

# 分散マルチメディアシステムにおける アプリケーション QoS 管理手法の提案とその実装

加藤 由花<sup>†</sup> 佐々木 徹<sup>†</sup> 箱崎 勝也<sup>†</sup>

コンピュータネットワーク上に多くの分散マルチメディアシステムが導入されるにつれ、利用者に対して提供されるサービスの品質 (Quality of Service: QoS) に対する注目が高まってきている。我々はこのサービス品質をアプリケーション QoS と定義し、システム環境に対して適応的なアプリケーション QoS 制御を実現する、アプリケーション QoS 管理システム (application QoS Management System: QMS) の研究を進めている。QMS はアプリケーション QoS を制御対象とするが、計算機やネットワーク資源が動的、静的に変化する実システム環境に適應するため、複雑なモデル化をとまなわないフィードバック型の簡易な制御方法を採用する。そのため、ネットワークレベルの制御とアプリケーションレベルの制御を組み合わせることによって、システム全体としてのアプリケーション QoS の向上を図る。ここで QoS の管理はシステム管理者が QMS に設定する QoS 管理ポリシーに基いて行われるが、複数のアプリケーション間で QoS 要求に対する整合性を保ちながら制御の優先順位を決定することは困難である。本稿では、この設定作業を支援する QoS 管理ポリシー設定手法を提案し、DV 転送システムを利用した実験システム上への実装結果を紹介する。さらに、この実験システムを利用して QMS の性能評価実験を行ったので、その結果を報告する。

## Application QoS Management for Distributed Multimedia Systems and Its Implementation

YUKA KATO,<sup>†</sup> TORU SASAKI<sup>†</sup> and KATSUYA HAKOZAKI<sup>†</sup>

As a large number of distributed multimedia systems are deployed on computer networks, Quality of Service (QoS) for an application becomes more important. This paper defines it as application QoS, and proposes the application QoS Management System (QMS). Its control targets are application QoS and it controls the QoS according to the system environment. At that time, the system adopts simple control methods without complicated modeling in order to adapt QMS to practical system environment. This is because system resources always change dynamically and statically on the practical environment. In addition, QMS aims to manage application QoS in the whole system; therefore, it does the QoS control by combining network level control method with application level control method. This QoS management is done according to QoS management policies which the system administrator sets to QMS. However, it is difficult to decide the control priority while negotiating QoS among many applications. This paper proposes the method setting the QoS management policy, and describes the implementation results on the prototype system using digital video transport system. Moreover, we show the experimental results of QMS performance evaluation by using this prototype system.

### 1. はじめに

近年、基幹業務における通信を IP (Internet Protocol) 系に移行する利用者が急増し、IP 技術は実質的に通信インフラとしての役割を果たすようになってきた。それにともない、インターネット上で様々な種類のアプリケーションサービスが提供されるようになってお

り、この状態がさらに進むと Everything on IP<sup>1)</sup> という形態に発展すると考えられる。これは音声、映像をはじめすべてのメディアをインターネットでサポートしようという考え方であり、現在でも技術的には実現可能な形態である。しかし、本当の意味で Everything on IP を実現するには、IP ネットワーク上でサービス品質 (Quality of Service: QoS) を制御する技術の確立が不可欠である。

ここで制御対象となる QoS は、アプリケーションの利用者が直接体感するサービスの品質で、たとえば

<sup>†</sup> 電気通信大学大学院情報システム学研究科  
Graduate School of Information Systems, University of  
Electro-Communications

映像配信サービスにおける映像品質, FTP サービスにおけるファイル転送時間などがあげられる。本稿では, このアプリケーションレベルの QoS を IP パケットの損失や遅延のようなネットワークレベルの QoS と区別して扱い, 前者をアプリケーション QoS, 後者をネットワーク QoS と定義する。ネットワーク QoS がサービスの内容に依存せず IP パケットレベルで定義可能なのに対し, アプリケーション QoS は利用者ごと, システム環境ごとに様々に定義される。そのため, サービスの提供者はアプリケーションごとに利用者の要求する品質のレベルをアプリケーション QoS として定義し, 利用者の満足度が高くなるようにこの QoS を制御しようとする。ここで問題となるのは, インターネット上では IP 自体に通信品質を保証する機能がないため, この制御の実現が困難であるということである。

このような背景から, IP ネットワーク上で QoS を保証する技術として様々な研究開発が行われてきた。まず, IP 層における QoS 制御手法として, IETF において IntServ<sup>2)</sup> や DiffServ<sup>3)</sup>, MPLS<sup>4)</sup> などの標準化が行われてきた。また, ネットワーク利用者が動的にネットワークを制御する技術として Active Network<sup>5)</sup> の研究が進められてきた。しかし, これらの技術の制御対象はネットワーク QoS であるため, アプリケーションの利用者が要求するアプリケーション QoS を制御対象とする場合, このアプリケーション QoS をネットワーク QoS に変換する必要がある。さらに, 各利用者や各アプリケーションはそれぞれに要求する QoS のレベルが異なるため, 複数のアプリケーションが共存するシステムでは, これらのアプリケーション間でシステム資源 (CPU 時間やネットワーク帯域など) の割当てに対する交渉が必要である。しかし, これらの技術ではこの交渉を実現する手段が提供されていない (以降, この交渉作業を QoS 交渉と呼ぶ)。

これらの問題を解決するため, QoS 交渉を実現しながらアプリケーション QoS をネットワーク QoS に割り当てる手法の研究も行われている<sup>6)~9)</sup>。これらの研究では, 利用者の QoS に対する要求やシステム環境 (ネットワーク上のトラフィック量など) を入力とし, 各アプリケーションへのシステム資源の割当て結果が出力となる最適化問題が解かれている。そのため様々なモデル化が行われており, たとえば, 市場モデルを利用したもの<sup>6)</sup>, Broker を利用したもの<sup>7)</sup>, マルチエージェントを利用したものがある<sup>8),9)</sup>。しかし現在のインターネット環境は, 新しいアプリケーションが頻りに追加, 削除され, トラフィック需要を予測

