

QOS に基づくマルチキャスト通信*

尾上裕子

NTT 情報通信網研究所

近年、動画像や音声データなどの連続メディア情報を広域分散環境で処理するために、エンドツーエンドで時間制約を保証するリアルタイム通信プロトコル技術が重要視されている。このリアルタイム通信におけるマルチキャスト機構では、グループ全体に統一的な転送制御を提供するために、受信処理能力の低いメンバがいると全体の QOS(Quality Of Service) が低下するという問題点があった。そこで本研究では、従来のエンドツーエンドの統一的な QOS 制御からポイントツーポイントの QOS 制御へと、制御の粒度を細かくし、中継エージェントにおける QOS 制御機能と、各 QOS のデータレートと受信処理能力に応じて動的に決定されるサービスレベルを導入した、MPC(Multicast Protocol for Continuous Media) を提案する。さらに、予備実験を行った結果、MPC 適用の効果として (1) 受信処理能力の多様性を考慮した転送制御、(2) 余分なネットワーク負荷の軽減、(3) アプリケーションレベルでの質の安定の 3 点を示す。

QOS-based Multicast Communication

Yuko Onoe

NTT Network Information Systems Laboratories
1-2356 Take, Yokosuka-shi, Kanagawa, 238-03 Japan

In these days, realtime communication protocols which guarantee end-to-end timing constraints have been applied to process continuous media information, such as video and audio data, in widely distributed environments. Multicast schemes in these realtime communication protocols provide unified transport control for every group member. For this, any member with a low receiving capacity lowers QOS(Quality of service) of whole group. To avoid this problem, this study changes the QOS control from end-to-end unified control to point-to-point control, and then proposes MPC(Multicast Protocol for Continuous Media) which introduces (1)QOS control functions at intermediate agents and (2)service levels specified with data rate of each QOS and receiving capacities. Moreover, the results of preliminary experiment, such as (1)transport control regarding varieties of receiving capacities, (2)reduction of invalid network load and (3)stabilization of quality at application level have been shown as effects of MPC application.

*この研究は、情報処理振興事業協会 (IPA) が実施している開放型基盤ソフトウェア研究開発評価事業「マルチメディア統合環境基盤ソフトウェア」プロジェクトのもとに行われました。本稿に含まれる見解や結論は、著者自身のものであり、IPA 自身の見解や結論を示すものではありません。

1 マルチキャスト通信技術の利用

近年、マルチキャスト通信はデータ通信を大規模ネットワーク上で効率良く行うための技術として注目されている。このマルチキャスト通信は、受信者の集合で成り立つグループに対してデータ転送を行うことにより、送信者に1対1通信と同程度の負担で、複数の受信者への伝搬サービスを提供する。このとき、ネットワーク資源や帯域幅を共有することが出来、同一情報の複製パケットがネットワーク上を伝搬されることの冗長性を避けることができる。また、ブロードキャスト通信と異なり、グループメンバー以外のホストに不必要なデータ転送を行わず、無駄にCPU負荷をかけることもない。

このようなマルチキャスト通信技術は、これまで様々な形態で利用されてきた。例えば、分散型複製データベースの保持およびそれへの応答を処理するために、ファイルサーバは協調的なグループを形成し、対故障性を考慮したファイルサービスを提供する。また、複数の計算機に点在するプロセスグループ内では、メッセージ交換により一連のオペレーションを提供する。さらに、映像・音声などをリアルタイムに多地点間で交換、または一方向的に放送するための分散マルチメディア処理などにも適用されている。

特に分散マルチメディア処理技術に関しては、インターネットネットワーク上の20サイトをIPマルチキャスト通信を用いて接続し(MBONEネットワーク)、年4回開かれるIETFの会議の様子を放送するIETF audio実験が始められている([1])。また、衛星通信とインターネットネットワーク(WAN, LAN)との組を基盤とした通信アーキテクチャWMA(WIDE Multicast communication Architecture)では、一方向型の放送型通信の実験などもなされている([12][16])。ここでは、広域分散環境での資源共有の効率化を図るため、ニュース(NNTP)やメール(SMTP)、PDSなどのファイル配送(FTP)のためのマルチキャスト通信技術の確立を目的とする。

2 リアルタイム通信プロトコルにおけるマルチキャスト機構

一般にマルチメディア情報は、動画・音声などの時系列的に変化する連続メディアデータである。このような情報を分散環境で処理するには、遅延時間およびその変化量の最小化、サーバの周期やデッドラインの指定による時間制約の保証という、これまでのテキスト処理技術には求められていなかった新たな条件を満足する必要がある([8][14])。そのため、エンドツーエンドで時間制約を保証し、予測可能な転送サービスを提供するリアルタイム通信プロトコル技術が求められている。このリアルタイム通信プロトコルを基盤とするTV会議システム・放送システムなどのマルチメディアシステムには、一対多・多対多の通信を司るマ

