

マルチキャストサーバにおける動的 QOS 制御*

尾上裕子
NTT 情報通信網研究所

藤井敬三
(株) システムコア

徳田英幸
慶應義塾大学環境情報学部
カーネギーメロン大学計算機科学部

リアルタイム通信プロトコルにおいて、従来の統一かつ固定的転送制御に対し、混成のネットワーク形態での相対的かつ動的な QOS を提供する転送制御方式の確立を目標とした。そのため、トランスポートレイヤでのエンドツーエンドの QOS 制御と、ネットワークレイヤでのポイントツーポイントの QOS 制御との協調制御方式を提案する。具体的には、(1) ネットワーク毎に異なる QOS を提供するための中継エージェントにおける QOS 制御機能と、(2) 転送フローの状態に応じてエンドツーエンドの QOS を変更するためのフローモニタリング・フロー調整機能を導入する。このフロー調整機能では、転送しようとするメディア情報の特性・エンドツーエンドのネットワーク性能特性・受信側アプリケーションの要求条件の3点を総合的に評価し、QOS レベルとフロー制御方式を決定する。

次に、本稿で提案する MPC を適用したマルチキャストサーバを、RT-Mach 上で実装し、フロー調整機能のうちの前者の2点について適用実験を行った。実験結果から、マルチキャスト通信に QOS レベルを導入することによって、複数のクライアントが同一メディアフローの転送サービスに対して個々に異なる性能パラメータ値を要求することができ、その結果例えばグループへの参加形態に応じた相対的な要求条件を満足することが可能となると考える。

Dynamic QOS Control Schemes on Multicast Servers

Yuko Onoe Keizo Fujii Hideyuki Tokuda
NTT Network Information Systems Laboratories System Core Co. Ltd. Institute of Environmental Information, Keio University
School of Computer Science, Carnegie-Mellon University

This study aims to establish transmission control methods in realtime communication protocols that provide a relative Quality Of Service(QOS) to allow a variety of receiving capacities of group members and their requests, and a dynamic QOS to cope with temporary CPU overload, and network congestion.

Therefore, this study proposes a cooperative control method for an end-to-end QOS control at a transport layer such as flow monitoring and adjusting functions, and a point-to-point control at a network layer such as flow control functions.

Next, the multicast server applying a Multicast Protocol for Continuous media (MPC) have been implemented on a RT-Mach. The results support that introducing QOS levels into multicast communication allows various performance parameters for transport services of shared media flows and satisfies relative requirements according to the styles of participation in group communications.

*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。

1 はじめに

近年、動画や音声などの連続メディア情報を、広域分散環境で統合的に処理する分散マルチメディア処理技術への要求が高まっている。この連続メディア情報を扱うには、メディアのタイムクリティカルな性質上、処理サーバの周期やデッドラインの指定、サービスの優先順位制御やスケジューリングポリシーによる時間制約の保証と処理の決定性、予測可能性といった、リアルタイム性を考慮したリアルタイム OS が適用されている [5][6]。さらに、連続メディア処理と通信との融合を図るために、リアルタイム OS ではユーザレベルアプリケーションが要求する処理性能の特性を、広域分散環境下でのエンドツーエンドで満足する必要がある。そのため、転送データの遅延時間およびそのばらつきの最小化、各ルータにおける優先度を加味したスケジューリングなどをネットワークワイドに行うための、リアルタイム通信プロトコルが検討されている [2]。

2 リアルタイム通信プロトコルにおける転送制御

連続メディア情報を扱う通信プロトコルにおいて、IP などのベストエフォート型のサービスモデルでは、パケット到着遅延時間のばらつきや CPU の負荷・ネットワーク輻輳によりパケット損失が生じ、音声がとぎれたり映像が止まるといった問題が挙げられる。そこで、XTP, ST-II, HeiTP などの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、あらかじめネットワークや CPU の必要量を予約しておく資源予約方法と、転送状況に応じて転送量を動的に変更する転送レート制御方法の二つの転送制御が行われている。

2.1 本研究の転送制御

従来の転送制御では、(1) グループメンバー全員に一意の QOS を提供する統一的転送制御であること、(2) 一度決定した帯域幅は変更しづらい固定的転送制御であることが問題点として挙げられる。そこでここでは、以下の二点を満足する転送制御方式の確立を目標とする。

