

インターネット上の動画転送を意識した動的 QoS の制御

山内 長承、河内谷 清久仁、串田 高幸

日本アイ・ビー・エム株式会社 東京基礎研究所

資源の確保状況に応じて、転送を拒否するのではなく、動画・音声データの圧縮率を変更するなどして、柔軟に転送サービスを実現する動的 QoS の技術が試みられている。動的 QoS の概念は、予約できない資源へも拡張できるので、予約できる資源とできない資源の混在する環境を統一的に扱える可能性を持っている。本報告では、初めに資源管理を利用して QoS 制御をする技術を整理し、帯域予約できないネットワーク上での動的 QoS 技術の初期実験の概要と結果を報告し、問題点を整理する。

Dynamic QoS Control for the Internet Environment

Nagatsugu Yamanouchi, Kiyokuni Kawachiya, Takayuki Kushida

Tokyo Research Laboratory, IBM Japan

The dynamic QoS technology, which, instead of refusing the service because of the short available resources, continues video/audio transmission by changing the data rate according to the available resource amount, has been experimented in several places. Since the concept of the dynamic QoS can be expanded to the resources where no reservation mechanism is implemented, it will be a reasonable candidate to manage services using both reservable and non-reservable resources. This report will first sort out the concepts and ideas about the dynamic QoS control, and then explain about our initial stage experiment and its results with a list of the problems to be solved in the next step.

1 はじめに

近年、動画・音声等のマルチメディアデータをネットワーク転送・表示するための、帯域保証技術や QoS (サービス品質) 制御技術が、多く論じられている。特に、資源の確保状況に応じて、転送を拒否するのではなく、動画・音声データの圧縮率を変更するなどして、柔軟に転送サービスを実現する動的 QoS の技術が試みられている。動的 QoS の概念は、予約できない資源へも拡張できるので、予約できる資源とできない資源の混在する環境を統一的に扱える可能性を持っている。我々は、マルチメディア転送サービスが将来のシステムの基盤の一部となるものと考え、多様な環境下で、エンドエンドの転送をユーザの品質要求とコストのバランスの上で実現するための枠組の一部として、動的 QoS 技術を検討したい。

本報告では、初めに資源管理を利用して QoS 制御をする技術を整理し、見通しを明らかにした上で、我々の目標とするシステムの一部をなす、帯域予約できないネットワーク上での動的 QoS 技術の初期実験の概要と結果を報告し、問題点を整理する。

2 マルチメディア転送と QoS

動画像などを転送する場合、あるデータの到着が締切時間に間に合わないと、画像や音声途切れたり乱れたりする。これを防ぐために、一定帯域を当該の通信に割り当てる/占有させる技術が実現、普及し始めている。ネットワークが、利用者に対して提供する転送サービスの品質を保証しようとするもので、品質項目として上げられているのが、(1) 帯域/転送レート、(2) 遅れ、(3) 誤り率/誤挿入率/損失率、などである (2)。これらの項目を保証するネットワークとしては、ATM 網における QoS 保証や TCP/IP ネットワークの枠組で提案されている帯域予約型プロトコル ST-II [3]、RSVP [4] などが知られている。他方、インターネットや既存の LAN で普及している TCP/IP プロトコル (TCP, UDP など) は、帯域を保証しない。最近 RSVP を備えたルーターも出始めており、新しい標準 IPv6 でも帯域保証について積極的に言及しているの、インターネットの一部は近いうちに帯域保証機能を持つようになるであろうが、当面は保証するネットワークと保証しないネットワークが混在する。

