

## インターネット上において オーディオ及びビデオを配送するプロトコル

串田高幸\* 河内谷清久仁 山内長承

日本アイ・ビー・エム株式会社  
東京基礎研究所

最近、インターネットのような QoS(Quality of Service) が保証されていないパケット交換ネットワークの利用が急激に増大してきている。このような QoS 非保証型のパケット交換ネットワークでは、途中のノードにおいて QoS を保証するために特別な機能を必要としない。そのため、ネットワークノードを容易に設置することができる。本稿では、ビデオやオーディオを含んだマルチメディアデータをインターネットのような QoS を保証していないパケット交換ネットワークにおいて、ネットワークの状況の変化を動的に検知して、その結果を随時アプリケーションに知らせ、アプリケーションデータの質を変更することによって実時間でマルチメディアデータを転送するトランスポートプロトコルについて述べる。

## Realtime Multimedia Protocol with Deadline Driven and Prioritized Data on Packet Networks

Takayuki Kushida(kushida@trl.ibm.co.jp)  
Kiyokuni Kawachiya  
Nagatsugu Yamanouchi

IBM Research,  
Tokyo Research Laboratory

The installation of a packet exchange network such as the Internet which doesn't guarantee QoS(Quality of Service), is increasing in these days. There is no advanced mechanism to handle QoS parameters in intermediate nodes such networks. If we want to guarantee QoS on the network, QoS parameters have to be guaranteed between end-to-end nodes. In this paper we introduce a new transport protocol to support the multimedia data with a realtime on the non-QoS guarantee network such as the Internet. The protocol detects the network status dynamically and tells its status to the application. This protocol allows us to transfer a multimedia data in time.

はじめに

マルチメディアのデータをネットワークで転送する場合、実時間制御とユーザへのサービスの質が、今までのデータ通信と異なっている。

最近インターネットのように QoS(Quality of Service)を保証しないパケット交換ネットワークの利用が拡大してきている。インターネットのように QoS 非保証型のパケットネットワークでは、途中のノードにおいて QoS を保証するための特別な機構を持っていない。そのため、ネットワークノードの設置が容易である。インターネットは、コンピューター間のデータ転送のうち、特にバッチ形式のファイル転送やリモートログインのような実時間を必要としないアプリケーションに対して盛んに利用されてきている。

一方、インターネットのような QoS 非保証のパケットネットワークでは、時間によってネットワークの状態が変化している。このような状況においても実時間でオーディオ及びビデオのようなマルチメディアデータを転送することが、ネットワークの利用の拡大とともに要求されるようになってきた。ネットワークが混雑した状況では、途中の経路において転送されているパケットに遅延や損失が起こってしまう。その結果として、ネットワークからアプリケーションに提供されるサービスの品質が時間により異なってくる。

パケット交換ネットワークでの実時間制御の根本的な解決案として、Clark[2]らは、新しいアーキテクチャとして ISPN(Integrated Services Packet Network)を提案した。また Zhang は、パケット交換ネットワークで転送するトラフィックに対して各ノードのスループットを保証するために Virtual Clock 方式 [3] を提案した。しかし、これらのまったく新規なアーキテクチャをもとにしたパケット交換ネットワークは、既存のネットワークに対して大規模な変更を必要とするため、その実現が極めて困難である。既存のパケット交換ネットワークを改良せずに行なう方式として End-to-end で制御する方式がある。

本稿では、我々の前に行なった研究 [10] を拡張して、非 QoS 保証型のパケット交換ネットワークにおいて、ネットワークの混雑のためにデータの遅延や損失が存在する場合であっても End-to-end の制御によってアプリケーションから見たときにネットワークのサービスの質に合った適正な品質のマルチメディアデータを実時間で転送することができるトランスポートプロトコルについて述べてゆく。

まず、背景として QoS 保証及び QoS 非保証サービスについて述べ、そこでなぜ実時間が保証されないかについて述べる。また、どこが QoS の選択権を持つかについても述べる。さらに QoS を保証する方式として、(2)End-to-end と Hop-by-hop の方式の長所及び短所について述べる。また、この QoS の変動と関連して、(3)メディアスケューリング技術について述べる。これらの解析をもとにして、(4)QoS 非保証型ネットワークに対す

