動画像を用いたコミュニケーションシステム向けのアプリケーション層マルチキャスト

山口 弘純　中村 嘉隆　廣森 聡仁　安本 慶一
東野 輝夫　谷口 健一

本論文では，エンドホスト間で映像データを実時間交換するようなコミュニケーションシステム向けのアプリケーションレベルマルチキャストプロトコル Emma を提案する．Emma は，複数の実時間データを，ユーザの優先度に応じてマルチキャスト転送する制御プロトコルであり，完全な分散制御プロトコルとして実現されているだけでなく，優先度要求に応じた動的なデータ配信制御が短い反応時間で実現でき，かつ制御メッセージ集約によるトラフィック削減などの工夫も図られている．数十ユーザを仮定したシミュレーション実験による性能評価の結果，Emma は少ないプロトコルオーバヘッドで高いユーザ満足度を得られることがわかった．

1 はじめに

高速ネットワークの普及は，近い将来，比較的小規模（数十人〜数百人程度）のグループによるビデオチャットのようなグループ通信アプリケーションの需要をもたらすと予想される．しかし，そのようなグループ通信が多数同時に行われる場合，全グループのデータ交換処理を限られた数のサーバが行うことは，サーバ側のネットワーク資源及びコネクション数の観点から現実的ではない．

グループ通信は主にグループ内ユーザへの同報通信からなるため，特定のサーバを必要とせず，ネットワーク資源の利用効率も高い IP マルチキャストが有用な通信の 1 つであると考えられる [1]．しかし，IP マルチキャストは現状では組織内での利用が多く，異なる組織間のユーザが全て IP マルチキャストで相互通信可能な環境を仮定することは現実的ではない．一方，全ての 2 ユーザ間のユニキャストでそのような同報通信を実現することは，グループの規模に関するスケーラビリティが欠如する．

このような問題に対する現実解として，マルチキャストをアプリケーション層で実現する通信形態（アプリケーションレベルマルチキャスト，以下 ALM）が注目を集めている．ALM では，エンドホストがユニキャストトンネリング（文献 [13] などいくつかの研究ではトンネリングの代わりに IP マルチキャストを認めるものもある）によるいわゆるオーバレイネットワークを形成する．そしてそれらエンドホストがオーバレイネットワーク上でマルチキャスト配信木を構築及び管理し，トンネリング間のパケット複製及び転送を行うことで実現される．例えば，図 1(a) ではホスト B は A からのデータを複製し，C，D に転送している（図 1(b) はその際のマルチキャスト配信木を表す）．ALM では，実リンクでのパケットの重複配送（例えば図 1(a) のリンク B-t では同一内容のデータ

An application level multicast protocol for multi-party visual communication systems.
Hirozumi Yamaguchi, Yoshitaka Nakamura, Akihito Hiromori, Teruo Higashino, Kenichi Taniguchi, 大阪大学 大学院情報科学研究科, Graduate School of Information Science and Technology, Osaka University.
Keiichi Yasumoto, 奈良先端科学技術大学院大学 情報科学研究科, Graduate School of Information Science, Nara Institute of Science and Technology.
コンピュータソフトウェア, Vol. 21, No. 2 (2004), pp. 1–11.
[論文] 2003 年 5 月 9 日受付.

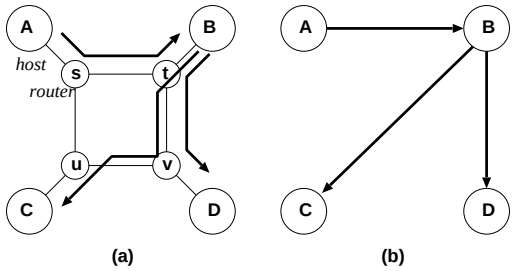


図 1　アプリケーションレベルマルチキャスト

が 3 回配送される）や，ホップ数の増加などのオーバヘッドがあるが，ネットワーク層でのマルチキャストサポートを必要とせず，ネットワーク資源の利用効率の点でユニキャストより効率的であるといった実用性から，近年になり多くの研究がなされている．

ALM に関する従来研究は，それぞれ異なる設計目標に基づく．Overcast [7] 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，帯域利用効率を考慮した共有経路木を構築する方法を提案しており，CAN [10] は仮想アドレス空間にホストをマッピングすることで経路木構築プロセスの簡単化を図っている．RMX [11] は利用可能帯域が異なる端末を組織化した ALM を構築し，効率よくデータを配信する方法を提案している．HBM [8] は経路木の安定性のため，バックアップリンクをどのように構成するかの方法論を主体とし，Yoid [13] はオーバレイネットワークの堅牢性向上のため，共有木とメッシュ状のオーバレイネットワークを併用している．また，ALMI [6] は短遅延での配信を目指し，総遅延が最小である被覆木を構築する．これらの文献では，それぞれの設計目標に対し優れたプロトコル設計を与えている．しかし，動画像など比較的大容量のデータストリームが複数同時並行的に配信されるようなビデオ会議などのアプリケーションにおいて，それら複数のデータストリームがオーバレイネットワーク上の限られた資源（オーバレイリンク容量やホストのストリーム転送能力）を競合する際の動的な制御方法については考慮されていない．Narada [4] [5] はビデオ会議などのアプリケーション向けに，メッシュ上に構築されたオーバレイネットワーク上に帯域及び遅延をメトリックとした各ノードからの経路木群を分散制御で構築，維持することで，データストリーム間の経路重複を（静的に）ある程度回避する．しかし，Narada においても同様に動的な制御は実現されていない．