することは困難になっている。したがって, 膨大な数のシステム資源と不確定な需要予測結果を入力に, 複雑なモデルを用いて最適解を求める方法を適用するのは困難である。そのためこれらの手法では, 実システムへの適用が困難であるという問題と, 新たなアプリケーションの追加が困難であるという問題が発生する。

このような背景から我々は, 複雑なモデル化をとまわらないフィードバック型の簡易な制御方法を採用する, アプリケーション QoS 管理システム (application QoS Management System: QMS) の研究を進めてきた<sup>10)</sup>。QMS はアプリケーション QoS をネットワーク QoS に割り当てることをせずに, システム環境に対して適応的な QoS 制御を実現する。ここで QMS は, ある IP ネットワーク, たとえば大学の研究室 LAN や企業における部署内 LAN などを管理する人が, その管理対象のシステムにおいてアプリケーション QoS を管理することを目的に設計される。管理者の業務としては, 管理対象のシステム全体としてのアプリケーション QoS の管理指針を決定すること (ここではこれをシステム運用ポリシーと呼ぶ), システム運用ポリシーを基に各アプリケーションごとに QoS の管理指針を決定すること (QoS 管理ポリシーと呼ぶ), これらのポリシーに従って新たなアプリケーションを管理対象のシステムに導入/削除すること, などを想定している。このような管理者のことを本稿ではアプリケーション QoS 管理者 (AP 管理者) と呼ぶ。また管理対象のシステムを構成する要素としては, アプリケーションサービスを提供する複数のサーバ計算機, そのサービスを受ける複数のクライアント計算機, システムを集中管理する 1 台の管理用計算機を想定し, これらの計算機が IP ネットワークによって互いに接続されている形態を対象とする。1 台の管理用計算機で管理可能な範囲であれば, 単一のネットワークセグメントであっても複数のネットワークセグメントであってもかまわない。ただし同一のシステム運用ポリシーに基づいて管理する必要があるため, 同一の管理グループによって管理されているシステムを対象とする。また, QMS はアプリケーション QoS を制御対象とした利用者指向のシステムなので, WWW の閲覧のようにインターネット上のアプリケーションを管理対象のシステムから利用する形態は対象とするが, 管理対象のシステム上のサーバを外部から利用する形態は対象から除く。

これまで我々は, アプリケーション QoS の適応的な制御を目的に, 3 種類のモジュールから成る QMS のシステム構成を提案し, シミュレーション実験によっ

て QoS 制御の有効性を確認してきた。しかし QMS の開発を進めるにつれ、AP 管理者によるポリシーの設定を支援する機能がないため、管理ポリシーを QMS に設定しそれを実際の制御に反映させる作業の負荷が大きくなってきた。また、既存のアプリケーションにどのように QMS を適用していくかという実装に対する課題も残った。

今回、この QoS 管理ポリシー設定機能を QMS に組み込み、容易に制御を実現する仕組みを構築した。さらに Digital Video (DV) 転送システムを利用した実験システムを構築し、これに QMS を実装することによって QoS 管理の実現方法を明らかにした。本稿ではこれらの結果を報告する。以下、2 章で QMS のシステム構成を述べた後、3 章で QoS 管理ポリシーの設定方法を提案する。4 章では実験システムの構成について説明し、5 章で実験システムを利用した性能評価実験の結果を示す。最後に 6 章で本稿をまとめる。

## 2. アプリケーション QoS 管理システム

### 2.1 システムの概要

QMS の設計目標は以下の 3 点である。

- アプリケーション QoS を制御対象とする：QMS では、たとえばネットワーク上でパケットの損失が発生しても、利用者が満足する品質のサービスが提供されていれば問題ないと考える。
- 実システムへの適用を目指し、複雑なモデル化をとまなわないフィードバック型の簡易な制御を行う：QMS では、アプリケーション QoS の劣化を検出したとき、これをネットワーク QoS に割り当てず単純なある決まった制御を実施する。たとえばビデオの映像品質の劣化を検出した場合、ある決まった帯域ずつビデオの送出帯域を減少させていく制御などを行う。
- 複数アプリケーション間で QoS 交渉を実現する：QMS では、システム全体としてすべてのアプリケーションが QoS 管理ポリシーに従って運用されることを目指す。そのためすべてのアプリケーションに対して制御の優先順位を決定し、その順位に従って制御を行う。

我々はこれらの設計目標を実現するため、QMS を、アプリケーション QoS を測定、監視する *Notificator* モジュール、QoS 管理ポリシーに従って制御方法を決定する *Manager* モジュール、実際の制御を行う *Controller* モジュールの 3 種類のモジュールで構成した。

### 2.2 システムの構成

QMS のシステム構成を図 1 に示す。QMS の 3 種類のモジュールはそれぞれいくつかのオブジェクト

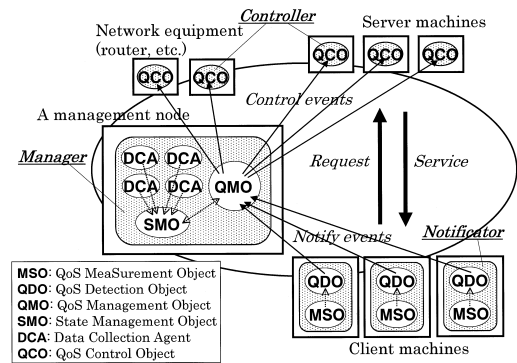


図 1 システムの構成  
Fig. 1 System architecture.