ルチキャスト性の考慮も不可欠となる。そのため、既存のリアルタイム通信プロトコルでも、様々な形でマルチキャスト機構が取り入れられている。

例えばXTPでは、データ転送のフロー、レート、エラーの制御時にマルチポイントからの応答の最小値を保持し、定期的に更新を行うbucket algorithmなどが適用されている([7])。

また、ネットワークレイヤに位置する接続型通信プロトコルのST-IIでは、ストリーム確立に際し、必要なネットワーク・CPU資源の予約を行うことで、ある一定の転送サービスを保証する。このとき、送信者をルートとしたマルチキャストルーティングツリーを構築し、一対多の片方向マルチキャスト通信が提供される([10])。

さらにこのST-IIの上位プロトコルとして提供されるHeiTPでは、ストリームを接続タイプによってサブストリームに分割し、あるメンバが転送量に追いつけなくなったら、途中からサブストリームを切断するというアプローチを採っている([4])。

最後に、IPマルチキャストプロトコルでのリアルタイム通信に関しては、送信者と受信グループ間で同期を取るためのパラメータとして、転送データの送信間隔(heartbeat)、この時間内に転送されるデータ数(window)、再送処理用のバッファ量(retention)のそれぞれがグループ加入処理において決定される([3])。

3 マルチキャスト機構における転送制御の問題点

マルチメディア情報を扱うための通信プロトコルにおいて、IPプロトコルをはじめとするベストエフォート型のサービスモデルでは、遅延時間のばらつきやネットワークオーバーフローによりパケット損失が生じ、音声かとぎれたり映像が止まるといった問題が挙げられる。その要因の一つとして、ネットワークおよびCPUの一時的な負荷や輻輳などが考えられるが、リアルタイム通信プロトコルではこれらに対処するために、以下の二つの転送制御が行われている。

・資源予約

リアルタイム通信においてある一定の転送サービスを保証するために、CPUやネットワークの必要な帯域幅・バッファ領域をあらかじめ予約しておくという、資源管理方法の一つ。落とせない情報、例えばMPEGのIフレームなどの転送に利用される。

・転送レート制御

転送状況を常にモニタリングし、状況に応じてデータの転送量を動的に変更する。このとき、提供する情報の質を変化させることによりデータ処理量を制御するメディアスケール技術が重要となる([4])。このメディアスケールにより転送データの階層化を行い、負荷や輻輳時には高優先度のデータのみを転送す

るよう制御する。

3.1 統一的な転送制御

XTP では、ネットワーク負荷によるパケットロスなどの事態を抑制するために、受信側のデータ受信処理能力などに合わせて送信側のフロー、レート、エラー処理する制御機構が取り入れられている。例えば前述の bucket algorithm では、受信グループからの受信処理状況の報告に対し、ある一定時間内に収集できた応答のみを採用し、それ以上遅れて到着したパケットや応答の無い受信者は制御対象から省かれる。さらに、その中の最悪値に合わせてグループ全体のデータ転送が制御される。

また ST-II では、転送サービスの保証量である QOS(Quality of Service) を反映した予約資源量は、遅延時間とその変化量・パケットサイズ・転送レート値を FlowSpec として指定することで確保される。ストリーム確立時に FlowSpec には、送信エージェントの要求値および限界値が指定され、この FlowSpec の値によって、順次中継エージェントは資源を予約していく。また、遅延時間の総和などの情報が付加され、パケットサイズや転送レートの要求値は各中継エージェントを通過するごとに更新される。この情報に基づき、受信エージェントはコネクションの受信を決定し、送信エージェントにストリーム確立受理メッセージを返す。

このようにして FlowSpec は、隣接中継エージェント間をポイントツーポイントで設定されいく。しかし最終的には、送信エージェントがエンドツーエンドでの帳尻合わせを行い、全てのグループメンバーに対し最小値に合わせた同一の QOS の転送サービスを提供している。

3.2 従来技術の問題点

従来の帯域予約では、あらかじめ予約可能な帯域の最小値をとって、グループメンバーまでの全経路を予約するというアプローチを採っている。そのため、仮に途中に帯域幅の狭い回線が入ると、そこに合わせて送信側でグループ全体の QOS の制御を行うために、そこから先の受信者だけでなく、同一グループのメンバー全員がそこに合わせた帯域のみの利用に制約されることになる。また、ネットワークや CPU に一時的な負荷が発生した場合、送信側で QOS を制御することにより転送量を低下させるメディアスケリングの技術は、グループ全体に適用される。ここで考えられる問題点を以下に挙げる。

