

## 推薦論文

## 移動エージェントによる QoS 保証機能

橋本浩二<sup>†</sup> 柴田義孝<sup>†</sup> 白鳥則郎<sup>††</sup>

分散マルチメディアシステムを利用して、オーディオやビデオなどのリアルタイムメディアを使い、複数の利用者がマルチメディア会議を行う場合、3 者以上の利用者が相互にマルチメディアデータを送受信する際のエンド間 QoS (Quality of Service) 保証や適合の仕組みが必要となる。従来のフィードバックメッセージによるメディアデータの制御や QoS 要求の適合方法はメッセージ数の増加とハンドリングの煩雑さを招き、ネットワーク負荷をも引き起こす可能性がある。本稿では、複数利用者を想定したマルチメディア会議サービスにおいて、マルチキャストセッションにおける QoS パラメータ集合を定義し、移動エージェントを利用した QoS 保証機能を提案する。ネットワーク上のホストを移動しながらタスクを遂行する移動エージェントは、相互にマルチメディア通信を行う複数利用者の QoS 要求やメディア処理状況を収集するための有効な手段となる。

## Mobile Agent Based QoS Functions

KOJI HASHIMOTO,<sup>†</sup> YOSHITAKA SHIBATA<sup>†</sup> and NORIO SHIRATORI<sup>††</sup>

Using distributed multimedia system that can integrate various realtime and non-realtime media data, it is required end-to-end quality of service (QoS) guarantee functions including QoS adaptation function, when the system users hold a multimedia conference and use realtime audio video data in computer networks. However it is difficult for the system to adapt more than three users required QoS to suitable QoS, when each user communicates to the other user respectively using realtime audio video data. The current feedback message algorithms to control and adjust some QoS requests can cause of increase of control messages and there is some possibility of network congestion. This paper describes a new QoS guarantee functions adopted mobile agent in multimedia teleconference service that has more than three users. In order to collect QoS requirements and state information for media processing, concepts of mobile agent that can transfer host to host is one of valid methods for efficient QoS adaptation and reduction of the number of the feedback messages.

## 1. はじめに

コンピュータの高性能化やネットワークの広帯域化に加え、音声や動画圧縮技術の向上により、安価なパーソナルコンピュータでもオーディオやビデオを利用したリアルタイムメディアの送受信が可能となった。現在、IP を利用した電話やラジオ、Video-on-Demand システム、マルチメディア会議システムなどコンピュータネットワークにおけるマルチメディア通信アプリケーションが日常的に利用されるようになり、遠隔地とのマルチメディアコミュニケーションが容易に行えるようになった。既存のマルチメディア通信アプリケーションの中には様々な機能を有し利便性の高いものも存在

する。しかしながら、現在のところ利用者のサービス要求 (Quality of Service) を保証する機能、または適合させる機能を備えているものはほとんど存在しない。ネットワークレベルでは ATM<sup>1)</sup> を中心にして帯域幅の保証や遅延<sup>2)</sup>、ジッタの制御<sup>3)</sup> を行う研究がさかに行われており、メディアデータ転送時の QoS 保証を考慮した RSVP<sup>4),5)</sup> や RTP<sup>6)</sup> プロトコルを実装した通信アプリケーションも存在する。しかしながら、マルチキャスト転送を想定した 3 者以上の利用者が相互にマルチメディアデータを送受信する際のエンド間 QoS 保証や適合の仕組みは確立されていない。

筆者らはこれまでに、クライアント-サーバ方式の Video-on-Demand System において、そのシステムアーキテクチャとエンド間 QoS 保証機能、オーディオ・ビデオデータの packets およびフレームレート

<sup>†</sup> 岩手県立大学  
Iwate Prefectural University

<sup>††</sup> 東北大学電気通信研究所  
Research Institute of Electrical Communication,  
Tohoku University

本論文の内容は 2000 年 12 月のマルチメディア通信と分散処理ワークショップにて報告され、DPS 研究会主査により情報処理学会論文誌への掲載が推薦された論文である。

制御方法について研究を行ってきた<sup>7)</sup>。また、利用者の要求するサービスを動的に構成し、環境や資源利用状況を考慮して利用者の QoS 要求をエンド間で保証するフレームワークとなるやわらかいマルチメディアシステム<sup>8)</sup> (FMS) の研究を進めている。これまでに、FMS における複数利用者の QoS 要求をとりまとめる方式<sup>9)</sup> について報告しているが、複数の利用者がマルチキャストによるメディア転送を相互に行う場合、従来のメッセージパッシングによる QoS の適合やフィードバックを用いたメディア処理を行うことは困難である。

本稿では、複数利用者を想定したマルチメディア会議における、移動エージェントを基盤とした QoS 保証機能について述べる。近年、実行プログラムの移動を可能とする移動エージェントの研究<sup>10),11)</sup> や開発が行われており、ネットワーク上のホストを移動しながらタスクを遂行する移動エージェントは、複数利用者が相互にマルチメディア通信するような場合における、QoS 要求やメディア処理状況を収集するための有効な手段である。移動エージェントを利用することにより、メッセージ数の削減と効率的な QoS 適合が可能となる。まず、2 章では我々が提案しているやわらかいマルチメディアシステムを概観し、3 章では複数の QoS 要求をとりまとめるための合意ポリシーとマルチキャストによるメディア転送における QoS の定義と保証機能について述べ、続く 4 章では、移動エージェントを用いた QoS 保証のためのプロトコルフローを示し、5 章でプロトタイプ構築とその評価について述べる。