1. 混成ネットワーク形態において、グループメンバーの受信処理能力および要求の多様性を許容した相対的な QOS の提供
2. アプリケーションからの要求や、ネットワークや cpu の一時的な負荷および輻輳時に対処するための動的な QOS の提供

以上の二点を満足するために、従来のエンドツーエンドの統一的 QOS 制御に対し、トランスポートレイヤでのエンドツーエンドの QOS 制御と、ネットワークレイヤでのポイントポイントの QOS 制御との協調制御方式を提案する。具体的には、(1) ネットワーク毎に異なる QOS を提供するための中継エージェントにおける QOS 制御機能と、(2) 転送フローの状態に応じてエンドツーエンドの QOS を変更するためのフローモニタリング・フロー調整機能を導入する。これにより、前述の二点を満足すると共に、メンバーの受信処理能力の範囲内で高優先度のパケットのみを送信するように送り手側で制御することにより、受信処理能力の低さを緩和することが可能になる。

2.2 転送制御のための QOS レベル

リアルタイム通信プロトコルでは、遅延時間とその変化量・転送レート・パケットロス率などを FlowSpec として指定することで、転送サービスの保証量である QOS を満足しようとする。この QOS と FlowSpec の関係を以下に示す。

表 1: QOS と対応する FlowSpec

QOS	FlowSpec	アプリケーションとの関連
情報の新鮮度	delay	TV 会議と放送型会議
情報の連続性	jitter	動画像のなめらかさ
情報の解像度	throughput	時間的解像度と空間的解像度
情報の信頼性	error rate	音声データ・圧縮データ

これらの QOS をグループ通信で取り扱うために、グループアドレス機構に QOS を反映した QOS レベルを導入する。この QOS レベルは、固定的なマルチキャストグループ中に動的にサブグループを形成し、各サブグループに応じた QOS を提供するものである。このとき、グループのメンバーや中継となるエージェントでは、

(session ID(group address, port), QOS_level)

の組をもって、パケット伝搬処理を行う。この QOS_level は、ネットワークや CPU の処理能力などの潜在的な要因と、負荷の状況などの動的な要因の両者に依存して決定される。また、転送制御の対象となるパケットフォーマットには、各 QOS の値を指定する QOS フィールドを付ける。この QOS フィールドを多次的に表現することにより、同時に複数の要因の QOS を採作することも可能となる。以下にメディアスケジューリング技術における QOS レベルの例を示す [1]。

表 2: QOS レベル

対象	Q1	Q2	Q3
音声			
信号	L+R(mono)	L-R	
周波数領域	10-4k 電話程度	4k-10k AM 放送	10k-20kHz FM 放送
音部	有声音部	無声音部	
JPEG			
NTSC フィールド	偶数	奇数	
フレーム送出周期	偶数	奇数	
変化の度合い	変化大	変化中	変化小
MPEG			
ピクチャ	I-frame	P	B

3 MPC(Multicast Protocol for Continuous media) の設計

次に、マルチキャスト通信に QOS 制御を取り入れた、MPC の設計について述べる。この MPC では、トランスポートレイヤとネットワークレイヤとで、以下のように管理の住み分けを行った。

a. トランスポートレイヤ

- ・フローモニタリング機能
- ・フロー調整機能
 - ジッタ処理
 - パケットフィルタリング
 - QOS に基づくキュー管理
 - スケジューリング調整

リアルタイムサービスを求めるアプリケーションレベルと、通信サービスを提供するネットワークレベルの間に位置し、アプリケーションが要求するサービス品質を解釈し、ネットワークで保証される転送サービス性能の各パラメータとのマッピングを図る。エンドツーエンドでの QOS を保証するため、FlowSpec に基づきデータ転送フローのモニタリングおよびフローの再調整を提供する。フローの状態を常に把握し、QOS の要求や交

涉、選択などを行う。

b. ネットワークレイヤ
・フロー制御機能

受信エージェントの処理能力および要求に従った相対的な QOS を提供するため、トランスポートレイヤの決定に従い、実際にデータ転送フローの制御を行う。

3.1 フロー調整機能

この MPC では、インターネットワーク上で転送されるマルチメディアフローを、送ろうとするメディア情報の特性・ネットワークから受ける転送サービス性能・エンドポイントの要求条件の 3 点によって定義する。そこでフロー調整機能では、QOS レベルの決定のために以下の 3 つの情報を保持する。

