

# 実時間動画像マルチキャストのための アクティブ動画像品質調整ノードの実装と評価

山田 達也<sup>†</sup> 若宮 直紀<sup>††</sup> 村田 正幸<sup>††</sup> 宮原 秀夫<sup>††</sup>

<sup>†</sup> 大阪大学 大学院基礎工学研究科 〒560-8531 大阪府豊中市待兼山町1-3

<sup>††</sup> 大阪大学 大学院情報科学研究科 〒560-8531 大阪府豊中市待兼山町1-3

E-mail: ††yamada@ics.es.osaka-u.ac.jp, ††{wakamiya,murata,miyahara}@ist.osaka-u.ac.jp

あらまし 動画像マルチキャストにおいて、それぞれのユーザの通信環境、システム性能、再生動画像に対する好みなどに応じた適切な動画像データを配信するためには、アクティブネットワーク技術を利用して、ネットワーク内で動画像の品質調整を行うのが効果的である。本報告では、利用者の人数や位置、ネットワークの状態によって変化する品質要求に対応するため、ネットワーク内で動画像レートを動的に変更できる動画像品質調整ノードについて検討、設計し、ネットワークプロセッサ上に実装した。実動画像データを用いた通信実験の結果、動画像品質調整ノードにおいて、動的かつ適切なレート制御が実現可能であることを示している。

キーワード 動画像品質調整, ネットワークプロセッサ, アクティブネットワーク, 動画像マルチキャスト

## Design, Implementation, and Evaluation of Active Video-Quality Adjustment Method for Heterogeneous Video Multicast

Tatsuya YAMADA<sup>†</sup>, Naoki WAKAMIYA<sup>††</sup>, Masayuki MURATA<sup>††</sup>, and Hideo MIYAHARA<sup>††</sup>

<sup>†</sup> Graduate School of Engineering Science, Osaka University

<sup>††</sup> Graduate School of Information Science and Technology, Osaka University

1-3 Machikaneyama, Toyonaka, Osaka 560-8531, Japan

E-mail: ††yamada@ics.es.osaka-u.ac.jp, ††{wakamiya,murata,miyahara}@ist.osaka-u.ac.jp

**Abstract** By introducing video-quality adaptation mechanisms into intermediate network equipments using active network technologies, we can provide users with video distribution services taking into account client-to-client heterogeneity in terms of available bandwidth, performance of client systems, and user's preferences about video quality. In this paper, we designed and implemented an active video-quality adjustment node on a network processor-based network equipment. The active video-quality adjustment node can adapt the video rate to the requested level on a packet-by-packet basis. The target rate is specified in an ANEP header of a video packet. We conducted experiments and verified the practicality of the video-quality adjustment within a network.

**Key words** video quality adjustment, network processor, active network, heterogeneous video multicast

### 1. はじめに

動画像配信サービスにおいて、動画像サーバから複数のユーザに対して動画像を配信する場合には、ユーザごとに異なるアクセス回線容量やネットワーク内での利用可能帯域、クライアントマシンの性能、さらには、再生動画像の品質に対するユーザの好みなどについて考慮し、それぞれのユーザにあった適切な品質、符号化レートの動画像を適切な転送レートで提供しなければならない。我々の研究グループでは、データ転送遅延に関する厳しい制約のもとで多数のユーザに同時に動画像を配信

しなければならない動画像マルチキャスト通信について、ネットワークに大きな負荷を与えることなくさまざまな品質要求を同時に満足することができる制御手法の研究を行ってきた。特に文献 [1] では、新しい通信アーキテクチャであるアクティブネットワーク技術を用いた効率のよい動画像マルチキャスト通信を行う手法を提案している。

アクティブネットワークは、アクティブルータあるいはアクティブノードと呼ばれる高機能ノードをネットワーク内に配置することにより、アプリケーションやユーザが必要とするネットワークサービスを柔軟かつ容易に開発、導入することがで