る新しいプロトコルについて述べてゆく。(5)最後にこのプロトコルの考察とまとめを行なう。

## 背景

### QoS 保証型、QoS 非保証型

一般にネットワークでは、QoS 保証型と QoS 非保証型の 2 つのサービスに分けられる。QoS 保証型のネットワークの場合、QoS を保証するには、次の手続きを取る必要がある。まず、接続前にエンドノードから要求すべき QoS パラメータをネットワークに渡す。ネットワークでは、要求された QoS パラメータを保証することができるかどうかを評価する。そして、その評価結果をネットワークからエンドノードへ返す。その結果、コネクション中は交渉結果の QoS パラメータが保証されることになる。このような QoS 保証型ネットワークの代表例として、ATM(Asynchronous Transfer Mode)ネットワークがある。ATM では、UNI(User-Network Interface)のシグナリングによってエンドノードとネットワークの間で QoS パラメータを交換することができる [5]。しかし、QoS 保証型のネットワークは、ネットワークの構造が複雑となり、また現状において広く一般的に利用されていないという問題点がある。

一方、QoS のパラメータをエンドノードからネットワークに対して渡すことができないネットワークがある。このネットワークを QoS 非保証型ネットワークと呼ぶ。この QoS 非保証型ネットワークの場合、ネットワークのサービスとして QoS を保証する機構がないために、実時間でマルチメディアデータの転送サービスを保証することが難しい。途中ノードの間の混雑のためにネットワークの状態が変化して、遅延や損失がネットワーク内に起こると、アプリケーション間において転送しているデータが、その遅延や損失の影響を受け、結果として実時間の制約を守ることができなくなる。インターネットは、この QoS 非保証型ネットワークの代表的な例である。インターネットにおいて使用している IP プロトコルの場合、中間にあるルーターでは、パケットの宛先アドレスとルーターがもっている経路情報によって、次のルーターに単純に転送しているだけで、目的ホストであるエンドノードへの QoS やパケットの到達性は、全く保証していない。そのため、上位の TCP のようなトランスポートプロトコルで到達しなかったパケットを再送することによりデータの到達の信頼性を保証している。しかし、TCP では実時間の制約を保証することができない。

我々の以前の実験では、受信側からのフィードバック情報により動的な QoS 制御をおこなうことによってビデオデータの配送が、適切な品質で転送できることを報告した [10]。また、Kanakia らは、実験によってフィードバックされる情報に遅延があったとしてもフィードバック制御を行なわない方式に比べてフィードバック制御を行なう方式が、よい結果を示していることを報告している [9]。

本研究では、次のような方式を使って QoS 非保証型

ネットワークの上で QoS を保証するための方式を提案する。まず、QoS 非保証型のネットワークにおいて、それぞれのエンドノードでネットワークの QoS をモニターするための機能をプロトコルモジュールに入れる。そのモニター機能は、エンドノードにおいて、QoS 非保証型のネットワークの QoS パラメータを実時間でモニターしている。そして、モニターしたネットワークの変化を QoS パラメータとして知らせる機能を持つ。ネットワークが変化すると両端のエンドノードにおいて、その変化を検知する。そして受信側で検知した変化を随時、送信側にフィードバックする。また送信側では、受信側からの情報によって送出するデータ量を動的に変更してゆく。

このように受信側から送信側へのフィードバック制御を行なって、ネットワークへの送出データを動的に変更することによって、たとえネットワークの QoS が変化して、ネットワークの中において一時的に低いサービス提供できなくなったとしても、エンドノードでは、送出データ量を変更しながら、その状況に追従して転送の保証を行なうことができる。その結果、ネットワークの状況が悪くなった時でも実時間でマルチメディアデータを転送し続け、またアプリケーションにおいては、マルチメディアデータを実時間で表現することができる。