1. グループのメンバー全員に対して統一的な転送サービスを保証するために、その転送サービス量に追い付けない受信者およびそれ以上の転送サービスを要求する受信者はグループのメンバーにはなれない。

2. グループ内のあるメンバーの受信能力が低いと、グループメンバー全員の転送サービスがその値に制限され、全体的なサービス量 (QOS) が低下する。

3. 相手によって相対的に転送制御を行うと、一対一のユニキャスト型になってしまい、ネットワーク資源の共有というマルチキャストの利点を生かせない。

4 QOS に基づくマルチキャスト通信機構

ここでは、従来のエンドツーエンドの統一的 QOS 制御から、ポイントツーポイントの QOS 制御へと制御の粒度を細かくすることを提案する。それによって、受信能力の違いを隠蔽し、受信者のネットワーク形態や計算機処理能力に応じた通信を、同一グループの他のメンバーに意識させずに提供する。さらに、様々なレベルの QOS を求めるターゲットの要求を効率良く満たし、グループメンバーの多様性を維持することが可能になる。そこで、(1) 中継エージェントにおける QOS 制御機能と、(2) 各 QOS のデータ発生レートと受信処理能力に応じて動的に決定されるサービスレベルを導入し、従来のエンドツーエンドの通信からポイントツーポイントの通信へと、より下位のレイヤレベルでの機能実現について検討する。

4.1 中継エージェントにおける相対的 QOS 制御

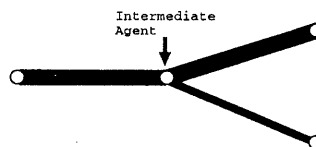


図 1: 中継エージェント

分岐点となる中継エージェントでは、受信処理能力に応じて相対的に QOS を制御する。そのためには、受信側のエージェントが自分の受信能力を中継エージェントに示す必要がある。ここでは、データパケットの受信処理状況を中継エージェントに報告する。この報告に従って中継エージェントでは、受信者ごとに転送データを制御してデータを送信する。すなわち、受信者が送信されたデータのうち受信できたデータのみを処理するのではなく、送信側でデータを選択的に転送するという、送信側主導の QOS 制御を提供する。これにより、受信能力の低さがもたらす送信データ品質低下を緩和することになる。ここで、もしこれらの中継処理量の最大値が中継エージェント自信の受信処理量よりも小さくなったら、それを中継エージェントの新たな受信処理能力とし、その前の中継エージェントに転送する。このようにして、隣接中継エージェント間で受信能力を示すステータスメッセージを経路を逆方向に伝搬していき、無駄なデータ転送を避ける。

4.2 サービスレベルの導入

受信者の受信処理能力に応じた相対的な QOS 制御を行っていく上で、相対性を重要視するがゆえに中継エージェントで対一通信を行っているのは、ネットワーク資源の共有というマルチキャスト通信の利点を維持できない。そこでマルチキャストグループに対し、QOS 保証レベルを反映したサービスレベルを導入する。このサービスレベルは、各 QOS のデータ発生レートと受信者の受信処理能力とから決定される。このようにして、固定的なマルチキャストグループの中に、このサービスレベルに応じて動的にサブグループを決定し、各サブグループに応じた QOS を提供する。このとき受信者は、

(group ID, service_level)

の組を保持してパケットの受信処理を行う。

4.2.1 QOS とサービスレベル

転送制御が扱うマルチメディア情報の QOS として、以下の 2 つが挙げられている ([2][9][13][15])。

1. 時間的解像度

動画の場合は単位時間当たりのパケット数、音声の場合はサンプリング周波数が挙げられる。例えば動画像に関しては、一定時間内に転送するフレーム数を制御することで QOS が変化し、情報内容の時間的変化をより詳細に追うことが可能になる。

2. 空間的解像度

動画像の場合はフレームサイズ、音声の場合は量子化ビット数が挙げられる。動画像では、各フレームのピクセル数を制御することにより、画像の鮮明度が変化する。

実際のシステムでは、以上のような QOS の代表値を QOS クラスとして設定する ([15])。ここで提案するサービスレベルは、これらの各 QOS とそのクラス値との組合せで構成される。