から成り、オブジェクト間の通信によってアプリケーション QoS の制御を行う。このとき実装を容易にするため、CORBA (Common Object Request Broker Architecture)<sup>1)</sup>などの分散オブジェクト環境を利用する。以下に各モジュールの機能と構成オブジェクトを記す。

- *Notificator*モジュール：クライアント計算機上に実装し、アプリケーション QoS の劣化を検出する。QoS を測定する MSO (QoS Measurement Object)、劣化を検出する QDO (QoS Detection Object) から成る。
- *Manager*モジュール：管理用計算機上に実装する。全アプリケーションの QoS 管理ポリシーを保持し、制御方法を決定する QMO (QoS Management Object)、制御方法の決定に必要なデータを収集する DCA (Data Collection Agent)、そのデータを蓄積する SMO (State Management Object) から成る。
- *Controller*モジュール：ネットワーク装置 (ルータなど) やサーバ計算機上に実装する。制御を実行する QCO (QoS Control Object) から成る。

### 2.3 オブジェクト間の通信

QMS におけるオブジェクト間の通信の手順を図 2 に示す。まず MSO が QoS を周期的に測定し、それを QDO に送信する。QDO は AP 管理者によってあらかじめ設定されたしきい値と受信したデータを比較し、QoS の劣化を検出する。QDO は劣化の検出結果を QMO に通知する。一方、DCA は制御方法を決定するために必要なデータを各計算機やネットワーク装置から周期的に収集し、SMO に蓄積する。QDO から劣化の通知を受信した QMO は、AP 管理者が設定した QoS 管理ポリシー、SMO に蓄積された各種データを基に制御方法を決定し、QCO に実際の制御を依頼する。

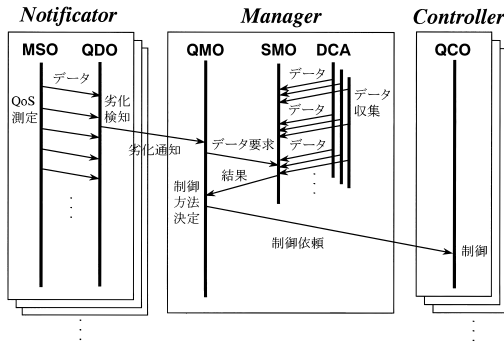


図2 オブジェクト間の通信  
Fig.2 Procedure in QMS.

### 3. QMSにおけるQoS制御

本章では、AP管理者がQoS管理ポリシーをQMSに設定する方法を提案する。

#### 3.1 基本方針

AP管理者は、管理対象のシステムに実装される全アプリケーションのQoS管理ポリシーを決定し、これをQMSに設定する。この設定は以下の2段階で行う。

- ネットワークレベルの制御の優先順位を決定する。
  - アプリケーションレベルの制御方法を決定する。
- ここでネットワークレベルの制御とは、IP層においてネットワークQoSを保証するために行われる制御で、たとえばルータのバッファあふれが発生したとき優先順位に従ってパケットを廃棄する制御などがある。これらの制御はネットワーク装置で実施される。QMSにおけるQoS制御はアプリケーションQoSの制御であるが、複数アプリケーション間でQoS交渉を実現するため、各アプリケーションのネットワークレベルの制御の優先順位を決定し、アプリケーションQoSの相対的な優先度を決める。しかし、それだけではアプリケーションQoSは保証されないので、アプリケーションレベルの制御によってアプリケーションQoSの維持を目指す。ここでアプリケーションレベルの制御とは、映像の送出帯域の変更やサーバの負荷分散など、各アプリケーションに依存した制御のことである。これらの制御は、アプリケーションの開発者によってあらかじめアプリケーションに組み込まれている場合（映像の送出帯域を変更する機能など）と、AP管理者によって運用時に設定される場合（サーバの負荷分散など）がある。以下、ネットワークレベルとアプリケーションレベルの制御方法、各制御におけるQoS管理ポリシー設定方法を述べる。

### 3.2 ネットワークレベルの制御

#### (1) 制御方法

ネットワークレベルの制御では、インターネットにおけるエンドツーエンドサービス品質として規定されるスループット、IPデータグラムの遅延、遅延ゆらぎ、損失の4種類のデータ<sup>12)</sup>を制御対象とする。それぞれ以下の制御を行う。

- スループットを確保するための制御：エンドツーエンドの転送帯域を保証する。たとえば、RSVP<sup>13)</sup>などを利用して帯域の予約を行う。
- 遅延を防ぐための制御：ネットワーク装置のキュー管理を行う。たとえば、CBQ(Class Based Queuing)などを利用してパケット送出の優先度を設定する。
- 遅延ゆらぎを防ぐための制御：トークンバケット方式などを利用したトラフィックシェーピングを行う。
- 損失を防ぐための制御：ネットワーク装置においてパケット廃棄の優先度を設定する。たとえば、Diff-Servなどを利用してパケットのクラス分けを行う。

QMSは管理対象のシステムに実装される全アプリケーションに対し、これら4種類の制御の優先順位を自動で決定する。このとき、アプリケーションをトラフィック特性を利用して分類し、分類ごとに制御の優先順位を決定する。これがネットワークレベルのQoS管理ポリシーになる。ただしシステム運用ポリシーによって優先順位の変更が必要な場合もあるので、AP管理者はQMSが決定した順位を変更することができる。

#### (2) QoS管理ポリシーの設定方法

ネットワークレベルの制御の優先順位は、アプリケーションのトラフィック特性による分類に従って決定されるが、このトラフィック特性は、管理対象のシステムに新しいアプリケーションを追加するとき、AP管理者がQMSに設定する。アプリケーションの分類方法、および優先順位を決定するための規則を以下に述べる。

- アプリケーションの分類：マルチメディアサービスにおけるトラフィック特性は、トラフィックの種類と応答時間に対する要求によって分類される<sup>14)</sup>。トラフィックの種類を表1に、応答時間に対する要求を表2に示す。

- QoS管理ポリシーを決定する規則：QMSでは、表1、表2の特性によって分類したアプリケーションに対し、表3に従って4種類の制御の優先順位を決定する。つまりQMOの内部に表3を保持し、AP管理者が指定したアプリケーションのトラフィック特性に従って、各制御の優先順位を自動で決定する。表1、

表 1 トラフィックの種類による分類  
Table 1 Application classification (traffic types).

トラフィックの種類	記号	特徴	AP 例
ブロック型	B	大きいサイズの情報 が低い頻度で発生	静止画
ストリーム型	S	中サイズの情報が 周期的に発生	映像
トランザクション型	T	小サイズの情報が 不規則に発生	コマンド

表 2 応答時間に対する要求による分類  
Table 2 Application classification (response time).

応答時間	記号	特徴	AP 例
リアルタイム	RT	数 100ms 以下が必要	電話
準リアルタイム	QRT	数秒以下が望ましい	WWW
非リアルタイム	NRT	100 秒以上も許容	e-mail

表 3 QoS 管理ポリシーを決定する規則  
Table 3 The rule setting QoS management policies.