ネットワークの QoS と、エンドユーザが見る動画像の「品質」の関係は必ずしも自明ではない。大別すると次のような考え方があ

- (1) 必要な資源を確保することができる場合は、予め決められたレベルの動画サービスを実現できる。たとえば、ネットワークが 1.5Mbps の帯域を保証できるならば、MPEG1 (@1.5Mbps) の動画/音声サービスができる。資源が確保できない場合は、セッションを開始できない。つまり、エンドユーザにとっては、MPEG1 (画像としては初期の VHS レベル) の動画が見られるか、サービスを拒否されるか、の 2 つのサービス品質がある。
- (2) 既存のインターネットのように、資源確保ができないネットワークを転送する場合、混雑して帯域が不足してくると、パケットの到着が遅れ、動画再生に間に合わなくなるので、絵や音が途切れたり乱れたりする。エンドユーザは、ネットワークがすいていて転送が間に合っている時は当初の MPEG1 の品質の動画を見られるが、混雑すると乱れた画像を見せられる。
- (3) 上記の (1) と (2) の間を柔軟に埋めるのが、動的 QoS + メディアスケールリングの考え方である ([6][7])。最高レベルのサービスに必要な資源を予め排他的に予約するのではなく、その時々で与えられた資源の量に応じて、動画/音声の品質を下げることによって (メディアスケールリング) データ量を減らす。これによって、サービスのレベルは低下するが完全には中断しない (図 1)。

具体的には、たとえば 1.5Mbps の MPEG1 データを転送するサー

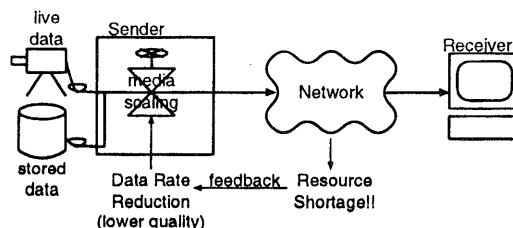


図 1: 動的 QoS の概念

ビスで、混雑のため帯域が 1Mbps しか利用できない時、画像の解像度などの品質を下げるにより必要帯域を 1Mbps にできれば、転送することができる。エンドユーザは絵の品質が若干低下したことに気づくが、サービスは継続される。

資源確保の変更を行なうことのできるタイミングによって、更に 2 通りに分類できる。すなわち (3-a) サービス開始時に必要な資源を管理者と交渉することにより、必要量を確保するが、セッションの途中で変更せず最後まで当初の資源を確保する場合、及び (3-b) サービス中にも資源とそれに対応する動画品質を変更する場合である。(3-b) には資源確保量をセッション中に変更するだけでなく、既存インターネットのように資源予約ができないシステムにおいて、その転送状況のみをフィードバックして動画品質を変更する場合を含めて考えることができる。

動的 QoS の概念はネットワークの他、CPU 資源についても適用され、実現機構が試されている他 ([11])、資源要求モデルが提案されている ([1])。

エンドユーザから見たサービス品質という観点で見直すと、

- (1) は、資源が確保できれば最高品質の画像を楽しむことができる。共有資源が少ないサービスを拒否される (「お話中」になる) 率が高くなり、極端な場合にはまったく繋がらないということになる。話中率を十分小さくするためには、多くの資源を用意する必要があり、繁忙時間帯以外では使われない無駄な資源が多くなる。
- (2) は動画や音声の場合途切れるという品質低下を招き、(3) に比べるとユーザは許容しにくい。
- (3) は動画の場合、個々の画面の解像度や色数が一時的に減少するような品質低下が実現でき、エンドユーザの不快感が (2) に比べて小さいと思われる。特に、品質低下の期間が短いと、ほとんど気づかない可能性がある。動的 QoS が機能する場合、一定帯域内に少し詰めることができる、バーストトラフィックによる影響を受けにくくなる、などの理由から、予め準備する資源を圧縮できる可能性がある。

更に、資源量によるサービス品質の動的変更に係わる別の視点として、一様でないネットワークへのマルチキャストの問題が上げられている ([5])。図 2 に見られるように、リンクの帯域が受信端によって異なる場合や、端末側の能力 (表示能力、処理能力) が異なる場合に、同一データをマルチキャストによりどのように配信するかという問題である。