本論文では，複数のホストが他のホストに向けて動画像などの実時間データストリームを同時配信するアプリケーション向けの ALM である Emma (End-user Multicast for Multi-party Applications) を提案する．Emma は，各ホストがそれらデータストリームに対し指定する優先度要求（プリファレンス）に基づき，限られたオーバレイリンク容量や各ホストのストリーム転送能力の中で，どのデータストリームを優先的に配信するかを動的かつ分散で制御できる新しい ALM であり，この点で既存のアプローチとは大きく異なる．例えばビデオ会議システムのユーザは中心となる話者達の映像に対しより強い興味を持つなど，ユーザが映像に対するプリファレンスを持つ状況はごく自然であることから，Emma の制御機能は，ユーザ満足度を向上させるための機能として有用であると考えられる．

シミュレーション実験による性能評価の結果，要求順に映像データを配送する単純な ALM と比較し，Emma は約 1.5 倍のユーザ満足度を達成できることがわかった．

本論文は以下のように構成する．2 章では Emma の機能とその実現について述べる．3 章ではシミュレーション実験による性能評価について述べ，4 章で本論文のまとめを述べる．

2　Emma プロトコル

以下では，各ノードが送信するデータソースを単にデータ（あるいはデータストリーム）とよぶ．Emma が想定するデータは実時間映像などの連続メディアである．

2.1　セッションへの参加

Emma では，セッションに参加するノードは，すでにそのセッションに参加しオーバレイネットワークを構築しているノード群の IP アドレスとポート番号の組のリストを，それらを管理する外部サーバから取

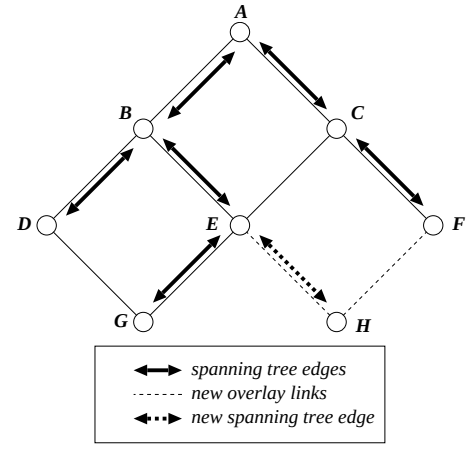
得するものとする．この機能は例えば WWW サーバなどで実現可能であり，さらに Emma の機能とは独立しているため，本論文では扱わないものとする．

参加するノードはリストを取得後，それらのノード間の遅延を ping などを用いて測定し，その遅延が小さいいくつかのノードに対して TCP コネクションを確立する．それらのコネクションを利用し，実際のオーバレイリンクとなる UDP のポート番号や，そのオーバレイリンク上に配送できる最大のデータストリーム数についてのネゴシエーションを行い，UDP による仮想的なオーバレイリンクを確立する．なお，オーバレイリンク上の最大データストリーム数は，自身の計算機資源，LAN の帯域，測定した遅延などから適宜決定する†1．

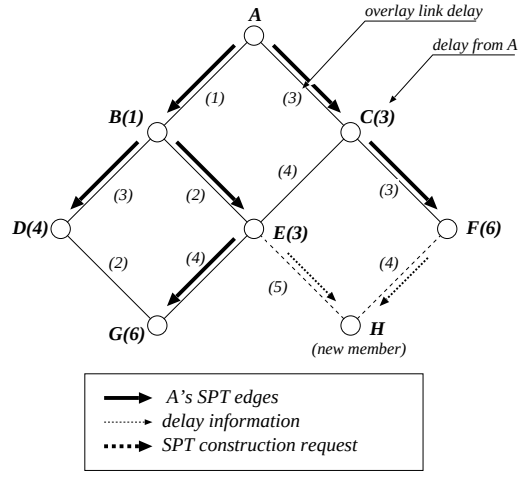
なお，Emma が想定する会議アプリケーションなどでは帯域をメトリックとすることが望ましいが，一般にエンドノード間の帯域測定は容易でないことと，遅延が短い経路は比較的帯域にも余裕があることを考慮し，遅延に基づく構築方針を採用する．

2.2 マルチキャスト経路木構築

セッションに参加しているノードは，オーバレイネットワークでの相互接続性維持や，各ノードの送信データのサムネイルの公告などの目的のため，共有被覆木を保持する．新たに参加したノードは，確立したオーバレイリンクを介して既存の共有被覆木に接続する．例えば，図 2(a) において，ノード *A..H* がこの順にセッションに参加したとする．もしノード *H* がオーバレイリンク *E-H* 及び *F-H* を確立したとすれば，ノード *H* は，そのうちの 1 つ（この例では *E-H*）を介して共有被覆木に接続する．なお，参加したノードは，それを介して共有被覆木に接続したノードを親ノードとみなし，親ノードが保持する祖先ノードのリストを受け取る．もし親ノードが離脱した場合は，オーバレイネットワークが分割されるのを避けるため，その子ノードは祖先ノードのいずれかとオーバレイリンクを確立し，共有被覆木に再接続する．



(a) spannig tree construction



(b) SPT construction

図 2 (a) 共有被覆木への接続，(b) SPT への接続

一方，データ配信向けには，各ノードが送信ノードになり得ることを考慮し，最短遅延経路木（Shortest Path Tree，以下 SPT）をノードごと構築する．新たに参加したノードは，各 SPT 上の根ノードから隣接ノードまでの遅延値を隣接ノードから受け取り，根ノードから自身まで最小遅延が実現できるノードとのオーバレイリンクを介して SPT に接続する．例えば，図 2(b) では，ノード *H* はノード *E* 及び *F* から，ノード *A* の SPT の（ノード *A* からの）遅延（この例ではそれぞれ 3 及び 6）を受け取り，さらにそれ