分類	スループットを 確保する制御	遅延を 防ぐ制御	ゆらぎを 防ぐ制御	損失を 防ぐ制御
B-RT	2	1	2	2
S-RT	1	1	1	1
T-RT	-	1	-	5
B-QRT	-	2	-	3
T-QRT	-	2	-	5
B-NRT	-	3	-	4
T-NRT	-	3	-	5

表 2 の分類を組み合わせると 9 種類の分類があるが、S-NRT 型と S-QRT 型に該当するアプリケーションは存在しないので、表 3 に示す 7 種類の分類に対して優先順位を決定した。表 3 の各要素は、数字が小さいほど制御の優先順位が高いことを、- の項目は該当する制御を実施しないことを示す。たとえばスループットを確保する制御では、まず S-RT 型の帯域予約を行い、次に B-RT 型の予約を行う。その他の分類では制御を行わない。

QMS は表 3 に従って制御の優先順位を決定し、QoS 管理ポリシーを決定する (AP 管理者は必要なら順位の変更を行う)。たとえば、QoS 管理ポリシーが { アプリケーション名, スループットを確保する制御の順位, 遅延を防ぐ制御の順位, ゆらぎを防ぐ制御の順位, 損失を防ぐ制御の順位 } と表されるとき、映像配信 (S-RT 型), FTP (B-QRT 型), 電子メール (T-NRT 型) の 3 種類のアプリケーションが実装されたシステムの QoS 管理ポリシーは、{ 映像配信, 1, 1, 1, 1 }, { FTP, -, 2, -, 2 }, { 電子メール, -, 3, -, 3 } となる。

### (3) ネットワーク装置への制御方法の設定

QoS 管理ポリシーは QMS が保持する QoS の管理指針だが、実際の制御はネットワーク装置で実施される

ため、制御の実装方法に依存しない QoS 管理ポリシーを、ネットワーク装置で実現可能な実装に依存した制御方法に変換する必要がある。このとき、ネットワーク装置に制御方法を設定するために必要なパラメータを QMO はあらかじめ保持しておく必要がある。そのため AP 管理者は、管理対象のシステムに新規アプリケーションを追加する場合、表 1、表 2 に示すトラフィック特性と合わせて、これらのパラメータを QMO に設定する。設定が必要なパラメータを以下に示す。

- アプリケーション名
- サーバ計算機とクライアント計算機の IP アドレス
- ポート番号とポート種別 (TCP/UDP/Other)
- サーバからの送出帯域 (平均値, 最大値)

また QMO は、管理対象のシステムに関する以下のパラメータも保持しておく必要がある。これは、システムに QMS を導入する時点で QMO に設定する。

- 管理対象のシステムのネットワークポロジ
- ネットワーク装置の IP アドレス

QMS はこれらのパラメータを利用して、ネットワーク装置に制御方法を設定する。ポート番号とポート種別によってアプリケーションの種類を識別し、ネットワークポロジから設定対象のネットワーク装置を決定する。スループットを確保する制御としては、サーバからの平均送出帯域を保証帯域と見なし、帯域予約を行う。遅延を防ぐ制御と損失を防ぐ制御としては、パケット送出と廃棄の優先順位を設定する。遅延ゆらぎを防ぐ制御としては、サーバからの最大送出帯域をバースト帯域と見なし、シェーピングの設定を行う。

### 3.3 アプリケーションレベルの制御

#### (1) 制御方法

アプリケーションレベルの制御対象は、システム運用中に準リアルタイムで変更可能な要素とする。これは QMS における QoS 制御が、フィードバック型の簡易な制御であることによる。以下の 4 種類の制御を行う。

- サービスレベルの変更: アプリケーションが提供するサービスの質を変更する制御で、たとえば映像配信サービスの送出帯域を変更する制御などがある。フィードバック型の制御を行うため、劣化を検出したらある単位ずつサービスの質を低下させ、回復したらある単位ずつサービスの質を回復させていく制御などを行う。
- サーバの負荷分散: サーバ計算機の負荷分散を行う制御で、低負荷のサーバを選ぶ制御などを行う。
- エラー訂正: アプリケーションが元々持つ機能を利用する。冗長パケットによるエラー検出などを行う。

• 提供サービスの変更：アプリケーションが提供するサービスの内容を変更する制御で、たとえばテレビ会議サービスの映像を静止画に変更する制御などがある。

アプリケーションによっては4種類すべてを実装できるわけではないが、アプリケーションごとに実現可能な制御を実装する。AP管理者はQMSに実装された制御の中から、システム運用ポリシーに従って実行する制御を選択する。ここで選択された制御の実現方法が、アプリケーションレベルのQoS管理ポリシーになる。

#### (2) QoS管理ポリシーの設定方法

AP管理者は、アプリケーションレベルの制御方法を、QMSのオブジェクトとして実装する。これらのオブジェクト間の通信によって制御が実現するが、この通信手順がアプリケーションレベルのQoS管理ポリシーであり、これがQMOに保持される。以下に管理ポリシーの設定手順を述べる：(i) AP管理者はアプリケーションQoSを決定し、MSO、QDOとしてオブジェクト化する；(ii) AP管理者は、実装可能な制御をQCOとしてオブジェクト化する；(iii) AP管理者は、オブジェクト間の通信手順をQMOに実装する；(iv) QMSは、MSO、QDO、QCOを各計算機に実装する。

### 4. DV転送システムへの実装

QMSの実装検証を行うため、研究室に Digital Video (DV) 転送を行う実験システムを構築し、その上にQMSを実装した。本章ではその結果を示す。

#### 4.1 システム構成

実験システムの構成を図3に示す。本システムは、2台のレイヤ3スイッチで接続された2つのネットワークで構成され、2台のクライアント計算機、2台のサー

バ計算機、1台の管理用計算機が図3のように接続されている。ネットワークはすべて100Mbpsのイーサネットである。各計算機の仕様を表4に示す。DV転送システムとしては、WIDEプロジェクトで開発されたDVTS (Digital Video Transport System)<sup>15)</sup>を利用した。DVTSは市販のデジタルビデオカメラとIEEE1394ボードを利用することによって、安価で容易なビデオ転送システムの構築を可能にする。QMSの各オブジェクトはJava言語によって実装し、各モジュール間の通信にはJava RMIを利用した。

