

# 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

中村嘉隆 山口弘純 廣森聡仁 安本慶一† 東野輝夫 谷口健一

大阪大学 大学院情報科学研究科 † 奈良先端科学技術大学院大学 情報科学研究科

## A study on performance evaluation of application layer multicast protocol Emma

Yoshitaka Nakamura, Hirozumi Yamaguchi, Akihito Hiromori, Keiichi Yasumoto†,

Teruo Higashino and Kenichi Taniguchi

Graduate School of Information Science and Technology, Osaka University

† Graduate School of Information Science, Nara Institute of Science and Technology

### 1 はじめに

高速ネットワークや計算機の普及は、近い将来、比較的小規模（数人から数十人程度）の多数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におけるデータ交換を限られた数のサーバが処理することは、ネットワーク資源及びコネクション数の制限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マルチキャストが有用な通信技術の一つであると考えられる。しかし、特に広域網（バックボーン）でのインフラストラクチャの整備の問題から、現状での利用は難しい。一方、すべての 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、グループサイズのスケールの点で非効率である。

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, ALM）が注目を集めており、近年になり多くの研究がなされている [5, 6, 7, 10]。ALM はエンドホスト間ユニキャストトンネリングによるオーバーレイネットワーク上で、それらエンドホストがマルチキャスト配信木を管理し、トンネリング間のパケット複製及び転送を行な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。また、図 1(b) はマルチキャスト配信木を表す。

トンネリングであるために特定のインフラストラクチャを必要としないこと、ユニキャストトランスポートプロトコルの制御機能（信頼性保証、フロー/レート制御など）が利用可能であるなど利点が多い。しかしその一方で、エンドホスト付近での帯域制約や実リンク上でのパケット重複などオーバーレイネットワーク上での制約やオーバーヘッドを考慮した性能のよいプロトコルが要求される。さらに、アプリケーション層のプロトコルであることから、ユーザアプリケーションの特性を考慮した柔軟な設計が望まれる。

本稿では、我々が文献 [1] において電子ビデオ会議などのグループ通信向けに提案した ALM プロトコルである Emma (End-user Multicast for Multi-party Applications) の性能評価を行ない、その有用性を評価する。Emma は、(1) 数台から数十台のエンドホストがそれぞれ同時並行かつ継続的に他のエンドホストに映像データを送出し、(2) 各エンドホストが扱える（受信または転送できる）映像数は、処理能力及びエンドホスト付近の帯域を考慮した場合には一定の制限があり、(3) 各エンドホストが受信したい映像には偏りがある（プリファレンスがある）、といった特徴を持つユーザアプリケーション（主にビデオ会議など）を対象とし、オーバーレイネットワークの自律的構築とプリファレンスに応じた映像の動的転送制御を完全な分散制御で行なう。シミュレーション実験により、オーバーレイ

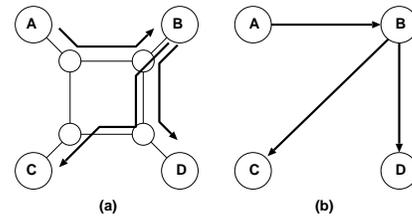


図 1: アプリケーションレベルマルチキャスト

ネットワークによるオーバーヘッド及び Emma の制御メッセージによるオーバーヘッドを測定することで、仮定する環境やアプリケーションにおいて妥当なオーバーヘッドで Emma が実現可能であることを示す。さらに、要求順に映像を配送する単純な方式と比較し、Emma で満足されるプリファレンスが十分に大きいことを示し、Emma がそれらのアプリケーションで有用であることを示す。

関連研究 トンネリングマルチキャストとしての ALM の概念は MBone をはじめ新しいものではないが、エンドホスト間での P2P 型（各ホストが対等な立場で処理を行なう方式）の ALM に関しては比較的最近の研究が多い。表 1 に関連研究の一部を主な特徴により分類しているため、参照されたい。

我々が提案する Emma は、主に複数ユーザによる映像の同時並行配信を動的かつ完全な分散で制御する点で他のアプローチとは異なっている。

## 2 プロトコル Emma の概要

### 2.1 Emma の機能

オーディオ配送制御機能 Emma は利用帯域が小さく遅延に敏感なオーディオのようなタイプのメディア向けに遅延をメトリックとした共有スパンニングツリーを、ノードがアプリケーションに参加する際に継ぎ木方式で構築する。この木上のエンドホスト間最大遅延がアプリケーションの許容遅延以下になるように接続先を選択する。

