

多対多ビデオ会議システムのための アプリケーションマルチキャストツリー構成手法の提案と実装

高野 健一郎, 北形 元, 菅沼 拓夫, 白鳥 則郎
東北大学電気通信研究所/情報科学研究科

近年, ストリーミング配信やビデオ会議など, ネットワークを介した複数人によるマルチメディアコンテンツの利用が増加している. このような通信では, 送信ノード(サーバノード)でのデータ送信処理や送信ノードのアクセスリンクにおいて冗長性が存在する. 本研究では, 複数人によるマルチメディア通信の中でも主に多方向通信である多対多ビデオ会議システムの利用時における冗長性を削減するため, 順次流量分配方式のアプリケーションマルチキャストツリー構成手法を提案・実装する.

Application Multicast Construction Method Tree for Many-to-Many Videoconferencing System

Kenichiro Takano, Gen Kitagata, Takuo Suganuma and Norio Shiratori
Research Institute of Electrical Communication / Graduate School of Information Science,
Tohoku University

Recently, the use of the multimedia contents such as the streaming delivery and videoconferencing for a number of persons through the network increases. In such a communication, redundancy exists in the data transmission processing and the access link of the transmission nodes. In this research, we propose the application multicast tree construction method with the sequential flow distribution to reduce the redundancy in the bi-directional multimedia communication for a number of people like many-to-many videoconferencing system.

1 序論

近年, ストリーミング配信やビデオ会議など, ネットワークを介した複数人によるマルチメディアコンテンツの利用が増加している. このような通信では, まったく同じマルチメディアデータを全ての受信ノード(クライアントノード)に向けて送信するため, 送信ノード(サーバノード)でのデータ送信処理や送信ノードのアクセスリンクにおいて冗長性が存在する. このような冗長性を削減するための手法として IP マルチキャスト [1] がある. IP マルチキャストは, ルータが受け取ったパケットを必要な分だけコピーし複数の受信ノードに向かって送信することによって, 送信ノードにおけるデータ送信処理やアクセスリンクの冗長性を削減することができる. しかし, IP マルチキャストを用いるには, マルチキャストアドレスの管理の手間やマルチキャストに対応したルータの必要性などから, 実際に使用するために必要なインフラストラクチャが十分に普及していないなど, 運用上の問題がある.

このような問題に対する現実的な解決策として, アプリケーション層でマルチキャスト通信を行うアプリケーションレイヤマルチキャストが注目されている [2] [3]. アプリケーションレイヤマルチキャストでは, エンドホスト間のユニキャストの連携によって形成されたオーバ

レイネットワーク上で, マルチキャストを実現する. 具体的には, 図 2 に示すように, IP マルチキャストでルータが行う”パケットをコピーし, 複数の受信ノードに向かって送信する”という動作を, 各受信ホストが行う. そのため, IP マルチキャストに比べてネットワークの利用効率が低いという欠点があるものの, エンドホストへの実装だけでよいために既存のインターネット上で実現可能であるなど, 運用が容易であるという利点を持つ.

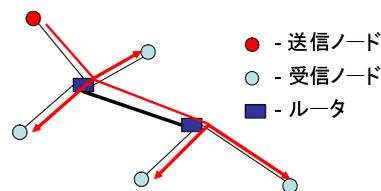


図 1 IP マルチキャスト

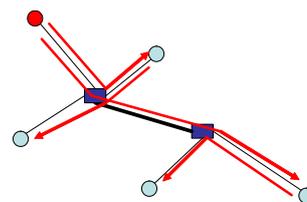


図 2 アプリケーションレイヤマルチキャスト

本研究では, 上述のアプリケーションマルチキャストに注目し, 複数人によるマルチメディア通信の中でも主

に多方向通信である多対多ビデオ会議システムの利用時における経路の冗長性を削減し、ネットワーク資源をより有効に利用するためのアプリケーションマルチキャストツリーの構成手法を提案する。

