

移動エージェントを利用したQoS保証機能の設計

橋本浩二[†] 柴田義孝[†] 白鳥則郎[‡]

[†] 岩手県立大学 ソフトウェア情報学部 ソフトウェア情報学科

[‡] 東北大学 電気通信研究所

分散マルチメディアシステムを利用してリアルタイムオーディオやビデオによるマルチメディア会議を行う場合、複数の利用者を考慮したエンド間 QoS(Quality of Service) 保証や適合の仕組みが必要となる。しかしながら、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、ネットワークトラフィックの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、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、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有効な 1 つの手段であり、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。

Design of QoS Guarantee Functions using Mobile Agent

Koji Hashimoto[†], Yoshitaka Shibata[†] and Norio Shiratori[‡]

[†] Faculty of Software and Information Science, Iwate Prefectural University

[‡] Research Institute of Electrical Communication, Tohoku University

Using distributed multimedia system which 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 which has more than three users. In order to collect QoS requirements and state information for media processing, concepts of mobile agent which 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、音声や動画画像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、遠隔地のと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。

このような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が、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[1] を中心にして帯域幅の保証や遅延、ジッタの制御を行う研究が盛んに行われており、メディアデータ転送時の QoS 保証を考慮した RSVP[2] や RTP[3] 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。

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 [4, 5] 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 適応が可能となる。

2 Flexible Multimedia System (FMS)

FMS は、やわらかさの概念 [6] 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 [7]。FMS 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実現する。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仕組みを有する。

ここで、FMS におけるエージェントを自律的に活動するコンピュータプログラムとする。また、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。

2.1 システムアーキテクチャ

FMS 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 [7] された後の構成を示している。

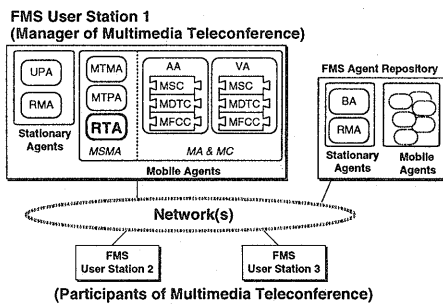


図 1: マルチメディア会議サービスにおける FMS のエージェント構成

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り、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 [8] 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 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 が受け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) を導入した。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。次節において、従来のメディア制御における問題点を述べ、RTA 導入の理由を述べる。

2.2 3 者以上の双方向通信

筆者らはこれまでに、クライアント-サーバ方式でオーディオ・ビデオ転送を行う際に、フィードバックメッセージを用いたレート制御 [8] の有効性について述べてきた。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てみる。

図 2 は、1 送信者に対し 2 人の受信者がいる場合のメディアデータストリームと、フィードバックメッセージを示している。この場合、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 個であるが、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。

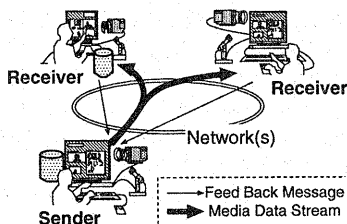


図 2: 1 送信者 - 2 受信者

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起こす可能性がある。従って、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、本稿では複数利用者の QoS 要求適合とメディア処理に必要となるフィードバック情報を収集するために移動エージェントを利用する。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。

次章ではマルチキャストによるメディアデータ転送を行う際の QoS を定義し、続く 4 章では移動エージェントが QoS の適合を効率よく実現するフローを示す。

3 合意ポリシーに基づく QoS 適合

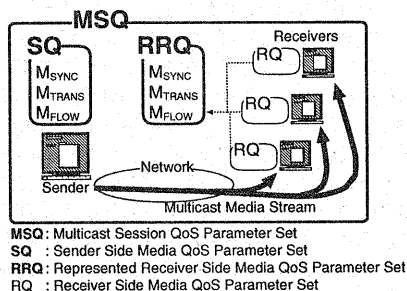
オーディオやビデオデータを利用して双方向通信を行う場合、送信側の QoS 要求と受信側の QoS 要求を考慮する必要がある。また、複数の受信者がそれぞれ異なる QoS を要求する場合も考慮する必要がある。マルチキャスト転送において送信側 QoS を優先する場合、各受信者は各々の環境や資源利用状況に応じてメディアデータを選択受信したり、受信したメディアデータを利用者に提供可能なデータへ変換することによりメディアを処理することも可能である。しかしながら、受信側 QoS を優先する場合、複数の QoS 要求を全て保証するメディアデータの転送を行うためには、送信側がそれぞれの受信者に対し個別にメディア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、各受信者の合意のもとにメディアデータの転送を行うため、合意ポリシー (POLICY) を導入する。

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。例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り、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。

3.1 QoS パラメータ



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 3: マルチキャストセッションにおける QoS パラメータ集合

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、図 3 はその概要を示す。

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

- 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)

また、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。

$$\begin{aligned} SQ &= \{M_{SYNC}, M_{TRANS}, M_{FLOW}\} \\ RRQ &= F_{rep}(\{RQ_i \mid 1 \leq i \leq n\}, POLICY) \\ RQ &= \{M_{SYNC}, M_{TRANS}, M_{FLOW}\} \end{aligned}$$

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。

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。

$$\begin{aligned} M_{SYNC} &= \{S_F, R_F, T_{START}\} \\ M_{TRANS} &= \{CODEC, FORMAT\} \\ M_{FLOW} &= \{S_{MDU}, S_{PEEK}, R_{MDU}, R_{LOSS}\} \end{aligned}$$