4.2.2 QOS フィールド

転送制御の対象となるパケットには、情報内容のサービスレベルを示す QOS フィールドを設定し、この QOS 値をネットワークレベルで解釈することによってデータパケットは制御される。例えば、図 2 に示すように 1 フレームを PDU に従って 4 パケットに分割した場合、(1) の空間的解像度においては同一フレーム内の各パケットには 1 から 4 の値が、(2) の時間的解像度においては同一フレーム内で同じ値が、それぞれ QOS 値として設定される。ここで、サービスレベルが 2 の時には、(1) の空間的解像度では値が 3 と 4 のパケットは間引かれ、残りの 2 パケットのみでフレームを構成し、(2) の時間的解像度では値が 2 と 4 のパケットを間引き、2 フレーム毎に 4 パケットでフレームを構成することによって、QOS 制御を提供する。

さらに、このサービスレベルは複数の要因で構成されることもある。例えば、空間的解像度と時間的解像度の 2 項目を同時に扱う場合、パケットの QOS フィールドの設定・解釈がそれぞれ異なる (図 2)。そこで、QOS フィールドを多次元で階層的に表現し、項目毎に QOS 制御を適用する。さらにここでは、階層性をパケットフォーマットの 1 フィールドで表現するために、QOS フィールドにマスクをかけることによって操作する。例えば、QOS フィールドが 4 ビットの場合、上 2 ビットを時間的解像度の値、下 2 ビットを空間的解像度の値とする。

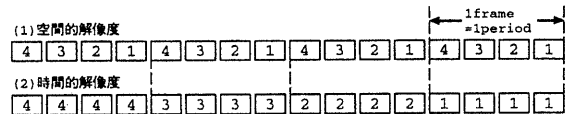


図 2: 時間的解像度と空間的解像度

5 MPC(Multicast Protocol for Continuous media) の設計と適用例

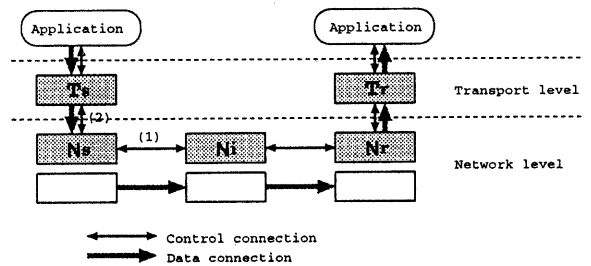


図 3: マルチキャスト機構の構成

前章で述べたマルチキャスト機構における QOS 制御は、トランスポートレイヤにおいてエンドツーエンドでの制御と、ネットワークレイヤにおけるポイントツーポイントでの制御の 2 形態で管理の住み分けが行われる。トランスポートレイヤでの制御は、以下の 2 つの機能により実現される。

- a. サービスレベル決定機能 (受信側)
- b. サービスレベル変更機能

また、ネットワークレイヤは、以下の 3 つの機能により実現される。

- a. サービスレベル設定機能 (送信側)
- b. サービスレベル設定機能 (受信側)
- c. サービスレベル伝搬機能

以下、各機能の詳細と適用例について述べる。

5.1 トランスポートレイヤ

- a. サービスレベル決定機能 (受信側)

受信者は、ネットワークレイヤから渡された受信処理能力 (FlowSpec) と、送信者からの各サービスレベルの

パケット発生レート情報(DataSpec)から、自分が受信するサービスレベルを選択する。また、常にエンドツーエンドのFlowSpecを監視し、最小値を切ったところでDataSpecとFlowSpecを突き合わせ、サービスレベルを変更する。

b. サービスレベル変更機能

サービスレベルが変更されたら、ネットワークレイヤへ変更処理の伝搬を要求する。

5.2 ネットワークレイヤ

ネットワークレイヤレベルで確立されるコネクションは、制御用コネクションと転送用コネクションの2系統で構成される([5][6])。制御用コネクションは、実際にデータを転送するコネクションの制御を行う。ここで提案するQOS制御は、この制御用コネクションに適用される。

a. サービスレベル設定機能(送信側)

a1. 各送信者は、ネットワークインタフェース毎にネットワーク上のメンバのサービスレベルを保持する。

a2. 転送データを受信したら、転送データのQOSフィールドを解釈してサービスレベルと対応付け、ネットワーク毎にそのサービスレベルのメンバが存在するかをチェックし、いればそのインタフェースにデータを転送する。

b. サービスレベル設定機能(受信側)

b1. 受信者は自分の受信処理能力を報告し、転送サービスレベルを割り当てを受ける。また、自分が要求するサービスレベルを申告する。そしてこのサービスレベルを保持する。

b2. 受信者はデータのサービスレベルと自分の属するレベルが一致すればそのパケットを受信する。

c. サービスレベル伝搬機能

ストリームを逆向きにサービスレベルを伝搬し、インタフェース毎に伝搬するサービスレベルを設定していき、転送データの切り詰めや変更を行う。

5.3 適用例

ここでは、送信側のエージェントにおける1フレーム内のパケット送信間隔が無視できるほど小さいこと、さらにパケット受信間隔を定義するために1フレームを構成するパケット数は2個以上であることが前提となる。