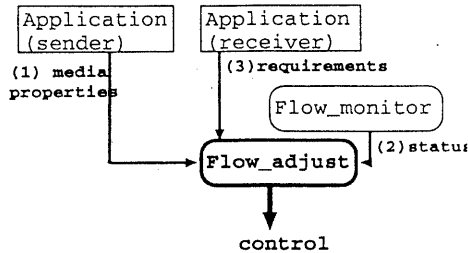


図 1: フロー調整機能

1. ここではマルチメディアデータを、最小限必要な基本要素とそれからの差分に従って階層的に符号化し、この階層に従って優先度付けを行ったパケットに分割して転送することを仮定している。この階層を QOS レベルとし、メディア情報の特性として、各 QOS レベル毎にパケットサイズや転送レートを保持する。

2. フローのモニタリング機能により、受信パケットのパケット番号やフレーム番号、チェックサムなどの情報から、スルーput・遅延をはじめとするエンドツーエンドの FlowSpec の実測を行い、フロー調整機能でこれを保持する。

3. エンドポイントの上位アプリケーションから、FlowSpec の各属性に対するアプリケーション側の感度、保証の度合いなど、受信フローに対するアプリケーション側の要求条件を得る。

フロー調整機能では、以上の 3 点を総合的に評価して QOS レベルと制御方式を決定し、実際の制御を依頼する。

3.1.1 エンドポイントの要求条件

前章で示したように、アプリケーションで要求されるサービス品質をネットワーク通信プロトコル部で保証しようとするには、アプリケーションで認識される QOS を、FlowSpec をはじめとする制御パラメータに対応付けする必要がある。そこでフロー調整機能では、従来より提案されているアプリケーションインタフェースに QOS レベルの項目を追加し、以下のよう定義する [4]。

1. 各性能パラメータに対する限界値

(1) 遅延時間に対する感度

TV 会議などのようにインタラクティブ性を重視する場合には遅延時間は大きく影響するが、放送型会議のように一方向型の場合には、アプリケーションは遅延時間よりもむしろその安定度合いを示す変化量やメディアの解像度に関与する。この遅延時間に関しては、モニタリングによって常に平均遅延時間を測

性能パラメータの限界値
worst-case delay
worst-case jitter
max error rate
max packet loss rate

定し、さらにそこから許容時間を加えた時間を静的な受信タイミングとして設定する。

(2) ジッタに対する感度

遅延時間の変化量であるジッタは、動画像の再生が一時的に停止したり逆に早まるといった現象につながる。このジッタは、前述の初期値的に決定された遅延時間と各パケットの実際の到着遅延時間の差分となる。

(3) エラーやパケットロスに対する感度

アプリケーションが圧縮データを扱う場合には、1 ビットのエラーでも解凍時に大きく影響し、フレーム全てが再生できない事態が生じる。これに対し、アプリケーションやパケット自身がハミングコードや ECC などのようなエラーコレクション機能を備えている場合には、ネットワークプロトコル部での多少のエラーは許されるものとする。

2. 保証の程度

QOS_level	guaranteed_level
Q1	guaranteed
Q2	predictable
Q3	best effort

ネットワークプロトコル部での保証の程度は、メディアフローの特性が、絶対的な帯域幅保証を要求するものであるか、平均値や割合を保証しある程度の変動は許容するような予測可能なものであるのか、さらにネットワーク性能の変動に対しアプリケーションがどの程度柔軟に対応する機能を提供するかなどにより決定される。

3. 時間・空間的解像度のトレードオフ、色情報の必要性

QOS_type	priority
temporal resolution	high
spatial resolution	low

メディアスケーリング技術の発展により、輻射時には緩衝的にサービス品質を低下させているが、このときの低下させる対象にも動画の場合時間的解像度・空間的解像度・色調など候補は様々であり、アプリケーションに指定される優先度に従って、各属性の QOS レベルのスケールダウンを行う。

