

## アプリケーション層マルチキャストミドルウェアにおける 携帯端末への映像配信機構の実装

佐藤 秀 則<sup>†</sup> 假家 直 樹<sup>†</sup> 山口 弘 純<sup>†</sup>  
安本 慶 一<sup>††</sup> 東野 輝 夫<sup>†</sup> 谷口 健 一<sup>†</sup>

本稿では、アプリケーション層マルチキャストミドルウェアにおける携帯端末への映像配信機構を設計及び実装する。ミドルウェアは、電子ビデオ会議などエンドホスト間で実時間ビデオを交換するグループコミュニケーション向けのアプリケーション層マルチキャストプロトコル Emma に基づいている。ミドルウェアは基本的にデスクトップ PC のような比較的処理能力などに余裕のある端末(固定端末)から構成されることを仮定しているが、今後の無線技術の発展や端末の小型化を考慮した場合、PDA のように通信能力や表示能力に制限のある移動端末からもアプリケーションに参加したいという要求が高まってくると考えられる。本稿で設計及び実装した配信機構では、ミドルウェアを構成する固定端末上に、PDA 向けの複数のビデオの合成トランスコードサービスを提供するエージェントを配置し、PDA はいずれかの固定端末に接続してビデオの受信を行う。また本稿では PDA ユーザの受信要求を全体としてなるべく満足する最適化問題も定義している。配信機構の実装を行い、性能評価を行った結果、少ない処理遅延で適切な品質で映像を提供できることが確認できた。

### Design and Implementation of Video Transcoding Functionalities for Mobile Terminals in Application Layer Multicast Middleware

HIDENORI SATOU,<sup>†</sup> NAOKI KARIYA,<sup>†</sup> HIROZUMI YAMAGUCHI,<sup>†</sup>  
KEIICHI YASUMOTO,<sup>††</sup> TERUO HIGASHINO<sup>†</sup> and KENICHI TANIGUCHI<sup>†</sup>

In this paper, we design and implement a video distribution mechanism for mobile terminals and integrate it into our application layer multicast middleware. The middleware is based on the application layer multicast protocol called Emma, which is designed to support multi-party video communication using multiple live video streamings. The middleware is assumed to be executed on terminals with marginal computation power such as consumer PCs. However, recent innovation of wireless technologies and small devices may let users have new requirements to join such communication using small devices such as PDAs. The mechanism presented in this paper adds a functionality for each terminal to provide transcoding service to PDA users, and each PDA user connects to one of those terminals to join the communication system. The experimental results have shown that our mechanism could provide appropriate quality of service for several PDAs.

#### 1. はじめに

近年の高速ネットワークの普及や計算機端末の著しい性能向上により、今後、ビデオ会議などのグループ通信アプリケーションが、比較的小規模な多数のグループに利用されるようになって考えられる。そのようなグループ通信の新しい形態としてマルチキャスト

をユニキャストコネクションから成る仮想ネットワーク(オーバーレイネットワーク)上でサーバレスで実現するアプリケーションレベルマルチキャスト(ALM)と呼ばれる新しい通信形態が、既存環境での実現の容易さとネットワーク資源の利用効率の観点から注目されてきている。<sup>3),4)</sup>

我々の研究グループでは、エンドホスト間でビデオなどの実時間データを交換するグループ通信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<sup>5)</sup>を提案している。Emma では、複数のエンドホスト間が送信する映像にそれぞれ互いに優先度要求(プリファレンス)を指定し、その優先度に基づきオーバーレイネットワーク上の帯域使用权を決定する枠

<sup>†</sup> 大阪大学 大学院情報科学研究科  
Graduate School of Information Science and Technology, Osaka University  
<sup>††</sup> 奈良先端技術大学院大学 情報科学研究科  
Graduate School of Information Science, Nara Institute of Science and Technology