2 関連研究

アプリケーションレイヤマルチキャストについての既存研究としては、代表的な手法として ALMI[2], Overcast[3] などが提案されている。

ALMI は短遅延での配信を目指し、各ノード間における遅延をツリー構築の指標とし、総遅延が最小である被覆木を構築する。また、Overcast はビデオオンデマンドなどの大容量ファイル配信アプリケーション向けに、ツリー構築の指標として各ノードの使用可能帯域を用いて、帯域利用効率を考慮した共有経路木を構築している。

上記の 2 手法を含む既存のアプリケーションレイヤマルチキャストにおける技術課題は、複数本のトラフィックを同時に、かつ、使用可能な帯域を有効に使用してツリー構築するための機構が存在しないことである。これは、これらの既存技術が、複数人の送信者が存在する場合を想定しておらず、ストリーミング配信のような一対多通信の環境で用いることを前提として設計されていることによる。

しかしながら、多対多ビデオ会議のような双方向アプリケーションでは、それぞれのノードが他の全てのノードに向けてマルチメディアデータを送信する。よって、既存のアプリケーションレイヤマルチキャストを使って多対多ビデオ会議システムの通信を行うと、ノード間での協調が行われなままそれぞれのノードが個別に自身のマルチキャストツリーを構築することとなる。そのため、全てのノードが同じツリー構築戦略を用いて各トラフィックのためのマルチキャストツリーを構築することになり、優先してデータの中継を行うようなノードに対してトラフィックの集中が生じる可能性がある。その結果、豊富なネットワーク資源を持つアクセスリンクほどボトルネックになりやすく、利用可能な通信帯域を十分に活用できなくなる場合がある。

3 順次流量分配方式の提案

3.1 想定するネットワークモデル

2 章で挙げた既存手法では共通して、図 4 のような無向完全グラフ型のネットワークモデルを用いてツリー構築計算を行っている。しかしながら、実際には図 3 のようにひとつのノードはひとつのアクセスリンクしか持たないケースが一般的であるため、このモデルではどのリンクに何本のトラフィックが通っているのかわかりづらく、計算が煩雑になってしまう恐れがある。よって本研究では、図 5 に示すネットワークモデルを用いてツリー

構築計算を行う。ここで、図中のリンクの横にある数字はそのリンクの上り帯域幅を表している。

また本研究では、コアネットワーク内ではボトルネックは生じないものとし、このシステムは全二重ネットワークで用いるものと仮定し、議論する。

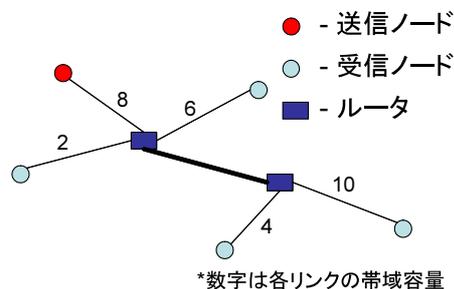


図 3 一般的なネットワークの例

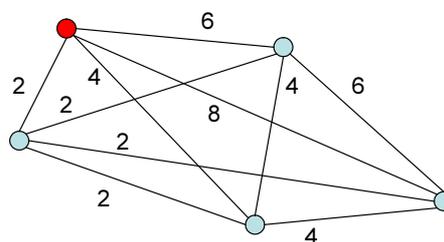


図 4 無向完全グラフ型ネットワークモデル

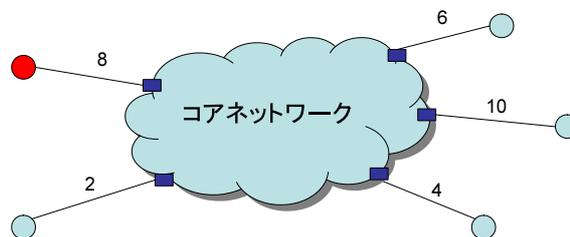


図 5 コアネットワーク型ネットワークモデル

3.2 順次流量分配方式の提案

本研究では、各送信ノードがそれぞれのデータ配信に使用する帯域を順次割り当てていくアプリケーションマルチキャストツリー構築手法を提案する。提案方式では、既存研究の技術課題を解決し、複数本のトラフィックに対して有効にツリー構築を行うために、以下のような 2 つのポリシーに従いツリーを構築する。