現在提案されている解決策は、ネットワーク内で帯域に応じてデータを選別し、配送する方法である。例えば、動画像データを図 4 のように帯域別に階層的に圧縮しておく。基本部分のみで再生すると最低品質での画像が得られ、順次上位の部分を追加して再生 (基本部

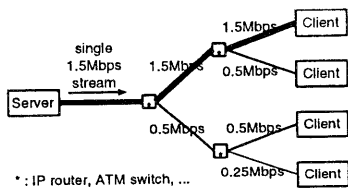


図 2: マルチキャストにおける QoS

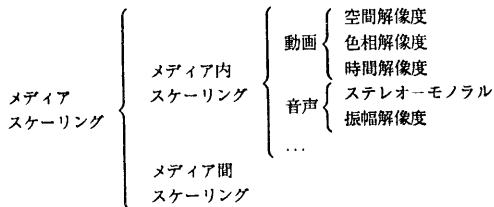


図 3: メディアスケールングの全体

分]+[追加レベル 1]+[追加レベル 2] など)すると、そのレベルに応じた高い品質が得られるとする。このデータを転送するとき、ネットワークの先の状態に応じて、必要な部分のみを転送するような中継をすると、端末毎に異なる品質で送ることができる。

これも、帯域管理とメディアスケールングの組合せであるが、大きな違いは、前のシステムでは帯域に合わせたデータの選択がエンドエンドで考えていたのに対し、マルチキャストの場合ではネットワーク中のリンク毎に行なわなければならない。この実現法として、ネットワーク途中にアプリケーションレベルまで解釈するノードを入れるか、ネットワークヘッダーの中に優先度のような形で記述することによりアプリケーションレベルまで解釈せずに構成するかは、議論の余地がある。

3 メディアスケールング

動的 QoS の実現に欠かせないのがメディアスケールング技術である。スケールングの手法は図 3 のように大別できる。

動画のスケールングでは空間解像度、色相解像度、時間解像度の 3 種類がスケールングの対象となる ([9])。これらはお互いに独立に変更できるので、ある転送レート/圧縮率が要求された時に、その 3 軸のなす空間の中で視覚上もっとも好ましい(影響の少ない)点を選んでスケールする必要がある。音声のスケールングは音声の圧縮方式に依存するが、例えば MPEG Audio の場合ステレオからモノラルへの削減、振幅解像度の削減が考えられる。

アプリケーションによっては、メディア自体が変わるスケールングも考えられる。たとえばニュースのような場合、動画と音声で伝えていた番組を音声だけにしてもある程度の用をなす。音声も通せないほどであれば、テレビの文字放送のように文字を低速で送ることも考えられる。

システムの実現方法は、大別すると図 4 のように環境によって 3 通りが考えられる。

- 実時間圧縮が可能な場合、圧縮エンコーダに与える圧縮パラメータを変えることによって出力のデータレートを変えることができる。

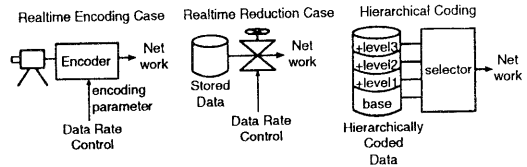


図 4: メディアスケールングの構成

今までの動的 QoS の実験は多くこの場合を用いて行なっている。この方法はデータレートをかなり細かく変更できることが利点となる。しかしビデオサーバーのように多数のセッションを同時にサポートするシステムでは、セッション毎にエンコーダを必要とすることが欠点となる。特にハードウェアを用いる場合、その個数が大きなものになると実用性を失う。ソフトウェアの場合も同時ストリーム数が増えると大きな CPU 負荷になる。

- 蓄積型のサービスの場合、圧縮済みの動画ストリームを加工することにより実時間でレートを変更することが考えられる。このとき、データは圧縮された形で蓄積できるので、蓄積媒体は節約できる。問題は圧縮された形のままでどこまで加工が可能かという点に絞られる。また、ストリーム毎に本機構が必要であることは、実時間圧縮の場合と同様である。この方法の一方の極端として、一旦デコードして再エンコードすることが考えられるが、データ量に損失のある (lossy な) 圧縮では画質が低下する恐れがある。この方法の代替案としては、異なる圧縮率のデータを何種類か持ち、それを切替えて使うことが考えられる。この手法は、蓄積スペースの量が余分に必要になること、複数のデータの一元的な管理が必要になること、切替時の同期が必要なこと、の 3 点を克服する必要がある。