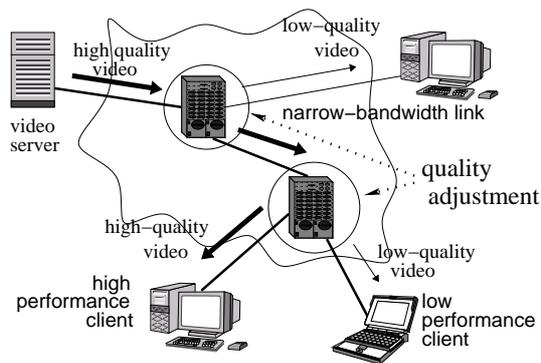


図 1 異種クライアントへの実時間動画画像マルチキャスト

きる通信アーキテクチャである [2, 3]。アクティブノードでは、パケットに対し、パケットに付加された、あるいはあらかじめノードに送り込まれたプログラムコードを適用し、ユーザやネットワーク管理者、アプリケーションの望むパケット処理を施すことができる。また、アクティブノードでは、経路制御など下位レイヤのパケット処理にとどまらず、ペイロードに含まれるアプリケーションデータに対して処理を施すこともできる。文献 [1] における提案手法では、図 1 に示すようにアクティブノードに動画品質調整機能を導入することにより、下流ノードやクライアントの要求に応じて品質調整、レート制御を行い、効率的かつ効果的な動画マルチキャスト通信を実現する。

我々の研究グループでは、さまざまな品質要求を同時に満たすことのできる実時間動画マルチキャストのためのいくつかの動画品質調整手法を提案、比較評価し、動画データの周波数成分を除去することによりデータ量やレート、動画品質を調整するローパスフィルタが、レート調整能力や動画品質劣化の抑制などの点において最も効果的であることを示した [4]。また、パケット処理に特化したアーキテクチャを有し、プログラマブルなネットワークシステムを構成可能な市販のネットワークプロセッサ上にローパスフィルタを実装し、実動画データを用いて通信実験を行うことで、パケットを単位とした動画の品質調整が可能であることを示した [5]。しかしながら、文献 [5] で実現した動画品質調整ノードはあらかじめプログラムされたレートでの品質調整しか行えず、利用者の人数や位置、ネットワークの状態の変化に応じて品質調整の度合いを制御することができない。また、ノードの有する複数のプロセッサの機能配置についても十分な検討を行っていない。

そこで本稿では、アクティブネットワーク技術により動的な品質制御が可能な動画品質調整ノードを設計、実装し、その実用性、実現性を評価する。設計に際しては、アクティブ動画品質調整ノードを構成する 4 つの機能層、すなわち、Active Application 層、Execution Environment 層、NodeOS 層、およびパケットの分類と転送処理のためのルータ層の、ノードの有するプロセッサへの配置について検討する。

以下に本稿の構成を示す。2. では、MPEG-2 符号化手法の概要と本稿で用いた MPEG-2 動画品質調整手法について述べる。3. では、アクティブノードのアーキテクチャとアクティブ動画品質調整ノードの機能設計について述べる。4. では、実装に用いた Intel IXP1200 ネットワークプロセッサの概要を

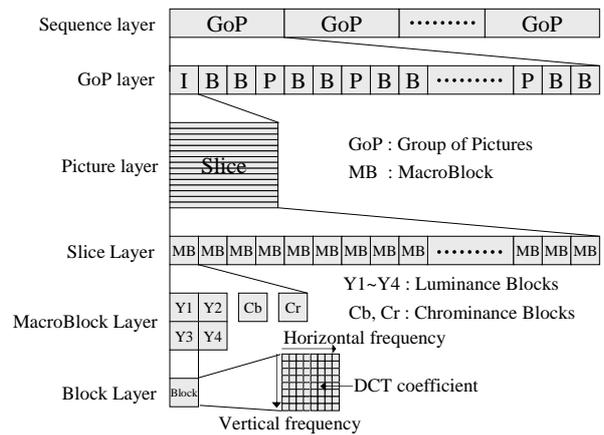


図 2 MPEG-2 動画画像の構造

述べて、ネットワークプロセッサへのアクティブ動画品質調整機能の実装方法について検討する。5. では、実験にもとづく実現性、実用性の検証を行い、最後に 6. では、まとめと今後の課題を述べる。