今回の実験では管理対象のシステムに、DV転送サービス、WWWサービス(インターネット上のWebページをクライアント2で閲覧する)、FTPサービス(サーバ1のファイルをクライアント2で受信する)の3種類のアプリケーションを実装した。アプリケーションQoSはそれぞれ、利用者が体感するDVの映像品質、Webページの表示時間、FTPサービスでファイルの取得を要求してから終了するまでの時間とした。

#### 4.2 Notificatorモジュールの実装

実験システムではNotificatorモジュールとして、DV転送サービスにおける映像品質の劣化を検出する機能のみを実装した。WWWサービスとFTPサービスではネットワークレベルの制御のみを行う。映像品質の評価は、システムの利用者がクライアント計算機上の評価用ウィンドウ(図4)を利用して行う。利用者はウィンドウ上のスライドバーと選択ボタンを利用し、好みのフレームレートとストリームの種類を選択する。

表4 計算機の仕様  
Table 4 Machine specifications.

計算機名	CPU/メモリ容量	OS
サーバ1, 2	PentiumIII 1G/256M	FreeBSD4.0
クライアント1	PentiumII 400M/256M	FreeBSD4.0
クライアント2	PentiumIII 1G/256M	FreeBSD4.0
管理用計算機	PentiumIII 750M/256M	Windows2000

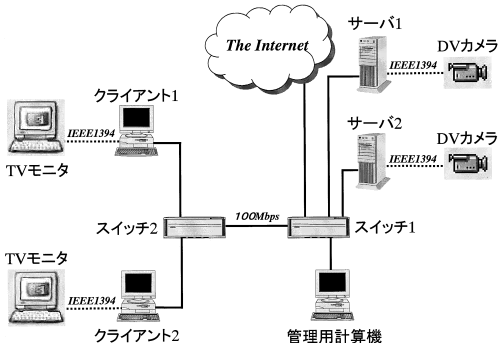


図3 実験システムの構成

Fig. 3 System architecture of the experiment system.

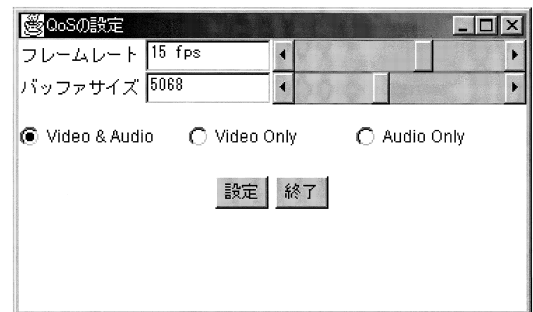


図4 評価用ウィンドウ

Fig. 4 The window for evaluation.

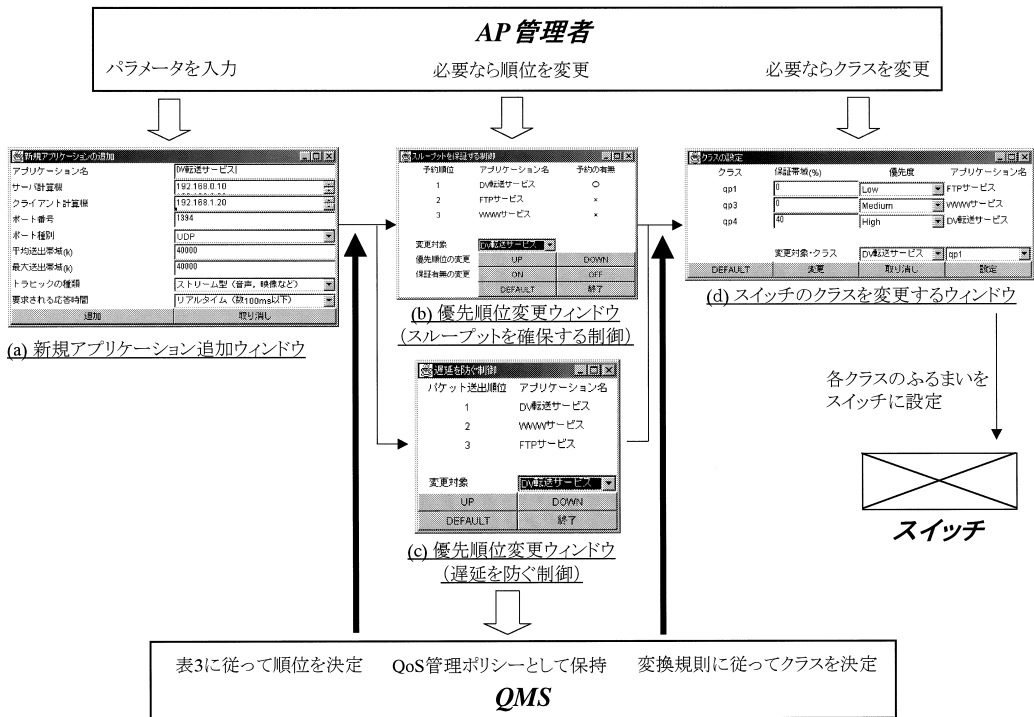


表3に従って順位を決定    QoS管理ポリシーとして保持    変換規則に従ってクラスを決定

**QMS**

**スイッチ**

図 5 ネットワークレベルの制御方法をスイッチに設定する処理の流れ  
 Fig. 5 The procedure setting network level control methods to the switches.