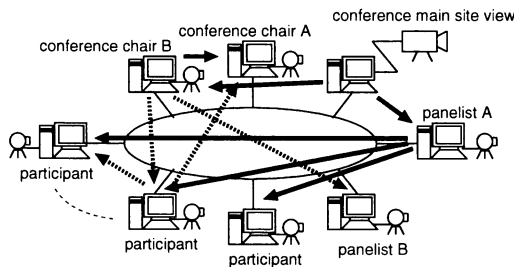


図 1 Emma アプリケーションの例

組みを持つ、多人数参加型ビデオコミュニケーションシステム向けのプロトコルである。Emma のアプリケーション例として図 1 に示されるようなビデオ会議システムがある。ビデオ会議に参加している各メンバは自身のデスクトップ上で会議全体の様子や議長の映像、もしくは他の参加者の映像などをビデオストリームとして受信、再生し、複数のウィンドウで同時に閲覧しようしたり、自分自身の映像を他の参加者へ配信する。また、各メンバは受信している複数の映像に対して、“会議全体の様子や議長の映像よりも、他の参加者の映像を優先して見たい”などの優先度要求も指定する。Emma ではこのような状況のもとで、限られたエンド間帯域をどの映像配信に割り当てるかを動的に最適化する。さらに、我々は Emma プロトコルを利用するアプリケーション開発者向けに、Emma の機能を提供する Emma ミドルウェアを構築している<sup>6)</sup>。

しかし、今後の無線技術の発展や通信料金の低価格化、さらに PDA のような小型携帯端末の普及に伴い、オフィスや自宅などからのみならず、移動先からそのような携帯端末を用いてアプリケーションに参加したいといった需要も増加すると考えられる。そのためには、デスクトップ PC などと比較し非力である携帯端末の通信環境を含む性能（動画復号能力及び送受信帯域）を考慮し、適切な品質をそのような携帯端末に提供するための枠組みがミドルウェアに望まれる。このような QoS 変換技術の 1 つとして中間ノードあるいはルータにおいて相手の環境に適するようにストリームを復号/再符号化し、シームレスに受信できるようにするトランスコーディングが利用されている<sup>8)~12)</sup>。

本研究では、複数ユーザによるビデオコミュニケーションシステム向けの ALM プロトコルのミドルウェアである Emma ミドルウェアに PDA などの携帯端末がシームレスに接続するための機能について述べる。Emma ミドルウェアは Java と Java Media Framework (JMF) を利用して実装され、各エンドホスト

(例えばデスクトップ PC) 上で動作し、Emma の基本機能（オーバーレイネットワークの構築/維持、マルチキャスト配送木構築/維持、優先度に基づく帯域使用権制御）を提供している。本稿で設計、実装した機能は、Emma ミドルウェアを用いるユーザエージェントの形で実現され、各 PDA からの複数ビデオの受信要求を受け取り、PDA に適切なビデオ (Motion JPEG) を提供する。この際、複数のビデオを個別に送信することは PDA 側の復号化のコストから適切ではないため、エージェントはそれらのビデオのフレームを圧縮、合成することで適切なフレームサイズ及びフレームレートの単一のビデオストリームを合成し、それを送信する合成トランスコーディングを行う。さらに、そのような PDA が多数存在する場合に、Emma ミドルウェアの負荷分散と各 PDA の優先度要求を満足するために各 PDA をどのエージェントに割り当てるべきかの問題を定式化している。実装実験による性能評価の結果、コンシューマレベルの PC において、数台の PDA に対しおおよそ遅延 70 ミリ秒で 3 ストリームが合成トランスコードされたビデオを提供することができた。

## 2. Emma と Emma ミドルウェア

### 2.1 Emma の概要

Emma<sup>5)</sup> はオーバーレイネットワークの構成要素である各エンドホスト（以下ノード）の振舞いに基づき以下の動作を行う。

#### オーバーレイネットワークの構築

グループに参加するノードは、まず始めに既参加ノードのプロファイル (IP アドレスやポート番号など) を、以下ロビーサーバと呼ばれる管理サーバから取得し、それを用いて既参加ノードとの遅延あるいは RTT を計測する。計測した遅延がなるべく小さいいくつかのノードと P2P リンク（以下オーバーレイリンクとよぶ）を構築することでグループメンバとなる。なお、Emma では、各ノードが発信するストリームの帯域はすべて等しいとし、オーバーレイリンクで利用可能なストリーム数をオーバーレイリンク容量と呼ぶ。オーバーレイリンク構築の際には構築相手のノードと交渉を行い、自身と相手の計算機資源や LAN の帯域、測定した遅延などからそのオーバーレイリンク容量を決定する。