相互接続性維持機能 各エンドホストはオーディオの配送木を元にエンドホストの離脱や障害に対してもエンドホスト間最大遅延とエンドホスト間の相互接続性を保証するように代替経路を計算しておく。

ビデオ配送機能 遅延と帯域を要するビデオのようなタイプのメディア向けには、遅延をメトリックとし、各エンドホスト

	Underlying	Tree	Metric on Tree Construction	Scale	Target Application	Other Features
Narada [6]	Mesh	SPT	Delay	Small group	Host-based application	
[5]	Mesh	SPT	Bandwidth, then delay	small group	Host-based Conferencing (one sender at any time)	
ALMI[7]	Constant Number of Neighbors	Steiner	Delay	Small group	Host-based application	Java middleware
HBM[9]	Central RP-based	N/S	N/S	Small group	Mobile	
M-RTP[10]	N/S	N/S	RTP Parameters	Small group	Multimedia application	Centralized
RMX(Scattercast)[11]	N/S	Spanning	N/S	N/S	Reliable Data Transfer	
Yoid[12]	Mesh	Spanning	N/S	Small group	Host-based Conferencing	Simple middleware (vic wrapper)
Emma[1]	Constant Number of Neighbors	SPT	Delay, Bandwidth, Preference	Small group	Host-based Conferencing (Simultaneous Video Transmission)	Distributed, Adaptive Control of Video Forwarding

N/S=No Specific

表 1: アプリケーションレベルマルチキャストの関連研究 (一部)

を根とするソーススペースの配送木を各エンドホストの受信要求をトリガとしたオンデマンドの継ぎ木方式で構築する。

なお、ビデオ配送においてはそれぞれがある程度の帯域を恒常的に消費するため、オーバーレイリンク上に空き帯域を十分に確保できない場合も多い。その場合、アプリケーションとそのユーザに依存するメトリックであるプリファレンス値に基づき、既存のメディアの配送停止によるプリファレンス値の損失をなるべく小さくしながら空き帯域を確保できる可能性のあるリンクを選択する。

**エンドホストの負荷削減** アプリケーションレベルマルチキャストでは制御メッセージ数、接続するオーバーレイリンク数、転送ビデオ数などがエンドホストの負荷に影響を与えると考えられる。そこで、一定の周期で各エンドホストを同期させ、周期内に一オーバーレイリンク上に送信される制御メッセージを集約することで制御メッセージ数の削減をはかり、周期時間の調整による制御トラフィックの調整を可能とする。

## 2.2 プロトコルの手順

### 2.2.1 オーバレイリンクの確立

エンドホスト(以下ノードと呼ぶ)はアプリケーションのセッション参加時にロビーサーバの役割を持つ適当なサーバに自身の IP アドレスを登録し、既にアプリケーションに参加しているノードの IP アドレスリストを入手する。それらへの遅延を計測し、遅延が小さい数個のノードへトランスポートレベルのコネクション設立要求を送出し、オーバーレイリンクを構築する。

### 2.2.2 オーディオ配送木の構築

オーディオタイプのメディア配送木はオーバーレイリンクを構築したノードが既に存在する木に継ぎ木する形で構築する。セッションに参加したノード  $v$  はオーバーレイリンクを構築したノード  $u$  から、既存の共有木の  $u$  までのノード間最大遅延を受け

取り、 $u-v$  間の遅延を加え、それがなるべく小さい  $u$  に対して共有木に接続する。

また、 $u$  を親ノードとした場合の  $u$  の親ノード  $u'$  ともオーバーレイリンクを構築しておき、これをノードの離脱時や障害発生時の代替経路とする。これによってノード間最大遅延及び相互接続性を保証できる。

### 2.2.3 ビデオ配送木の構築

**プリファレンス値集約** Emma では、各ノード  $v$  ごとにビデオ  $V_s$  に対するプリファレンスの程度を整数値  $pref(s,v)$  として持っているとする。各ノード  $v$  は受信中のメディア  $V_s$  に対し、 $pref(s,v)$  を含むメッセージ KeepAlive を、適当な時間周期間隔ごとに上流ノードに送出する。また  $V_s$  を子ノードに転送している場合はそれらのノードからの  $V_s$  に対するプリファレンスを含むメッセージを受信し、 $v$  以下の  $V_s$  に対するプリファレンス値総和(以下  $Pref(s,v)$  で表す)を計算し、それを含むメッセージを次の周期時間に上流ノード  $u$  へ転送する。

