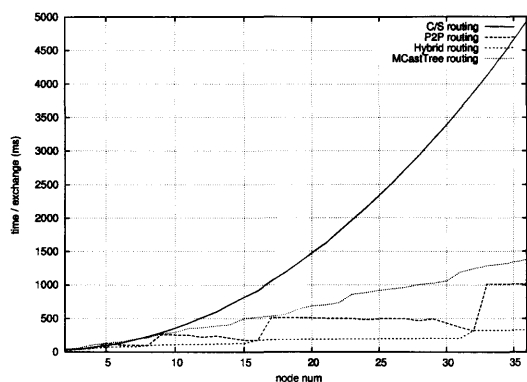


図 19 実験結果:上り 2M, 下り 10M, 2~36 ノード 図 20 実験結果:上り 2M, 下り 10M, 2~80 ノード

上り回線が 1Mbps の時と比べ、データ交換にかかる時間が約半分となっている。各手法のスループットがほとんど上り帯域に影響されていた事が分かる。

4.2.4 実験結果 (上り 5Mbps , 下り 40Mbps)

ADSL 回線のベストエフォートとされる速度で実験を行った。結果を図 21, 22 に載せる。

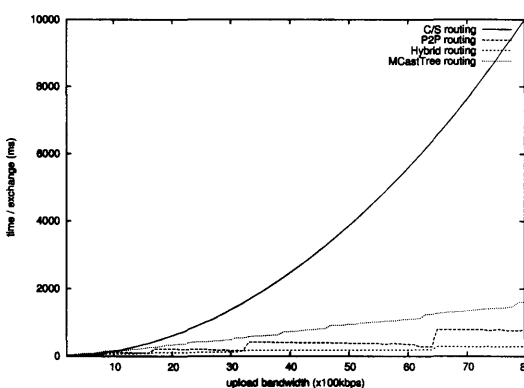
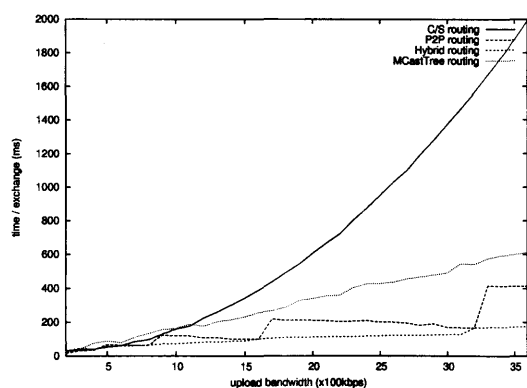


図 21 実験結果:上り 5M, 下り 40M, 2~36 ノード 図 22 実験結果:上り 5M, 下り 40M, 2~80 ノード

上り回線が 1Mbps, 2Mbps の時と比べ、データ交換にかかる時間が減少している。しかしながら各手法のスループットは相対的にはあまり変化しない。

4.2.5 実験結果 (上り 100Mbps, 下り 100Mbps)