#### 経路木の構築

各ノードは自身のデータを他ノードに配信するための経路木を、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構築する。また、途中参加したノー

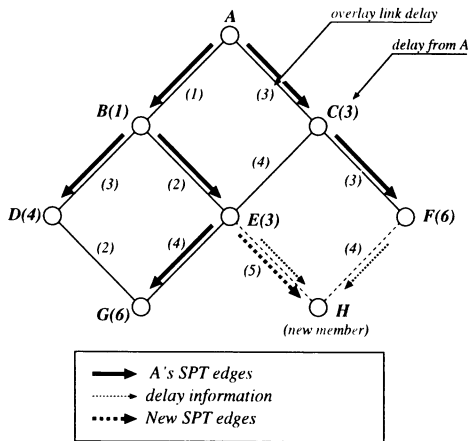


図2 経路木の構築

ドは既存の各経路木に、根ノードからの最小遅延を実現できるリンクを用いて参加する。例えば、図2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、さらにそれらとの間のオーバー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。

#### ストリームの受け入れ制御

ノードvの経路木に参加したノードは、ノードvを送信者とするストリーム(これを以下ストリームvと呼ぶ)の受信を要求できる。Emmaはオーバーレイリンク容量を考慮し、この要求を許可するか否かの受け入れ制御を行う。この際のメトリックとして、各ユーザが他のユーザのデータに対して指定する優先度要求(以下プリファレンス値とよぶ)を用いる。Emmaでは満足されるプリファレンス値がオーバーレイリンクの容量制限内となるべく大きくなるよう、複数のマルチキャストデータ転送をどのように制御するかを分散環境で効率よく決定する方法を提案している。この詳細については文献5)を参照されたい。

#### ノードの離脱時の回復処理

Emmaは、セッション途中にあるストリームの親ノードが離脱した場合、その子ノードが離脱ノードの親ノードに再接続することで、ストリーム配信を回復し、プリファレンス値の維持に努める。

### 2.2 Emmaミドルウェアの概要

我々は上記の機能をユーザアプリケーションにシームレスに提供するミドルウェアを設計、実装している。Emmaミドルウェアは各エンドホストで動作しEmmaプロトコルを実行するホストコントローラと、

参加しているエンドホストの情報を保持するロビーサーバから構成される。

#### ホストコントローラ

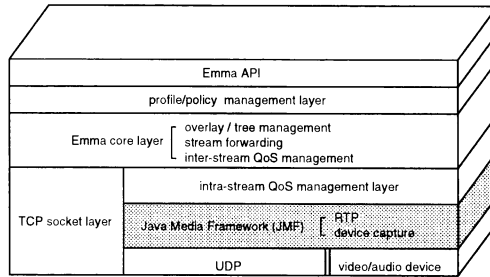


図3 ホストコントローラ構成

ホストコントローラを構成しているモジュール群を図3に示す。

ポリシー管理層(Profile/Policy Management Layer)はセッションポリシーとユーザポリシーを管理し、ユーザの振舞いがそれらのポリシーを満足しているかを判断する。セッションポリシーには、オーバーレイネットワークリンク数制限である次数制約や、ユーザが各ビデオに対して与える優先度要求(プリファレンス値)の上限などが含まれ、セッションの管理者により決定される。各ホストは後述するロビーサーバから参加時にセッションポリシーを受け取る。ユーザポリシーは他のユーザが自身とオーバーレイリンクを構築する際に適用される制約で、次数制約やオーバーレイリンク容量などが含まれる。また自身がビデオソースとなる場合には、提供するビデオに関する情報(符号化方法、フレームサイズ、フレームレートなど)を管理し、これをセッション参加時にロビーサーバに伝える。

Emma層(Emma Core Layer)はEmmaプロトコルの実行を行い、制御メッセージを交換することで、オーバーレイリンクやオーバーレイネットワーク上での経路木表を管理する。また、この層ではプリファレンス値に基づく複数のビデオストリームの受け入れ制御を行う。Emmaではオーバーレイリンク上で配送されている各ビデオストリームに対して、その配送を停止した場合に損失するプリファレンス値の総和をあらかじめ計算しておく。この制御は、すでに空き帯域が残っていない状況において、現在配送されていないビデオストリームに対する新しい要求があった場合、そのビデオストリームに対するプリファレンス値の合計が、現在配送中のビデオストリームを停止することで損失するプリファレンス値の総和よりも大きければ、配送