## 2. Flexible Multimedia System (FMS)

FMS は、やわらかさの概念<sup>12)</sup> に基づいたエージェント指向のマルチメディアシステム<sup>13)</sup> であり、エージェント技術をマルチメディア通信に応用することで多様な利用者環境間でのマルチメディア通信を実現する<sup>8)</sup>。このシステムは、利用者端末 (FMS User Station) とエージェントリポジトリ (FMS Agent Repository) により構成され、利用者の環境や資源の利用状況に応じたマルチメディア通信サービスを利用者に提供する。また、利用者のサービス要求に応じて必要なエージェントを動的に組織することが可能であり、利用者の QoS 要求をエンド間で保証するフレームワークを有する。

### 2.1 システムアーキテクチャ

FMS によりマルチメディア会議を実現するためのエージェント構成を図 1 に示す。この構成は、マルチメディア会議サービス開始時に FMS エージェント

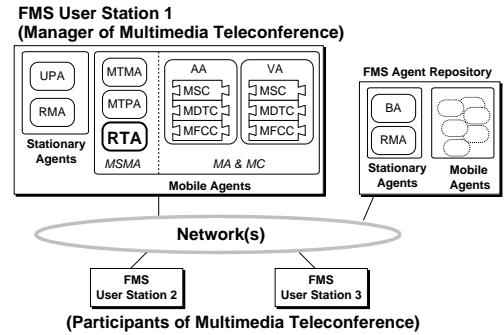


図 1 マルチメディア会議における FMS のエージェント構成  
Fig. 1 FMS agent configuration in multimedia teleconference.

リポジトリから利用者端末へ必要となるエージェントとコンポーネントが組織<sup>8)</sup>された後の構成を示している。FMS におけるエージェントは、自律的に活動するコンピュータプログラムである。エージェントは大別して 2 種類存在し、実行を開始したシステム上でのみ稼働するタイプのエージェントを位置固定エージェント (Stationary Agent) と呼び、実行を開始したシステムに拘束されないタイプのエージェントを移動エージェント (Mobile Agent) と呼ぶ。

*User Partner Agent (UPA)* は、利用者からのサービス要求や QoS 要求を受け付け、*Resource Management Agent (RMA)* は利用者端末のハードウェアおよびソフトウェア資源を管理し、必要な資源の確保や解放を行う。UPA と RMA の機能は利用者端末固有のものであるため、位置固定エージェントとして各利用者端末に常駐している。

一方、*Multimedia Service Management Agent (MSMA)* は、サービス固有の機能を利用者に提供し、*Media Agent (MA)* はメディア処理の監視や制御を行うエージェントである。MA は、実際にメディア処理を行う *Media Component (MC)* を有する。*Media Synchronization Component (MSC)* はメディア内およびメディア間同期処理<sup>14)</sup> を行い、*Media Data Transform Component (MDTC)* が、JPEG, MPEG1/2, H.261 などの圧縮/展開や、画像データのカラーフォーマット、オーディオデータの変調方式といったデータ変換処理を行う。そして *Media Flow Control Component (MFCC)* は、メディアのレート制御やパケット紛失の調整を行うために可変ビットレート転送やパケット間隔調整<sup>9)</sup> を行う。これらは、必要に応じて FMS エージェントリポジトリから *Broker Agent (BA)* により利用者端末へ組織される移動エージェントである。

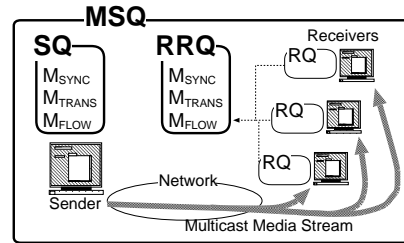
FMS においてマルチメディア会議サービスを利用者

に提供する場合、会議主催者または運営者の利用者端末には MSMA として *Multimedia Teleconferencing Management Agent (MTMA)* が組織され、参加者の受付や退出の管理、後述する合意ポリシーの管理や利用者からの QoS 要求をとりまとめる。一方、会議参加者の利用者端末には *Multimedia Teleconferencing Participant Agent (MTPA)* が組織され、UPA が受付ける利用者からの会議に関する要求を処理する。また、QoS の保証されたマルチキャストによるメディアデータ転送を効率よく行うために、*Round Trip Agent (RTA)* を利用する。RTA は利用者端末を巡回し、QoS 保証のために必要な情報の収集や、各利用者端末への適切な QoS の通知などを行う。

### 3. 合意ポリシーに基づく QoS の適合

オーディオやビデオデータを利用して双方向通信を行う場合、送信側の QoS 要求と受信側の QoS 要求を考慮する必要がある。また、複数の受信者がそれぞれ異なる QoS を要求する場合も考慮する必要がある。マルチキャスト転送において送信側 QoS を優先する場合、各受信者はおのおのの環境や資源利用状況に応じてメディアデータを選択受信したり、受信したメディアデータを利用者に提供可能なデータへ変換することによりメディアを処理することも可能である。しかしながら、受信側 QoS を優先する場合、複数の QoS 要求をすべて保証するメディアデータの転送を行うためには、送信側がそれぞれの受信者に対し個別にメディアデータ転送を行う必要がある。マルチメディア会議におけるオーディオやビデオデータの転送を想定すると、これは現実的な解決方法ではない。そこで、FMS では複数の受信側 QoS 要求をとりまとめるために、合意ポリシー (*POLICY*) を用いる。*POLICY* は最高 (*Highest*)、最低 (*Lowest*)、平均 (*Average*)、最多 (*Mode*)、中央 (*Medium*)、特定 (*Special*) のいずれかの値をとり、複数の受信側 QoS の代表を決定するための方針を与えるものである。たとえばマルチメディア会議を行う場合、会議の管理者が *POLICY* に応じて複数の受信側 QoS 要求をとりまとめることによって、合意された受信側 QoS によるメディア転送が可能となる。

ここで、1 送信者に対し複数の受信者が存在する場合の、マルチキャストによるメディア転送をマルチキャストセッションと呼ぶ。以下 3.1 節においてマルチキャストセッションごとの QoS を定義し、3.2 節ではマルチキャストセッションにおける QoS 保証機能について述べる。



MSQ: Multicast Session QoS Parameter Set  
 SQ: Sender Side Media QoS Parameter Set  
 RRQ: Represented Receiver Side Media QoS Parameter Set  
 RQ: Receiver Side Media QoS Parameter Set

図2 マルチキャストセッションにおける QoS パラメータ集合  
 Fig.2 QoS parameter set in a multicast session.