†1 Emma では簡単のため，オーバレイリンク容量をストリーム数で考える．なお利用帯域が大きいストリームは，通常ストリームの利用帯域を 1 とした単位帯域を複数利用すると見なすことで対応できると考えられる．

らとの間のオーバレイリンクの遅延を加えてより小さい遅延が実現できるノード E を介して A の SPT に接続する．また，自身の SPT に関しては，DVMRP と同様，オーバレイネットワークに遅延をメトリックとしたブロードキャストと枝刈りにより，SPT を構築する．

なお，SPT の根となる各ノードは SPT 維持のためのメッセージ送信を SPT 上で定期的に行う．ノードの離脱などによりオーバレイネットワークトポロジが変化し，あるノードがそのメッセージを受信できなくなった（SPT が破壊された）場合†2，SPT の根ノードに共有被覆木を介して再構築要求を行う．根ノードからのブロードキャストと枝刈りにより SPT が再構築される．

2.3 マルチキャストデータ転送制御

SPT に基づき，各ノードはマルチキャストデータ転送制御のための 6 種類のメッセージ（(a) MEDIA/Keep，(b) MEDIA/Join，(c) MEDIA/Leave，(d) MEDIA/Accept，(e) MEDIA/Reject 及び (f) MEDIA/Adaptation）を処理する．(a) は現在受信中のデータ転送を引続き行うことを定期的に依頼するため，(b) はデータの受信要求を行うため，(c) はセッションから離脱することを通知するため，(d) 及び (e) は (b) に対する応答のため，(f) は転送中のデータの品質が劣化したことを通知するためのメッセージである．

Emma では，各ユーザが他のユーザのデータに対して非負整数値で指定する優先度（以下プリファレンス値とよぶ）をもとに，あらかじめノード間でネゴシエーションにより決定されたオーバレイリンクの最大容量を超える数のデータ受信要求がされた場合，もしくはあるデータストリームに極度の品質劣化がみられた場合は，オーバレイリンク上のデータストリーム数を調整することによる QoS 制御を行う．この機能を含め，Emma の機能が上述のメッセージにより分散環境でどのように実現されるかを述べる．

†2 メッセージ受信待ち時間のタイムアウトにより判断する．

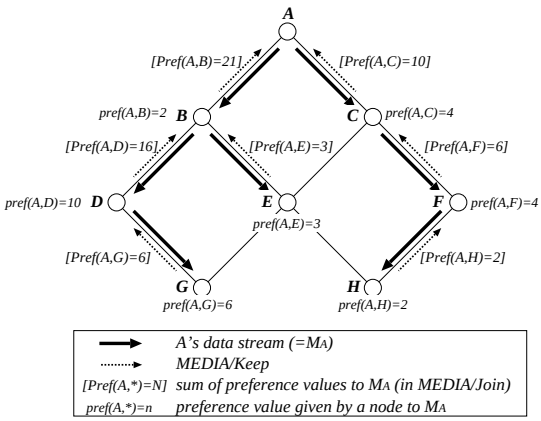


図 3 MEDIA/Keep メッセージの転送例

2.3.1 MEDIA/Keep メッセージ

以下 $pref(s,v)$ で，ノード v がノード s のデータ（以下 M_s で表す）に対し指定したプリファレンス値を表すとする．ノード v は，M_s を受信しているなら，$pref(s,v)$ を含む MEDIA/Keep メッセージをノード s の SPT（以下 SPT_s で表す）における上流ノードに定期的に送出する．またノード v が M_s を他ノードに転送しているなら，v はそれらのノードからの MEDIA/Keep メッセージを受け取り，それらに含まれる値に自身のプリファレンス値を加えた値を含ませた MEDIA/Keep メッセージを上流ノードに送出する．結果として，あるノード v が送出する MEDIA/Keep メッセージには，v 及び v の下流でそのデータを受信するノードのプリファレンス値の総和（以下集約プリファレンス値とよび，$Pref(s,v)$ で表す）が含まれることになる．なお，MEDIA/Keep を受け取ったノードはそれに含まれる集約プリファレンス値を一定期間保持しておく．このように，プリファレンス値を定期的に集約することで，後述する MEDIA/Join メッセージによる受信要求の受け入れ計算が短い時間で可能となるようにする．

図 3 に例を示す．ノード B はノード D，E からそれぞれデータ M_A に関する MEDIA/Keep メッセージを受け取り，それらに含まれる集約プリファレンス値 $Pref(A,D)$ 及び $Pref(A,E)$（16 及び 3）に自身のプリファレンス値を加えた集約プリファレンス値 $Pref(A,B)$（21）をノード A に向かう上流ノード

（ここではノード A 自身）に送出する．また，ノード B は受け取った集約プリファレンス値 $Pref(A,D)$ 及び $Pref(A,E)$ を保持しておく．

2.3.2 MEDIA/Join メッセージ

MEDIA/Join メッセージは，あるノード v が他ノード s からのデータ M_s の受信を要求する場合に SPT_s における上流ノードに送出するメッセージである．MEDIA/Keep メッセージと同様に，他のノードが M_s に対する MEDIA/Join メッセージを送出した場合にはそれらのメッセージは集約されながら SPT_s 上を s に向けて転送される．すでに M_s を受信しているノードに MEDIA/Join メッセージが到着した場合には，そのノードはメッセージを受け取ったノードへの M_s の転送を開始すると共に，MEDIA/Accept メッセージを送出する．データ M_s 及び MEDIA/Accept メッセージは，M_s を要求したノードに到着するまで順に転送される．