## 2. MPEG-2 動画品質調整

本稿では、動画の符号化圧縮方式として MPEG-2 を用いる。MPEG-2 は図 2 に示す階層構造となっており、最上位はシーケンス層と呼ばれる。シーケンスは複数のピクチャの集合である GoP (Group of Pictures) の繰り返しからなり、GoP はそれぞれ符号化アルゴリズムの異なる 3 種類のピクチャ (I, P, B) の組み合わせにより構成されている。ピクチャは高さ 16 画素の帯状の領域に分割され、これをスライスと呼ぶ。スライスは 1 個以上のマクロブロックから構成される。マクロブロックは  $16 \times 16$  画素の領域に対応し、それぞれ  $8 \times 8$  画素の 4 つの輝度ブロック (Y) と、2 つの色差ブロック (Cb, Cr) の 6 つのブロックから構成される。ブロックは離散コサイン変換 (DCT) により周波数成分に変換された後、低周波数の DCT 係数から順に並べられる。

ローパスフィルタは、目標とする符号化レートに応じてブロックを構成する 64 個の DCT 係数を高周波のものから順次削除する品質調整、レート制御手法である。

## 3. アクティブ動画品質調整ノードの機能設計

アクティブ動画品質調整ノードは、図 3 に示すようにパケットの分類のためのルータ機能と動画品質調整のためのアクティブノード機能の階層構造となっている [6]。さらに、アクティブノード層は NodeOS, EE (Execution Environment), AA (Active Application) の 3 つの機能層からなる [3]。

### 3.1 Active application 層

AA は、ユーザやネットワーク管理者から与えられ、パケットに適用されるプログラムであり、処理対象のパケット自身に付加したり、他のアクティブノードやサーバからダウンロードするなどして動的に注入することができる。アクティブ動画品質調整ノードでは動画品質調整プログラムを AA として実装する。なお、AA はマルチキャストグループ別に起動される。

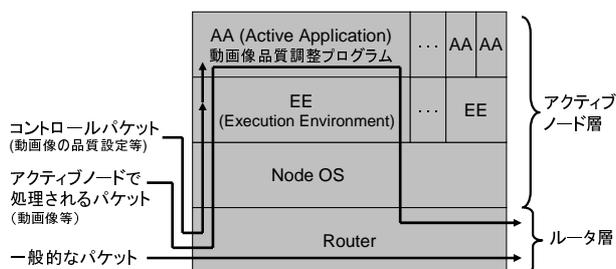


図 3 アクティブ動画品質調整ノードの機能構成

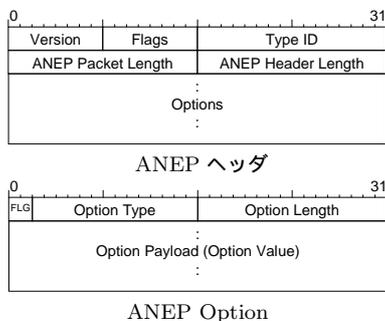


図 4 ANEP ヘッダフォーマット

### 3.2 Execution environment 層

EE はプログラムを実行するための実行環境、あるいは仮想マシンであり、プログラミングインタフェースを提供する。AA の起動、実行、終了などの操作および AA が利用可能な資源を管理することにより、AA が実行可能な処理を規定する。EE により、NodeOS は AA から隠蔽される。

### 3.3 NodeOS 層

ネットワークノードの OS である NodeOS は、プロセッサ、チャネルなどノード資源のスケジューリングを行う。パケットヘッダに含まれるプロトコル番号、送受信 IP アドレス・ポート番号対や ANEP (Active Network Encapsulation Protocol) ヘッダの Type ID にもとづいて、到着したパケットを適切な EE に配送する。また、EE からノード外に送信されるパケットをルータ層に配送する。

### 3.4 ANEP

ANEP (Active Network Encapsulation Protocol) [7] は、アクティブノード間の通信プロトコルである。ANEP ヘッダを付加されたユーザデータは、IANA (Internet Assigned Number Authority) によってアクティブネットワークに割り当てられたプロトコル番号 107 を用いて IP パケットにカプセル化される。ANEP ヘッダのフォーマットを図 4 に示す。