#### 3.1 QoS パラメータ

マルチキャストセッションの QoS を示すパラメータの集合を *MSQ* とし、以下のように定義する。また、その概要を図 2 に示す。

$$MSQ = \{P_{MS}, SQ, RRQ, P_{SR}\}$$

- $P_{MS}$ : マルチキャストセッションの優先順位
- $SQ$ : 送信側メディア QoS パラメータ集合
- $RRQ$ : 受信側を代表するメディア QoS パラメータ集合
- $P_{SR}$ : 送受信優先識別値 (0: 送信側優先, 1: 受信側優先)

次に、1 送信者に対し  $n$  受信者のマルチキャストセッションを想定し、*MSQ* に含まれる  $SQ$ ,  $RRQ$  を次のように定義する。

$$SQ = \{M_{SYNC}, M_{TRANS}, M_{FLOW}\}$$

$$RRQ = F_{rep}(\{RQ_i \mid 1 \leq i \leq n\}, POLICY)$$

$$RQ = \{M_{SYNC}, M_{TRANS}, M_{FLOW}\}$$

ここで、 $RQ$  は受信側それぞれが要求するメディア QoS パラメータ集合を示す。 $F_{rep}$  は、マルチキャストセッションにおける受信側の代表となるメディア QoS パラメータ集合を合意ポリシーを用いて算出する関数である。一方、 $SQ$  および  $RRQ$ ,  $RQ$  に含まれる  $M_{SYNC}$ ,  $M_{TRANS}$ ,  $M_{FLOW}$  は、それぞれメディア同期、メディアデータ変換、メディアフロー制御の QoS を示すパラメータの集合である。これらのパラメータを次のように定義する。

$$M_{SYNC} = \{S_F, R_F, T_{START}\}$$

$$M_{TRANS} = \{CODEC, FORMAT\}$$

$$M_{FLOW} = \{S_{MDU}, S_{PEEK}, R_{MDU}, R_{LOSS}\}$$

メディア同期のパラメータ集合  $M_{SYNC}$  に含まれる  $S_F$  [byte] はフレームサイズを示し、 $R_F$  [1/sec] はフレームレートを示す。また、 $T_{START}$  はメディア処理の開始時刻を示す。メディアデータ変換の QoS を示す  $M_{TRANS}$  はコーデックとフォーマット種別をパラメータとし、 $M_{FLOW}$  はメディアフロー制御に必要な

るパラメータの集合である． $S_{MDU}$  [byte] は FMS におけるメディアデータの送受信単位となるメディアデータユニット (Media Data Unit: MDU) の平均サイズを示し， $S_{PEEK}$  [byte] は，そのピークサイズを示す．また， $R_{MDU}$  [1/sec] は MDU のレートを示し， $R_{LOSS}$  [%] は MDU のロス率を示す．これらのパラメータをもとに各利用者端末のメディアコンポーネントはメディア処理を行い，対応するメディアエージェントは，その処理状況を監視する．

これらのパラメータを利用し，合意された QoS を保証するために，まず各マルチキャストセッションごとの  $MSQ$  を生成する必要がある．以下に示す関数  $F_{create}$  は， $m$  個のマルチキャストセッションにおける送信側および受信側すべてのメディア QoS パラメータと合意ポリシーから， $MSQ$  のベクトルを生成する関数である．

$$\{MSQ_j \mid 1 \leq j \leq m\} \\ = F_{create}(\{SQ_j, RQ_{ji}, P_{SR_j} \mid 1 \leq j \leq m, 1 \leq i \leq n\}, POLICY)$$

ここで， $RQ_{ji}$  はマルチキャストセッション  $j$  における  $i$  番目の受信側メディア QoS パラメータ集合である．また，関数  $F_{create}$  は各マルチキャストセッションの  $RRQ$  を決めるために関数  $F_{rep}$  を用いる． $MSQ$  のベクトルが生成されると，複数のマルチキャストセッションにおける QoS の適合が可能となる．

### 3.2 QoS 保証機能

FMS は合意された QoS 保証を行うために，以下 3 つの機能を有する．

(1) QoS マッピング機能：合意ポリシーに基づき決定された  $MSQ$  に含まれるパラメータの値はメディア処理において保証すべき QoS の値を示す．関数  $F_{map}$  は，それらの値を実際のメディア処理に必要な資源パラメータ (Resource Parameters) へマッピングする．

$$\begin{array}{ll} F_{map}(SQ) & \text{送信のみ} \\ F_{map}(RRQ_j \mid 1 \leq j \leq m) & \text{受信のみ} \\ F_{map}(SQ, RRQ_j \mid 1 \leq j \leq m-1) & \text{送受信} \end{array}$$

各利用者端末の会議参加者エージェント MTPA が  $F_{map}$  関数を利用する． $F_{map}$  関数はメディアごとに必要となる資源パラメータを取得するために，複数のメディアエージェント MA に対し，QoS マッピング要求を発行する．各 MA はメディア QoS パラメータと資源パラメータを関連づける表を所有しており，その表を参照して必要となる資源パラメータの値を取得する．そして， $F_{map}$  関数により資源パラメータの値を取得した MTPA は，各利用者端末の資源管理を行っ