ここで，オーバレイリンクのデータストリーム数制限（容量制限）により，大きいプリファレンス値で要求されたデータストリームを転送できない場合がある．例えば図 4(a) は，ノード G 及びノード H が M_A を要求する MEDIA/Join メッセージを送出した様子を示している．これらのメッセージはノード E で集約され，ノード B へと転送される．ノード B はすでに M_A を受信しているため，ノード E への M_A の転送を試みるが，ノード G，ノード H に配送するためには，オーバレイリンク B-E，E-G 及び E-H 上に少なくとも 1 データストリーム分の空き（空き容量 1）が必要となる．しかし，この例では各オーバレイリンクの容量は 2 であり，それらのリンク上ですでに 2 本のデータストリームが転送されているために空き容量がない．

Emma では，これらのオーバレイリンク上で空き容量 1 を生じるために既存のストリームの転送を最小のプリファレンス値損失で停止する方法を計算できる．Emma の利点の 1 つは，この計算が MEDIA/Join メッセージを転送するノード上で，メッセージの転送に伴いボトムアップで行われるため，計算時間や計算のためのデータ収集にかかる時間やメッセージ交換によるオーバヘッドをほとんど伴わないことである．以下でその方法を説明する．

以下，MEDIA/Join メッセージが転送された SPT_s の部分木を要求木とよぶ．例えば図 4(a) においては，ノード B を根とする SPT_A の部分木（B-E, E-G 及び E-H からなる木）が要求木となる．また，$loss(i,v)$ で，ノード v の SPT_s 上の親ノード（ノード u とする）を根とする要求木において，オーバレイリンク u-v 上のデータ M_i の配送停止を含む停止方法により実現される最小のプリファレンス値損失値を表すとする．$loss(i,v)$ は以下のように再帰的に定義することができる．

$$
\begin{aligned}
&loss(i,v)\\
&\quad = \quad pref(i,v)\\
&\qquad\quad + \sum_{w_a} Pref(i,w_a)\\
&\qquad\quad + \sum_{w_b} loss(i,w_b)\\
&\qquad\quad + \sum_{w_c} \min{}_j \{loss(j,w_c)\}
\end{aligned}
$$

この定義の基本アイディアは以下の通りである．まず，ノード v の各隣接ノードを，(a) データ M_i を v を介して受信しているが，M_s の要求木上にはいないノード w_a，(b) データ M_i を v を介して受信しており，かつ M_s の要求木上に存在するノード w_b，(c) データ M_i は受信していないが M_s の要求木上に存在するノード w_c，(d) 上記に該当しないノードに分類する．w_a およびその下流で M_i を受信しているノードは，v-w_a に空き容量を確保する必要はないが，v が M_i の転送を停止することで M_i に対する集約プリファレンス値 $Pref(i,w_a)$ を損失する．一方，w_b は要求木上に存在するために空き容量を必要とするが，v が M_i の転送を停止することで少なくともオーバレイリンク v-w_b 上には空き容量が生じる．したがって，w_b 以下の要求木上に空き容量を確保するための最小損失は $loss(i,w_b)$ となる．最後に，w_c は M_i 以外のデータストリームの転送を停止して空き容量を確保する必要があるため，損失が最小となるようなデータ M_j を選択した場合の $loss(j,w_c)$ が損失となる．以上より，ノード w がノード v に送出する MEDIA/Join メッセージには，v-w 上の各データス

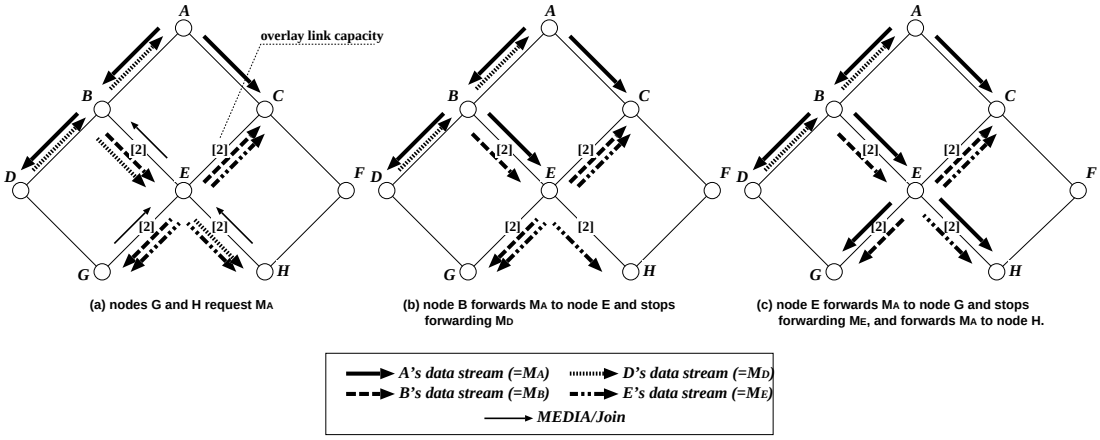


図 4 MEDIA/Join メッセージの転送例

トリーム M_k に対する $loss(k,w)$ を含めることにより，ノード v は，u-v 上の各データストリーム M_i に対する $loss(i,v)$ を計算することができる（なお，集約プリファレンス値は MEDIA/Keep メッセージにより収集したものを利用できる）．

上記の手順を例で示す．図 4(a) においてノード G と H が M_A を要求している状況で，ノード E はそれらからの MEDIA/Join メッセージを受取り，新たに MEDIA/Join メッセージをノード B に送出する．この場合，ノード E はオーバレイリンク B-E 上のデータストリーム M_B 及び M_D に対しそれぞれ $loss(B,E)$ 及び $loss(D,E)$ を計算する．ここで，$loss(B,E)$，$loss(D,E)$ はそれぞれ M_B，M_D の受信をノード E が停止した場合の最小のプリファレンス値損失である．なお，この例では B-E，E-G，E-H からなる M_A の要求木上に空き容量は存在しない．

$$\begin{aligned} loss(B,E) \\ = \ & pref(B,E) \\ & + Pref(B,C) \\ & + loss(B,G) \\ & + \min\{loss(D,H), loss(E,H)\} \quad (1) \end{aligned}$$