なお、 $v$  が同じ親ノードから複数のビデオを受信している場合は、それらに関するプリファレンスは 1 つのメッセージにまとめて送出する。これにより、一オーバーレイリンク上に一周期時間あたりに送出されるメッセージを一つにできる。

**ロス値集約** ノード  $v$  はノード  $s$  からのビデオ  $V_s$  の受信要求を行う場合、 $s$  への遅延時間がセッションの許容遅延以下でありかつオーバーレイリンクに空き帯域(スロット)がある隣接ノード  $u$  を一つ選んで適当な周期時間ごとにビデオ配送要求メッセージを送信する。

ただし、いずれのリンクにも空き帯域がない場合、各隣接ノード  $u$  と、 $(u,v)$  上を  $v$  へ配送されている各メディア  $V_i$  について、VIDEO/JoinREQ が転送されてきた各リンク上(これらのリンクで構成される木を  $V_s$  の要求木とよぶ)に少なくとも 1 つの空きスロットを確保するために、 $(u,v)$  上での  $V_i$  の

配送停止を含む既存のビデオの配送停止方法のうち、最もプリファレンス値損失が小さいような配送停止方法の損失を計算する (計算法の詳細は文献 [1] 参照)。

**配送の変更** ビデオ配送要求メッセージを受け取ったノード  $v$  上に  $V_s$  がすでに配送されている ( $v = s$  の場合も含む) 場合、 $v$  はその各下流ノード  $w$  に配送しているビデオ  $V_i$  のうち、 $V_i$  の配送停止によるプリファレンス値の損失が最小のものについて、その値が  $Pref(s, w)$  より小さいなら、 $V_i$  の  $V_s$  への変更命令を、そうでなければ変更不可通知メッセージを  $w$  に向けて送信する。

### 3 シミュレーションによる性能評価

ネットワークシミュレータ ns-2 [3] 上に Emma を実装し、性能評価を行なった。

#### 3.1 ネットワークモデルと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、3.2 節の実験 (2), (3) に関しては、電子会議を想定し、以下のシミュレーションシナリオのもとで実験を行った。

- 各ユーザのセッション参加時刻は、会議開始時刻で標準化した正規分布に基づく。
- 各ユーザは参加時刻より 20 秒後に 4 つのビデオ要求メッセージを送出する。
- 各ユーザの、ユーザ  $i$  ( $i = 1, \dots, N$ ) のビデオに対するプリファレンスを zipf モデルに基づき  $2N/i$  に設定した。これにより、プリファレンスの偏向性を表現する。例えば会議の議長や会議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、などである。
- ユーザはより高いプリファレンスを指定した未受信のビデオがあればそれを要求し続ける。

なお、制御メッセージのサイズは 200 バイトとした。

#### 3.2 評価項目

オーバーレイネットワークのオーバーヘッドに関する評価 Emma では、ビデオはホップ数 (リンク数) をメトリックとするオーバーレイネットワーク上の最短経路木上で配送されるが、一オーバーレイリンクは複数の実リンクからなるため、オーバーレイネットワーク上の 2 ユーザ間の最短経路に含まれる実リンク数 (以下オーバーレイ最短ホップとよぶ) は、最短経路木を用いるネイティブマルチキャストやユニキャストの最短経路に含まれる実リンク数 (以下最短ホップとよぶ) と比較した場合に多くなる。実験では Emma におけるその傾向を調べた。図 2 はノード数 122, ユーザ数 42 のネットワークにおけるすべての 2 ユーザ間の最短経路について、最短ホップに対するオーバーレイ最短ホップの比の分布 (10 回試行の平均) を表す。各値は存在割合 (左縦軸)、連続線はその累積割合 (右縦軸) を表す。

グラフより、比が 1.5 以下の経路数は全体の訳 50%, 2 以下の経路数は約 80% に達することがわかる。比が 2 であるとは遅延がおおよそ 2 倍であることを表すが、比較的遅延の大きい WAN を通る回数は高々 2 回であると考えられるため、国内規模程度のネットワークであれば十分妥当であるといえる。