以下、本論文では、「送信ノード」は一つのデータトラフィックに対して唯一存在するそのマルチメディアデータのオリジナルの配信元ノード（サーバノード）の事を指す。それ以外のクライアントノードは、全て「受信ノード」と呼び、その中で特に、自らが受信したデータを他の受信ノードに向けてコピー・送信するノードを「中継ノード」と呼ぶこととする。また、各ノードはマルチキャストに参加するノードの IP アドレス等をツリー構築の前に予め知っているものと仮定する。

1. 接続されたアクセスリンクの上り帯域幅の小さい送信ノードから順にツリーを構築する
2. 次の送信ノードがツリー構築を行う前に、使用する予定の帯域を予め使用可能帯域から減らす

このように、各マルチキャストツリーを構築する際に、ノード間の連携を実現することで2章で述べたようなトラフィックの集中を避け、ネットワーク資源を有効に利用するツリー構築を可能にする。次に提案手法によるツリー構築の手順を説明する。

3.3 ツリー構築の手順

各トラフィック毎に、その送信ノードは図6に示す4つの手順i.~iv.により自身のマルチキャストツリーを構築する。

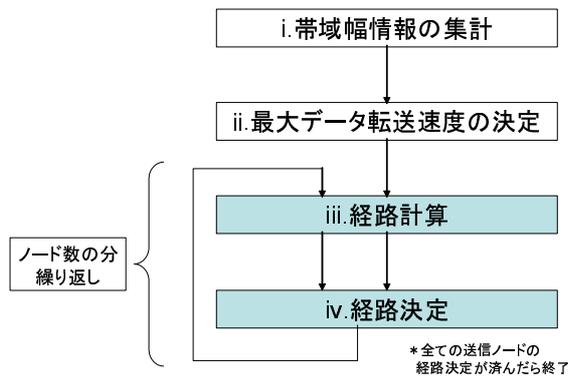


図6 ツリー構築の手順

- i. 帯域幅情報の集計
- ii. 最大データ転送速度の決定
- iii. 経路計算
- iv. 経路決定

このうち、i. 帯域幅情報の集計とii. 最大データ転送速度の決定は、ビデオ会議などのサービスを最初に起動したノードである多対多マルチメディア通信サービス開始ノードのみが行う項目で、残りのiii. 経路計算・iv. 経路決定は全ての送信ノードが自身のマルチキャストツリーを構築する際に行う項目である。よって、iii. 経路計算・iv. 経路決定は送信ノード数の分だけ繰り返して行われる。以下、特にii. 最大データ転送速度の決定とiii. 経路計算について説明する。

3.3.1 ii. 最大データ転送速度の決定

各ノードが上りトラフィックをいくつ持つことができるかを決定するため、ここで最大データ転送速度を決定する。

全てのトラフィックデータ転送速度の値は等しいものとし、サービス参加ノードの数を n 、最大データ転送速度を D 、各アクセスリンクの上り帯域幅を B 、其中最も小さいリンクの上り帯域幅を B_{min} 、各アクセスリンクの下り帯域幅のうち最も小さいものを B_{Dmin} とし、まず、i. 帯域幅情報の集計で得た情報を元に、以下の式(1)

を用いて最大データ転送速度の初期値 D_0 を決定する。

$$D_0 = \min(B_{min}, \frac{\sum B}{n(n-1)}, \frac{B_{Dmin}}{n-1}) \quad (1)$$

D_0 は、各アクセスリンクの帯域がちょうど D_0 の自然数倍になっている理想の場合の最大速度であり、実際の最大データ転送速度の値はこれ以下となる。そこで次に、 B を D_0 で割ることで各アクセスリンクで作ることが可能なこのクッション数を求める。これにより、各アクセスリンクで確立できる接続数の総数が求まる。システム全体に必要な接続の総数は $n(n-1)$ なので、この条件を満たすように D の値を決める。その際、 D が初期値 D_0 のままで総接続数が $n(n-1)$ 未達ならば、 D の値を小さくして必要な接続数を確保する。この条件を満たす D が、求める最大データ転送速度となる。