5.3.1 サービスレベル設定

まず最初に受信エージェント(T_r)は、送信アプリケーションからMDUとして1フレームのサイズとその発生周期(PERIOD)を得る。そしてそれをPDUごとに分割し、転送データの形態を示すDataSpecとしてパケットサイズ(PACKETSIZE)・1フレーム中の

パケット数(PACKETNUM)・1フレームの発生周期(PERIOD)を保持する。

受信エージェント(T_r)は、受信処理能力を知るために、1周期時間(Period)内に受信処理したパケット数(Packetnum)とサイズ(PACKETSIZE)から、以下のようしてFlowSpec中の実際のスループット値(T)を計算する。

$$T = \frac{PACKETSIZE \cdot Packetnum}{Period}$$

ここで、時間的解像度における受信処理能力に関しては、Periodは1フレーム分のパケットが受信処理された時間(PacketInterval・PACKETNUM)、Packetnumは1フレームのパケット数(PACKETNUM)となり、空間的解像度に関しては、PeriodはPacketnum分のパケットが受信処理された時間間隔(PacketInterval・Packetnum)で、最大値は1フレームの発生周期(PERIOD)となり、Packetnumはこの周期毎に受信処理されたパケット数となる。

このスループットの実測値から、次の式によって受信優先度群を決定する。

$$T \geq \frac{PACKETSIZE \cdot Packetnum'}{Period'}$$

時間的解像度に関してPeriod'はPERIOD・xでPacketnum'はPACKETNUM、空間的解像度に関してPeriod'はPERIODでPacketnum'はxとなると、xの限度値から受信エージェントが受信するQOS値を決定する。

上述の手段で得られたQOS値が共通の受信エージェントごとにサブグループを形成し、各サブグループに対しサービスレベルを割り振る。空間的解像度をX、時間的解像度をYとした例を表1に示す。

表1: サービスレベル設定例

受信エージェント	QOS 値	サービスレベル
A_1	$\{x_1, x_2, x_3\}$	X_3
A_2	$\{x_1\}$	X_1
A_3	$\{y_1, y_3\}$	Y_2

送信エージェントは、インタフェース毎に一つ先のネットワーク上に存在する受信エージェントが属するサービスレベルの最大値(X_3, Y_2)を保持しておく。また、各受信エージェントはデータ受信処理のためにサービスレベルを保持する。

5.3.2 転送制御

次に、データ送信時の転送制御について説明する。データ送信側のアプリケーションはデータ送信側エージェントに対し、データパケットの送信を依頼する。このとき、データパケットごとにQOSフィールドが

設けられている。送信エージェントは、サービスレベル設定手順であらかじめ保持するインタフェースとサービスレベルの対応表から、送信しようとするインタフェース側にその転送バケットを受信する受信エージェントが存在するかどうかを調べ、存在すればそのインタフェースからバケットを送信する。各データ受信エージェントは、送信されるバケットのグループアドレスと優先度フィールドから自分宛かそうでないかを判断し、自分宛のバケットを受信する。

5.3.3 資源予約

コネクション確立時の資源予約に、ここで提案するマルチキャスト機構を適用する場合の制御方法について述べる。

1. 送信側のエージェント (T_s) はネットワークレイヤのエージェント (N_s) に対し、コネクション確立要求メッセージを送信する。このコネクション確立要求メッセージには、転送データのバケット発生レート情報を含む DataSpec と FlowSpec が含まれている。エージェント (N_s) は受信側のエージェント (N_r) まで、FlowSpec に利用可能な帯域幅を加算しながらポイントツーポイントに確立要求を伝搬していく。

2. このメッセージを受信した受信エージェント (T_r) は、FlowSpec 内のメッセージ内の利用可能な帯域幅と DataSpec を突き合わせ、受信データの QOS を選択してサービスレベルを決定する。そして、受信データのバケット発生レート情報から必要な帯域幅を逆算し、返答メッセージに要求するサービスレベルと必要な帯域幅を指定した FlowSpec を含めて送信側エージェント (T_s) に転送する。