上式の意味は以下の通りである．ノード C は要求木上には存在しないが M_B を受信しているため，その下流ノードを含めた集約プリファレンス値 $Pref(B,C)$ を失う．ノード G は要求木上で M_B を受信しているために $loss(B,G)$ を失う．また，ノード H は要求木上に存在するが M_B は受信していないために，データストリーム M_D または M_E のうち損失が小さいものの配送を停止した際のプリファレンス値を失う．同様に，

$$\begin{aligned} loss(D,E) \\ = \ & pref(D,E) \\ & + loss(D,H) \\ & + \min\{loss(B,G), loss(E,G)\} \quad (2) \end{aligned}$$

なお，式（1）及び（2）の右辺の項は MEDIA/Join メッセージあるいは MEDIA/Keep メッセージからすべて得られるので，ノード E は $loss(B,E)$ 及び $loss(D,E)$ を計算することができ，これらを含んだ MEDIA/Join メッセージをノード B に送出する．

2.3.3 MEDIA/Accept・Reject メッセージ

M_s を要求する MEDIA/Join メッセージが，それをすでに受信しているノード（ノード z とする）に，その下流ノード（ノード u とする）から到着したとする．この場合，ノード z は M_s に対するプリファレンス値総和と MEDIA/Join に含まれるプリファレンス値損失の最小値を比較し，この受信要求を受け入れるか否かを決定する．例えば図 4(a) ではノード B がノード z に相当する．ノード B は $\min\{loss(B,E), loss(D,E)\}$（仮に $loss(B,E) > loss(D,E)$ とする）と，M_A に対するプリファレンス値を比較する．もし後者が大きければ，ノード B は，転送が停止されるデータとして M_D が指定された MEDIA/Accept メッセージを

ノード E に送信するとともに M_A のノード E に向けての転送を開始する（図 4(b)）．ノード E はその MEDIA/Accept メッセージを受け取ると，式 (2) の $loss(D,E)$ の計算式（仮に $loss(B,G) > loss(E,G)$ であったとする）に基づき，M_E のノード G への転送を停止し，M_A をノード G 及びノード H に向けて転送する（図 4(c)）．ノード E は，ノード G へは M_E が指定された MEDIA/Accept メッセージを，ノード H へは M_D が指定された MEDIA/Accept メッセージをそれぞれ送出する．

2.3.4 MEDIA/Adaptation メッセージ

一般にエンドホスト間の帯域は一定ではなく，輻輳などの影響を受けることも考えられる．したがって帯域減少がある一定期間継続する場合はオーバレイリンク上での適切なレート調整を行うことが品質維持の観点から望ましいといえる．

Emma では既存のデータストリームの品質劣化を検知した場合は，MEDIA/Join の場合と同じ計算方法を用いることで各オーバレイリンクのデータストリーム数を調整するレート調整方法を採用する．もしあるノードが受信データ M_s の品質劣化を検知した場合，そのノードは MEDIA/Adaptation メッセージを SPT_s における上流ノードへと送出する．これにより，M_s の配送を停止するか，或いは他のデータストリームの配送を停止するかを MEDIA/Join と同じ方法で計算及び決定することができる．

2.4 制御メッセージ量の調節

Emma では，制御メッセージによるオーバヘッドをなるべく少なくするため，各ノードで受け取る MEDIA/Keep メッセージや MEDIA/Join メッセージはある定められた期間（周期期間）ごとに集約して送信する方法を採用する．また，異なるデータに関する MEDIA/Keep メッセージや MEDIA/Join メッセージを同じオーバレイリンクに送出する場合はそれらを 1 つにまとめて送出することでメッセージ数を削減できる．

なお，上述の周期期間と，要求に対するプロトコルの応答速度にはトレードオフがあるため，我々は次章で，MEDIA/Join の処理にかかる周期時間数を測定し，アプリケーションの特性（許容応答時間，許容制御トラフィックなど）から適切な周期時間長を決定するための目安となるようにしている．測定結果の詳細は次章を参照されたい．

3 性能評価

Emma の性能評価のため，我々はスクリプト型言語 Ruby [3] により Emma を実装し，離散イベント型シミュレーションにより性能評価を行った．

3.1 シミュレーションシナリオ

ネットワークは LAN，MAN 及び WAN より構成される階層型トポロジを tiers モデル [2] に基づき生成し，LAN に属するノードの約 50%をエンドホスト（セッションに参加するユーザ）としている．また，LAN，MAN，WAN の帯域はそれぞれ平均で約 2Mbps，4Mbps 及び 10Mbps となるようランダムに決定し，各ユーザのデータストリーム（映像）は 200kbps であるとした．各ノードはセッションの参加時に 3 つのノードとオーバレイリンクを確立し，1 ノードのオーバレイリンクの最大数は 6，各オーバレイリンクの容量は 3，4，5 のいずれかとした．

また，シミュレーションシナリオとして，電子会議を想定した以下のシナリオを利用した．各ユーザはランダムな順番で順次セッションに参加し，参加後は 3 つのデータの受信要求を行うようにした．また，ユーザ i (i=1..N) のビデオに対し，各ユーザのプリファレンス値を Zipf の法則に基づき 2N/i と設定した．これにより，例えば会議の議長や本会議会場の映像は多くの人が受信しようとするためプリファレンス値は一般に高い，などプリファレンスの偏向性を表現する．さらに，ユーザはより大きいプリファレンス値を指定した未受信のデータがあればそれを要求し続けるとする．