- 階層コーディングによる方法は、データをたとえば最低品質部分、次レベルとの差分、更に次レベルとの差分、というように階層化して圧縮して持つ方法である。もし階層コーディングの結果のデータ量の和が、直接高いデータレートで圧縮したデータ量に比べてあまり増えないのであれば、各レートで圧縮したファイルを複数持つ方法に対してメリットがある。再生時に各層のデータを同期させる必要がある。

この方法では、予めエンコードしておくことができるので、スケールング操作はどのデータ部分を送るかを選択するだけで良く、多数のセッションを持つサーバーでは負荷軽減のメリットがある。他方、スケールングのレベルは予めエンコード時に決められてしまうので、細かく制御することはできない。

4 動的 QoS 実現のための仕組み

動的 QoS を含めて、システム全体に QoS 概念を導入するためには、資源を管理する機構だけでなく、QoS を管理する機構が必要となる。これを実現するために図 5 のような仕組みが考えられる。

資源管理機構 (図中 per-resource manager) はいまままでの資源予約型のシステムで実現されているようなものに沿って考える。まず、資源確保機能のレベルとしてたとえば次の 3 レベルが考えられる。

- (1) 資源を予約できないが状況は通知する。既存の TCP/UDP や UNIX OS などに資源の状況を通知できるような機構を追加すれば、この仕組みに加わることができる。
- (2) 指定値を予約できるが、セッションの期間中は変更できない。たとえば ATM の CBR(Constant Bit Rate) モードなどのネットワーク

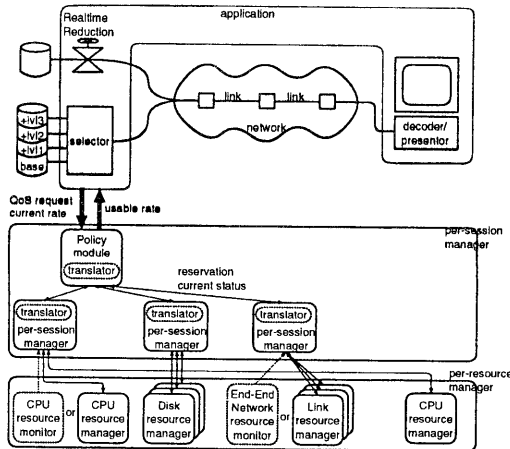


図 5: QoS のアーキテクチャ

クはこれに該当する。状況のフィードバックとしては予約できたかできなかったかを返すことになる。

- (3) [最低確保値, 希望値] の対など、複雑なモードで予約でき、セッションの間中もある範囲では変更交渉が可能である。最低確保値は譲れない予約値とし、もちろん希望値分だけを使いたいのだが、最低確保値までの間の値に譲ることができる。

いずれにしても、最低限(1)のように状況を通知する機能が必要である。

ネットワークにおいて資源予約をする場合、各リンクの転送帯域やルータ/スイッチの転送容量が資源となるので、セッションに対して経路を固定しその経路沿いの全てのリンクの資源をまとめて確保する必要がある。従って交渉可能な予約システムの場合は、リンク毎の管理の上で、セッション毎に管理する必要がある。他方、予約しない場合は、後述の実験の中で見られるようにエンド-エンドの状況だけでも判定できることがある。ネットワーク以外ではたとえば CPU 資源、ディスク資源 [10] が必要になる。