光回線のベストエフォートとされる速度で実験を行った。結果を図 23, 24 に載せる。

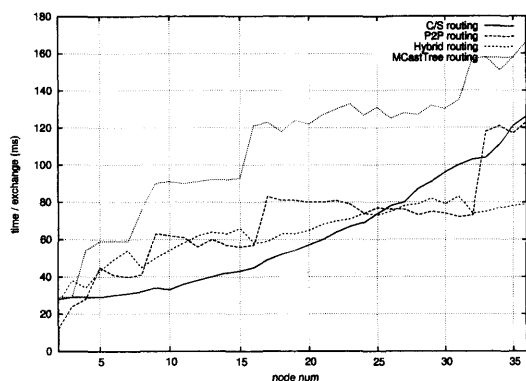


図 23 実験結果:上り 100M, 下り 100M, 2~36 ノード

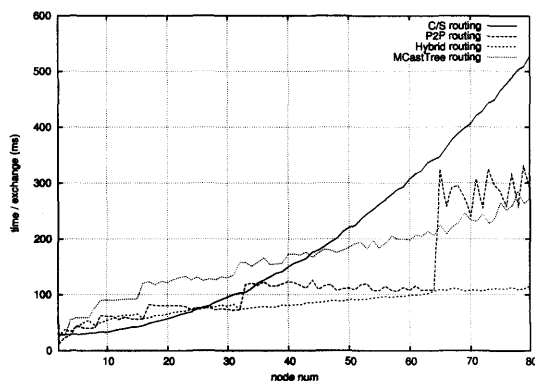


図 24 実験結果:上り 20M, 下り 100M, 2~80 ノード

ノード数が 24 ぐらいまでは従来手法である C/S 型が一番良い結果を出している。上り帯域が広い場合は P2P を用いたネットワークによって上り帯域の負荷を各ノードで分散させるよりも、C/S 型でホップ数を少なくした方が有効である事が分かる。

4.3 上り帯域幅変化における実験

前節での実験により、データ交換が各ノードの上り帯域幅に多大な影響を受ける事が分かった。本節では上り帯域幅の変化に対して、各手法におけるデータ交換に必要な時間がどのように変化するか実験を行った。

4.3.1 実験方法

各手法において、各ノードがビデオフレームデータを想定した 1KB のデータを生成し、それをそれぞれの手法を用いて全ノードが共有するまでにかかった時間を測定している。C/S routing は従来手法におけるクライアントサーバ型、P2P routing は提案手法において代行ルーティング法を採用した場合、Hybrid routing は提案手法においてハイブリッド法を採用した場合、MCastTree routing は完全二分木のマルチキャスト木を利用したアルゴリズムの場合の結果を表している。

4.3.2 実験結果 (12 ノード, 下り 10Mbps)

12 ノードに固定し, 上り帯域幅を変化させながらデータ交換にかかった時間を測定した。結果を図 25, 26 に載せる。

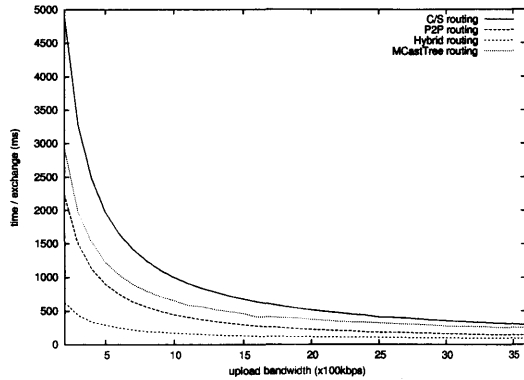


図 25 実験結果:12 ノード, ~3.5Mbps

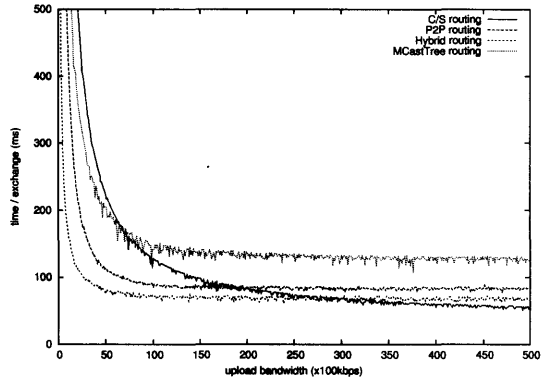


図 26 実験結果:12 ノード, ~50Mbps

提案手法である P2P ルーティング手法のハイブリッド法が 17Mbps 以下では最も高いスループットである事が分かる。従来手法の C/S ルーティング以外は 10Mbps を越えたあたりからほぼ横ばいの軌跡を描いている。上り帯域の上昇が, スループットの向上に結び付かなくなっている。これは, 提案手法の P2P ルーティング手法においてもマルチキャストツリーを利用した手法においても, データを受信して集めるまで次のノードに対してデータ送信が出来ないため, ルータのルーティング遅延の影響の方が大きく出てしまうためであると考えられる。従来手法の C/S 型ルーティングが 17Mbps を越えたあたりから一番高いスループットを出している。C/S 型ルーティング手法はサーバ役ノードが各クライアントノードへユニキャストで一斉にデータ送信するため, 論理ネットワークでのホップ数は 1 となり, ルータのルーティング遅延の影響を提案手法に比べ比較的多くは受けないためであると考えられる。