図5のようなネットワークの場合、上述の手順により求めた最大速度 D は図7ようになる。

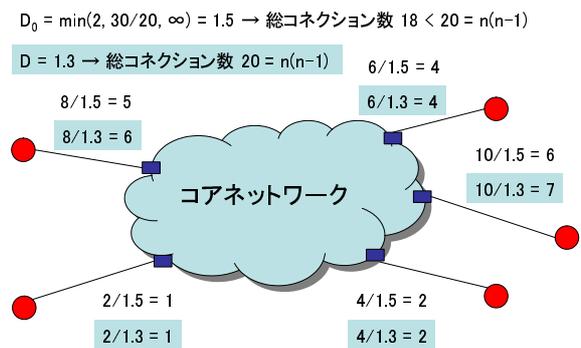


図7 最大データ転送速度の決定

3.3.2 iii. 経路計算

本研究の目的は多対多ビデオ会議システムでの利用であり、それゆえ、フレームレートや映像サイズ等の動画の品質を一定以上に維持するために十分な帯域での通信を実現するだけでなく、円滑なビデオ会議のインタラクションのために遅延を最小限に抑えることも必要である。そこで本手法では、中継ノードによるデータ中継の回数を1回以内に制限する。しかし、この条件は単にiii. 経路計算において最大データ転送速度と各ノードの受け持てる上り接続数を決めただけでは完全に満たすことはできない。そこで、各ノードは以下のルールに基づいて自身からの送信ツリーを決定する。

- 経路決定済みノードに接続確立可能な量の使用可能帯域を残さない

これにより、図8のように全てのトラフィックが確実に中継回数1回以内で他の全ての参加ノードに到達することができる。

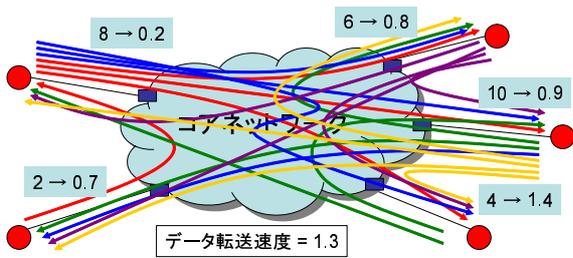


図8 ツリー構築成功時の経路

4 設計と実装

4.1 エージェント設計

本提案手法は多対多ビデオ会議システムでの利用を想定しており、このような各参加者が対称的な環境においては、各参加者のノードが自立分散的に他のノードと協調し、提案手法に基づきマルチキャストツリーを構成することが望ましい。そこで本研究では、3.3節で示した機能を、自律的に動作する多数のエージェントの協調的動作により問題を解決するマルチエージェントシステムとして設計を行った。

以下に、設計したエージェントと、そのエージェントによって実現する機能を示す。

- MulticastCommunicator
 - i. 帯域幅情報の集計
 - iv. 経路決定
- RouteDecisionMaker
 - ii. 最大データ転送速度の決定
 - iii. 経路計算

なお、サービスの動的構成、名前解決部には井戸端LAN[5]を用いている。

4.2 エージェント構成

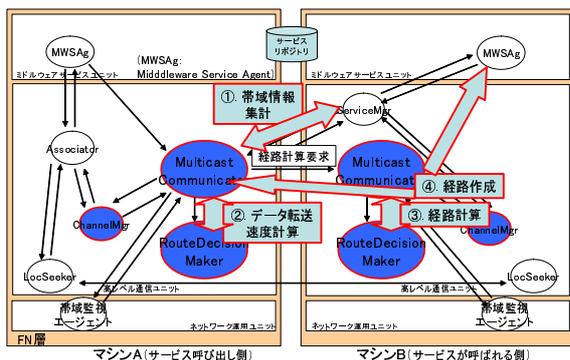


図9 エージェント構成

設計したエージェントの構成を図9に示す。以下、MulticastCommunicatorとRouteDecisionMaker以外のエージェントの役割を示す。

次に、3.3節の機能をこれらのエージェント構成でどのように実現するかを説明する。

i. 帯域幅情報の集計

表1 他のエージェントとその役割

エージェント名	役割
MWSAg	サービス提供エージェント
Associator	接続要求獲得
ServiceMgr	他ホストからの接続要求獲得
ChannelMgr	Communicator生成
LocSeeker	位置解決