QoS 管理機構はおよそ次のような要件を満たすことが必要になる。

- アプリケーションとのインターフェースを持ち、アプリケーションからの要求を聞くと同時に、集めた資源状況からあるポリシーを持って推奨転送レートを割出し、アプリケーションへ返す。ここでポリシーはアプリケーション毎に異なる可能性がある。
- 各資源の資源管理機構とインターフェースする。各資源の確保に関して交渉し、更に状況を得る。ネットワーク内には資源確保機能のレベルの異なる様々な資源があるので、それらとインターフェースしなければならない。

このようなシステムでは、異なる資源に対する QoS パラメータ異なる。それらの間の翻訳を資源管理機構で行うか、QoS 管理機構の一部とするかは、今後の研究課題である。

5 予約できないネットワークの動的 QoS 実験

5.1 実験システム

これまでの議論を踏まえて、動的 QoS の実装上の問題を検討する第一歩として、図 6 のような実験システムを作成した。本実験では、

- クライアントサーバーモデルで、ビデオサーバーをモデルとする。

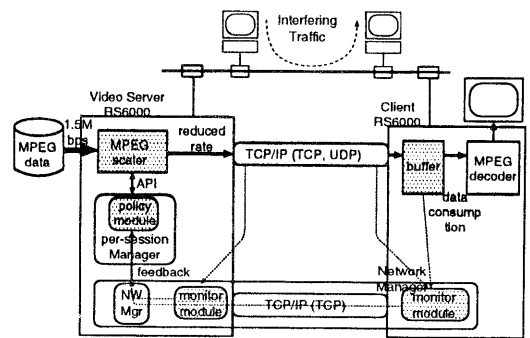


図 6: 実験システム

- (マルチキャスト型でなく) 1対1型の通信を用いたサービスを考える。
- ネットワークの QoS だけを検討する。
- ネットワークは帯域予約ができない。状況データを収集するモニターモジュールを装着する。
- 送出サーバー側で、ディスクに格納された MPEG1 データを読み出し、推奨レートにしたがって動的にメディアスケールングをし、転送レートを変化させる。

という構成例で実験した。実験の主眼として

- 帯域予約できないネットワークにおいて、このようなフィードバックによる動的 QoS 制御が可能であることを実証する。
- 特に、MPEG1 の実データを使うことにより、フィードバック系に与える影響を調べる。
- QoS アーキテクチャの中で、メディアスケールングと QoS 管理部分との分離、QoS 管理部のポリシーモジュールの分離などが可能であるか実証する。
- QoS アーキテクチャの中で、フィードバックのパラメータの妥当性を検討する。

などが挙げられる。

動的 QoS 機構 (DQoS) の第一歩としては、ネットワーク状況をモニタするモニターモジュール、モニターモジュールからの情報をマネージャに転送するプロトコル、状況情報を受けてポリシーを計算するポリシーモジュールといくつかのサンプルポリシー、メディアスケールへのインターフェースなどを作成した。サンプルポリシーとしては、例えば次のようなものを考えた。

- (1) 受信側バッファの残量をモニタし、高水位基準を越えればデータレートをあげ、低水位基準を下回ればデータレートを下げる方法。通常のフロー制御と逆向きに見えるが、このデータレートはバッファから出て行く側のフローレートを指定することになるので、これで良い。
- (2) 受信側でのパケット到着レート (パケット長/到着間隔) を求め、データのレートをこの到着レートに合わせる方法。

Vosaic ([12]) ではフレームが締切時間に間に合わない割合を検出し、15%のスレシールドを越えたとするとレートを下げ、5%を下回るとレートを上げることを試みている。

メディアスケールングの部分 (MPEG スケラ) は、時間軸スケールングを用いた暫定的な機構であるので、レベルは 4 通りしか許されない。実転送レートがレベル 3 を下回ると、デコーダのバッファが枯渇して一時停止し、音が途切れる。