4.3.3 実験結果 (24 ノード, 下り 10Mbps)

24 ノードに固定し, 上り帯域幅を変化させながらデータ交換にかかった時間を測定した。結果を図 27, 28 に載せる。

概ね 12 ノードの場合と同じ傾向を示している。提案手法では, 12 ノードの場合と変わらず, 10Mbps を越えたあたりから平坦となっている。これは 12 ノードの場合と同様に, ルータのルーティング遅延の方が大きな影響を与えるためであると考えられる。12 ノードの場合からの変化として, 従来手法の C/S 型ルーティングが提案手法よりも高性能を示すタイミングが 22Mbps を越えたあたりとなっており, 12 ノードの場合よりも遅い。これはクライアント・サーバ手法の送信

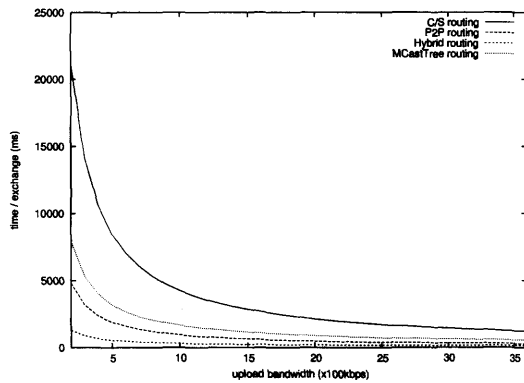


図 27 実験結果:24 ノード, ~3.5Mbps

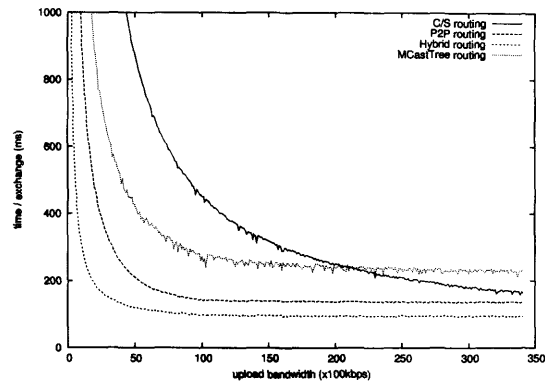


図 28 実験結果:24 ノード, ~50Mbps

量オーダーが $O(n^2)$ のため、24 ノードに増えた事で他の手法よりも送信量が大きく増えてしまったためであると思われる。

4.4 データ量変化における実験

オンラインゲームを想定して、データ量が 20Byte の場合の実験を行った。また、高解像度でのビデオ会議システムを想定して、データ量が 1Mbit の場合の実験を行った。

4.4.1 実験方法

各手法において、各ノードがオンラインゲームの制御情報や高解像度でのビデオ会議システムのフレームデータを想定したデータを生成し、それをそれぞれの手法を用いて全ノードが共有するまでにかかった時間を測定している。なお、各ノードはインターネットのバックボーンに上り 1Mbps, 下り 10Mbps で接続されているものとする。C/S routing は従来手法におけるクライアントサーバ型, P2P routing は提案手法において代行ルーティング法を採用した場合, Hybrid routing は提案手法においてハイブリッド法を採用した場合, MCastTree routing は完全二分木のマルチキャスト木を利用したアルゴリズムの場合の結果を表している。

4.4.2 実験結果: データ量が 20Byte, 1Mbit の場合

結果を図 29, 図 30 に載せる。

20B の場合は、1KB のときと同じような傾向を示している。しかし、15 ノード辺りまでは従来手法の C/S 型ルーティングが一番良い結果となっている。データサイズが小さく、参加者数がそこまで多くないオンラインゲームでは従来手法の方が適していると言える。

1Mbit の場合は従来手法の C/S 型では 13 ノード, 提案手法の P2P 型ルーティングの代行ルーティング法では 9 ノード, P2P 型ルーティングのハイブリッド法では 16 ノード, マルチキャスト