#### 実時間が保証されない理由

一般にインターネットのようなパケット交換ネットワークのエンドノードにデータの実時間で到達性が保証されない原因について次のように考えることができる。

- 混雑がある  
途中のノードが混雑しているためにデータの転送のバンド幅が足りない。一時的に途中の経路のバンド幅が足りなくなるため、データを時間内に転送することができなくなる。
- パケットの順番が変わってしまう。  
伝送の途中でパケットの順番が変わってしまう。原因は、途中のノードにおいて順序変更があったことが考えられる。パケットの順番の変更があると TCP のような単純なウィンドウ制御型のトランスポートストリームの場合、後から到着した順番が異なっているパケットまでのすべてのパケットを再送する必要がある。そのため、実時間制御に影響がでる。
- 損失がある。  
インターネットでは、通常、約 2% から約 10% の損失が起こることがわかっている [4]。また我々が行なった実験においても約 2% から約 7% ぐらいの損失が常にあった。この損失の原因は、伝搬中のデータ損失及び途中ノードでのオーバーフローである。Bolot[12] は、インターネットの遅延と損失を UDP echo ツールを使って測定し、その解析を行なった。その解析の結果、(1) パケットがランダム

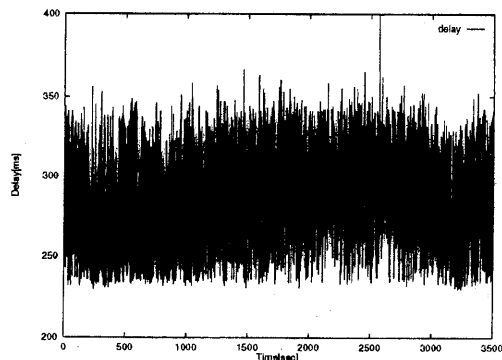


図 1: インターネットにおける遅延

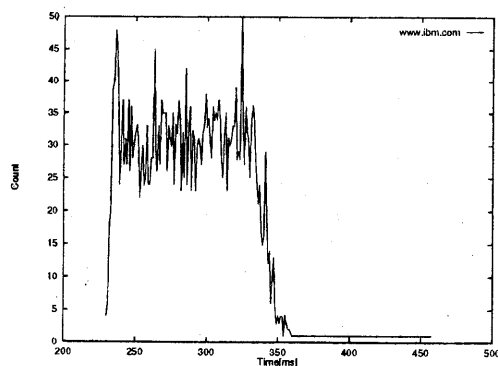


図 2: インターネットにおける遅延の頻度に損失していること、(2) またモデルとして、単純な一つのキューイングサーバーに対して検査の検査するためのトラフィックとインターネットのトラフィックが入力されるものが、うまく適用できることを報告している。

- 遅延がある。  
インターネットでは、遅延がある。これは、途中のノードにおいてバッファとパケットの転送処理の遅延の加算が End-to-end において現れるためである。その遅延は、途中のルータの混雑によって刻々と変化する。そのため、遅延は、時間の分布となる。また、ネットワークにおいて損失が起こると信頼性のあるストリームのプロトコルでは、再送により損失したデータを回復する。その結果、アプリケーションでは遅延が起こる。実験として、ホスト orbiter.imnet.tri.ibm.co.jp から www.ibm.com へ ICMP エコーを 1 秒おきに出して、その RTT(Round Trip Time) を測定したグラフを図 1 に示す。この実験の最大遅延は 457ms であり、また最小遅延は 230ms であった。また平均値は、287.6(ms) であった。図 2 は、図 1 の頻度を表にしている。

## どこが QoS の選択権を持つか

Patridge[1] は、インターネットのドラフトの RFC1363 においてインターネットに特別なサービスを要求するためのフロースペックと呼ばれるデータ構造について提案した。このデータ構造は、クライアントやサーバーのアプリケーションからネットワークに対して特別なタイプのトラフィックを流すときに、ネットワークの制御を行なうためのデータとして使用する QoS パラメータについて述べている。一般にクライアント・サーバー型を構成する場合、サーバー、ネットワーク及びクライアントの3箇所において QoS パラメータを決めることができる。本稿では、その3箇所において QoS の選択を行なうための要件について述べる。

### ・サーバー

例として、すでに圧縮された画像データを考える。圧縮された画像データが、すでに入っている場合には、そのバンド幅や表示時間データが決まっており、圧縮されたデータがディスクに入っている。そのため、QoS のパラメータは、サーバーからネットワークとクライアントに要求される。

### ・ネットワーク

QoS 非保証型ネットワークの場合、遅延や損失のパラメータ、バンド幅のような QoS パラメータがネットワークの状態が変化することにより動的に変化してゆく。そのため、QoS は、時間によって変化しているパラメータとしてサーバー及びクライアントに渡される。