3.1.2 ネットワーク性能

MPC において、下位レイヤで提供される帯域制御は、遅延時間やスルーputなどの FlowSpec の最悪値を保証するものであることを想定する。つまり、ネットワーク輻射時にはある一定値が保証されるが、通常はトラフィックの総計が回線容量までは使用可能とする。このとき、メディアフローが受けるネットワーク性能は常に予測不可能であり、状況変化に迅速に対応するためのモニタリングを必要とする。リアルタイムサービスに対するネットワーク性能パラメータを以下に示す。

上で示すネットワーク性能を測定するにあたっては、必要な属性をパケットヘッダに付け、エンドノードではその値によ

性能パラメータの実測値
measured throughput
measured delay
measured jitter
measured error rate
measures packet_loss rate

てエンドツーエンドのネットワーク状態を測定する。パケットヘッダを以下に示す。

表 3: パケットヘッダ

GROUP ADDRESS	DEST PORT
FRAME SEQ NUM	PACKET SEQ NUM
TIMESTAMP	
CHECKSUM	

パケットヘッダ中の GROUP ADDRESS と DEST PORT の組が Session ID となる。また、FRAME SEQ NUM は上位アプリケーションレイヤで認識されるフレームを単位としたときのフレームシーケンス番号、PACKET SEQ NUM はこのフレームを PDU に従って分割した場合の 1 フレーム内のパケット番号、さらにビットエラー検知のための CHECKSUM と遅延時間およびジッタ測定のための TIMESTAMP が含まれる。

3.1.3 メディア情報の特性

アプリケーションが保持するメディア情報には、ファイルフォーマットのバージョン番号・フレームの縦横幅・圧縮時の Q-factor・単位時間フレーム数・シーケンス番号・カラーマップ中のカラー数・音声のボリューム情報や音声トラック数などがある。ここでは送ろうとするメディアデータを、階層的符号化によりベースとなる基本情報とそれからの差分情報に分割する。これにともない、アプリケーションレベルで保持されるメディア情報のうち、フレームサイズやフレームレート・帯域必要量などの各階層毎に異なる値を取る階層固有の情報と、階層間で共通の情報から分離し、トランスポートレイヤで階層制御のキーとするために前者を以下のようなテーブルとして保持する。

QOS レベル	フレームサイズ	転送レート
Q1	16 KBytes	256 Kbps
Q2	8 KBytes	128 Kbps
Q3	8 KBytes	128 Kbps

3.2 フロー制御機能

次に、フロー調整機能で保持されたメディア情報の特性・ネットワーク性能・エンドポイントの要求条件の 3 点の評価方法と、その結果決定されるメディアフローの制御方式について述べる。

3.2.1 ジッタ処理

特に連続メディアを扱う場合、動画や音声再生時の遅延は、視聴者が知覚するサービス品質を決定する重要な要因とされている。このエンドツーエンドの遅延時間の変化量であるジッタ処理に関して、表示時の遅延を低く抑える方式 (I-policy) では、遅れたフレームは表示しないのでフレームギャップを引き起こ

す確率が高くなる。また、フレームギャップを避けレートは一定で遅延時間を最悪値に合わせる方式 (E-policy) では、輻輳終了後もキューは長く遅延時間も大きい状態が続く。そこで、I-policy を適用した上に、アプリケーションが指定するジッタの範囲内で、一時的な再生時の遅延を許容する方式を採る。この方式では、輻輳時には許容範囲内でジッタが生じるが、その分フレームギャップを低くできる。さらに、処理の起動時間は定期的なため、E-policy と異なりパケット毎に遅延対処を行っているため、輻輳終了後すぐに正常状態に戻ることが出来る。

3.2.2 パケットフィルタリング機能

次に、ネットワークレイヤでのフィルタ設定手続きと転送処理について述べる。

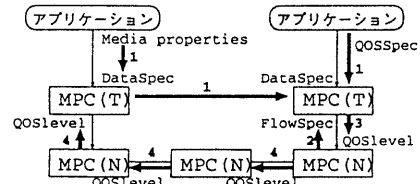


図 2: QOS レベルの設定

1. DataSpec, QOSSpec の保持

送信側のトランスポートエージェントは、アプリケーションから転送するマルチメディア情報の特性を DataSpec として受け取ると、それを受信側のトランスポートエージェントに伝達する。さらに、受信側のトランスポートエージェントは、受信側のアプリケーションから、メディアフローに対する要求条件を得てそれを QOSSpec として保持する。