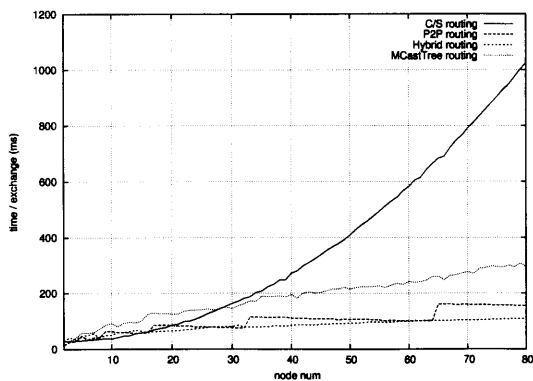


図 29 実験結果:上り 1M, 下り 10M, データサイズ 20Byte, 2~80 ノード

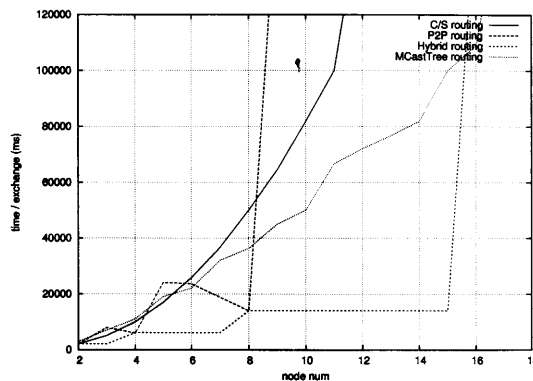


図 30 実験結果:上り 1M, 下り 10M, データサイズ 1Mbit, 2~16 ノード

トツリーを利用した手法では 17 ノードの時点でデータ交換にかかる時間が 120 秒を越えてしまった。ビデオ会議システム等にはリアルタイム性が求められるため、論外な結果となってしまっている。データサイズの大きな場合についての新たな手法の考察が今後の課題として挙げられる。

4.5 考察

n 対 n でのデータ交換が上り回線によってかなりの性能影響を受ける事が分かる。提案手法は上り回線の容量やデータサイズに関わらず $O(n)$ の高いスケーラビリティを示しており、従来手法のクライアント・サーバ型手法の $O(n^2)$ に比べかなりの性能改善が達成されたと言える。特に P2P 型ルーティング手法のハイブリッド法方式では提案方式の中でも著しい高スループットを出している。しかしながら、上り回線が極端に大きい場合 (100Mbps) やデータサイズが極端に小さい場合 (20B) では、あるノード数までは従来手法であるクライアント・サーバ型手法が高い性能を出す事もあり、今後のネットワーク環境の変化によっては提案手法よりも従来手法の方が有利となる可能性もある。

おわりに

ビデオ会議システムに着目し、従来では実現が難しかった多人数でのビデオ会議システムを実現するための n 対 n でのデータ交換の効率的な手法についての提案と、シミュレータによる評価実験を行った。

本研究では従来では1台のノードに負荷が集中してしまう問題に対し、参加者同士でP2P型の論理ネットワークを構成し、それを利用した動作アルゴリズムでデータ交換をすることによって、負荷を分散させることをはかった。

また、従来の手法との比較を行う実験を行い、従来の手法よりもオーダ次元を削減しスケーラビリティを飛躍的に向上できた事や、低ノード数でも高いスループットを出す事を示した。

今後の課題として、以下のものがあげられる。

- 会議グループへの参加・脱退手法の定義
- 実環境における評価実験

本研究では、数百ものノード数を想定した実験を行っていたため、実インターネットでの実験は困難であった。また、LANにおいてはデータリンク層では階層型となっている場合が多く、データの衝突等が多く発生するため提案手法のP2P型手法の実験を正しく評価出来ない。よって実インターネットを模倣したモデルを用いてシミュレーションによる実験を行った。他の実験評価方法としてはネットワークのシミュレーションツールであるNS2[16]や、実環境での実験を行うPlanetLab[17]がある。こうしたツールや設備を使っただけの実験はより正確なデータの計測が出来ると思われがちだが、実装が難しかったり、複数研究者での共同利用になるといった、扱いが難しい側面もある。今度P2Pネットワークを用いた研究を行うには、より簡易で精度の高い実験ツールが必要であると考えられる。