ている RMA に対して必要となる資源が利用可能かどうかを確認する．

(2) 資源管理機能：マルチキャストセッション開始時には QoS を保証するために必要な資源を確保し，終了時には資源を解放する．必要な資源の確保が可能かどうかを知るために，資源利用状況を管理する必要がある． $F_{admit}$  関数は，資源パラメータの値として示される資源に対しアドミッションテストを行い，その資源が利用可能かどうかを確認し，現在の資源利用状況 ( $UTIL$ ) を算出する．

$$UTIL = F_{admit}(Resource\ Parameters)$$

FMS では，各利用者端末の RMA がこれらの資源管理を行う．RMA はビデオカメラやオーディオデバイスなどの物理的な資源を管理する一方，動的に変化する CPU 占有率やロードアベラージュ，メモリの利用状況を管理する．

(3) QoS 適合機能：必要となる資源が常に利用可能であるとは限らない．一方，あるマルチキャストセッションが終了する場合には現存するマルチキャストセッションの QoS を向上させることが可能かもしれない．また，動的な外部負荷による一時的な QoS の劣化が生じた場合は合意ポリシーとパラメータの優先順位に基づく QoS の適合を行う必要もある．これらの場合，FMS は関数  $F_{adapt}$  によって，各  $MSQ$  に含まれる送受信 QoS を適合する．また， $MSQ$  に含まれる送信側メディア QoS パラメータ集合  $SQ$  や受信側の代表  $RRQ$  を，送受信優先識別値  $P_{SR}$  に従って適合する．

関数  $F_{adapt}$  では， $t$  時刻における  $MSQ$  のベクトルを適合し， $t+1$  時刻における  $MSQ$  を生成するために，各送受信メディア QoS の保証状況 ( $ASQ_j, ARQ_{ji}$ ) と各利用者端末の資源利用状況  $UTIL$  および合意ポリシーを用いる． $ASQ, ARQ$  はメディア QoS パラメータ集合であり，送受信処理状況を示す従来のフィードバックメッセージに相当する．

$$\{MSQ_j(t+1) \mid 1 \leq j \leq m\} \\ = F_{adapt}(\{MSQ_j(t), ASQ_j, ARQ_{ji}, UTIL_{jk}, POLICY \mid 1 \leq j \leq m, 1 \leq i \leq n, 1 \leq k \leq us\})$$

( $us$  は利用者端末数)

FMS におけるマルチメディア会議サービスにおいては，会議主催者の利用者端末に組織されている MTMA が，関数  $F_{adapt}$  を用いて  $MSQ$  の適合を行う．

これら 3 つの QoS 保証機能は利用者端末間の連携により実現される．図 3 は，各エージェントと QoS 保証機能の関係を示している．図に示すよう，MSMA および MA はマッピングテーブルを所有しており，利用者が要求するアプリケーション QoS をメディア QoS

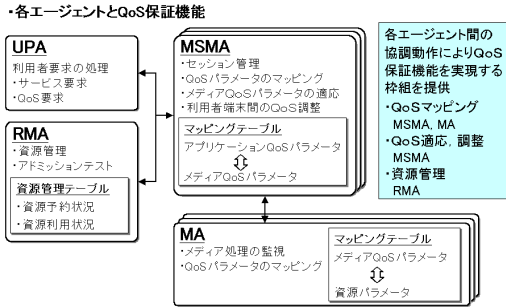


図 3 エージェントと QoS 保証機能  
Fig. 3 Agents and QoS functions.

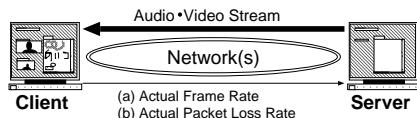


図 4 フィードバックメッセージによるレート制御  
Fig. 4 Rate control using feed back messages.

へ、そして必要となる資源へのマッピングを行う。一方、RMA は利用者端末内の資源を管理するために、資源管理テーブルを所有している。このテーブルで資源予約状況と資源利用状況を把握し、アドミッションテストが可能となる。また、利用者端末間の QoS 調整は次章で述べる移動エージェントを利用したプロトコルにより実現される。

#### 4. 移動エージェントによる QoS 保証

##### 4.1 3 者以上の双方向通信

図 4 は、クライアント-サーバ間でオーディオ・ビデオ転送を行う場合の、フィードバックメッセージによるレート制御の概要を示している。クライアントは一定間隔でフレームレートやパケットロス率サーバに通知することにより、動的な負荷変動に対応したレート制御<sup>9)</sup>が可能となる。このようなフィードバックメッセージによるメディア制御をマルチキャスト転送を用いたビデオ会議に適用した場合、1 送信者に対し 2 人の受信者を想定すると、レート制御を行うためのフィードバックメッセージ数は 2 個である。しかし、10 人のビデオ会議を想定し、10 人それぞれが送受信を行う場合、フィードバックメッセージ数は  $10 \times (10 - 1) = 90$  個必要となる。N 者それぞれが送受信を行う場合のフィードバックメッセージ数は  $N \times (N - 1)$  であり、 $N^2$  でメッセージ数が増加する。このように、従来のフィードバックメッセージによるメディアデータの制御はメッセージ数の急激な増加とハンドリングの煩雑さを招き、ネットワークトラフィックの負荷をも引き