2. FlowSpec の評価

トランスポートレイヤにおけるフローモニタリング機能により、パケットヘッダのフレーム番号・パケット番号・チェックサムなどの情報から、エンドツーエンドの FlowSpec の実測を行う。

(1) throughput の実測

(2) if(measured delay > worst_case delay)

パケットの無効化
packet_loss++

3. QOS レベルの決定

アプリケーションレベルの DataSpec, QOSSpec とネットワークレベルからの FlowSpec から、エージェントが受信する QOS レベルを選択する。また、一時的な負荷状態などにより FlowSpec が変化したら、再度それらを突き合わせ、QOS レベルを変更する。

(1) if(measured packet_loss rate > max packet_loss rate)

優先度の低い QOS type を選択
measured throughput 範囲内の QOS level の決定

(2) set_filter(SessionID, QOS type, QOS level)

4. QOS レベルの設定

トランスポートレイヤにおいて決定された QOS レベルを、ネットワークレイヤではパケット中継処理のために、ネットワークインタフェース毎のテーブルとして保持する。また、テーブルにはフィルタ設定者のフィールドが付加され、フローを逆流

して伝搬しながら各ルータに以下のフィルタを設定していく。このフィルタ設定者には、SRC, DEST, INTER などがある。また、Next はフィルタが伝搬される際のエージェントを示す。この Next フィールドは、同一インタフェース上の複数の次エージェントからのフィルタをマージする時に使用する。実

表 4: フィルタ

Session	NetIF	QOSlevel	Filter	Next
SID-1	IF-1	Q3	DEST	NEXT1
	IF-2	Q1	DEST	NEXT2
	IF-3	Q3	DEST	NEXT3, NEXT4

際にデータを転送する場合は、送信側のネットワークエージェントは、パケットの QOS フィールドとネットワークインタフェース毎に設定されている QOS レベルとを対応付けパケットの中継処理を判断する。また、受信側のネットワークエージェントは、グループアドレスに QOS レベルを加味してパケットの受信処理を判断する。例えば上記のフィルタを持つルータは、IF-1 からは Q3 レベル以下のパケットのみ受信し、IF-2 へは Q1 レベルのみ、IF-3 へは Q3 レベル以下全てのパケットを中継処理する。

4 MPC を適用したマルチキャストサーバ

4.1 RT-Mach の利用

本研究の基本ソフトウェアプラットフォームとして、米国カーネギーメロン大学で開発されている Real-Time Mach 3.0(以下、RT-Mach と略す。)を適用し、マルチキャストサーバをリアルタイムスレッドを用いて実装中である。全体構造は、トランスポートレイヤ部、ネットワークレイヤ部の二部で構成され、ネットワークレイヤ部はマルチキャスト IP にフィルタ機能を横付けする形とした。

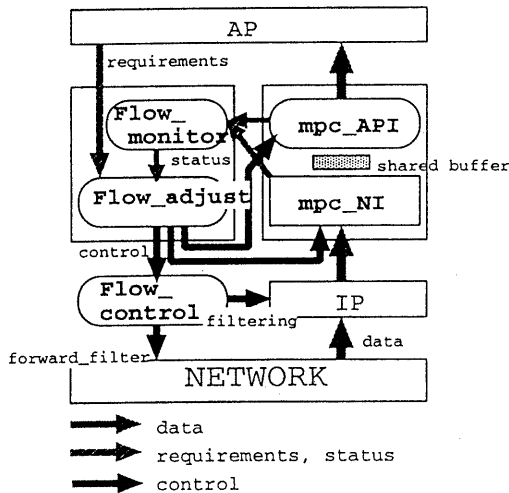


図 3: 全体構成

このシステムでは、トランスポートレイヤ部の AP とのインタフェースを図る部分 (mpc_API) を周期的スレッド、ネットワークレイヤ部との間 (mpc_NI) を非周期的スレッドとして実現した。mpc_API は、アプリケーションとバッファを共有し、あらかじめアプリケーションによって指定される QOS パ