3. この返答メッセージに従い、中継エージェント (N_i) では随時資源予約とサービスレベルの登録を行いながら、送信側に返答メッセージを伝搬する。このとき、既存のセッションへの追加であれば、中継エージェントは各インタフェース毎に既に予約した帯域幅を保持しており、予約量を上回る場合のみ帯域幅の再予約を行い、インタフェースにサービスレベルを登録する。

5.4 既存のプロトコルとの比較

前述のように ST-II では、3回のネゴシエーションによって、全ての受信メンバに統一的な FlowSpec で資源予約を行っている。これに対し、MPC では2回のネゴシエーションにより、受信能力の違いを加味した相対的な資源予約がなされる。さらに転送制御に関しても、MPC では中継エージェントでの QOS 制御機能により、メンバ毎に異なる転送制御を行う。

また RSVP では、(予約資源量 (FlowSpec)、予約方法 (固定・動的・無し)、フィルタ) を指定したメッセージを受信者から送信者方向に伝搬していき、各ルータではメッセージに従って一定の資源量を予約していく ([11])。この RSVP では、予約量に応じてあらゆる

優先度のデータバケットが均等に制御されることになる。そのため、ある一定の転送量を保証することになるのに対して、MPC ではデータの QOS を解釈し、予約量に応じて転送制御は行うことにより転送の質の保証を目的とする。

前述の HeiTS では、あらかじめストリームを QOS に従ってサブストリーム化し、複数のサブストリームを同期させることによりセッションを構成する。すなわち、MPC がポイントツーポイント型の QOS 制御を行うのに対して、HeiTS ではエンドツーエンド型の QOS 制御となる。これらの2形態のプロトコルにおいて、以下のような相違点が考えられる。

- ・転送制御の粒度
- ・サブストリーム管理のオーバーヘッド
- ・複数の要因の QOS の取扱い
- ・スケールアップの処理
- ・複数経路への分割

6 適用実験

本論文で提案するマルチキャスト機構を取り入れたマルチメディア通信システムのプロトタイプを、UNIX 上で UDP/IP を用いて実装した。データ転送に際してデータ転送用コネクションと制御用コネクションの2つのコネクションを張り、制御用のコネクションではコネクション確立要求、再確立要求およびサービスレベル変更要求などのコントロールバケットが交換される。このコントロールバケットには、制御タイプ・サービスレベル・フレーム周期と画像データに関する情報が含まれている。

6.1 実験環境

実験環境として、慶応大学藤沢キャンパス IPA プロジェクトと横須賀通研を実験用に接続した 1.5M の ISDN と通研内の FDDI の二つのネットワークにつながるゲートウェイマシン (SUN WS) と、FDDI 上のマシン A (DEC ステーション)・マシン B (SUN WS) および ISDN のマシン C (SUN WS) を用いた。ここで、ゲートウェイから 80ms 毎に 500 x 8 Bytes のバケットを転送し、各マシンでは受信したバケットを 4K Bytes のフレームに組み立て、X ライブラリを通して画面に表示する実験を行った。この時、送信側のサーバの周期性に対して受信側で同期をとるように、80ms を受信側での表示周期として設定し、周期内で受信できたバケットのみでフレームを組み立てて表示した。

6.2 実験結果

まず、複数の受信者に対して転送制御を適用した場合、単位時間当たりのデータバケット受信処理量を測定した。その結果、マシン B およびマシン C では転送制御によって、それぞれ 1、3 バケットの受信処理に抑