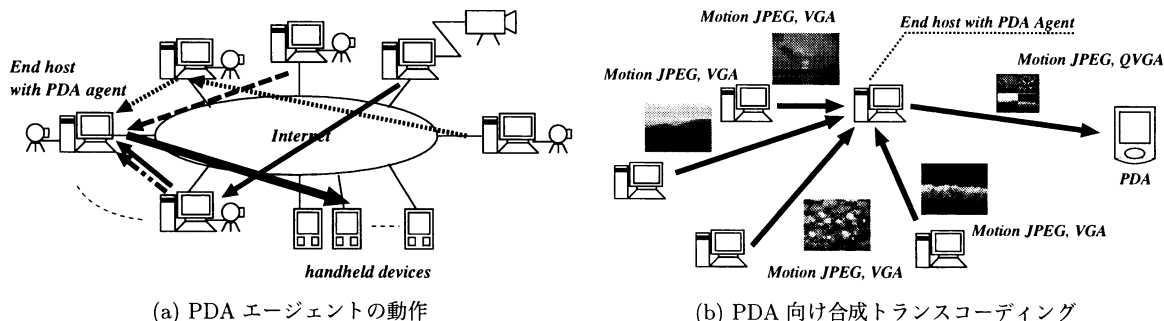


図 4 配信機構の概要

中のビデオストリームの配送を停止し、新しく要求されたビデオストリームの配送を開始する。

Emma ミドルウェアではオーバーレイリンクごとに、ビデオストリームの受信ノードの受信状況をモニタしており、CPU 負荷や帯域不足などにより定められた送信レートでの配送が困難であると判断した場合に送信レートを下げる機能を持つ。ストリームレート制御層 (intra-stream QoS Management Layer) はこれを実現する。ミドルウェアではこの品質制御を実現するために RTP<sup>2)</sup> を利用している。JMF ではバージョン 2.0 以降において RTP/RTCP の実装を提供しており、送信側でインタフェース RemoteListener を実装することで、受信側のロス率などを含む受信状況に関する RTP/RTCP 受信者レポートを RemoteEvent の形で受け取ることができる。さらに JMF では Control インタフェースを利用して、ソースノードにおける送出レートを調節することができる。例えば、Motion JPEG を送信する場合、Control インタフェースのサブインタフェースである QualityControl の setQuality メソッドを利用し、引数に 0.0 から 1.0 の範囲で指定することで送信する JPEG の画質を調節できる (1.0 の場合が最高品質となる)。この 2 つの機能を利用すれば、送信の相手側においてロスが検出された場合に、送信レートを落とすことで相手の受信状況に合わせる事が可能となる。

#### ロビーサーバ

ロビーサーバは、セッション参加に必要な初期情報 (ノード ID, IP アドレス, ポート番号など) の提供及びセッション/ユーザポリシーの管理を行う。メンバーの参加や離脱に応じて、情報の追加や削除を行う。

### 3. Emma ミドルウェアにおける携帯端末への映像配信機構

本稿では、Emma ミドルウェアに PDA のような

小型携帯端末から参加するために提供される機能を述べる。

#### 3.1 基本方針

Emma ミドルウェアが対象とするアプリケーションは (全ノード対全ノードではなくとも) 少なくともグループのメンバー同士の多対多通信である。したがって、一般にユーザは複数の動画を同時に受信しようと試みるが、PDA では表示能力やネットワーク帯域に制約があることから、適切なトランスコードを行うことが望ましい。また PDA がエンドホストの役割を果たすのはその制約上困難である。

そこで本稿では、各エンドホストに PDA エージェントを配置し、PDA はいずれかのエンドホストの PDA エージェントに接続する (図 4(a))。PDA エージェントはあたかも Emma ミドルウェアのユーザアプリケーションのように振舞い、PDA からの複数ストリームの受信要求を受け取り Emma ミドルウェアにその受信要求を渡す。また受信要求が受け入れられ、複数のビデオストリームを渡す場合、それらのフレームを縮小後に合成し、単一の低解像度ストリームで送信することで、PDA の表示領域の制約や復号処理性能の制約、ネットワーク帯域制約のもとでも PDA にビデオストリームを提供する (図 4(b))。