ラメータに従い、周期的サーバの周期やデッドラインなどのマルチキャストサーバとしてのパラメータを指定する (表 5)。また mpc_NI では、ネットワークレイヤから受信したパケットを、MPC ヘッダのフレーム番号に従い mpc_API との共有バッファとして順次配置する。mpc_API はこの共有バッファを周期的にアクセスし、パケット番号に従ってフレームを組み立て、アプリケーションとの共有バッファに書き込む。

表 5: マルチキャストサーバのパラメータ

FlowSpec	サーバのパラメータ
delay	受信開始時間
jitter	デッドライン
frame_rate(max, avg)	周期
frame_size(max, avg) (throughput)	バッファ領域
error_rate	チェックサム
packet_loss_rate	受信パケット数

4.2 適用実験 1: ジッタ処理

適用実験として、Intel i-486 を搭載した IBM PC AT 互換機を 2 台使用し、マシン A からマシン B に対して、0.1 sec 毎に 500 x 8 Bytes のパケットを転送し、送信側のマシン A にある周期で負荷を与えた場合に発生するジッタやパケット受信処理量を測定した。

本システムでは、周期的スレッドを用いつつ、ネットワーク輻輳による転送パケットの遅延時間のジッタをある程度考慮するために、RT-Mach で提供されるデッドラインハンドラとシグナルを利用した。アプリケーション側の mpc_API は周期的に起動されるが、ネットワーク側の mpc_NI で予想されるパケットの受信処理が間に合わない場合、mpc_NI 側ではパケットの受信処理を続行し、mpc_API は周期スレッドのデッドラインをミスするか、予想される全てのパケットを受信または次のフレームのパケットが先に届き mpc_NI からのシグナルを受けるまで wait し、そこまで受信できたパケットを処理する。このときのデッドライン値とデータ受信のジッタおよびパケット受信処理量を以下に示す。

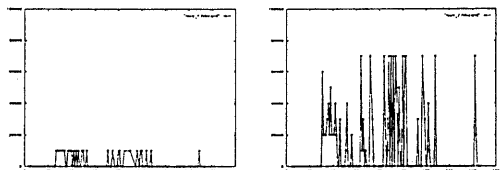


図 4: 周期的スレッドのデッドラインに 0.01sec/0.07sec を指定したときのジッタ

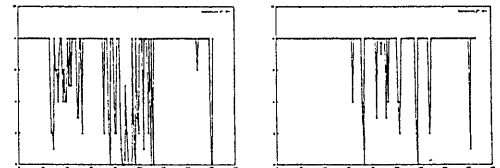


図 5: 周期的スレッドのデッドラインに 0.01sec/0.07sec を指定したときのパケット受信処理量

実験結果より、デッドライン値を 0.01 sec に設定した側は

パケットロス率が高いのに対し、0.07 secの方は受信処理ジッタは0.07sec以内で変動している代わりに、パケットロス率は0.01 secの場合に比べて小さい。しかし、程度の差はあれ周期的なパケット受信処理量は不安定で、アプリケーションレベルにおける画像の質の変動となって顕在化する。これより、ネットワークレベルでの転送性能制御が、受信パケットのジッタ処理だけでは不十分であると考えられる。

4.3 適用実験 2: パケットフィルタリング処理

次に、ネットワークレイヤでのパケットフィルタリング処理には、IPのTOS(Type Of Service)フィールドのうち、2 bitsを時間的解像度、2 bitsを空間的解像度として解釈する。このIPパケットの中継エージェントでは、あらかじめ決定されたQOSレベルとネットワークインタフェースの組を保持しており、このQOSレベルをMachで提供されるネットワークフィルタの値に設定する。この値を、フロー調整機能によるQOSレベルの変更に従って動的に変えることにより、受信パケットのフィルタリング処理を行うことができる。

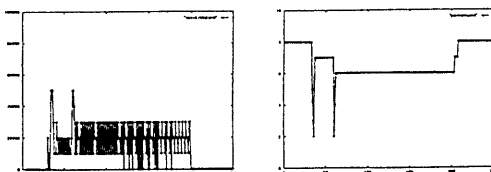


図 6: パケットフィルタリング機能を適用した場合のジッタとデータパケット受信処理量

デッドラインには0.05 secを指定し、適用実験1と同様の実験を行った結果、ジッタは0から0.05 secの間で変動しているが、パケット受信処理量の方は、適用実験1と異なり負荷を与えた後ある一定の値(6)で安定していることが示されている。