この場合 MSO の機能は利用者自身が持つことになり、ウィンドウを実現するオブジェクトが QDO になる。

### 4.3 Controllerモジュールの実装

WWW サービスと FTP サービスではネットワークレベルの制御のみを行うため、Controllerモジュールとして DV 転送サービスの QoS を制御する機能のみを実装した。ここでは DVTS 自身が持つフレームレートを変更する機能とストリームの種類を変更する機能を利用する。具体的には、サービスレベルの変更(フレームレートの変更)、サーバの負荷分散(2台のDVサーバ間で視聴開始時に CPU 使用率の低いサーバを選択する)、提供サービスの変更(ストリームの種類の変更)の3種類の制御を行うこととした。Notifierモジュールで直接フレームレートとストリームの種類を指定できるので、DVサーバにコマンドを送るオブジェクトとして、サーバ計算機上に QCO を実装した。

### 4.4 Managerモジュールの実装

ネットワークレベルの制御の優先順位とアプリケーションレベルの制御方法を決定する機能を QMO に、DVサーバの CPU 使用率を収集する機能を DCA に、収集データを保持する機能を SMO に実装した。

#### (1) ネットワークレベルの制御

AP 管理者は、管理用計算機上のウィンドウを利用してアプリケーションごとの QoS 管理ポリシーを決定し、ポリシーに基づく制御内容をネットワーク装置(スイッチ 1, 2)に設定する。ネットワークレベルの制御方法をスイッチに設定する処理の流れを図 5 に示す。

管理対象のシステムに新しいアプリケーションを追加する場合、新規アプリケーション追加ウィンドウ(図 5(a))において、アプリケーション名、サーバ計算機とクライアント計算機の IP アドレス、ポート番号と種別、サーバからの送出帯域(平均値、最大値)、トラフィックの種類、要求される応答時間を入力する。すると、QMO が表 3 に従ってネットワークレベルの制御の優先順位を決定する。決定された順位は 4 種類の優先順位変更ウィンドウ(たとえば図 5(b)や図 5(c))で確認でき、AP 管理者は必要なら順位の変更を行う。

QMS は決定した優先順位を QoS 管理ポリシーとして保持するとともに、実際の制御内容をスイッチに設定する。実験システムではスイッチの持つ機能を利用して制御を実現するため、優先順位をスイッチで実現可能な制御内容に変換する必要がある。今回利用したスイッチは、アプリケーションを最大 4 つのクラス(qp1 ~ qp4)に分類することが可能で、クラスごとに最低保証帯域、パケット送出の優先度(Low, Normal, High)を設定することが可能で、クラスごとに

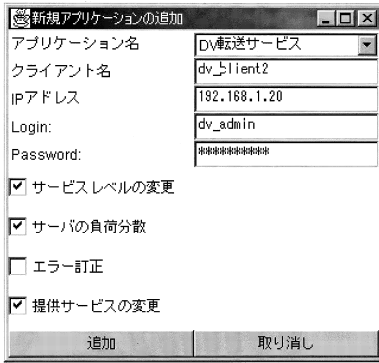


図 6 新規アプリケーション追加ウィンドウ (AP レベル)

Fig. 6 The window in adding a new application.

Medium, High の 4 レベル) を設定する。そのため QMS が決定した優先順位を以下の手順に従ってスイッチのクラスに変換し、その結果をスイッチに設定することとした (スイッチの機能の制約から、遅延ゆらぎを防ぐ制御と損失を防ぐ制御は実装しない): (i) クラス qp1 は帯域保証を行わずパケット送出の優先度が Low のデフォルトクラスとする; (ii) スループットを確保する制御の有無を確認し、制御ありのアプリケーションを優先順位の高い順に qp4 ~ qp2 に割り当てる。このときの保証帯域はアプリケーションの平均送出帯域とする; (iii) 各アプリケーションのパケット送出の優先度は、遅延を防ぐ制御の優先順位の順に High, Medium, Normal とし、それ以下の順位のアプリケーションではすべて Low とする (ii) で割り当てたアプリケーションのクラスにこの送出の優先度を設定する。この時点で割り当てられていないクラスが残っていたら、パケット送出優先度が高い順に残りのクラスを割り当てる。

この手順はあらかじめ QMO に設定しておく。そのため AP 管理者は各制御の優先順位を決定するだけで、容易にスイッチに制御方法を設定することができる。優先順位をスイッチのクラスに変換した結果はクラスを変更するウィンドウ (図 5 (d)) で確認可能であり、AP 管理者は必要に応じてその内容を変更できる。

## (2) アプリケーションレベルの制御

AP 管理者は、管理用計算機上の新規アプリケーション追加ウィンドウ (図 6) を利用して、アプリケーションレベルの制御を行うオブジェクトを QMS に実装する。まず AP 管理者はアプリケーションごとに QDO と QCO を実装し、管理計算機上の決められたディレクトリにこれらのオブジェクトを置く。その後、図 6 の追加ボタンを押すと、これらのオブジェクトがクライアント計算機、およびサーバ計算機に送信される。

今回、QDO が DV 転送サービスのフレームレートとストリームの種類を直接設定できるので、その要求を受けた QMO が QCO に制御を依頼することをアプリケーションレベルの QoS 管理ポリシーとした。実験システムにおける QoS 制御方法を以下に示す。

- (i) DCA がデータの収集を開始する。
- (ii) 利用者が映像の視聴開始を QMS に要求する。
- (iii) QMO が DV サーバを決定し映像送出を開始する。
- (iv) 利用者は要求する QoS を QMS に通知する。
- (v) QMO は利用者からの要求を QCO に通知する。
- (vi) QCO は DVTS のフレームレート、またはストリームの種類の変更を行う。

## 5. 性能評価

実験システムを利用して QMS の性能評価を行った。運用ポリシーに従った制御が可能になることを示すための実験 1、アプリケーション QoS の制御が可能になることを示すための実験 2 の 2 種類の実験を行った。

### 5.1 評価方法

#### (1) 実験 1

以下の 3 種類のシステム運用ポリシーを設定し、様々なポリシーを満足する運用が可能になることを示す。  
運用ポリシー 1: DV 転送サービスを優先し、DV の映像品質の維持を目指す。

運用ポリシー 2: FTP サービスを優先し、ファイル転送の時間が短くなることを目指す。

運用ポリシー 3: WWW サービスを優先し、Web ページの表示時間が短くなることを目指す。

各運用ポリシーに対し、制御を行わなかった場合、ネットワークレベルの制御のみ行った場合、ネットワークレベルとアプリケーションレベルの制御を組み合わせで行った場合 (QMS) の 3 種類の実験を行った。ネットワークレベルの制御の優先順位は、QMS が決定した順位 (図 5 (b), 図 5 (c)) を運用ポリシーに従って変更し、それを QMS がスイッチのクラスに変換した。各運用ポリシーに対して設定されたスイッチのクラスを表 5 に示す。またアプリケーションレベルの制御は全ポリシー共通で、図 4 を利用し、品質劣化を感じなくなるまでフレームレートを下げる制御を行った。

#### (2) 実験 2

QMS によってアプリケーション QoS が維持されることを示すため、ネットワークふくそう時にアプリケーションレベルの制御を行った場合と行わなかった場合の映像品質を比較した。アプリケーションレベルの制



表 5 各運用ポリシーにおけるスイッチのクラス  
Table 5 Switch classes for each policy.