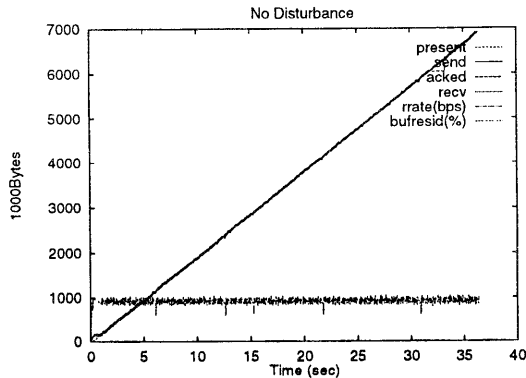


図7: トラフィック負荷がない場合

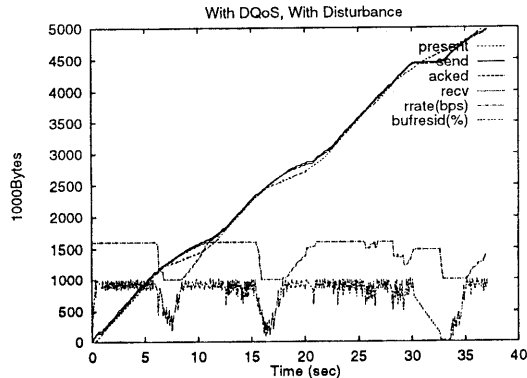


図9: DQoSがあるシステムにトラフィックをかけた場合

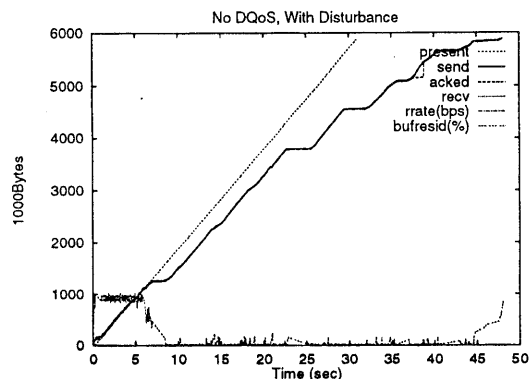


図8: DQoSがないシステムにトラフィックをかけた場合

レベル0	レベル1	レベル2	レベル3
30 フレーム/秒	10	2	音声のみ
約 1.5 Mbps	約 1.0 Mbps	約 0.6 Mbps	約 0.25 Mbps

機材は、サーバー、端末とも RS6000 ワークステーションを用い、ネットワークは Ethernet を用いた。また負荷トラフィックは別ワークステーションやスニファーにより生成した。

5.2 実験の結果

図7はトラフィック負荷がないので円滑に1.5Mbpsで転送できた場合、図8はDQoS機構がないためトラフィック負荷がかかると1.5Mbpsで途切れた場合、図9はDQoS機構が働くためトラフィック負荷があっても圧縮率を高めて送り切れた場合の、転送の有り様を示している。横軸は経過時間、縦軸はデータ量の積算値を示す。各グラフの意味は、

present その時刻に再生/表示されるデータの位置。各データのMPEGデコーダで再生する時にかかる時間は、データ長をその時の再生レートで割った時間でわかる。それを積算したものがこのデータを表示する時刻になる。

send その時刻に送出されるデータ。同様に **ack** は端末からサーバへ **ack** が返された時刻とデータの対応、**received** は端末側でデータを受信した時刻とデータの対応を示す。スケールに比べてこれらの値の差が小さいので、グラフ上は重なっている。present より上

にあれば端末側での受信が追い付いていることになる。水平に推移している所はこの区間データが到着していないので、端末側の受信バッファが枯渇してしまう。

rrate DQoS がメディアスケラに対して通知している推奨レートの値 (バイト/秒)
bufresid 端末側受信バッファ内のデータ量 (%)。バッファが空になると絵や音が途切れる。

図7では、ネットワークに負荷がほとんどなく、バッファ内もほとんど詰まっており、受信量 **recv** はデコーダの消費量にしたがって、直線的に伸びている。

ちなみに、本実験ではフロー制御は別途行なわれており、受信バッファが詰まっているとデータを転送しない。したがって、ネットワークがすいている定常状態では、バッファが消費されて空いた分だけのデータが送られるので、送信レートは定常的にMPEGのレート(1.5Mbps)になる。