また、同じ状況において、一つの実リンク上を通る最短経路数の分布を 10 回試行の平均の値で図 3 に示す。ここで、ユニキャストの最短経路数はユーザ数を  $N$  (ここでは 42) とすると  $N \times (N - 1)$ , ネイティブマルチキャスト及びオーバーレイマルチキャストのそれは  $N$  である。ただし、オーバーレイネットワークでは二つ以上の異なるオーバーレイリンクが実リンク一つを共有する場合があるため、一つの実リンク上の経路数は  $N$  より大きくなる場合がある。

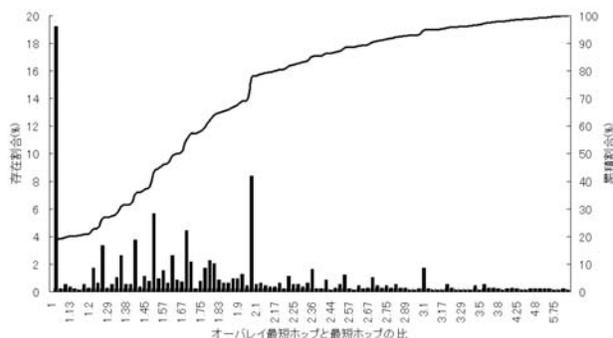


図 2: オーバレイの最短ホップの存在割合

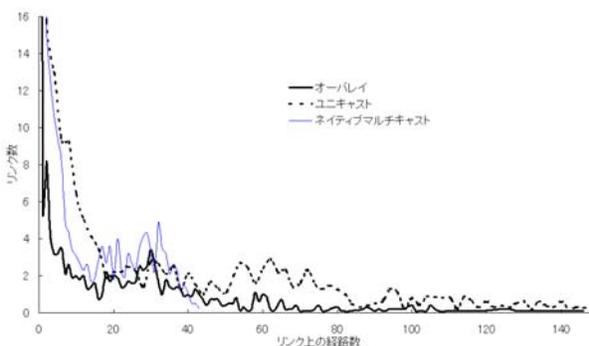


図 3: 実ネットワーク上の経路数

グラフより、経路数はネイティブマルチキャストに比較的近い傾向にあり、Emma におけるオーバーレイの経路重複によるオーバーヘッドは比較的抑えられているといえる。

**ユーザ満足度の評価** Emma が、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、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、各ネットワークノード数ごとに、2 回の試行を行い測定した。図 4 にその結果を示す。

いずれもノード数 (ユーザ数) の増加にほぼ比例して増加しているが、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 1.2 倍程度の値を達成している。

**プロトコルオペレーションに関する実験結果** Emma ではノードが送出する制御メッセージを一定の周期時間で集約し、制御トラヒックによるプロトコルオーバーヘッドを削減、さらに周期時間の制御によるトラヒック量の制御を行うとしている。しかし、周期時間を増大させた場合には、受信要求処理の遅れなどがプロトコル応答速度に影響を与える可能性が懸念される。そこで、実験ではこの周期時間を変化させた場合に、受信要求送信からその応答を受信するまでの時間 (以下応答時間とよぶ) と、実リンク上の制御トラヒックの平均値を測定した。なお、ネットワークノード数は 105, ユーザ数は 54 とした。測定結果を図 5 に示す。