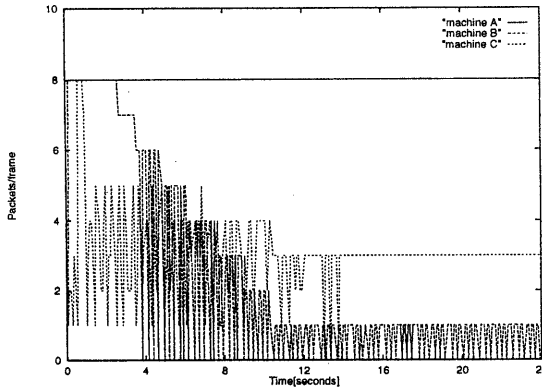


図 4: データパケット受信処理量

えられているが、マシン B と同一ネットワーク上の受信処理能力の高いマシン A では、同じ画像データに対し常に 8 パケットの受信処理が可能となっている。

る。これは、マルチキャスト通信と回線を共有する他のコネクション、例えば rlogin 等の応答時間に与える影響を示している。

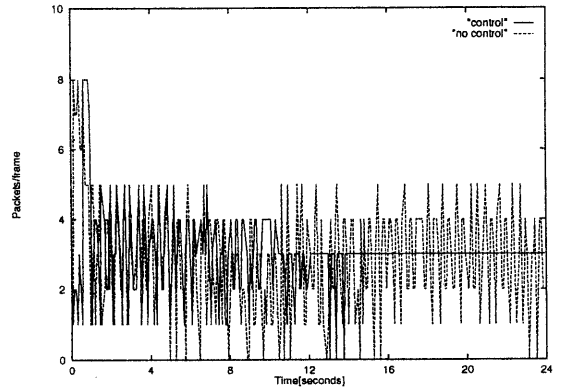


図 6: 質の変動

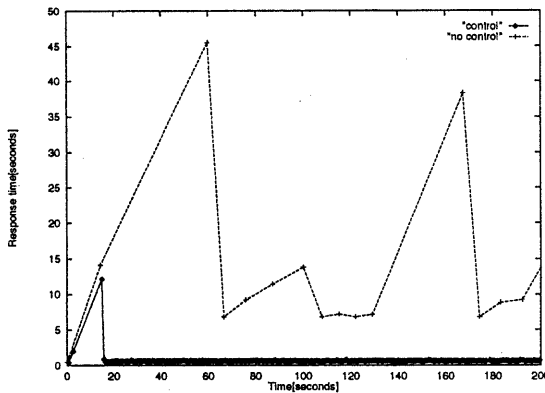


図 5: TCP の遅延時間

次に、1.5M の ISDN 上のマシン C において、図 4 の転送制御を行った場合 (control) と、この制御を行わなかった場合 (nocontrol) とで、TCP コネクションの遅延時間を測定した (図 5)。その結果、転送制御無しでこのマルチキャスト通信を行うと、転送量に追いつけない回線ではネットワークに輻輳が生じ、その他のコネクションの遅延時間に影響が出てくる。これに対し、受信能力に依存して転送制御を行うと、帯域幅の低いメンバのネットワークに輻輳を生じさせる事もなく、しかもこの時図 4 で示すように、他のメンバの転送量は維持される。つまり MPC の転送制御を用いた場合、FDDI 上のマシン A の様な他の受信者の転送サービスの高さは維持したまま、一方の ISDN の様な細いネットワーク上の不必要な輻輳の軽減につながる

同時に、ネットワークレベルでの転送制御が上位アプリケーションに与える影響を調べるために、上記の場合の単位時間に受信処理したパケット数を測定した。その結果、転送制御無しの場合は単位時間に受信処理したパケット数に大きな変動が見られる。これは、受信アプリケーションが一定の周期でフレームを表示するとき、画像の解像度の変動となって表れる。このように、ネットワークレベルでの転送制御の有無によって、アプリケーションレベルで表示される画像の質が頻繁に変化することが示されている。

6.3 考察

マルチメディア通信システムの基盤となるリアルタイム通信プロトコルにおいては、IP の様なベストエフォート型サービスモデルで転送を行い、輻輳時に転送制御にて対処する方法と、その一方で ST-II の様なあらかじめ必要量の資源を予約してしまう方法の二つの傾向が見られる。ここでインターネットでの利用を考慮すると、マルチメディアシステムの利用者に対して有効なサービスを提供すると同時に、そのサービスがネットワークを共有する他の利用者を思いやるサービスであることも必要である。この意味で、リアルタイム通信プロトコルを設計するにあたっては、図 5 のような視点も重要であると考える。

7 課題

本研究で提案する MPC では、以下に示す課題について検討する必要がある。

1. ネットワークレイヤでのサービスレベル決定機能
今回は、サービスレベルの決定処理を、トランスポー

トレベルでのエンドツーエンド型で行っているが、今後はポイントツーポイント型の FlowSpec の総計としての転送制御について、考察する。

2. 多次元的サービスレベルの扱い

前述のような多次元的要因をネットワークレイヤで提供する場合、全体で提供されるサービス量が一定であることの副作用として、各メンバが受信するサービス量の低下と、ネットワークを共有する他のメンバの動向の影響などが考えられる。