ポリシー名	クラス	保証帯域	優先度	AP 名
ポリシー 1	qp1	0%	Low	WWW
	qp3	0%	Normal	FTP
	qp4	50%	Normal	DV 転送
ポリシー 2	qp1	0%	Low	WWW
	qp3	0%	Normal	DV 転送
	qp4	80%	High	FTP
ポリシー 3	qp1	0%	Low	FTP
	qp3	0%	Normal	DV 転送
	qp4	0%	High	WWW

表 6 実験結果

Table 6 Experimental results.

ポリシー	制御方法	DV 転送 (損失率)	FTP (完了時間)	WWW (表示時間)
1	制御なし	0%	580 s.	2.6 s.
	NW のみ	0%	908 s.	3.2 s.
	QMS	0%	886 s.	3.2 s.
2	制御なし	0%	580 s.	2.6 s.
	NW のみ	78%	457 s.	5.3 s.
	QMS	34%	461 s.	4.9 s.
3	制御なし	0%	580 s.	2.6 s.
	NW のみ	0%	582 s.	2.5 s.
	QMS	0%	496 s.	3.4 s.

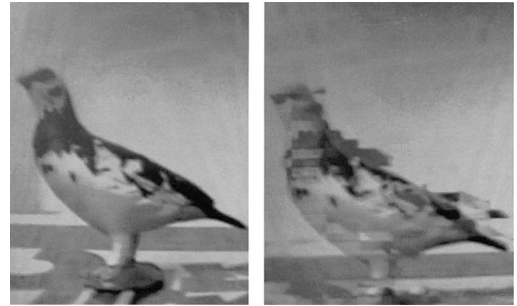
御としては実験 1 と同じ制御を行い、ふくそう状態を模擬するためスイッチ間を 10 Mbps で接続した。

## 5.2 評価結果

### (1) 実験 1

それぞれのシステム運用ポリシーに対し、DV 転送サービスにおける損失率(クライアントでの受信帯域/サーバの送出帯域)、FTP サービスの完了時間、WWW サービスの表示時間を測定した。DV 転送サービスにおける映像品質を定量的に評価するため、映像品質と相関の高いパケットの損失率を測定することとした。FTP サービスでは約 4.2 G のファイルを転送し、WWW サービスでは決まったページを再読み込みする時間を測定した。結果を表 6 に示す。DV 転送サービスでは損失率が低いほど、FTP サービスでは完了時間が短いほど、WWW サービスでは表示時間が短いほど QoS が高いと見なせる。ここで、制御を行わなかった場合の DVTS の映像送出帯域は約 33 Mbps であった。

まず、DV 転送サービスの映像品質を優先した運用ポリシー 1 では、ネットワークに余裕があるため、DV 転送における損失は発生していない。そのため QMS での制御の有無にかかわらず、運用ポリシーを満足している。次に、FTP サービスの完了時間を優先した運用ポリシー 2 では、制御なしの場合に FTP に長い時



映像 1 (制御あり) 映像 2 (制御なし)

図 7 実験結果

Fig. 7 Experimental results.

間がかかっておりポリシーを満足していない。ネットワークレベルの制御を行うと、FTP サービスに対して帯域予約が行われ FTP の完了時間が短くなる。しかしその結果、転送に広帯域を必要とする DV 転送が圧迫され、多くの損失が発生している。同時に WWW の品質も劣化する。この状態ではアプリケーションレベルの制御によってネットワークに流入するパケット数を制限しないと、優先される FTP サービス以外の QoS が大きく劣化する。ここでは QMS によって、優先度の低い DV 転送サービス、WWW サービスともに QoS が向上している。そして、WWW サービスを優先した運用ポリシー 3 でも、QMS での制御によって優先度の低い FTP の完了時間が短くなっている。

この実験から、QMS によってシステム運用ポリシーに従った運用が可能になること、さらに運用ポリシー 2 のように優先度の高いサービスによって他のサービスが圧迫されるような場合でも、優先度の低いサービスの QoS が維持されることが分かった。

### (2) 実験 2

アプリケーションレベルの制御を行った場合と行わなかった場合の映像を図 7 に示す。映像 1 と映像 2 ではスイッチ間の映像の転送帯域は約 10 Mbps でほとんど同じである。しかし、映像 2 ではスイッチのバッファでパケット損失が発生しているため、映像品質が劣化している。デジタル映像の場合、パケットの損失によって映像品質が著しく劣化する場合がある。そのため、本実験で採用したような映像サーバにおける送出帯域の制御が重要になってくる。本実験から、QMS によって DV 転送サービスのアプリケーション QoS (ユーザが体感する映像品質) が向上することが分かった。

ただし、映像 1 における制御はフレームレートを下げることによって実現しているため、動きのなめらかさという観点からの品質は劣化する。動きのなめらかさを優先するサービスの場合は、映像のエンコード

レートを変更する制御などを採用する必要がある。

## 6. ま と め

本稿では、アプリケーション QoS 管理システム (QMS) に QoS 管理ポリシーを設定する方法を提案した。本設定方法はアプリケーションのトラフィック特性を選択することにより自動で制御の優先順位を決定する機能を持ち、AP 管理者が QoS 管理ポリシーを QMS に設定する作業を支援する。また本設定方法に基づく制御は、ネットワークレベルとアプリケーションレベルの制御を組み合わせた制御であり、これにより複数アプリケーション間の QoS 交渉が実現し、システム全体として QoS 管理ポリシーに従ったシステム運用が可能になる。本稿ではまた、QMS を DV 転送システムを利用した実験システムに組み込み、QMS の実装検証を行った。その結果、QMS によって QoS 管理ポリシーに従ったシステム運用が可能になること、特に相対的に優先度の低いアプリケーションの QoS が向上することが分かった。