### ・クライアント

マルチメディアデータのクライアントの場合、QoS パラメータは、ユーザーや使用しているアプリケーションにより決まってくる。例えば、ビデオの画質は、ビデオを見ているユーザーの知覚に依存している。アプリケーションにおいて使用可能な資源に関しても QoS パラメータとなる。

QoS 非保証型のネットワークでは、ネットワークの中で QoS が保証されないために、サーバーやクライアントからネットワークに対して要求される QoS パラメータをネットワークにおいて受けつけることができない。サーバーとクライアントで要求している QoS パラメータは、ネットワークの状態によって常に変化してしまう。つまりクライアントとサーバーでは、アプリケーションのサービスの質を守るためにネットワークの変化に対して追従していく必要がある。

## Hop-by-hop と End-to-end

ネットワークの QoS 制御を行なう方式として、End-to-end と Hop-by-hop の2つがある。End-to-end 方式の場合、ネットワークにおける途中経路の状況は、すべてエンドノードにおいて取得される。そのためネットワークの途中経路において損失や遅延が起こったときに、エンドノードでは、その状態の詳細な情報がわからない。つまり、エンドノードでは、途中経路のネットワークの状況が縮退して伝わってしまう。そのため、エンドノ

ードでネットワークの状況を測定すると位置のパラメータを取得できない。しかし、End-to-end 方式は、インターネットのような既存の packets 交換ネットワークに対して、すぐに適用できるため、広範囲の利用が可能となる。

一方、Hop-by-hop 方式の場合、途中経路のノードごとに QoS を保証するためのモジュールを入れることになる。途中経路のノードごとに Hop-by-hop の QoS を保証すれば、結果として、エンドノードに対して QoS の保証をすることになる。この方式では、途中のノードのすべてに対して QoS をサポートするための機構を組み込む必要がある。TCP/IP プロトコルと並立させて実時間ストリームをサポートするために TENET では、ネットワーク層とトランスポート層を提案及び実装し、実験結果を報告している [7]。また、Hop-by-hop 方式でレート制御を使った混雑の制御は、End-to-end 方式に比べてパフォーマンスがよいことが報告されている [8]。さらに広域ネットワークにおいて、Hop-by-hop 方式による途中ノードのスケジューリングポリシーに関して、HL(Hop-Laxity) スケジューリングが、よい結果を示すことが報告されている [11]。

## メディアスケーリング技術

マルチメディアデータを実時間で動的に QoS が変化しているネットワークにおいて転送する場合にメディアスケーリングの技術と QoS 技術との連携が必要になってくる。例えば、ネットワークにおいて、データの転送が可能なバンド幅が一時的に減少してくれば、送出するときにビデオデータのフレームを一時的に落して、送出するデータのバンド幅を一時的に低下させる。さらにネットワークにおいて転送可能なバンド幅が低下してくれば、ビデオデータの転送を一時中断して音声データだけを転送する。また、ネットワークのバンド幅が、もとに戻れば、転送するデータの送出量を追従して増やしてゆく。このように使用可能なバンド幅に追従してメディアスケーリングを行なって追従していけば、バンド幅が現象したとしても実時間でビデオ・オーディオデータを見ることができる。

一般に動画のメディアスケーリング技術では、空間解像度、色相解像度、時間解像度の3軸がスケーリングの対象となる。これらの解像度は、お互いに独立に変更することが可能であり、3次元空間の点として表現される。このメディアスケーリングの3次元空間の点の移動は、ユーザーからの音声と画像の認知度から決定されるサービスの質に依存することになる [10]。

## プロトコル

本稿で述べるトランスポートプロトコルは、サーバに蓄積されたマルチメディアデータをクライアント側へ実時間で転送することが目的である。そのため図3のようにプロトコルは、サーバからクライアントへの非対称型のトランスポートプロトコルになる。また、逆にクライアントからサーバーに対しては、単にプロトコルの制御データだけが転送される。つまり、転送されるデータ量