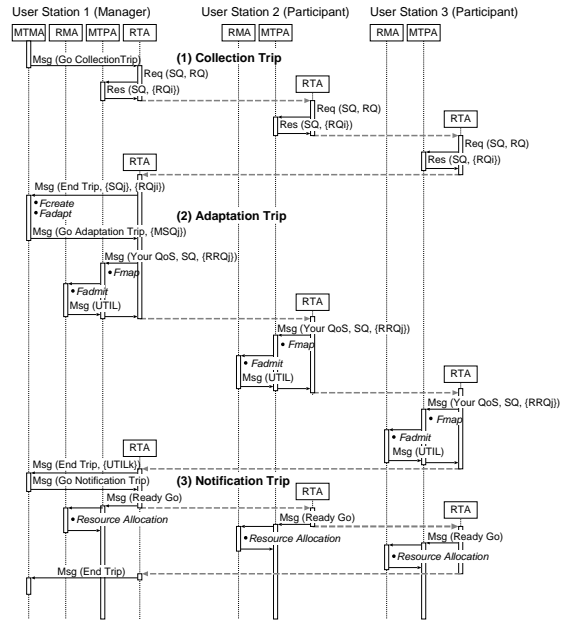


図 5 QoS 要求の収集と適合：マルチキャストセッション開始時  
Fig. 5 Collection and adaptation of requested QoS beginning of multicast sessions.

起こす可能性がある。したがって、マルチキャスト通信を利用する N 人の QoS 保証は非常に困難である。

そこで、FMS におけるマルチキャストセッションでは、従来のフィードバックメッセージ数削減と効率的な QoS の適合を行うため、移動エージェントを利用する。セッション開始時に会議主催者の利用者端末へ組織された巡回エージェント RTA が定期的に各利用者端末を巡回し、QoS を保証するために必要となる情報収集や合意ポリシーによって適合された QoS の通知を行うことにより、QoS 保証機能を実現する。以下、マルチキャストセッション開始時と期間中に分け、そのプロトコルフローを示す。

##### 4.2 マルチキャストセッション開始時

図 5 にマルチキャストセッション開始時の QoS 要求収集と適合フローを示す。ここで、各利用者は利用者エージェント UPA を通して QoS 要求を行っているものとする。また、合意ポリシーはあらかじめ MTMA により決定されているものとする。

セッション開始時における RTA の巡回は以下 3 つのフェーズで構成される。

##### (1) Collection Trip

まず、MTMA が RTA に対し Collection Trip 開始メッセージを送る。RTA は MTPA に送受信メディア QoS パラメータ集合取得要求を発行し、送信側メディア QoS パラメータ集合 SQ と受信側メディア QoS

パラメータ集合  $RQ_i$  を取得する．RTA は取得した  $SQ, RQ_i$  を保持し，次の利用者端末へ移動する．すべての MTPA から  $SQ, RQ_i$  を取得したら，RTA は会議主催者の利用者端末へ戻り，収集した  $SQ, RQ_i$  とともに Trip 終了メッセージを MTMA に送る．

Collection Trip が終了すると，MTMA は各マルチキャストセッションにおける  $MSQ$  のベクトルを生成するために，関数  $F_{create}$  を実行する．次に，MTMA は  $MSQ$  に含まれる送信側メディア QoS パラメータ集合  $SQ$  や受信側の代表  $RRQ$  を，送受信優先識別値  $PSR$  に従って適合するために，関数  $F_{adapt}$  を実行する．これにより，合意ポリシーに基づき適合された  $MSQ$  のベクトルが生成される．

## (2) Adaptation Trip

適合された  $MSQ$  のベクトルが生成されると，次に MTMA は RTA に対し Adaptation Trip 開始メッセージを送る．RTA は MTPA に適合された  $SQ$  および  $RRQ_i$  を送る．ここで MTPA は関数  $F_{map}$  によりメディア QoS パラメータのマッピングを行い，必要となる資源パラメータ値を取得する．次に MTPA は取得した値を RMA に送り，RMA は関数  $F_{admit}$  によりアドミッションテストを実行する．その結果として資源の利用状況  $UTIL$  が MTPA 経由で RTA へ送られる．RTA はこれを保持し，次の利用者端末へ移動する．以後これを繰り返し，すべての MTPA から  $UTIL$  を取得したら，RTA は会議主催者の利用者端末へ戻り，収集した  $UTIL_k$  とともに Trip 終了メッセージを MTMA に送る．Adaptation Trip が終了すると，MTMA は各利用者端末において合意のとれた QoS によるマルチキャストセッションを開始可能かどうかを確認する．もし不可能な場合， $UTIL_k$  をもとに関数  $F_{adapt}$  を実行し，再度  $MSQ_j$  の適合をし，RTA は再び Adaptation Trip を実行する．

## (3) Notification Trip

合意のとれた QoS によるマルチキャストセッションが開始可能である場合，MTMA は RTA に対し Notification Trip 開始メッセージを送る．RTA は移動を繰り返しながら，各 MTPA に対しセッション開始準備が完了したことを伝える．

各利用者端末の RMA は必要となる資源を確保し，以後マルチキャストセッションが開始される．

### 4.3 マルチキャストセッション期間中

セッション期間中，MTMA は RTA に対し定期的に Collection Trip 開始メッセージを送る．メッセージを受けた RTA は各利用者端末を巡回し，MTPA から以下の情報を収集する．

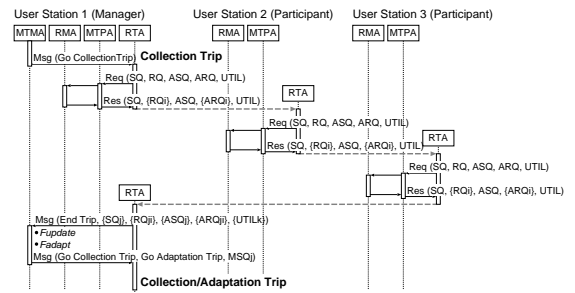


図 6 セッション期間中の情報収集

Fig. 6 Collection Trip during multicast sessions.