なお、エンドホスト数に対し PDA 数が多い場合、一つの PDA エージェントは複数の PDA を扱うこともある。それらの PDA が受信要求するストリームがそれぞれ異なる場合、エージェントは多数のストリームを受信及びトランスコードする必要があるため、その負荷は大きくなり、また多くの PDA がその要求を満足できないことも考えられる。本稿では、PDA の集合とそれらのストリーム受信要求及びプリファレンス値が与えられた場合に、全体として満足されるプリファレンス値を最大とするための PDA エージェントへの PDA の割り当て問題を定式化する。

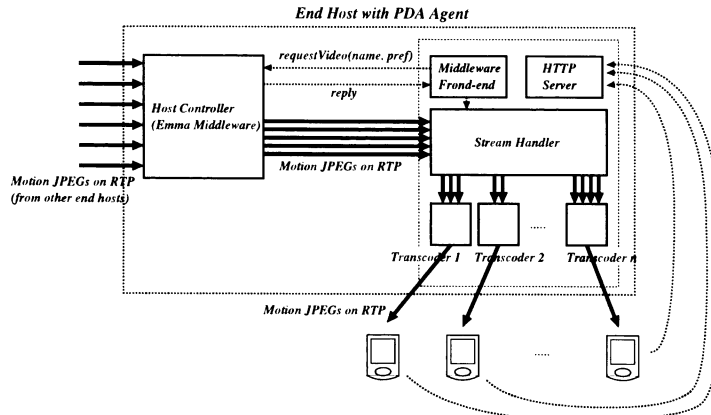


図 5 PDA エージェントの構成

### 3.2 PDA エージェント

PDA エージェントの構成を図 5 に示す。PDA エージェントは PDA からの要求 (HTTP) を扱う HTTP サーバプロセス、ミドルウェアを呼び出すフロントエンドプロセス、ミドルウェアからのストリームを受け取るストリームハンドラプロセス及びストリームの合成トランスコーディングを行うトランスコーダからなる。これらは Emma ミドルウェア同様、Java 及び JMF を用いて実装されている。

PDA からの接続要求を受け取った PDA エージェントは、このエージェントが管理している PDA 端末が最大処理台数を超えていない範囲で要求を受け付ける。接続が許可された場合、この PDA のトランスコーディングを行うトランスコーダプロセスが生成され、ミドルウェアフロントエンドプロセスを介して Emma ミドルウェアのホストコントローラより受け取った、現在受信可能なビデオソースのリストが返される。ユーザはそのリスト中のビデオにプリファレンス値を設定し受信要求を行う。PDA 側から受信要求を受けた PDA エージェントは、必要なビデオを Emma ミドルウェアに要求し、受け取ったビデオをこの PDA に対応するトランスコーダプロセスに渡す。

ストリームハンドラは、Emma ミドルウェアのホストコントローラから送られるビデオストリームを (必要に応じて複製し) トランスコーダプロセスに渡す。トランスコーダでは、ストリームハンドラから送られるビデオを用いてトランスコーディングを行う。すでに述べたように、PDA に対して複数のビデオを渡す場合、それらのフレームを PDA の表示サイズに適するように縮小及び合成し、単一のストリームに変換する。この合成には次のような方法を用いる。

トランスコーダが利用する各ビデオは、そのフレームを Image オブジェクトに変換してリングバッファに保存する。JMF では、ビデオフレームにアクセスするために、Codec インタフェース実装し利用する方法がある。TrackControl インタフェースの setCodecChain メソッドにより、メディアのトラックにコーデックを設定することで、コーデックを通してメディアのデータにアクセスできる。ビデオトラックの場合、コーデックの入力サポートフォーマットを正しく設定 (今回は RGBFormat) しておけば、フレームの Buffer オブジェクトにアクセスすることができる。このフレームを BufferToImage オブジェクトを利用して変換している。

トランスコーダでは、順次バッファリングされていく、各ビデオのフレームイメージを取り出し、PDA の表示サイズで生成した BufferedImage オブジェクトに位置及び縮小サイズを調整し書き込んでいくことで、合成フレームイメージを生成する。図 6 にこの合成フレームをストリームとして PDA に表示させている様子を示している。