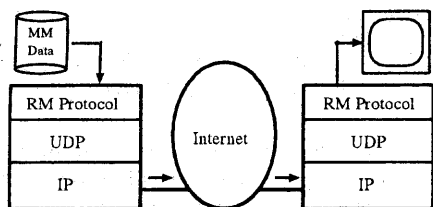


図 3: プロトコルの構成  
の観点からみると、このプロトコルは非対称になる。

以下にプロトコルの目的を上げておく。

- エンドノードにおいて損失や遅延等の QoS 非保証型ネットワークの状態を検出する。
- 再送機構を使ってネットワークの状況に応じて、実時間で動的に転送量を制御する。
- リアルタイムでマルチメディアデータの転送ができるようにエンドノードにおいて実時間の制御を行なう。
- End-to-end で制御を行なう。
- 送信パケット及び受信パケットにおいて優先順位の制御を行なう。
- 損失した場合の再送方式がデータの内容によって異なる。
- 実時間制御のために Deadline を最優先する。

#### プロトコルの特徴

1. UDP を使ったトランスポートプロトコル  
TCP では、実時間の配送をすることができない。また、UDP では損失しても再送を行わないために信頼性がない。そのために、実時間と信頼性の両方をサポートするプロトコルが必要となる。このプロトコルは、UDP を利用してユーザー空間に実装される。
2. エラー制御  
このプロトコルは、UDP の上に実装するのでエラー制御の機能を必要としない。もしデータにエラーがあれば、UDP のインターフェイスからデータを取得できないためである。一般にマルチメディアデータの場合、送信したデータが間違っていたとしてもデータを捨てずに上位層に、その間違っただけのまま転送するオプションがあってもよい。この機能により上位層では、間違っただけのままアプリケーションに渡すことができる。
3. 損失制御  
それぞれのパケットヘッダーについているシーケンス番号により損失制御を行なう。次に受信できるパケットのシーケンス番号を予測しておき、異なったシーケンス番号を受信した場合、損失したとみなして、損失したデータをサーバーから再送する。これとは別にデータの内容ごとに再送基準を設けて、その基準に従って、再送がどの程度必要であるかを決めてゆく。つまり、まったく再送の必要がないデータから、必ず再送しなければならないデー

タまでのクラスがある。

#### 4. 優先順位の制御

配送モジュールでは、データを転送する場合に、ビデオ、オーディオ及びテキスト等のデータの内容によって送信時に優先順位をつける。例えば、音声とビデオのデータの場合、音声は、損失しないように必ず届くようにするようにする。また、ビデオデータは、損失しても再送せずに、そのまましておく。次のようにデータの内容によって優先順位を決める属性を持つようにする。

- (a) 時間を優先すべきデータ、再送をして信頼性を維持する
- (b) 時間を優先すべきデータ、再送を行わない
- (c) 時間を優先していないデータ、再送して信頼性を維持する
- (d) 時間を優先していないデータ、再送を行わない

(例えば、オーディオは、(a)、テキストは、(c)、ビデオは、(d)になる。)

#### 5. 再送基準

データパケットの属性に信頼性の項目を設けて、その属性によって再送する基準を決めてゆく。次の項目が再送基準である。

- (a) 何度でも届くまで再送する。
- (b) x 回再送する。届かなければやめる。
- (c) 再送しない。
- (d) 同じデータを必ず y 回送る。

またこの項目に加えて、プロトコルに時間の項目を設けて「デッドラインまでは再送を繰り返して、デッドラインを過ぎると再送をやめる。」という方式をとる。

#### 6. 混雑制御を行なう

RTT により単純な混雑制御を行なう。

- RTT が大きくなってゆく。  
ネットワークが混雑している。  
ネットワークへの送出量を少なくしてゆく。
- RTT が小さくなってゆく。  
ネットワークが空いてきている。  
ネットワークへの送出量を多くする

また、送出量の増減の変化量は変数となる。つまり、増減する割合は、固定値ではなく随時変化させる。

#### 7. 流量制御を行なう

基本的には、レート制御とウィンドウ制御を状況に応じて使い分ける。さらに明示的な Start/Stop コマンドによって流量の制御を行なう。

- サーバにおいてクライアントからフィードバックがない場合、サーバの送信は、実時間の制約によって自走する。

- バンド幅に余裕があって早く送ってしまうことができるならば、先に送る。この場合、レート制御ではなく、受信バッファの量に依存するウィンドウ制御になる。(できる限り早く送ることになる。)
- バンド幅が足りないか、あるいは表示時間から計算した時間になったならば、レート制御によってデータの転送を行なう。(ここまで遅くてもよいという時間である。)