- $SQ, RQ_i$  : 利用者からの QoS 要求
- $ASQ, ARQ_i$  : 送受信 QoS 保証状況
- $UTIL_k$  : 各利用者端末の資源利用状況

$ASQ, ARQ_i$  はそれぞれ送信 QoS 保証状況と受信 QoS 保証状況を示すメディア QoS パラメータである．

図 6 は，セッション期間中の Collection Trip を示している．RTA は各利用者端末を一巡すると MTMA に収集した情報を送信する．MTMA は収集した情報から  $MSQ$  のベクトルを適合するかどうか判断する．

- (1) 利用者からの QoS 更新要求を検出した場合：関数  $F_{update}$  により  $MSQ$  のベクトルを更新し，関数  $F_{adapt}$  を実行する．
- (2) 送受信メディア QoS 保証状況が劣化した場合：保証状況を示す  $ASQ, ARQ_i$  と各利用者端末の資源利用状況  $UTIL_k$  をもとに関数  $F_{adapt}$  を実行する．

上述した 2 つの場合，RTA は Adaptation Trip を開始する．これにより，動的な負荷変動によって QoS が一時的に劣化した場合でも，利用者からの QoS 要求が更新された場合においても，合意された QoS によるマルチキャスト通信が可能となる．

## 5. プロトタイプシステム

やわらかいマルチメディアシステムの応用として，マルチメディア会議サービスを実現するためのエージェントを実装した．プロトタイプの構築環境は，図 7 に示すとおりである．開発は Sun Ultra1 ( Soralis 2.6, UltraSPARC 167 MHz, 128 Mbyte Memory ) 上で Java 1.2.2 を用いて各エージェントをスレッドで実装した．また，エージェントの移動を実現するために RMI とオブジェクトシリアライゼーション機能を利用し，JMF 2.1 を利用してオーディオやビデオデータを処理するエージェントを実装した．

図 8 はプロトタイプシステムのクラス構成を示している．図中，FmsPlace クラスはエージェントの生

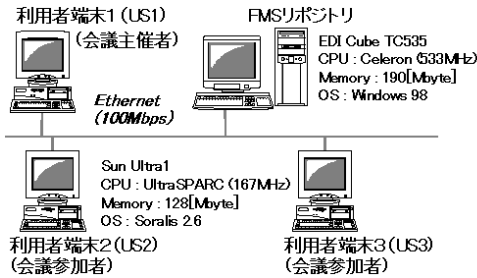


図 7 プロトタイプ構築環境  
Fig. 7 Environment of prototype system.

成, 移動, 終了処理を行うエージェントの動作環境である. また, java.rmi.Remote インタフェースを継承した FmsPlaceInterface クラスを実装することにより FmsPlace 間の RMI による通信が可能となる.

FmsPlace クラスは複数の FmsAgentManager クラスインスタンスを管理する. FmsAgentManager クラスは, FMS におけるエージェントを管理するクラスであり, エージェントごとに存在する. FMS におけるエージェントは, FmsAgent クラスにより実現される. このクラスは移動エージェントの基盤クラスとなり, そのインスタンスは 1 つのスレッドとして実現される.

UserPartnerAgent, ResourceManagementAgent クラスは FmsAgent クラスを継承しており, システムが提供するエージェント間通信機能を利用することが可能となっている. また, FmsFrameAgent クラスは GUI を含む移動エージェントの基盤クラスであり, このクラスを継承することによって, エージェントの基本的な GUI を利用することが可能となる.

そして, FmsMSMA, FmsMA クラスはそれぞれ MSMA と MA の基盤クラスである. マルチメディア通信サービスやメディアに応じてこれらのクラスを継承した特定のクラスを実装することにより, サービスやメディア固有の機能のみを追加することでマルチメディア通信を実現することが可能となる.

図 7 の環境において利用者端末でビデオ (Motion JPEG) の送受信処理を行った場合の CPU 占有率を表 1 に示す. たとえば 16fps のビデオフレームを, JPEG Quality 0.6 で送信する場合, 表 1 より, CPU 占有率は 24% であることが分かる. 一方, 8fps のビデオフレームを, JPEG Quality 0.4 で受信および再生する場合, CPU 占有率は 93% である. 本プロトタイプシステムにおける Motion JPEG の受信処理は, ソフトウェアによるフォーマット変換を含んでおり, 処理遅延が生じる. したがって, 送信処理に比べて受信処理のほうが多くの CPU 処理を必要とする.

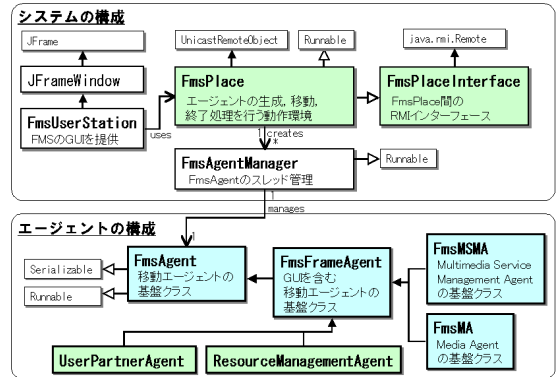


図 8 プロトタイプシステム — クラス構成図  
Fig. 8 Prototype system — Class configuration.

表 1 利用者端末におけるビデオ (Motion JPEG) 処理  
Table 1 Processing of Motion JPEG in the prototyped user station.