NodeOS は、ANEP ヘッダの Type ID によって指示された EE にパケットを配送する。Type ID は、MIT の ANTS [8] に対して 19、コロンビア大学の NetScript [9] に対して 31-35 のように、組織名および使用目的別に、ANANA (Active Networks Assigned Numbers Authority) によって割り当てられ、Type ID Registry [10] として公開されている。

ANEP Option の最初の 1 ビットは、オプションが ANANA が指定するグローバルフォーマットに従うか、個々の EE が独自に決定できるプライベートフォーマットであることを示す。プライベートオプションについては NodeOS は解析しない。Option Payload のフォーマットは Option Type によって指定される。

### 3.5 ルータ層

ルータ層は、アクティブノード層で処理するパケットの選別機能と、入力ポートおよびアクティブノード層から受信したパケットを転送処理するための基本的なルータ機能を有する。動画パケットに ANEP ヘッダを付加した場合には、ルータ層では IP ヘッダ内のプロトコル番号によりパケットを分類することができる。この手法は、パケット分類処理の負荷が小さいが、動画サーバ、クライアントへの ANEP 処理機能の導入、あるいは動画サーバ、クライアント近傍のアクティブノードでの ANEP ヘッダの付加、除去が必要となる。一方、ANEP を用いない場合には、ルータ層では送受信 IP アドレス・ポート番号によってパケットを選別しなければならず、そのためのテーブルの参照、管理が必要となる。本稿では、ノード実装の容易さから ANEP ヘッダを用いたパケット分類を採用する。

### 3.6 シグナリングプロトコルの設計

動画品質調整 AA の起動、終了、および目標レートの指定・変更は ANEP Option のプライベートフォーマットを用いて行う。起動、終了、レート指定には Option Type として 1, 0, 2 を割り当てる。Option Payload には動画データ受信元および送信先のマルチキャストグループアドレス、ポート番号と目標レートを記述する。同一の動画から異なる品質の動画を同時に複数生成する場合には、送信先マルチキャストグループアドレスとポート番号、目標レートを変えた ANEP Option を必要数付加すればよい。EE は指定されたマルチキャストグループのための AA が存在しない場合にはこれを起動し、パラメータとして目標レートを渡す。以降、EE は NodeOS から受け取った動画パケットをマルチキャストグループアドレスとポート番号にもとづいて適切な AA に渡す。

## 4. アクティブ動画品質調整ノードの実装

### 4.1 Intel IXP1200 ネットワークプロセッサの概要

IXP1200 ネットワークプロセッサ [11-13] は、232 MHz で動作する Intel StrongARM プロセッサコアと、同じく 232 MHz で動作する 6 つのマイクロエンジンからなる (図 5 太枠内)。StrongARM では機能制限された組み込み用 linux が動作する。マイクロエンジンはパケット転送などの単純な処理を対象としたシンプルな RISC プロセッサであるため、レジスタ数や命令行数などの制限が大きい。4 つのスレッドを並列実行可能である。したがって、6 つのマイクロエンジンで合計 24 個のスレッドを動作させることが出来る。実験に用いたボードには 4 つの全二重 10/100Base-TX イーサネットポートと 2 MB の SRAM, 102 MB の SDRAM が搭載されている。それぞれのマイクロエンジンは IX バス経由で全二重イーサネットポートに接続されている。マイクロエンジンと StrongARM プロセッサコアは SDRAM 内共有メモリ, SRAM, またはスクラッチパッドメモリを介してデータの受け渡しが可能である。より高度な処理を行うため IXP1200 をコンピュータに接続した場合には、PCI バスを經由して StrongARM プロセッサコアと Pentium プロセッサとのデータのやり取りが可能である。

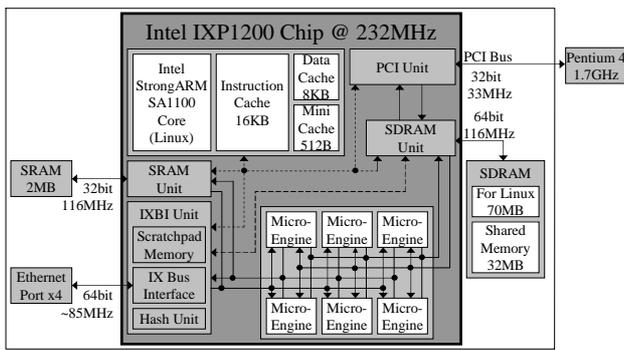


図 5 IXP1200 の構成

#### 4.2 アクティブ動画画像品質調整ノードの機能配置

マイクロエンジン, StrongARM プロセッサコア, Pentium プロセッサに, アクティブ動画画像品質調整ノードの 4 つの機能, すなわち, ルータ, NodeOS, EE, AA を配置する.