### 3.3 PDA エージェントへの PDA の割り当て

各トランスコーダが利用できるビデオは、そのトランスコーダを生成したエンドホストが受信可能なビデオのみである。従って、PDA はどのエンドホストの PDA エージェントに接続するかで満足される要求が異なる。以下では、ユーザからのプリファレンス値に基づきユーザ全体で満足されるプリファレンス値の合計ができるだけ高くなるように PDA をどのようにエージェントに割り当てるかの問題を整数線形計画問題に定式化する。

具体的には以下の 0-1 整数変数及び定数を導入する。

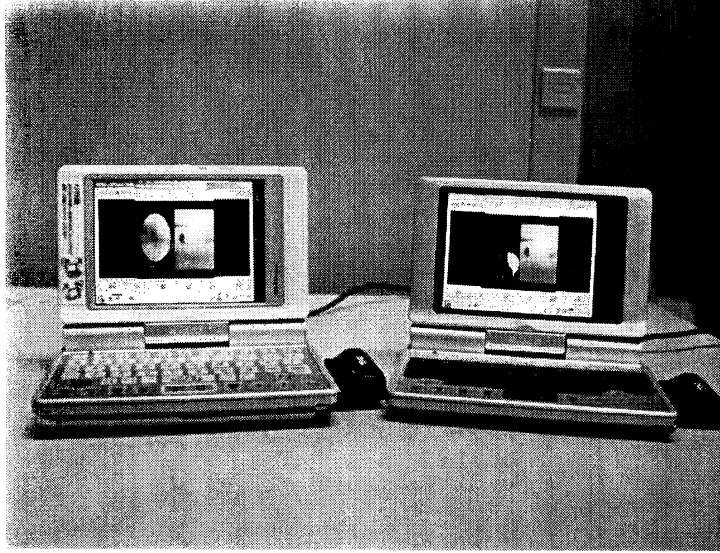


図 6 PDA における合成ストリーム受信の様子

[変数]

- $rcv_{i,j,v}$ : PDA  $i$  がエンドホスト  $j$  のエージェントに接続し、トランスコードされたビデオにストリーム  $v$  が含まれている場合に 1, そうでなければ 0
- $alc_{i,j}$ : PDA  $i$  がエンドホスト  $j$  に接続されている場合に 1, そうでなければ 0

[定数]

- $P_{i,v}$ : PDA  $i$  がストリーム  $v$  に対し要求するプリファレンス値
- $V_{j,v}$ : エンドホスト  $j$  がストリーム  $v$  を受信しているなら 1
- $M_i$ : PDA  $i$  に対し合成トランスコーディングできる最大ストリーム数 (4 分割トランスコーディングであれば 4)
- $N_j$ : エンドホスト  $j$  が扱える PDA の最大台数  
このもとで、以下の制約式が成り立つ。

$$\forall i, j, v \quad alc_{i,j} \geq rcv_{i,j,v} \quad (1)$$

$$\forall i, j, v \quad V_{j,v} \geq rcv_{i,j,v} \quad (2)$$

$$\forall i, v \quad \sum_j rcv_{i,j,v} \leq 1 \quad (3)$$

$$\forall i, j \quad \sum_v rcv_{i,j,v} \leq M_i \quad (4)$$

$$\forall j \quad \sum_i alc_{i,j} \leq N_j \quad (5)$$

各式は、それぞれ

- PDA  $i$  がエンドホスト  $j$  からストリーム  $v$  を

受信するためには、いずれかのエンドホストに接続しなければならない (式 (1))

- PDA  $i$  がエンドホスト  $j$  からストリーム  $v$  を受信するためには、 $j$  において  $v$  を受信していなければならない (式 (2))

- PDA  $i$  はストリーム  $v$  を高々 1 つのエンドホストから受信する (式 (3))

- PDA  $i$  がエンドホスト  $j$  より受信できる最大数は  $M_i$  である (式 (4))

- エンドホスト  $j$  において、PDA の接続台数は最大  $N_j$  である (式 (5))  
ことを意味している。

これに対し、

$$\max \sum_i \sum_v P_{i,v} * \sum_j rcv_{i,j,v} \quad (6)$$

を目的関数とすることで、最適な PDA の配置を決定できる。

ただし、エンドホストの受信ストリームや PDA のプリファレンス値は時間と共に変化するため、このような全体の最適化が適用できる場合は限られる。従って、それらの動的な変化に対応できるように問題を定義し、それをエンドホストが協調して解を求め、PDA を適切に再割り当てするようなプロトコルを設計することが今後の課題である。