Frame Rate	JPEG-Q	CPU (送信)	CPU (受信)	Bandwidth
16 fps	0.6	24%	-	960794 bps
16 fps	0.4	21%	-	757709 bps
16 fps	0.2	20%	-	638534 bps
10 fps	0.6	19%	-	642834 bps
10 fps	0.4	18%	-	504600 bps
10 fps	0.2	18%	-	426396 bps
8 fps	0.6	16%	93%	552682 bps
8 fps	0.4	15%	89%	433382 bps
8 fps	0.2	15%	87%	361680 bps
4 fps	0.8	39%	-	317413 bps
4 fps	0.6	13%	53%	288758 bps
4 fps	0.4	13%	52%	227449 bps
4 fps	0.2	12%	48%	189669 bps
2 fps	0.8	28%	27%	201195 bps
2 fps	0.6	10%	47%	149930 bps
2 fps	0.4	10%	33%	118777 bps
2 fps	0.2	11%	32%	97631 bps

### 5.1 機能評価

プロトタイプシステム上で 3 者によるマルチメディア会議を行うことで, QoS 機能の評価を行った. 表 2 は, 各 3 つのセッションの属性と利用者の QoS 要求を示す. また, 表 3 は, 各区間における送受信状態を示す.

表 2 の QoS 要求で 3 人の利用者がビデオ送受信を行う. まず, 区間 1 において利用者端末 2 (US2) がビデオ送信を開始し, 利用者端末 1 と 3 (US1, US3) がビデオ受信のみを開始する. 次に区間 2 において, US1 がビデオ送信を開始する. 続く区間 3 では, US3 がビデオ送信を開始する. 最後の区間 4 において, US2 がビデオの送信を中断する. このようなシナリオのもとに, セッション 1 における QoS 機能の評価を行った.

図 9 はセッション 1 における送受信フレームレートを示す. QoS 保証機能を利用しない場合, 図 9 (a) に示すように 2 者以上の送受信を行うとフレームレートが安定しない. これは 3 者それぞれ最善努力型の送受信を行うために, 必要となる資源を適合することがで



表 2 マルチキャストセッションと利用者の QoS 要求

Table 2 Multicast sessions and QoS requirements of users.

Session ID	Session Priority	送信者	受信者	送信/受信優先	Frame Rate	M-JPEG Quality	優先属性
1	1	US1	US2, US3	送信側優先 (SQ1)	8fps - 2fps	0.6 - 0.2	画質優先
2	1	US2	US1, US3	受信側優先 (RRQ1)	8fps - 2fps	0.6 - 0.2	画質優先
3	0	US3	US1, US2	受信側優先 (RRQ2)	8fps - 2fps	0.6 - 0.2	滑らかさ優先

表 3 各区間における送受信状態

Table 3 State of sending and receiving in each section.

		US1	US2	US3
区間1	送信	—	○	—
	受信	US2	—	US2
区間2	送信	○	—	—
	受信	US2	US1	US1,2
区間3	送信	○	—	○
	受信	US2,3	US1,3	US1,2
区間4	送信	○	—	○
	受信	US3	—	US1

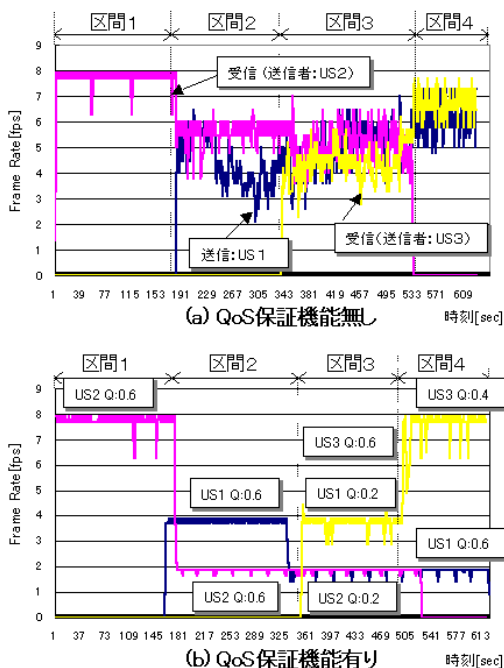


図 9 利用者端末 1 における送受信フレームレート  
Fig.9 Sending and receiving frame rate (US1).

きないためである。この場合、ある瞬間は高フレームレートを記録するが、レートが安定しないことにより滑らかさの保証ができない。ある瞬間、ビデオは滑らかに動くが、次の瞬間には非常にぎこちないフレームを表示することになる。

一方、表 1 に示す CPU 占有率をもとに、RTA を 1 秒ごとに巡回させ、QoS 保証機能を利用した場合、図 9 (b) に示すとおりフレームレートが安定する。この場合、画質優先が滑らかさ優先かといった優先属性

に従い、メディア処理に必要となる CPU 占有率を考慮し、フレームレートまたは JPEG Quality の設定値を変え、適切な QoS の適合が行われたことがわかる。

## 6. まとめ

本稿では、やわらかいマルチメディアシステムにおけるマルチキャストセッションの QoS パラメータ集合を定義し、複数利用者の QoS 要求収集と適合を行うために、移動エージェントを利用する手法を提案した。そして、プロトタイプシステムの構築を行い、3 者によるビデオ通信時における QoS の適合が可能であることを確認した。移動エージェントを用いて QoS 要求の適合やメディアデータ制御に必要な制御情報を取得することにより、メッセージ数の増加やマルチキャスト通信における QoS 適合の煩雑さを抑えることが可能となる。

しかしながら、資源の負荷変動が少ない環境や利用者の QoS 要求更新が行われない状況では、移動エージェントを巡回させる間隔を調整したり、巡回するための適切なルートを決定する必要があり、検討の余地がある。また、今後、フィードバックメッセージを用いたシステムとの定量的な比較評価を行うとともに、複数のネットワークが相互接続された環境における移動エージェントの利用についても検討を進める。

## 参考文献

- 1) Okubo, S., Dunstan, S., Morrison, G., Nilsson, M., Radha, H., Skran, D.L. and Thom, G.: ITU-T Standardization of Audiovisual Communication System in ATM and LAN Environments, *IEEE Journal on Selected Areas in Communications*, Vol.15, No.6, pp.965-982 (1997).
- 2) Georgiadis, L., Guerin, R., Peris V. and Rajan, R.: Efficient Support of Delay and Rate Guarantees, *ACM SIGCOMM*, Vol.26, No.4 (1996).
- 3) Kadur, S., Golshani, F. and Millard, B.: Delay-jitter control in multimedia applications, *ACM Multimedia Systems*, Vol.4, pp.30-39 (1996).