図8では、トラフィックをかけているので受信 **recv** が遅れており、表示 **present** に対して追い付いていない (**recv** が **present** より下にある) ので、受信バッファがほとんど0 (枯渇) の状態が続いてしまう。実際はバッファが枯渇した時デコーダが停止するので、消費がなくなり、パケットが到着すればバッファに溜り、しばらくすると再開できる。たとえば6-8秒時点で **recv** が水平になっており、パケットが到着していない。バッファは8秒時点ぐらいで枯渇するが、8秒時点ぐらいで元の再生速度と平行な状態に戻っており、この時点で再生が復活した様子がわかる。このケースではレートの制御をしていないので立ち上がると元通り1.5Mbpsで再生を始めている。

図9はDQoSが働いている場合である。5秒時点ぐらいでバッファ量 **bufresid** の低下を検出したので、MPEGのデータレート **rrate** が1.6Mbpsから1.0Mbpsに下げられている。それによって5-11秒ぐらいの期間は表示バイト位置 **present** の時間変化が鈍っている。それによって **recv** の傾きも鈍っているが、バッファ量 **bufresid** は再生速度が遅くなったために復活し、8秒時点ぐらいで一杯になっている。データレート **rrate** は徐々に復活し、**present** が元の傾き (1.5Mbps、高品質状態) に徐々に戻り、それを追いかけて **recv** の傾きも元に戻っている。

これらの結果から、動的QoSのフィードバック機構は効果があることがわかるが、同時に次のような問題も判明した。

- ネットワークの状態として収集するデータとして何が適当か。特に、様々なポリシーを書くことを許すとした場合、ポリシーからの参照上必要で、しかし十分に普遍的であるパラメータとは何だ

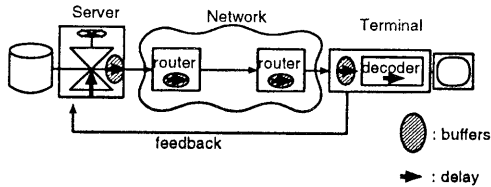


図 10: 遅れの要素とフィードバックループ

らうか。

- どのようなポリシーが適当か。アプリケーションに依存することは良いとして、個々のネットワークに依存せず普遍的なポリシーが存在するのか。
 - アプリケーションに依存せず普遍的なポリシーが存在するのか。アプリケーションの側で推奨レートを受けとって解釈する仕方はポリシーの一部ではないか？
 - フィードバックループ内の遅れが、改善を妨げていると思われるが、実際にどうなのか。図 10にあるように、メディアスケララの遅れ、ネットワーク内のバッファリングがフィードバックループの内側にある。
- また、MPEG 動画の特質として、フレームごとのデータ量の変動を吸収するバッファを受信側デコーダ内にもっている。デコーダ内のバッファリングはフィードバックループの外側にあるように見えるが、たとえば受信バッファの容量を元にフィードバックしていると、デコーダの消費量の追従の遅れが影響する。
- QoS パラメタの翻訳は本当に可能なのだろうか。ネットワーク、CPU、ディスクなどの資源に共通のパラメタを定義できるのか。
 - QoS パラメタを誰が翻訳するのが適当なのか。資源の側で翻訳し、全ての資源に共通な言語でセッション対応のマネージャに渡せるのだろうか。それともセッション/アプリケーション毎に解釈を変えなければならぬのだろうか。

多様な環境での QoS 管理に関する疑問は、ATM など帯域予約のできるネットワーク、帯域予約できる OS 下での CPU、できない OS 下での CPU など、いろいろな資源を実際に組み合わせて、ポリシーを書いてみた段階で結論できると考えている。