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毎の 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全てのメディア 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。

$$\begin{cases} F_{map}(SQ) & \text{send only} \\ F_{map}(RRQ_j \mid 1 \leq j \leq m) & \text{receive only} \\ F_{map}(SQ, RRQ_j \mid 1 \leq j \leq m-1) & \text{send and receive} \end{cases}$$

F_{map} 関数は各利用者端末の会議参加者エージェント MTPA が発行する。 F_{map} 関数はメディア毎に必要な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。

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を算出する。

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

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レージ、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知れない。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_{admit} 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。

$$\{MSQ_j(t+1) \mid 1 \leq j \leq m\}$$

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

(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。

上述したQoS保証機能は利用者端末間の連携により実現される。次章では、移動エージェントを用いてこれらのQoS保証機能を実現するプロトコルを示す。

4 移動エージェントによるQoSの適合フロー

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おける、そのプロトコルフローを示す。

4.1 マルチキャストセッション開始時

図4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。

以後これを繰り返し、全ての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

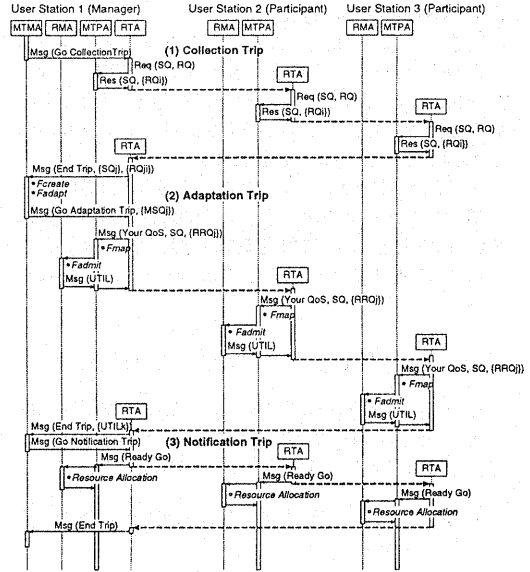


図4: マルチキャストセッション開始時におけるQoS要求の収集と適合

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 P_{SR} 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、全ての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に出発する。

(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。以後、RTA は定期的に各利用者端末を巡回し、利用者からの QoS 更新要求や送受信 QoS の保証状況を収集する。

4.2 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。RTA は移動しながら各 MTPA から以下の情報を収集する。

- $SQ, \{RQ_i\}$: 利用者からの QoS 要求更新を検出するため
- $ASQ, \{ARQ_i\}$: 送受信 QoS 保証状況を把握するため
- $\{UTIL_k\}$: 各利用者端末の資源利用状況の把握するため

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し、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。そして、以下の場合、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。

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。

5 まとめ

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 の適合は複数の利用者が同時に送受信を行う場合、制御メッセージが n^2 で増加する。これはメディアデータの制御や QoS の適合を煩雑にするだけでなく、ネットワークトラフィックにも影響を与える可能性がある。本稿では、マルチキャストセッションにおける QoS を定義し、やわらかいマルチメディアシステムに移動エージェント RTA を導入し、RTA が各利用者端末を移動しながら QoS パラメータの適合を行う手順を示した。

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、問題であるメッセージ数の増加や QoS 適合の煩雑さを抑えることが可能となる。

現在、筆者らは Java 1.2 によるやわらかいマルチメディアシステムの実装を Sun Workstation/Solaris 7 上で行っている。各エージェントをスレッドで実現し、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ている。今後、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、移動エージェントによる QoS 保証機能を実現する予定である。

参考文献

- [1] Sakae Okubo, Stuart Dunstan, Geoff Morrison, Mike Nilsson, Hayder Radha, Dale L. Skran and Gary Thom.: *ITU-T Standardization of Audio-visual Communication System in ATM and LAN Environments* IEEE Journal on Selected Areas in Communications, Vol.15, No.6, pp.965-982, Aug. 1997.
- [2] Tsipora P. Barzilai, Dilip D. Kandlur, Ashish Mehra and Debanjan Saha.: *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, Apr. 1998.
- [3] H. Schulzrinne, S. Casner, R. Frederick and V. Jacobson.: *RTP: A Transport Protocol for Real-Time Applications*, RFC 1889, January 1996.
- [4] Pauli Misikangas and Kimmo Raatikainen.: *Agent Migration between Incompatible Agent Platforms*, Proceedings of the 20th IEEE International Conference on Distributed Computing Systems (ICDCS2000), pp.4-10, Apr. 2000.
- [5] Shinji Tanaka, Hirofumi Yamaki, and Toru Ishida *Mobile-Agents for Distributed Market Computing*, Proceedings of the 1999 International Conference on Parallel Processing, pp.472-479, Sep. 1999.
- [6] Shiratori N., Sugawara K., Kinoshita T. and Chakraborty G.: *Flexible Network: Basic Concepts and Architecture*, IEICE Trans. Communication, Vol.E77-B, No.11, pp.1287-1294, 1994.
- [7] Koji Hashimoto, Yoshitaka Shibata and Norio Shiratori.: *The System Organization and QoS Functions for Flexible Multimedia System*, Proc. of the 6th International Conference on Distributed Multimedia Systems (DMS'99), pp.209-216, Jul. 1999.
- [8] 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, Feb. 2000.