ルータ層は, ルーティング, パケットフォワーディング用に開発された RISC プロセッサであるマイクロエンジンに実装する. パケット送信処理とパケット受信処理にそれぞれ 1 マイクロエンジン, 4 スレッドずつ割り当てた. 各スレッドは, 1 つのイーサネットポートの入力, または出力を担当する. パケットの送受信マイクロエンジン間, および, ルータ層・アクティブノード層間のパケットの受け渡しは, SDRAM 内共有メモリにパケットを, SRAM 上にパケット管理キューを置き, パケットの SDRAM アドレスを通知することで実現する.

NodeOS, EE, AA は, 4 つのマイクロエンジン, StrongARM プロセッサコア, および IXP1200 と PCI バス経由で接続された Pentium プロセッサに配置される. 最も処理負荷の大きい動画画像品質調整機能について, StrongARM のみを用いた場合, StrongARM とマイクロエンジンを用いた場合, および Pentium プロセッサのみを用いた場合について実装した. なお, 動画画像品質調整のための AA はあらかじめアクティブ動画画像品質調整ノードに導入するものとする.

##### 4.2.1 StrongARM のみを用いた品質調整

StrongARM 上の linux に NodeOS, EE, AA を実装する. NodeOS は, linux のカーネル機能およびドライバを用いて SRAM のパケット管理キューおよび SDRAM 内共有メモリにアクセスし, マイクロエンジンに実装されたルータ層とパケットの受け渡しを行う. NodeOS・EE 間と EE・AA 間のパケットの受け渡しにはローカルアドレスを用いたソケット通信を利用する.

##### 4.2.2 StrongARM とマイクロエンジンを用いた品質調整

文献 [5] と同様, 動画画像品質調整のうち単純かつ処理量が多い DCT 係数の演算処理をマイクロエンジンに割り当てる. 動画画像品質調整の残りの処理と NodeOS, EE を StrongARM に実装する. 動画画像品質調整のための StrongARM とマイクロエンジンとの通信にはスクラッチパッドメモリと SDRAM 内共有メモリを利用する. まず, StrongARM はマイクロエンジンによる演算が必要な DCT 係数を SDRAM 内共有メモリに置き, そのアドレスをスクラッチパッドに書き込む. マイクロエンジンのスレッドはスクラッチパッドのそれぞれ定められたア

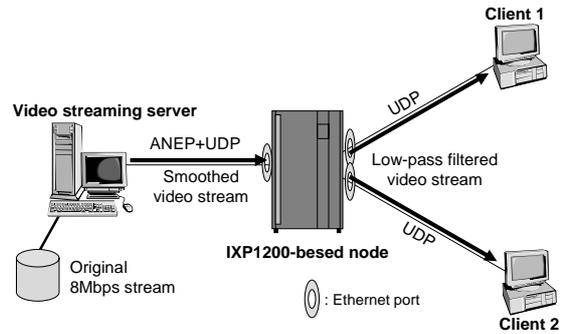


図 6 実験システム 1

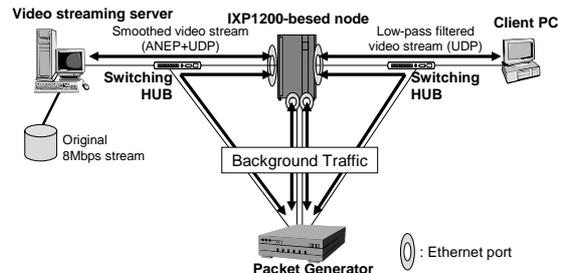


図 7 実験システム 2

ドレスを定期的に確認し, SDRAM アドレスが書き込まれていれば SDRAM 内共有メモリから演算データを, また, スクラッチパッド内の別の領域から量子化テーブルを読み出し, 演算処理する. 演算結果は同一アドレスに書き込まれ, スクラッチパッド内の割り当てられたアドレスを用いて StrongARM に処理完了を通知する. StrongARM もマイクロエンジンと同様にスクラッチパッドを定期的に確認し, 演算済の DCT 係数を SDRAM から読み出す.