3.2 オーバレイネットワークによるオーバヘッドの測定

Emma ではオーバレイリンク上での SPT 同士の重複度が少ない方がより効率的であると考えられる．したがってオーバレイリンクあたりいくつの SPT が

存在するかを測定した．

さらに，オーバレイネットワークでのノード間の経路は実際のネットワーク上の単一リンクを複数回含んでいる可能性があるため，単一のパケットが単一実リンクを複数回配送される可能性がある．したがって，1 つの SPT が実リンクあたり何回含まれるかの重複度を調べた．

最後に，Emma では SPT を採用しているため，オーバレイネットワーク上での 2 ノード間遅延は最小になるが，実ネットワークでの 2 ノード間遅延と比較した場合，一般にはそれより大きくなる．したがって，実ネットワークでの最小遅延（ユニキャスト遅延）に対する Emma でのノード間遅延の比を，ホップ数をもとに比較した．

測定はノード数が 56，104，146 のネットワークについてそれぞれ 10 回の試行を行い，その平均値を用いた．

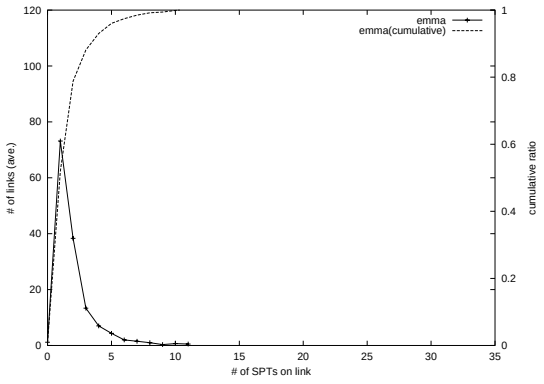
3.2.1 オーバレイリンクあたりの SPT 重複度

オーバレイネットワークの 1 リンクあたりいくつの SPT が存在するかを測定し，その分布を図 5 に示した．図の X 軸は SPT 数を，Y 軸は全オーバレイリンク中その SPT 数が存在するリンク総数をそれぞれ表す．

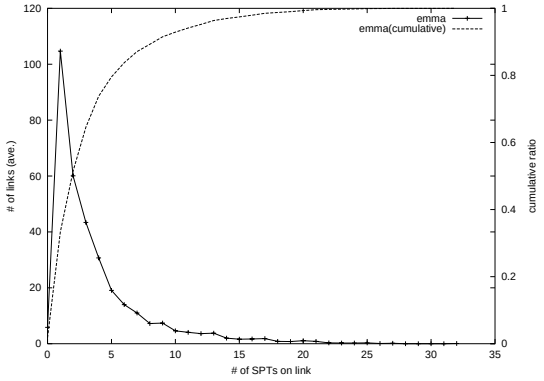
図 5(b) より，ノード数 104（平均ユーザ数 35）のオーバレイネットワークにおいて，オーバレイリンクの約 80%の SPT 数は高々5，ノード数 146（平均ユーザ数 66）においても 10 程度に抑えられていることがわかる．

3.2.2 実リンクあたりの単一 SPT の重複度

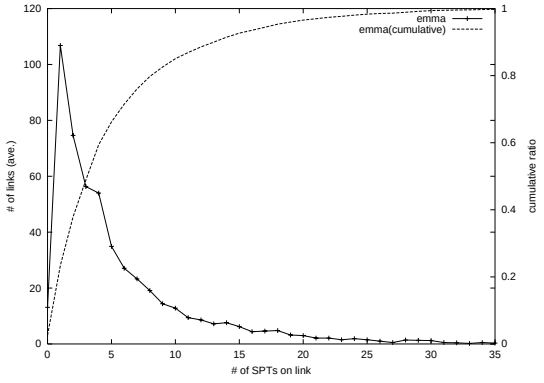
3.2.1 項と同じ実験において，各実リンクが単一の SPT に何回含まれるかを測定し，その分布を図 6 に示した．X 軸は 単一の SPT に含まれる回数を，Y 軸が全実リンク中その回数含まれる実リンクの総数を表す．なお，比較のため，その SPT の根となるノードから他の各ノードへのユニキャスト最短経路群について同様の測定を行った．特に図 6(c) からわかるように，ネットワークの規模が大きくなるにつれて，ユニキャストは非常に重複度の高い実リンクが全体の数%から 5%程度みられる．これはユニキャストにおいてはボトルネックとなり得るリンクが存在すること



(a) 56 ノード（ユーザ数平均は 17）



(b) 104 ノード（ユーザ数平均は 35）



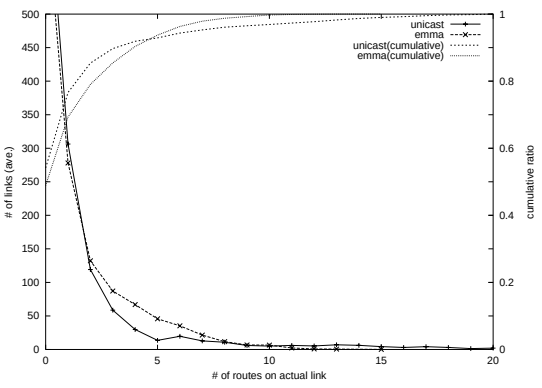
(c) 146 ノード（ユーザ数平均は 66）

図 5 オーバレイリンクあたりの SPT 数の分布

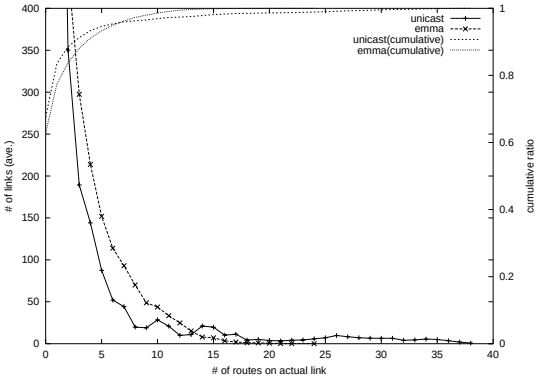
を表しており，これと比較した ALM の有用性がわかる．なお，図 6(a)，(b)，(c) の X 軸の目盛単位はそれぞれ異なることに注意されたい．