また、メディアスケラリングの側に対しては、次のようなことが言える。

- 実用上は、動的 QoS のフィードバックポリシーと並んで、品質低下を目立たせないメディアスケラリングの手法が重要である。今回は敢えてフレームレートの変更という目に見える形でスケールしたが、色相の低下など、より目立たない手法を検討する必要がある。
 - 動的 QoS のフィードバックを受けるためには、この実験のように 4 段階というような大きなステップではなく、ある程度スムーズなスケールがあるといいのではないか。ポリシーを書く場合、スレシールドがあると、それをどのように扱うかが難しい。色相軸や空間軸のスケラリングである程度細かいステップのスケラリングが可能ないように思われる。
- 但しこれは、階層的なコーディング方式では刻みが細かいだけ、データの持ち方が難しくなる欠点がある。
- 今回は実時間の加工が MPEG1 のデータレートに追い付いた。これはデータレートが 1.5Mbps と低いこと、サーバーがワークステーションで CPU 処理能力が大きいこと、同時セッション数をごく少数に限定したこと、によっている。これらの条件が外された時、たとえば MPEG2 での利用、PC サーバの利用、同時セッ

ション数を実用的な数まで増やした場合など、実時間で処理が追い付かなくなる可能性がある。

6 おわりに

本報告では、動的 QoS 技術を整理するとともに、インターネットのように帯域予約のできないネットワークでのモニタリングだけによる QoS 制御の実現可能性を検証し、併せて問題点を抽出した。今後は実験結果の問題点として上げられたより良い、一般的なポリシーの設計の他、ATM などの予約可能なネットワーク、予約可能/不可能な CPU 資源など、他の種類/性格の資源との共存環境での制御について検討していきたい。

本研究を進めるに当たり貴重な助言をいただいた、当研究所の椎尾研究員、根岸研究員に感謝する。なお、本報告で述べられている見解は筆者の個人的な見解であり、日本アイ・ビー・エムおよび IBM Corporation を代表するものではない。

参考文献

- [1] 河内谷、緒方、西尾、徳田: 連続メディアの QoS 制御のための「QoS チケット」モデル. 第 50 回情処全大論文集 (1N-6). 1995.
- [2] The ATM Forum: ATM User-Network Interface Specification UNI 3.1. Prentice Hall, 1995.
- [3] Delgrossi, L., Berger, L.: The Internet Stream Protocol version ST-2+. IETF RFC 1819.
- [4] Zhang, L., Deering, S., Estrin, D., Shenker, S., Zappala, D.: RSVP: A new resource reservation protocol. IEEE Network, Sept. 1993. pp. 8-18.
- [5] 尾上裕子: QOS に基づくマルチキャスト通信. 情報処理学会マルチメディア通信と分散処理研究会報告 63-7, Jan. 1994. pp. 49-56
- [6] Tokuda, H., Tobe, Y., Chow, S.T.C., Moura, J.M.F.: Continuous media communication with dynamic QOS control using ARTIS with an FDDI network. Proceedings of ACM SIGCOMM '92, Oct. 1992. pp. 88-98.
- [7] Campbell, A., Eleftheriades, A., Aurrecochea, C.: Meeting end to end QoS challenges for scalable flows in heterogeneous multimedia environments. Proceedings of High Performance Networking VI, Sept. 1995. pp. 101-115.
- [8] ISO: ISO/IEC 11172-1/2/3 (MPEG1) 1993.
- [9] ISO: ISO/IEC 13818-1/2/3 (MPEG2) 1994.
- [10] Haskin, R. L.: The Shark continuous-media file server. Proc. IEEE COMPCON Spring '93, pp. 12-15. 1993.
- [11] 河内谷、緒方、徳田: Real-Time Mach 上での QOS 制御サーバの実験. 第 47 回情処全大論文集 (4V-3). pp.2-355 - 2-356. 1993.
- [12] Chen, Z., Tan, S., Cambell, R.II., Li, Y.: RealTime Video and Audio in the World Wide Web. <http://choices.cs.uiuc.edu/>. 1995.
- [13] Nahrstedt, K., Smith, J. M.: The QOS Broker. IEEE Multimedia Vol.2, No.1. 1995.