最後に、QMS の設計目標の 1 つである実システムへの適用を目指し、解決しなくてはならない問題、および今後の課題をまとめておく。まず問題点の 1 つめは、アプリケーション QoS の劣化をどのように検出するかということである。今回、クライアント計算機上に実装した評価用ウィンドウによって劣化の検出を行ったが、QMS が自動的に劣化を検出する方法と本手法との比較、QoS 要求に対する課金の仕組みの導入などが今後の課題として残っている。

もう 1 つの問題は規模適応性に関するものである。本稿では同一の管理グループによって管理されるネットワークシステムを制御対象としたが、QMS をインターネット上での応用を考えたシステムに適用する場合、異なる管理グループによって管理されるネットワークシステムの間での QoS 交渉の方法、複数の Manager モジュールの連携方法などの検討が必要である。QMS はアプリケーションレベルの制御方法としては、フィードバック型の簡易な制御方法を用いるため規模適応性に対応できると考えられる。しかし、ネットワークレベルの制御方法の中には RSVP のように規模適応性に欠ける方法も存在するので、どのような範囲でどのような制御をどのように組み合わせて適用すべきかを検討する必要がある。さらに、セグメントをまたいだ通信では、同じアプリケーションでもセグメントによって QoS 管理ポリシーが異なるのがふつうであり、その場合、他のセグメントにおける QoS をどのように保証するかという問題が出てくる。現在

でも、自セグメント内では RSVP などで帯域保証を行い、他セグメントを経由するトラフィックに関しては DiffServ などで対処する手法が提案されている<sup>16)</sup>。ここにさらに、QMS における『優先度の低いアプリケーションのサービスレベルを故意に劣化させる(ただしアプリケーション QoS は維持する)』という考え方を導入する必要があると考える。そのため今後は、複数のセグメントから成るネットワーク構成に対してシミュレーション実験、および実装実験を行っていく予定である。

## 参 考 文 献

- 1) 青山友紀：ネットワークの進化と IP 技術，電子情報通信学会誌，Vol.83, No.4, pp.248-256 (2000).
- 2) Braden, R., Clark, D. and Shenker, S.: *Integrated Services in the Internet Architecture: an Overview*, IETF RFC1633 (1994).
- 3) Brake, S., Black, D., Carlson, M., Davies, E., Wang, Z. and Weiss, W.: *Architecture for Differentiated Services*, IETF RFC2475 (1998).
- 4) Rosen, E., Viswanathan, A. and Callon, R.: *Multiprotocol Label Switching Architecture*, IETF RFC3031 (2001).
- 5) Tennenhouse, D.L., Smith, J.M., Sincoskie, W.D., Wetherall, D.J. and Minden, G.J.: A survey of active network research, *IEEE Commun. Mag.*, Vol.35, No.1, pp.80-86 (1997).
- 6) 八幡博史，マイケル P. ウェルマン，石田 亨：市場モデルに基づくアプリケーション QoS の制御，電子情報通信学会論文誌 D-I，Vol.J81, No.5, pp.540-547 (1998).
- 7) Nahrstedt, K. and Smith, J.M.: The QoS broker, *IEEE MultiMedia*, Vol.2, No.1, pp.53-67 (1995).
- 8) 橋本浩二，野村尚央，柴田義孝，白鳥則朗：QoS 保証を考慮したやわらかいマルチメディアシステム，情報処理学会論文誌，Vol.40, No.1, pp.113-123 (1999).
- 9) Yamazaki, T. and Matsuda, J.: Adaptive QoS management for multimedia applications in heterogeneous environments, *IEICE Trans. Commun.*, Vol.E82-B, No.11, pp.1801-1807 (1999).
- 10) Kato, Y. and Hakozaiki, K.: Application QoS Management for Distributed Computing Systems, *Proc. ICC'2001*, Helsinki, IEEE (2001).
- 11) Object Management Group: *The Common Object Request Broker: Architecture and Specification*, Object Management Group, Framingham, Massachusetts (1998).
- 12) Paxson, V.: Towards a framework for defining Internet performance metrics, *Proc. INET'96*,

p.3 (1996).

- 13) Braden, R., Zhang, L., Berson, S., Herzog, S. and Jamin, S.: *Resource ReSerVation Protocol (RSVP)*, IETF RFC2205 (1997).
- 14) 間瀬憲一：マルチメディアネットワークとコミュニケーション品質（社）電子情報通信学会（編），コロナ社（1998）.
- 15) Ogawa, A., Kobayashi, K., Sugiura, K., Nakamura, O. and Murai, J.: Design and Implementation of DV based video over RTP, *Proc. Packet Video'2000*, Italy (2000).
- 16) Stoica, I. and Zhang, H.: Providing Guaranteed Service Without Per Flow Management, *Proc. ACM SIGCOMM'99* (1999).

(平成 13 年 5 月 8 日受付)

(平成 13 年 10 月 16 日採録)



加藤 由花（学生会員）

1989 年東京大学理学部卒業。同年日本電信電話（株）入社。ATM 網のトラフィック制御に関する研究，双方向マルチメディアシステムの研究開発に従事。1999 年電気通信大学大

学院情報システム学研究科博士前期課程修了。現在，同後期課程在学中。分散システムにおける QoS 制御技術の研究に従事。電子情報通信学会，IEEE 各会員。



佐々木 徹（学生会員）

2000 年電気通信大学電気通信学部情報工学科卒業。現在，同大学院情報システム学研究科博士前期課程在学中。QoS の制御手法に関する研究に従事。



箱崎 勝也（正会員）

1941 年生。1963 年九州大学工学部電子工学科卒業。同年日本電気入社。中央研究所，ソフトウェア開発グループにおいて，システム性能評価，コンピュータアーキテクチャ，OS，ネットワークの相互接続性等の研究開発に従事。1994 年から電気通信大学大学院情報システム学研究科教授。工学博士。分散システム技術，マルチメディア利用技術，モバイルコミュニケーション，情報システムアーキテクチャ等の研究に従事。情報処理学会論文賞受賞（1982）。分散システム運用技術研究会主査。電子情報通信学会，IEEE，ACM 各会員。