##### 4.2.3 Pentium プロセッサを用いた品質調整

IXP1200 に PCI バス経由で接続された Pentium プロセッサに NodeOS, EE, AA を実装し動画画像品質調整を行う. マイクロエンジン上にのルータ層と Pentium プロセッサ上の NodeOS 層との通信を可能にするため, StrongARM にパケットを中継するプログラムを実装した. プログラムは, linux のカーネル機能およびドライバにより SRAM のパケット管理キューおよび SDRAM 内共有メモリを通じてルータ層とパケットをやり取りし, PCI バス上に実現された仮想 IP ネットワークにより, NodeOS 層とソケット通信を行う. また, NodeOS・EE 間と EE・AA 間のパケット配送にはローカルアドレスを用いたソケット通信を利用した.

## 5. 実験と検証

図 6 に示す 1 台の動画サーバと 2 台のクライアントおよび 1 台のアクティブ動画画像品質調整ノードからなる実験システム 1 において, IP パケット化された MPEG-2 動画データの品質調整を行った. 本実験では, 動画画像品質調整によるレート制御の精度, 動画画像品質調整のスループットについて評価し実用性を検証する. また, 図 7 に示す実験システム 2 では, パケットジェネレータを用いて動画画像品質調整を必要としないバックグラウンドトラフィックを生成し, 動画画像品質調整とパケット転送処理が互いに与える影響について検証する. パケットジェネ

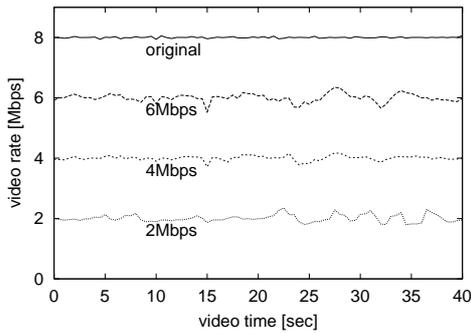


図 8 目標レート固定時の品質調整

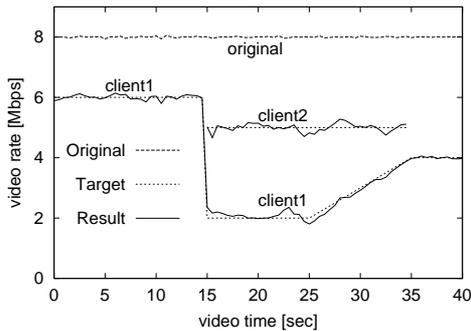


図 9 目標レート可変時の品質調整

レータの有する 4 つのイーサネットポートのうち、2 ポートはアクティブ動画品質調整ノードに直接接続し、残りの 2 ポートは各々スイッチングハブを経由してアクティブ動画品質調整ノードに接続する。2 台のスイッチングハブには、動画サーバ、またはクライアントを接続し、同一イーサネットポートにバックグラウンドトラフィックと動画データが入力または出力されるようにした。

MPEG-2 動画データは、8 Mbps の固定ビットレートであり、720 × 480 画素、30 fps の動画を CBR 符号化し、192 Kbps の音声データとあわせて Program Stream フォーマットで多重化している。GoP の長さは 0.5 秒 (15 フレーム) とし、GoP 単位で目標レートの変更が行える。なお、MPEG-2 動画データは、スライス層ごとに IP パケット化する [5]。動画サーバは、動画パケットを UDP、ANEP、IP の順にカプセル化し、アクティブ動画品質調整ノードに転送する。アクティブ動画品質調整ノードがクライアントにパケットを転送する際には ANEP ヘッダを取り除く。