#### 4. 評価実験

PDA 端末へのトランスコーダ機能の性能評価を行

うため、複数の Motion JPEG ビデオを同時に送信する実験をいくつか行った。各実験において、使用したエンドホストの仕様 (CPU, メモリ, OS) は表 1 の通りである。

表 1 使用したエンドホストの仕様

Host	CPU	Main Memory	OS
a	PentiumIV 2.6GHz	2GB	WindowsXP
b	PentiumIII 1.2GHz	512MB	WindowsXP
c	PentiumIII 866MHz	384MB	WindowsXP

また、実験において使用したネットワークを図 7 に示す。ノード (エンドホスト)  $a, b, c$  は送信者であり、これらのビデオストリーム (Motion JPEG, 320x240, 23.8fps) はノード  $a$  で合成トランスコードされ、単一のストリームとして送られる。PDA はノード  $a$  の PDA エージェントに接続し、トランスコードサービスを受ける。ノード  $a$  において PDA の数や PDA が要求、受信するストリーム数及びストリームのレートを変えて、それらにおける処理遅延 (フレームが切り出されて、合成されるまでの処理遅延) を測定した。ただし、ノード  $a$  の Emma ミドルウェアはすでに 3 つのストリームを受信しているため、Emma ミドルウェアが新たにストリームを要求、受信するまでの時間は含まれていない。

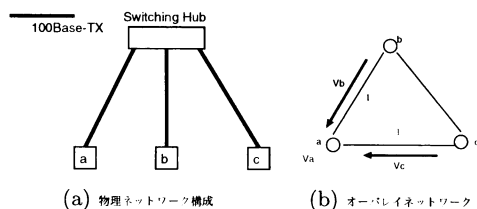


図 7 実験で使用したネットワーク

#### 4.1 実験 1: 単一 PDA のストリーム数などが処理時間に与える影響

実験 1 では、PDA 端末数を 1 とし、ストリーム送信レート、合成するストリームとその数を変化させた場合の処理遅延の変化を測定した。

[ストリームの送信レートに対する処理時間の増加] PDA エージェントは  $a, b, c$  の 3 つのストリームの合成を行っている状態で、それらのストリームの送信レートを Motion JPEG の品質を変えて 1Mbps, 1.4Mbps, 2.4Mbps と変化させた。フレームレートは全て 23.8 fps である。測定結果を以下に示す。

rate	quality	time (ms)	
		Ave.	Max
1Mbps	0.4	11.2	23
1.4Mbps	0.6	13	35
2.4Mbps	1.0	15.0	41

この結果、送信レートの処理時間の増加は十分大きいレートに対しても小さく、送信レートがパフォーマンスに与える影響は比較的小さいと予想される。

[ストリーム切り替えに要する処理時間] PDA が  $a, b$  のストリームを合成して受信している状態で、ストリーム  $b$  をストリーム  $c$  に変更するのに要した時間を測定した結果、平均 4.3 秒で切り替えが完了した。この時間は体感的には十分短いと感じられるが、より高いレスポンス性能が要求される場合はバッファ長を調整するなどの工夫も考えられるため、現在チューニングを行っている。

[ストリーム数の増加に対するトランスコード時間の増加] ストリーム数を変化させた場合の合成トランスコードの処理時間の変化を測定した。その結果を以下に示す。

# of Videos	time (ms)	
	Ave.	Max
1	17.0	35
2	25.3	42
3	33.0	48

結果より、ほぼストリーム数に比例したトランスコード処理時間の増加が見られた。これは各フレームの切り出しやサイズ変更の処理の増加を考慮した場合、ほぼ予想された通りの結果であると思われる。

#### 実験 2: PDA 数が処理時間に与える影響

実験 2 では、PDA 数を増加させることにより、PDA エージェントにおける処理遅延がどのように増加するかを測定した。ここで、各 PDA が受信するビデオストリームは 3 とし、フレームが切り出されてから合成されるまでの処理遅延を測定した。その結果を以下に示す。