#### 8. デッドライン制御

実時間転送に必要とされる機能である。現在の時間、転送時間とバッファの時間の和が、クライアントにおける表示時間が過ぎている場合には、送出してもアプリケーションにとってデータの意味がないので、そのデータを捨ててよい。

パケットが表示時間よりも早く到着すると、そのパケットをバッファに溜めることができるのでバッファ量の時間分は、事前にサーバーからクライアントへ送ることができる。

送出すべき時間 (td) : 絶対時間

転送時間 (tt) : 相対時間

表示時間 (tp) : 絶対時間

バッファ量の時間 (tb) : 相対時間

td は、 $td < tp - tt - tb$  とならなければならない。この場合、 $td = tp - tt - tb$  が、デッドラインタイムとなり、td に間に合うように送信側からは送り出さなければならない。

また送信側では、独自にレート制御によって走る場合に、どこまでも自分で勝手に走るわけではなく、先送りの条件として、無限の先まで送ることができるわけではなく、受信側のバッファ量が一杯になったところで止まる。

#### 9. バンド幅の算出

バンド幅は、X 秒の間にデータ転送量 Y を計測し、そのデータからバンド幅 (bytes/s) = Y (bytes) / X 秒を算出する。

#### 10. 交換される情報について

クライアントとサーバーで転送されている情報

- シーケンス番号
- ACK/NACK 情報
- 時間情報 (RTT 等)
- バッファの残量
- データ消費量
- ネットワークでの損失率

#### 結論

本稿では、インターネットのような QoS 非保証型ネットワークにおいてオーディオ及びビデオを実時間で配送するための問題点を分析し、その対処方法について述べた。そして、その方法をもとにして新しいプロトコルの設計を行なった。今後、このプロトコルの実装を行ない、その性能評価を行なう予定である。

#### 参考文献

- [1] C. Partridge, A Proposed Flow Specification, *Network Working Group Request for Comments: 1363, 1992.*
- [2] D. D. Clark, S. Shenker and L. Zhang, Supporting Real-Time Applications in an Integrated Services Packet Network: Architecture and Mechanism, *In Proc. ACM SIGCOMM 92, 1992.*
- [3] L. Zhang, Virtual Clock: A New Traffic Control Algorithm for Packet Switching Networks, *In Proc. ACM SIGCOMM 90, 1990.*
- [4] D. Sanghi, A. K. Agrawala, O. Gudmundsson, B. N. Jain, Experimental Assessment of End-to-end Behavior on Internet, *In Proc. INFOCOM 1993, 1993.*
- [5] The ATM Forum, ATM User-Network Interface Specification, *Prentice Hall 1993*
- [6] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, RTP: A Transport Protocol for Real-Time Application *Audio-Video Transport Working Group Request for Comments: 1889, 1996.*
- [7] A. Banerjee, D. Ferrari, B. A. Mah, M. Morgan, D. C. Verma, and H. Zhang, The Tenet Real-Time Protocol Suite: Design, Implementation, and Experiences *IEEE/ACM Transaction on networking, vol.4, No.1, February 1996.*
- [8] P. P. Mishra and H. Kanakia, A Hop by Hop Rate-based Congestion Control Scheme, *In Proc. ACM SIGCOMM 92, 1992.*
- [9] H. Kanakia, P. Mishra and A. Reibman, An Adaptive Congestion Control Scheme for Real Time Packet Video Transport, *IEEE/ACM Transaction on networking, vol.3, No.6, December 1995.*
- [10] 山内長承, 河内谷清久仁, 串田高幸, インターネット上の動画転送を意識した動的 QoS の制御, 情報処理学会マルチメディア通信と分散処理研究会, *96-DPS-75, 1996.*
- [11] H. Schulzrinne, J. Kurose and D. Towsley, An Evaluation of Scheduling Mechanisms for Providing Best-Effort, Real-Time Communication in Wide-Area Networks, *In Proc. INFOCOM 1994, 1994.*
- [12] J-C. Bolot, End-to-end Packet Delay and Loss Behavior in the Internet, *In Proc. ACM SIGCOMM 93, 1993.*