### 5.1 レート制御の精度

図 8 は、1 台のクライアントに対して動画データを転送し、アクティブ動画品質調整ノードにおいて、StrongARM と 4 個のマイクロエンジンを使い、目標レートをそれぞれ 2, 4, 6 Mbps として品質調整を行った結果である。横軸は、動画データ内の時刻を示す。なお、動画品質調整によるレート制御の精度の検証に際しては、アクティブ動画品質調整ノードがすべての動画データを受信、品質調整、および送出処理できるよう、動画サーバからの送出レートを 1 Mbps に平滑化して実験をおこなった。図 8 に示されるように、小刻みにレートが変動しているものの、指定された目標レートにあわせた動画品質調整が実現されている。

図 9 は、1 本の動画データから 2 台のクライアントに対し、それぞれ 2 個のマイクロエンジンを用いる 2 つの AA を起動し、それぞれ異なるレートの動画データを同時に生成した結果である。時刻 0 ~ 14.5 秒は 8 Mbps の動画データから 6 Mbps の動画データを生成する動画品質調整 AA が動作している。時刻 15 秒に 30 番目の GoP の ANEP ヘッダによって目標レートが 2 Mbps に変更されると同時に、同じ入力動画データを 5 Mbps に品質調整する新たな AA が起動されている。この新しい AA は時刻 35 秒に終了される。目標レートが 2 Mbps の AA は時刻 25.5 秒 ~ 30 秒まで GoP 単位で 0.1 Mbps ずつ目標レートを増やしている。図に示される通り、アクティブ動画品質調整ノードは目標レートが動的に変更された場合にも適切に品質調整を行い、また、複数の品質調整要求に対してもそれぞれの目標レートに応じた動画データを生成可能である。

### 5.2 動画品質調整のスループット

図 6 に示す実験システム 1 において 1 台のクライアントに対して動画品質調整のスループットを測定した。目標レートを 4 Mbps とし、動画サーバからの送出レートを 0.3 Mbps から 50 Kbps 単位で増加させながら 100 秒間の転送実験を繰り返し行った。クライアントでのパケット受信状態からパケット棄却を検出し、100 秒間の転送実験中にパケット棄却の発生しなかった試行のうち最も高い送出レートを動画品質調整のスループットとみなす。サーバ、ネットワーク、クライアントは十分なパケット処理能力を有するが、パケット棄却が観測された場合には 4 回の追加実験をおこない、3 回以上の追加実験でパケットの棄却が発生しなかった場合にはそのレートでの動画品質調整が可能とする。実験結果を図 10 に示す。

図の示すとおり、StrongARM のみを用いて品質調整を行った場合のスループットは 0.8 Mbps であった。

StrongARM と 1 個のマイクロエンジンの組み合わせにおいて、スレッドの数を変化させて実験を行った。StrongARM とマイクロエンジン間のデータ交換のオーバーヘッドのため、少ないスレッド数では十分なスループットがえられないが、図 10 に示されるとおり、3 個以上のスレッドを用いることで StrongARM のみによる品質調整と比較して性能が向上することがわかる。

また、図 10 に示されるとおり、StrongARM とそれぞれ 4 つのスレッドが動作するマイクロエンジンを複数組み合わせた場合、本稿の実装システムにおいては、3 個以上のマイクロエンジンを用いてもほとんど性能が向上しないことがわかる。原因として、スクラッチパッドメモリや SDRAM の帯域不足、あるいは StrongARM の処理能力不足などが考えられる。空きスレッドは定期的にスクラッチパッドメモリを参照するため、マイクロエンジン数が増えるとオーバーヘッドが大きくなる。スクラッチパッドメモリの帯域がボトルネックであれば、オーバーヘッドにより演算処理中のスレッドのスクラッチパッド内データ参照が妨害され、スループットが低下すると考えられるが、図 10 ではそのような現象が発生していない。また、後述のとおり、バックグラウンドトラフィックのある環境での動画品質調整でも最大スループットは変わらない。ルータ層では、入出力処理のためのマイクロエンジン間のデータ交換のため、