3.2.3 ユニキャストに対するオーバレイ SPT のホップ数

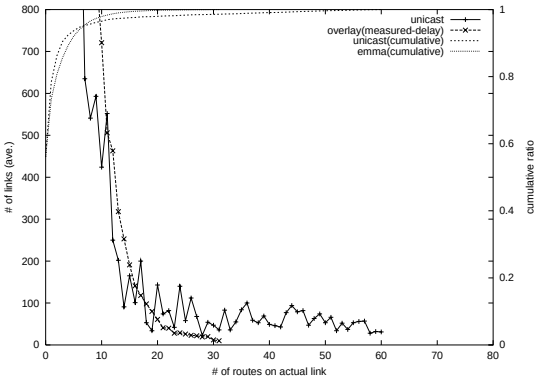
最後に，ユニキャストの最短経路でのホップ数及び



(a) 56 ノード（ユーザ平均は 17）



(b) 104 ノード（ユーザ数平均は 35）



(c) 146 ノード（ユーザ数平均は 66）

図 6　実リンクあたりの単一 SPT の重複度の分布

オーバーレイネットワークでの最短経路（SPT）でのホップ数を測定し，前者に対する後者の比の分布を図 7 に示した．X 軸は比を，Y 軸はその比となる経路数を表す．

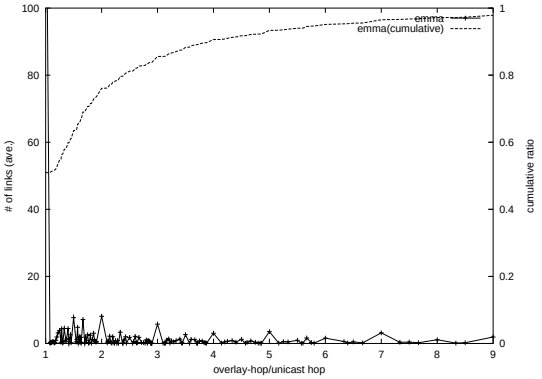
文献 [16] では，TV 電話を用いた自由会話における許容限度は往復で約 400ms と報告されている．これに基づきオーバレイネットワーク上でのホスト間の許容遅延をおおよそ 200ms と仮定する．一方，我々の組織から国内の著名ないくつかの組織に対して ping による往復遅延の測定を行ったところ，その多くは 100ms 以下であったことから，ユニキャスト遅延をおおよそ 50ms 程度とする．これらから，ホップ比の許容値はおおよそ 4 であると考えられる．ここで，グラフより，比が 4 以下の経路数は全体の約 80%を占めていることから，Emma における遅延は十分妥当であるといえる．

3.3　ユーザ満足度

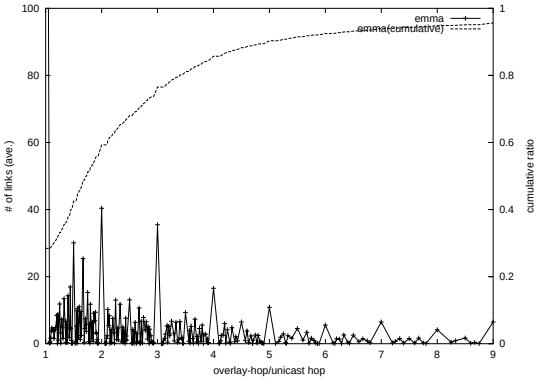
（リンクに空きがある限り）要求順に受け入れ，空きがない場合は受け入れない方式（First-Come-First-Serve 方式，以下 FCFS 方式とよぶ）に対し，Emma がどの程度のユーザ満足度を達成しているかを，ネットワークノード数（及びそれに伴うユーザ数）を変化させ測定した．図 8 にその結果を示す．いずれもノード数（ユーザ数）の増加に比例して増加するが，Emma ではプリファレンスに基づく動的な制御を行っているため，FCFS 方式の 1.5 倍程度の値を達成していることがわかる．

3.4　応答時間

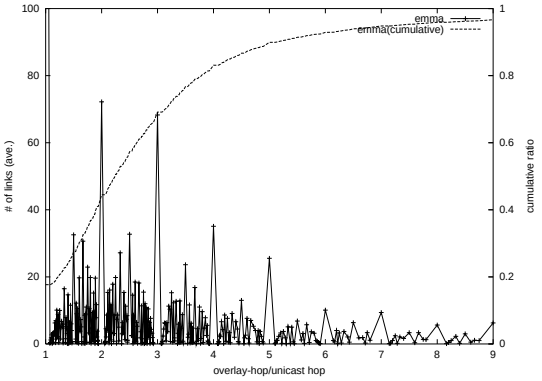
Emma では，各ノードが制御メッセージを一定の周期時間ごとに集約し，次の周期に送出することで制御トラフィックによるプロトコルオーバヘッドを削減し，またトラフィック量の見積りを可能としている．しかし，受信要求メッセージが隣接ノードに転送されるのには周期時間長の時間が必要となるため，受信要求に対し受け入れが完了するまでの時間は，受信要求メッセージの往復ホップ数と周期時間長の積で決定される．ここで，アプリケーションごと異なる許容応答時間と許容制御トラフィック量に応じた周期時間長を設定するためには，受信要求メッセージが処理されるおおよそのホップ数がわかっている必要がある．そこで，受信要求メッセージ（MEDIA/Join）メッセージを送出した際にそれが上流に転送され，受け入れ可能性の判定計算がなされるまでのホップ数を測定した．結果を図 9 に示す．この結果より，アプリ