- 4) Zhang, L., Berson, S., Herzog, S. and Jamin, S.: *Resource ReSerVation Protocol (RSVP) — Version 1 Functional Specification*, Braden R., (Ed.), RFC 2205 (1997).
- 5) Barzilai, T.P., Kandlur, D.D., Mehra, A. and Saha, D.: Design and Implementation of an RSVP-Based Quality of Service Architecture for an Integrated Services Internet, *IEEE Journal on Selected Areas in Communications*, Vol.16, No.3, pp.397–413 (1998).
- 6) Schulzrinne, H., Casner, S., Frederick, R. and Jacobson, V.: *RTP: A Transport Protocol for Real-Time Applications*, RFC 1889 (1996).
- 7) 橋本浩二, 知念 正, 佐藤 純, 柴田義孝: 圧縮ビデオデータ転送のためのパケットおよびフレームレート制御法, *情報処理学会論文誌*, Vol.39, No.2, pp.337–346 (1998).
- 8) Hashimoto, K., Shibata, Y. and Shiratori, N.: The System Organization and QoS Functions for Flexible Multimedia System, *Proc. 6th International Conference on Distributed Multimedia Systems (DMS'99)*, pp.209–216 (1999).
- 9) 橋本浩二, 柴田義孝, 白鳥則郎: やわらかいマルチメディアシステムによるマルチメディア会議サービス, *情報処理学会論文誌*, Vol.41, No.2, pp.387–395 (2000).
- 10) Misikangas, P. and Raatikainen, K.: Agent Migration between Incompatible Agent Platforms, *Proc. 20th IEEE International Conference on Distributed Computing Systems (ICDCS2000)*, pp.4–10 (2000).
- 11) Tanaka, S., Yamaki, H. and Ishida, T.: Mobile-Agents for Distributed Market Computing, *Proc. 1999 International Conference on Parallel Processing*, pp.472–479 (1999).
- 12) Shiratori, N., Sugawara, K., Kinoshita, T. and Chakraborty, G.: Flexible Network: Basic Concepts and Architecture, *IEICE Trans. Comm.*, Vol.E77-B, No.11, pp.1287–1294 (1994).
- 13) 唐橋拓史, 勝倉 真, 菅沼拓夫, 菅原研次, 木下哲男, 白鳥則郎: やわらかいビデオ会議システムの協調プロトコルの拡張と評価, *情報処理学会論文誌*, Vol.39, No.2, pp.178–187 (1998).
- 14) Shibata, Y., Seta, N. and Shimizu, S.: Media Synchronization Protocols for Packet Audio/Video System on Multimedia Information Networks, *Proc. HICSS-28*, pp.594–601 (1995).

(平成 13 年 4 月 27 日受付)

(平成 13 年 9 月 12 日採録)

## 推 薦 文

この論文は、分散マルチメディアシステムに関して、

移動エージェントを利用したエンド間 QoS を保証する方式を提案している。提案方式では、FMS(やわらかいマルチメディアシステム)の概念をベースにして、送受信側双方の QoS 要求の適合を合意ポリシーにより行っている。また、本論文では、提案方式を具体的な対象としてマルチメディア会議に適用実現して評価を行い、その有効性を検証している。以上のように本論文は、QoS に関する新しい知見を与えるもので、会員にきわめて有用であり、推薦論文とした。

(DPS 研究会主査 宮部博史)



橋本 浩二 (正会員)

1970 年生。東洋大学大学院工学研究科電気工学科専攻博士前期課程修了。同年(株)CSK 総合研究所に入社。1998 年より岩手県立大学ソフトウェア情報学部助手。2001 年東北大学大学院情報科学研究科博士後期課程修了。博士(情報科学)。移動エージェントを利用した分散マルチメディアシステムの構築とエンド間 QoS 保証の研究に従事。



柴田 義孝 (正会員)

1950 年生。1985 年 UCLA コンピュータサイエンス学科修了。Ph.D. in Computer Science。1985 年～1988 年まで Bellcore(旧 AT&T ベル研究所)にて専任研究員としてマルチメディア情報ネットワークの研究に従事。1989 年より東洋大学工学部情報工学科助教授。1997 年同大学教授。1998 年より岩手県立大学ソフトウェア情報学部教授。同大学メディアセンター長。高速パケットビデオ、マルチメディアプロトコル、ハイパーメディアシステム、感性情報処理等の研究に従事。IEEE, ACM, 電子情報通信学会各会員。



白鳥 則郎 (正会員)

1946 年生。1977 年東北大学大学院博士課程修了。1984 年同大学助教授(電気通信研究所)。1990 年同大学教授(工学部情報工学科)。1993 年同大学教授(電気通信研究所)。情報通信システム、ソフトウェア開発環境、ヒューマンインタフェースの研究に従事。1993 年本会マルチメディアと分散処理研究会主査。IEEE Fellow。