謝辞

日ごろから多くの御指導を頂きました太田義勝教授、鈴木秀智准教授に深く感謝いたします。そして、日頃何かとお世話になりました落合美子事務員に感謝いたします。また、本論文作成にあたって特にお世話になりました太田義勝教授に深く感謝いたします。最後に、日頃から熱心に討論して頂いた研究室の諸氏に感謝いたします。

参考文献

- [1] 堀内 英斗, 若宮 直紀, 村田 正幸, "多人数参加型 P2P テレビ会議システムのための論理網構築手法の提案と評価", 電子情報通信学会技術研究報告 Vol.106, No.577 pp.143-148, 2007.
- [2] 福永 茂, 中井 敏久, 吉田 哲雄, "LAN における分散マルチポイントテレビ会議システムの検討", 電子情報通信学会技術研究報告, Vol.93, No.224 pp.1-8, 1993.
- [3] 安藤真介, 稲葉良二, 林 一生, 野原健美, 鷹羽秀宣, 河口信夫, 高橋秀行, 石川雅彦, 小池 誠, 橋本 裕, 松原伸晃, 大田 渡, 大石和也, "環境に適応可能でスケーラブルな多地点間マルチメディア通信基盤ソフトウェア-効率的で使い易い多地点間通信をめざして", <http://www.ipa.go.jp/about/jigyoseika/06fy-pro/jise/2006-1210d.pdf>
- [4] 三村和, 中内清秀, 森川博之, 青山友紀, "RelayCast:ピアツーピア型ストリーム配信のためのミドルウェア", 電子情報通信学会技術報告, Vol.102, No.214 pp.7-12, 2002.
- [5] Bjorn Knutsson, Honghui Lu, Wei Xu, Bryan Hopkins: "Peer-to-Peer Support for Massively Multiplayer Games", Department of Computer and Information Science, University of Pennsylvania, Proc. Infocom, 2004.
- [6] Ashwin R.Bharambe Carnegie Mellon University, Venkata N.Padmanabhan Microsoft Research, Srinivasan Seshan Carnegie Mellon University, "Supporting Spectators in On-line Multiplayer Games", In Proceedings of ACM SIGCOMM Workshop on Hot Topic in Networks, November 2004.
- [7] "ITU-T Recommendation H.323", <http://www.itu.int/rec/T-REC-H.323-200606-I/en>
- [8] "入門 H.323 規格", QUALITY STAR, <http://www.kowa.co.jp/interstream/h323/index.htm>
- [9] タミー・コスタンゾ, "マルチキャストのビデオを実現する H.323", インターネット最新テクノロジー: 第 10 回, INTERNET magazine 1997/11.
- [10] 川村 卓也, 波谷 尚久, 船戸 康雄, "映像会議を実現する多地点接続装置 MCU", 東芝レビュー Vol.61 No.4, 2006.
- [11] R.Boivie, N.Feldman, Y.Imai, W.Livens, D.Ooms, OParidaens, E.Muramoto, "draft-ooms-xcast-basic-spec-06", IETF, 2004.
- [12] Suman Banerjee, Christopher Kommareddy, Koushik Kar, Bobby Bhattacharjee and Samir Khuller, "OMNI; An Efficient Overlay Multicast Infrastructure for Real-time Ap-

- plications”, Computer Networks, Volume 50, Issue 6, Pages 739-876 13 April 2006.
- [13] X.Zhang, J.Liu, B.Li, and T.-S,P.Yum, ”CoolStreaming/DONet;A data-driven overlay network for effcent media streaming”, in IEEE INFOCOM 2005 Miami US, Mar 2005.
- [14] 荒川 伸一, ”インターネットトポロジに現れる Power-Law 特性と経路制御への応用”,
<http://www.anarg.jp/achievements/web2005/slide/arakawa05IEICESociety-modelingISPtopology-slide.pdf>
- [15] Ellen W. Zegura, Kenneth L. Calvert, Samrat Bhattacharjee:”How to Model an Internet-work”, College of Computing Georgia Institute of Technology Atlanta, GA 30332-0280, 2000.
- [16] The Network Simulator - ns2, <http://www.isi.edu/nsnam/ns/>
- [17] PlanetLab, <http://www.planet-lab.org/>