(a) 56 ノード（ユーザ数平均は 17）



(b) 104 ノード（ユーザ数平均は 35）



(c) 146 ノード（ユーザ数平均は 66）

図 7　ユニキャストに対するオーバレイ SPT のホップ数の比の分布

ケーション開発者はアプリケーション特性（許容最大遅延）やネットワーク特性（帯域など）に応じて周期時間長を決定することができると考えられる．

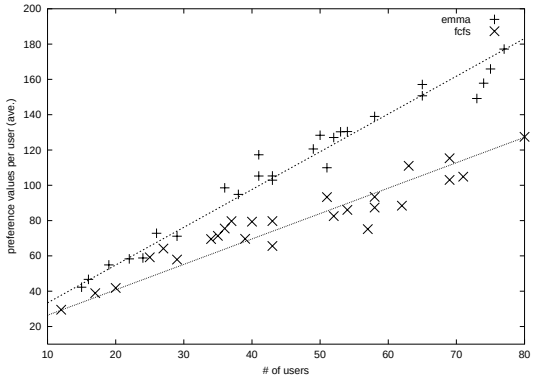


図 8　ユーザあたりの平均プリファレンス値の変化

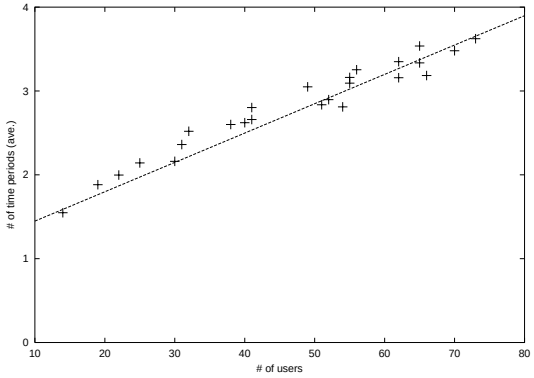


図 9　MEDIA/Join メッセージが処理されるのに必要なホップ数

4　おわりに

本論文では，映像を利用した複数人によるコミュニケーションシステム向けのアプリケーションレベルマルチキャストプロトコル Emma の提案を行い，シミュレーション実験によりその性能評価を行った．実験結果から，要求順に受け入れる単純な方式と比較し，Emma は 1.5 倍程度のユーザ満足度を達成できることがわかった．また，オーバレイネットワークによるオーバヘッドも十分許容できる程度に収まっていることが確認できた．

現在我々は Emma プロトコルを元に，映像を利用した複数人によるコミュニケーションシステム向けのミドルウェアを設計し，Java により実装中である．そのミドルウェアを用い，実ネットワークにおいて会

議アプリケーションなどを動作させる実証実験を行い，Emma の有効性を確認していくことが今後の課題である．

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 (1997), pp. 277–290.

[2] Calvert, K. L., Doar, M. B. and Zegura, E. W. : Modeling Internet Topology, *IEEE Communications Magazine*, pp. 160–163, 1997.

[3] Ruby Home Page. `http://www.ruby-lang.org/`

[4] Chu, Y. -H., Rao, S. G., Seshan S. and Zhang, H. : Enabling Conferenceing Applications on the Internet using an Overlay Multicast Architecture, *Proc. of ACM SIGCOMM*, 2001.

[5] Chu, Y. -H., Rao, S. G. and Zhang, H. : A Case for End System Multicast, in *Proc. of ACM SIGMETRICS*, 2000.

[6] 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.

[7] Jannotti, 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.

[8] Roca, V. and El-Sayed, A. : A Host-Based Multicast (HBM) Solution for Group Communications, in *Proc. of 1st IEEE Int. Conf. on Networking* (*ICN'01*), 2001.

[9] Cohen, R. and Kaempfer, G. : A Unicast-based Approach for Streaming Multicast, in *Proc. of IEEE INFOCOM2001*, 2001.

[10] Ratnasamy, S., Handley, M., Karp, R. and Shenker, S. : Application-level Multicast using Content-Addressable Networks, in *Proc. of 3rd Int. Workshop on Networked Group Communication*, 2001.

[11] Chawathe, Y., McCanne, S. and Brewer, E. A. : RMX: Reliable Multicast for Heterogeneous Networks, in *Proc. of IEEE INFOCOM2000*, 2000.

[12] Zhuang, S. Q., Zhao, B. Y., Joseph, A. D., Katz, R. H. and Kubiatowicz, J. : Bayeux: An Architecture for Scalable and Fault-tolerant Wide-area Data Dissemination, in *Proc. of ACM NOSSDAV 2001*, 2001.

[13] Francis, P. : Yoid: Extending the Internet Multicast Architechture, *Unrefereed Report*, 2002. `http://www.isi.edu/div7/yoid/`

[14] Baccelli, F., Kofman, D. and Rougier,J. L. : Self Organizing Hierarchical Multicast Trees And Their Optimization, in *Proc. of IEEE INFOCOM2001*, 2001.

[15] 中村，山口，廣森，安本，東野，谷口： 映像による複数人のコミュニケーションシステム向けのアプリケーションレベルマルチキャスト Emma の性能評価, マルチメディア，分散，協調とモバイル（DICOMO2002）シンポジウム論文集，2002, pp. 253–256.

[16] 栗田，井合，北脇： オーディオビジュアル通信における伝搬遅延の影響, 電子情報通信学会論文誌，Vol.J76-B-I，No.4 (1993), pp. 331–339,