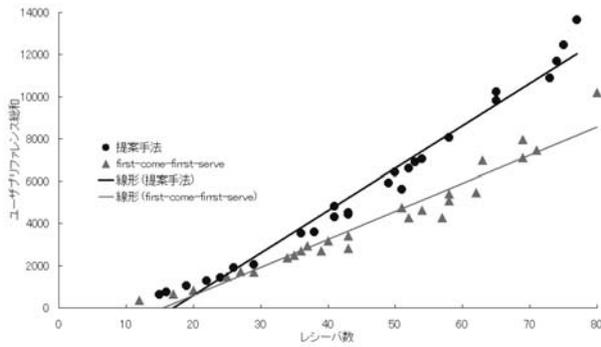


図 4: ユーザ満足度の評価

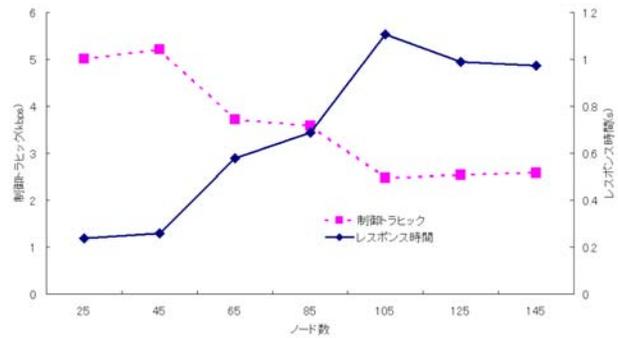


図 6: ノード数を变化させた場合

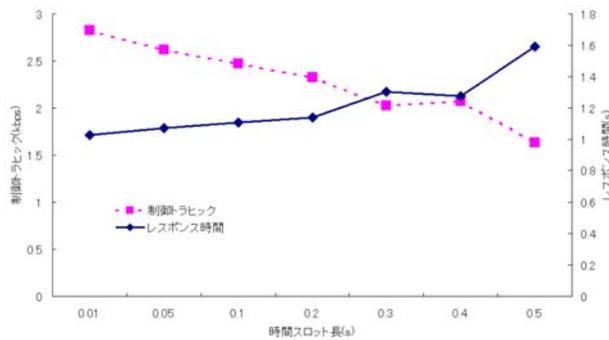


図 5: 時間スロット長を变化させた場合

グラフより、両者のトレードオフの関係は読み取れるが、周期時間に依存した大きな影響はなかった。

また、周期時間を 0.1 秒に固定した場合の、ノード数 25 から 145 までのネットワーク（ノード数は 20 ずつ増加）上での制御トラヒック量と応答時間の測定結果を図 6 に示す。

グラフより、制御トラヒック量は高々 5kbps であり、ビデオ送信には影響のない範囲であるといえる。また、応答時間に関しては、以上の結果に基づき、ネットワーク環境やアプリケーション特性に応じた周期時間の決定を行えばよいと考えられる。

#### 4 まとめ

本稿では、ユーザプリファレンスを考慮したアプリケーションレベルマルチキャスト Emma の性能評価を行なった。

今後の課題は、Emma によるメディア配信時のジッタやパケットロスの計測等、より詳細なシミュレーション実験による性能評価、Emma のワイヤレスアドホックマルチキャストへの応用などがあげられる。

#### 参考文献

[1] 山口弘純, 中村嘉隆, 廣森聡仁, 安本慶一, 東野輝夫, 谷口健一, “ユーザプリファレンスに基づく転送制御を行うアプリケーションレベルマルチキャストの一方式,” 情報処理学会研究報告, Vol.2002, No.32 (2002-DPS-107), pp43-48, 2002.

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

[3] Berkeley MASH Research Group, University of California, “The Network Simulator ns-2,” <http://www-mash.ce.berkeley.edu/ns/>, 2000.

[4] P. Parnes, K. Synnes and D. Schefstrom, “Lightweight Application Level Multicast Tunneling using mTunnel,” *Computer Communication*, 1998.

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

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

[7] D. Pendarakis, S. Shi, D. Verma and M. Waldvogel, “ALMI: An Application Level Multicast Infrastructure,” *Proc. of 3rd Usenix Symp. on Internet Technologies & Systems*, 2001.

[8] J. Jannotti, D. Gifford, K. Johnson, M. Kaashoek and J. O’Toole, “Overcast: Reliable Multicasting with an Overlay Networks,” *Proc. of the 4th Usenix Symp. on Operating Systems Design and Implementation (OSDI)*, 2000.

[9] V. Roca, A. El-Sayed, “A Host-Based Multicast (HBM) Solution for Group Communications,” *Proc. of 1st IEEE Int. Conf. on Networking (ICN’01)*, 2001.

[10] R. Cohen and G. Kaempfer, “A Unicast-based Approach for Streaming Multicast,” *Proc. of IEEE INFOCOM 2001*, 2001.

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

[12] P. Francis, “Yallcast: Extending the Internet Multicast Architecture,” <http://www.yallcast.com>.

[13] F. Baccelli, D. Kofman and J. L. Rougier, “Self Organizing Hierarchical Multicast Trees And Their Optimization,” *Proc. of IEEE INFOCOM 2001*, 2001.