サービス呼び出し側の MulticastCommunicator が相手の ServiceMgr (接続要求獲得機構) に接続・帯域幅情報要求メッセージを送信する。応答メッセージを受け取ることで参加ノードの帯域幅情報を取得する。

ii. 最大データ転送速度の決定

MulticastCommunicator は i. 帯域幅情報の集計で獲得したデータを RouteDecisionMaker に受け渡す。RouteDecisionMaker は 3.3.1 節の手法で最大データ転送速度 D を計算し、MulticastCommunicator に伝える。

iii. 経路計算

経路計算要求を受け取った MulticastCommunicator は使用可能帯域幅情報を RouteDecisionMaker に渡す。RouteDecisionMaker は自身の送信ツリーを計算し、結果を MulticastCommunicator に返す。

iv. 経路決定

計算結果を受け取った MulticastCommunicator は MWSAg (サービス提供エージェント) と相手の MulticastCommunicator にチャンネル通知する。

以下の表2に、設計の一例としてメッセージのパフォーマティブの一部を示す。

表2 パフォーマティブの一部

メッセージ	:performative	:content
接続要求	multiCom	(spec :name ?cname :user ?user :owner ?myname)
相手 MC 生成要求	findMultiCom	(MultiCom :id ?id :spec ?spec)
帯域情報	replyFindMultiCom	(MultiCom :id ?id :spec ?spec :ag ?ag :bandwidth ?bw)
ツリー要求	calculate	()
転送速度計算要求	hostdata	(data :size ?size :hostlist ?hostlist)
転送速度計算結果	startRouteCal	(start :to ?ag :data (data ...))
経路計算要求	routeCal	(data :size ?size :hostlist ?hostlist)
経路計算結果	replyHostdata	(result :arrival ?arr :end ?end :data (data ...) :nextAg ?newAg :route ?route)
チャンネル通知 (MWSAg)	serverChannelReady	(channel :channel ?channel :id ?id)
チャンネル通知 (MC)	path	(channel :channel ?channel :host ?host :port ?port :id ?id)
	relayPath	(relay :child ?child :channel (channel ...) :id ?id)

次にメッセージの種類について概明する。

- 接続要求
 - 起動されたサービス名、自身と相手の利用者名を通知する。
- 相手 MC 生成要求 (サービス参加要求)
 - 起動されたサービス名を通知する。
- 帯域情報
 - サービス参加要求への応答として、自ノードの

MulticastCommunicator の名前とアクセスリンクの使用可能帯域を返す。

- ツリー要求
 - 参加ノードの帯域情報が全て集まった後、MulticastCommunicator にマルチキャストツリーを要求する。
- 転送速度計算要求
 - ii. 最大データ転送速度の決定のために他ホストから受け取った帯域幅情報を RouteDecisionMaker へ渡す。
- 転送速度計算結果
 - 最大データ転送速度の値と最初に iii. 経路計算を行うノードを計算結果として MulticastCommunicator に返す。
- 経路計算要求
 - 経路計算を行うノードの MulticastCommunicator に最大データ転送速度と使用可能帯域の値を渡す。
- 経路計算結果
 - iii. 経路計算の結果として、ツリー構築メッセージ、経路計算後の使用可能帯域、次に経路計算を行うノードの情報と自分が経路計算を最後に行ったノードであるかどうかなどを MulticastCommunicator に返す。
- チャンネル通知 (MWSAg)
 - 入力ストリームを取得するためのチャンネル名を渡す。
- チャンネル通知 (MC)
 - path は受信ノードに対するメッセージのパフォーマティブで、IP アドレスとポート番号を渡す。relayPath は中継ノードに対するメッセージのパフォーマティブで、それに加えて中継先ノードの情報も渡す。