次に、マルチキャスト通信において複数のクライアントが異なるジッタ値(0.01と0.07 sec)を指定した場合の受信ジッタとデータパケット受信処理量を測定した結果を示す。

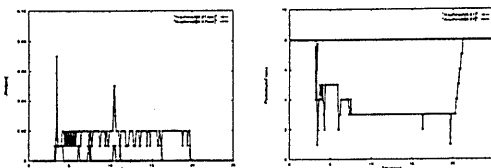


図 7: マルチキャスト通信において複数のクライアントが異なるジッタを指定した場合のジッタとデータパケット受信処理量

複数のクライアントが同一メディアでかつパケット到着時間のばらつきを同じように受けた場合、ジッタ許容範囲を大きく設定したクライアント(0.07 sec)ほど、QOSレベルは高い値(8)で安定する。このジッタ許容範囲は、ユーザレベルアプリケーションが個別に指定でき、ジッタの許容範囲を広げるほど動きのなめらかさが妨げられる一方、QOSレベルは高いままなので画像の解像度は維持できる。このように、複数のクライアントが共有するメディアに対して異なるジッタ値を指定したとしても、それがQOSレベルに変換され、クライアント毎のQOSレベルの差異は、中継エージェントで制御される。こ

れは、遅延時間に関しても同様のことが言える。例えば、マルチキャスト会議において、参加者にとっては早いレスポンスが求められるため、遅延時間は小さいほうが望ましい。これに対し、ただ会議を聴講している聴講者にとっては、遅延時間は多少大きくても構わない。さらに、会議を同時にファイルサーバ上に保存し、非同期的なアクセスへサービスしようとする場合には後者の傾向がより強くなり、遅延時間を犠牲にしても質の高いほうが望まれる。このとき中継では、クライアント毎の遅延時間の差異を反映したQOSレベルのフィルタリングを行うと同時に、QOSレベルに基づいて遅延制御も行い、QOSレベルの低いパケットよりも後から到着したQOSレベルの高いパケットを優先して中継し、遅延時間に関与しないアプリケーションのためのQOSレベルの低いパケットは、遅れて中継するよう制御することもできる。

5 まとめ

本稿では、QOS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。マルチキャスト通信にQOSレベルを導入することによって、複数のクライアントが同一メディアフローの転送サービスに対して個々に異なる性能パラメータを要求することが出来、その結果例えばグループへの参加形態に応じた相対的な要求条件を満足することが可能となる。さらに、ネットワーク輻輳などに対する転送制御では、受信処理が間に合わなくなったら単にQOSを下げるといった一元的な制御方式でなく、エンドノードの要求条件といった複数の要因を加味した多次元的な制御方式により、いたずらにQOSを低下させることが避けられると考える。

6 謝辞

本研究を行うにあたりご協力頂いた齋藤信男教授、萩野達也助教授、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。また、NTT情報通信研究所の和佐野哲男部長、大町雄一グループリーダー、グループメンバーの皆様へ感謝いたします。

参考文献

- [1] L. Delgrossi, C. Halstrick, D. Heilmann, R.G. Herrtwich, O. Krone, J. Sandvoss, and C. Vogt. Media Scaling for Audio-visual communication with the Heidelberg Transport System. *ACM Multimedia93*, 1993.
- [2] S. Kihara, Y. Onoe, S. Moriai, A. Mituzawa, A. Nambu, and H. Tokuda. An implementation of ST-II Protocol as a user-level server on RT-Mach. In *Abstracts of the 4th International Workshop on Network and Operating System Support for Digital Audio and Video*, 1993.
- [3] Y. Onoe. QOS-based Multicast Communication. In *Abstracts of the 2nd Workshop on Advanced Teleservices and High-Speed Communication Architectures*, 1994.
- [4] C. Partridge. A Proposed Flow Specification, 1992. RFC1363.
- [5] 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.
- [6] 西尾信彦, 追川修一, 緒方正陽, 尾上裕子, 河内谷清久仁, 堀野崎敦, 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.