3. アプリケーションとの協調

今後は、上位レイヤが指定する MDU の幅から最適値を動的に決定し、変更に応じてサービスレベルの仕切り直しを図るような処理が望まれる。

4. 取り扱う QOS の拡大

今回は、特に動画像のみを対象として検討したが、音声も含めた圧縮技術や色情報など、より多くの QOS が提供できるようにする。

8 まとめ

本稿では、連続メディアを扱うためのリアルタイム通信プロトコルにおいて、QOS に基づいた転送制御を取り入れたマルチキャスト通信機構 (MPC) の設計について述べた。これまで提案されているリアルタイム通信プロトコルでは、ネットワークや CPU の一時的な負荷に対処するために統一的な転送制御を適用していた。このため、混成ネットワーク形態では、受信処理能力の低いメンバによってグループ全体の転送サービス量が低下するという問題点があった。そこで本研究では、エンドツーエンドの統一的 QOS 制御からポイントツーポイントへと制御の粒度を細分化し、中継エージェントにおける転送制御と、データ発生レートと受信処理能力から動的に決定されるサービスレベルを導入した MPC を提案した。さらに、適用実験を行い、MPC 適用の効果として (1) 受信処理能力の多様性を考慮した転送制御、(2) 余分なネットワーク負荷の軽減、(3) アプリケーションレベルでの質の安定の 3 点を示した。

9 謝辞

本研究を行うにあたりご協力頂いた開放型基盤ソフトウェア研究開発評価事業「マルチメディア統合環境基盤ソフトウェア」プロジェクトの皆様へ感謝いたします。さらに、ご指導頂いている慶応義塾大学環境情報学部の齋藤信男教授、徳田英幸助教授に感謝いたします。また、いつも温かい励ましのお言葉を頂いている WIDE プロジェクトの皆様へ感謝いたします。

参考文献

[1] S. Casner and S. Deering. First IETF Internet Audio-cast. *ACM SIGCOMM Computer Communica-*

tions Review, vol.22, no.1.3, 1992.

- [2] S.T.-C. Chou and H. Tokuda. System Support for Dynamic QOS Control of Continuous Media Communication. In *Proceedings of 3rd International Workshop on Network and Operating System Support for Digital Audio and Video*, 1992.
- [3] S. Deering. Host Extensions for IP Multicasting, 1989. RFC1112.
- [4] L. Delgrossi, C. Halstrick, D. Heilmann, R.G. Hertrich, O. Krone, J. Sandvoss, and C. Vogt. Media Scaling for Audiovisual communication with the Heidelberg Transport System. 1993.
- [5] S. Kihara, Y. Onoe, A. Mituzawa, A. Nambu, and H. Tokuda. An implementation of ST-II Protocol as a user-level server on RT-Mach. In *Proceedings of 4th International Workshop on Network and Operating System Support for Digital Audio and Video*, 1993.
- [6] A. Shionozaki and M. Tokoro. Control Handling in Real-Time Communication Protocols. In *Proceedings of SIGCOMM93*, ACM Press, 1993.
- [7] W.T. Strayer, B.J. Demsey, and A.C. Weaver. *XTP: The Xpress Transfer Protocol*. Addison Wesley, 1992.
- [8] H. Tokuda, T. Nakajima, and P. Rao. Real-Time Mach: Towards a Predictable Real-Time System. In *Proceedings of Mach Workshop*, USENIX Association, October 1990.
- [9] H. Tokuda, Y. Tobe, S.T.-C. Chou, and J.M.F. Moura. Continuous Media Communication with Dynamic QOS Control Using ARTS with an FDDI Network. In *Proceedings of SIGCOMM92*, ACM Press, 1992.
- [10] C. Topolcic, S. Casner, C. Lynn, P. Park, and K. Schroder. Experimental internet stream protocol, version 2(ST-II), 1988. RFC1190.
- [11] L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala. RSVP: A New Resource Reservation Protocol. *IEEE Network*, 1993.
- [12] 寺岡文男 and 楠本博之. HDVMPR: 広域ネットワークに適したマルチキャスト経路制御プロトコル. 第10回大会論文集, 1993.
- [13] 盛合敏. リアルタイム・オブジェクト向けの IPC について. 情処研報 93-OS-60-2, 93-DPS-61-2, July 1993.
- [14] 西尾信彦, 追川修一, 緒方正陽, 尾上裕子, 河内谷清久仁, 塩野崎教, 南部明, 持田茂人, 和田英彦, and 徳田英幸. マイクロカーネルによる連続メディア処理の基盤技術. 第5回コンピュータシステムシンポジウム論文集, October 1993.
- [15] 船渡大地 and 徳田英幸. Real-Time Mach3.0 における連続メディアサーバの実験. 情処研報 93-OS-60-2, 93-DPS-61-2, July 1993.
- [16] 楠本博之, 寺岡文男, and 村井純. 広域ネットワークにおけるマルチキャスト通信アーキテクチャ. コンピュータソフトウェア Vol.10 No.4, 1993.