# of PDAs	time (ms)	
	Ave.	Max
1	32.3	50
2	52.3	72
3	65.4	68
4	70.2	71.5
5	73.3	73.5

PDA 数の増加に伴い、平均処理時間にも相応の増加が見られた。現在の実装では、PDA ごとにトランスコードプロセスを生成し、個別にトランスコードを行うため、例えば別の PDA が同じストリームを要求している場合も、フレーム切り出しやサイズ変更は重複して行われる。これは例えば帯域が制限される PDA にはフレームを棄却してフレームサイズを小さくするなど、PDA の性能やネットワーク状況に応じた処理が容易になる反面、処理のオーバーヘッドは増大する。これらの適度なバランスが実現されるよう、PDA ごとのトランスコード処理を適切に共通化することなどが今後の課題として残されている。

## 5. おわりに

本稿では、アプリケーション層マルチキャスト Emma のミドルウェアに対し、PDA 端末へビデオをトランスコード配信するための機構の設計、実装について述べた。Emma ミドルウェアはユーザ間で複数のリアルタイムビデオストリーミングを行うビデオチャットのようなアプリケーションの実現を容易にするためのプラットフォームである。これに対し、今後の無線技術の発展や小型携帯端末の普及を考慮し、PDA 端末のような小形端末もアプリケーションに容易に参加できるための枠組みを提供することで、機種間の性能差を可能な限り隠蔽したビデオコミュニケーションシステムの構築を可能にすることを目的としている。

現在、機構の性能向上のためのチューニングのみならず、PDA エージェントの負荷分散機能など新しい機能の追加により、全体のユーザ満足度を向上する枠組みの検討などを行っている。

## 参考文献

1) S. McCanne and V. Jacobson, "Vic: A Flexible Framework for Packet Video," *Proc. of ACM Multimedia*, pp. 511-522, 1995. Tools are available at Lawrence Berkeley National Laboratory WWW manual page, <http://ee.lbl.gov/vic/>.

2) H. Schulzrinne, S. Casner, R. Frederick and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," *RFC 1889*, 1996.

3) 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.

4) P. Francis, "Yoid: Extending the Internet Multicast Architecture," *Unrefereed Report*, 2002. <http://www.isi.edu/div7/yoid/>

5) 山口弘純, 中村嘉隆, 廣森聡仁, 安本慶一, 東野輝夫, 谷口健一, "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," コンピュータソフトウェア, 2003 (採録決定)

6) T. Yamashita, H. Yamaguchi, K. Yasumoto, T. Higashino and K. Taniguchi, "Emma Middleware: Application-level Multicast Infrastructure for Multi-party Video Communication," *Proc. of 15th Int. Conf. on Parallel and Distributed Computing and Systems (PDCS2003)*, 2003 (to appear)

7) 佐藤秀則, 山下剛, 山口弘純, 安本慶一, 東野輝夫, 谷口健一, "携帯端末を含む動画像コミュニケーションシステム向けミドルウェアの設計と実装," マルチメディア, 分散, 協調とモバイルシンポジウム (DICOMO'2003) 論文集, pp. 505-508, 2003

8) R. Lienhart, M. Holliman, Y. Chen, I. Kozintsev and M. Yeung, "Improving Media Services on P2P Networks," *IEEE INTERNET COMPUTING*, Vol.6, No.1, pp.73-77, 2002.

9) W. T. Ooi, R. Renesse and B. Smith, "DESIGN AND IMPLEMENTATION OF PROGRAMMABLE MEDIA GATEWAYS," *The Proc. of 10th International Workshop on Networks and Operating System Support for Digital Audio and Video (NOSSDAV '00)*.

10) 山崎達也, 福永茂, 佐藤範之 "トランスコーディングを用いた複数端末への QoS 調整動画像配送," 電子通信学会論文誌, Vol.J85-B, No.1, pp.50-59, 2002.

11) S. Roy, B. Shen, V. Sundaram and R. Kumar, "Application Level Hand-off Support for Mobile Media Transcoding Sessions," *Proceedings of the 12th International Workshop on Networks and Operating System Support for Digital Audio and Video (NOSSDAV '02)*.

12) B. Shen, and S. Roy "A very Fast Video Spatial Resolution Reduction Transcoder," in *To appear in Proceedings of ICASSP 2002*, 2002