4.3 実装

上述のような設計に基づき、以下に示す実装環境でプロトタイプを実装した。

- CPU : Pentium4 1.8GHz
- Memory : 512MB
- OS : Windows XP SP2
- 言語 : JDK 5.0
- エージェント実行・開発環境 : IDEA 1.2.4 [4]

なお、実装したエージェントの数は7個だった。

5 実験と評価

5.1 動作実験

本節では、ツリー作成・コネクション構築等の機能が正常に動作し、ノード間の多対多通信が可能となることを確認するために、図 10 に示すように 3 台のノート PCa, b, c をホストとして使い、図 11 に示すようなネットワー

クを構築した。また、多対多通信サービスとしてメッセージャーサービス提供エージェントを用いて動作実験を行った。

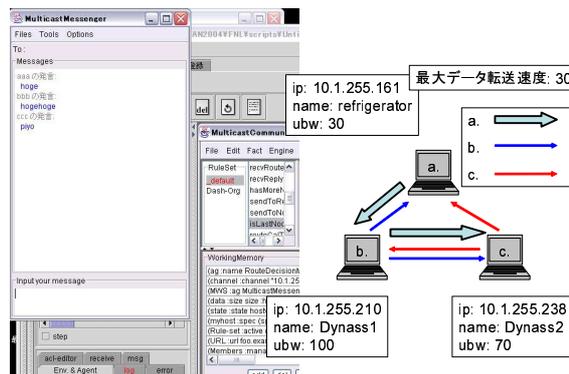


図 10 動作実験の様子

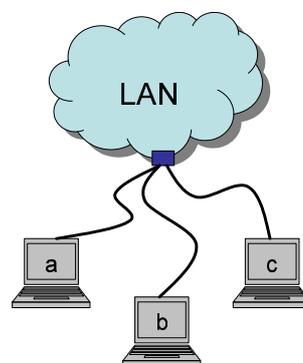


図 11 動作実験で用いたネットワーク

図 10 の右側は、動作実験を行った際の環境を表したもので、ip, name, ubw はそれぞれ各ホストの IP アドレス、コンピュータ名、上り帯域幅を表している。この環境で実際にツリー構築を行った結果、最大データ転送速度は 30、各ホストのツリーはそれぞれ矢印のような形になった。

なお、各ホストの下り帯域幅はそれぞれ充分大きいものとした。

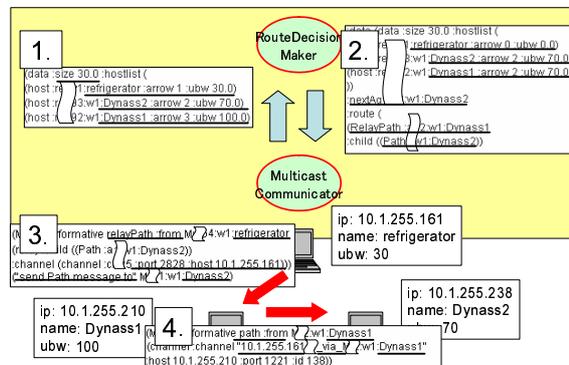


図 12 動作実験中のメッセージ

図 12 に、iii. 経路計算～iv. 経路決定までのメッセージの内容を追跡した結果を示す。

1. 経路計算要求メッセージ (MulticastCommunicator → RouteDecisionMaker)

- ii. 最大データ転送速度の決定で決まった最大データ転送速度の値と各ホストの受け持てる上りコネクション数, 使用可能上り帯域幅を RouteDecisionMaker に渡している. ここで RouteDecisionMaker は iii. 経路計算を行う.

2. 経路計算結果メッセージ (RouteDecisionMaker → MulticastCommunicator)

- 使用予定帯域の分を減らした後の各ホスト使用可能帯域情報と次に経路計算を行うノードのコンピュータ名と, :route の項にツリーの構築のための MulticastCommunicator への命令が書かれている. この場合は Dynass2 に対して「Dynass1 への中継要求メッセージ」を送らせているが, これは図 10 のホスト a が構築するツリーと一致する.

3. 中継ノードへのチャンネル通知メッセージ (MulticastCommunicator → 中継ノード側 MulticastCommunicator)

- 使用するポート番号と自ノードの IP アドレス, 中継先ノードのコンピュータ名を中継ノードの MulticastCommunicator に向けて送信している.

4. 中継先ノードへのチャンネル通知メッセージ (中継ノード側 MulticastCommunicator → 中継先ノード側 MulticastCommunicator)

- 中継ノードが使用するポート番号とその IP アドレス, さらに, 中継元ノードのコンピュータ名を中継先ノードの MulticastCommunicator に向けて送信している.

5.2 動作実験に対する考察

メッセンジャーサービスを用いた動作実験の結果, 図 10 の右図のようなツリーが構築された.

このツリーでは, 3.2 節で挙げた提案手法のポリシー「2 次の送信ノードがツリー構築を行う前に, 使用する予定の帯域を予め使用可能帯域から減らす」によって, それぞれのノードが使用する帯域をお互いに知らせあうことで, 一部のアクセスリンクへのトラフィックの集中を回避している. また, 「1 接続されたアクセスリンクの上り帯域幅の小さい送信ノードから順にツリーを構築する」によって, 比較的大きい帯域を持つノードが, 自分のマルチキャストツリー構築のためにアクセスリンクの帯域を大きく消費することで帯域の小さいノードのトラフィックを中継することができなくなり, 中継回数が増加することによって遅延が大きくなることを防いでいる. これらの効果は, ノード数が多くなり, データトラフィック

の総数が増えることでより顕著になると考えられる.

以上により, 提案手法によって構築されたマルチキャストツリーでは, 広い帯域を使った遅延の小さい通信が可能となることが示された.

6 結論

既存のアプリケーションレイヤマルチキャスト技術にはネットワーク資源を有効に活用した複数本のデータトラフィックのツリーを構築するための機構が存在しなかったため, 中継ノードのアクセスリンクなどでボトルネックが生じてしまうという問題があった. 本研究では, この問題を解決するため, ネットワーク資源を有効に利用した多対多ビデオ会議を行うためのアプリケーションレイヤマルチキャストツリーを構築する順次流量分配型のツリー構築手法を提案し, エージェント設計と実装を行った. さらに, 実現したシステムを用いた動作実験を行い, 実験結果よりネットワーク資源を有効に使用するツリーを構築できることを示した.

今後, 多対多ビデオ会議システムでの利用のために, 動画受信アプリケーションの実装を行い, それを用いた評価を行う予定である.

また, 本システムの課題として, 使用者や各ノードの使用帯域が変化した場合への対応, 無線など半二重通信への対応を検討している.

参考文献

- [1] Diot, C., Dabbous, W. and Crowcroft, J.: Multipoint Communication: A Survey of Protocols, Functions and Mechanisms, *IEEE Journal on Selected Areas in Communications*, Vol.15, No.3, pp.277-290 (1997).
- [2] Pendarakis, D., Shi, S., Verma, D. and Waldvogel, M.: ALMI: An Application Level Multicast Infrastructure, in *Proc. of 3rd Usenix Symp. on Internet Technologies & Systems*, 2001.
- [3] Jannotte, J., Gifford, D. K., Johnson, K. L., Kaashoek, M. F. and O'Toole, J. W.: Overcast: Reliable Multicasting With an Overlay Network, in *Proc. of the 4th Usenix Symp. on Operating Systems Design and Implementation (OSDI)*, 2000.
- [4] IDEA Web Site, "IDEA", <http://www.ka.riec.tohoku.ac.jp/idea/>
- [5] 北形元, 渡辺将一郎, 松島悠, 永井克幸, 長谷川大介, 木下哲男, 白鳥則郎. アドホックネットワーク向けサービス構成システム: 井戸端 LAN の設計と実装. 情報研報 MBL vol.2004, no.44, 2004.