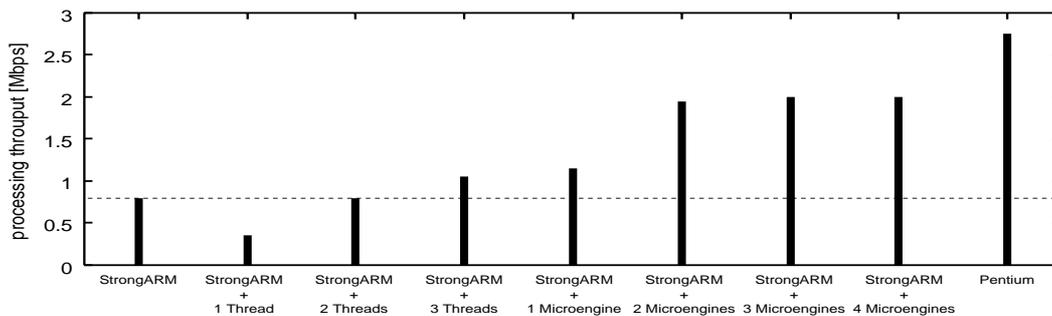


図 10 動画品質調整のスループット

SDRAM 内共有メモリを用いているため、SDRAM の帯域がボトルネックであれば、バックグラウンドトラフィック流入によるパケット処理量増加によりスループットが低下すると考えられる。したがって、StrongARM とマイクロエンジンを組み合わせた動画品質調整では、StrongARM の処理能力がボトルネックになっている可能性が高いと考えられる。

ただし、それぞれ 2 個のマイクロエンジン用いる 2 つの AA で 2 本の動画データに同時に品質調整をした場合のスループットはそれぞれ 1.00 Mbps であり、ノード全体のスループットは 1 本の動画データを 4 個のマイクロエンジンを用いて品質調整した場合と等しい。

また、図 10 に示すとおり、Pentium プロセッサを用いた場合のスループットは 2.75 Mbps であったが、プロセッサの利用率は平均 20 % 以下であり、StrongARM の処理能力や PCI バスの帯域がボトルネックであると考えられる。

さらに、バックグラウンドトラフィックと動画品質調整が互いのスループットに与える影響を検証するため、図 7 の実験システム 2 を用いて実験をおこなった。バックグラウンドトラフィックは SPIRENT 社製パケットジェネレータ SmartBit200 を用いて生成した。動画品質調整を行わない場合、2 つのマイクロエンジンには 4 つのイーサネットポートで同時に入出力それぞれ 95 Mbps、すなわち 760 Mbps のパケット処理能力があることがわかった。なお、ここでスループットは、プレアンブル、イーサネットフレーム、イーサネット CRC を合計した値である。StrongARM と 4 個のマイクロエンジンにより、送信レート 2 Mbps の動画データを目標レート 4 Mbps で品質調整した。同じポートに 93 Mbps の、残りのポートに 95 Mbps のバックグラウンドトラフィックを入力したが、バックグラウンドトラフィックと動画パケットはいずれも棄却されることなく転送、品質調整処理が行われた。トラフィック種別によらず、全二重のイーサネットポートの最大レートに近いスループットが得られたことにより、ルータ層へのマイクロエンジンの割り当てとしては 2 個が適当であることがわかった。

## 6. まとめと今後の課題

本稿では、ネットワーク内での動画品質調整の実用性、実現性を評価、検証するため、アクティブネットワーク技術を利用して動的に動画品質を変更できるアクティブ動画品質調整ノードを設計、ネットワークプロセッサに実装した。実験により、動的に変更される目標レートに応じて適切に品質調整

を行うことを示した。また、複数の動画データ品質調整、バックグラウンドトラフィックの流入下での品質調整についても良好に動作することを示した。

しかしながら、現在のシステムでは十分に実用的なスループットを達成できていない。今後は特にボトルネックと考えられる StrongARM への負荷軽減によるスループット向上について検討したい。さらに、動画品質調整 AA のノードへの動的な注入やノード機能の高度化にも取り組みたい。

## 文 献

- [1] H. Akamine, N. Wakamiya, and H. Miyahara, "Heterogeneous video multicast in an active network," IEICE Transactions on Communications, vol.E85-B, no.1, pp.284-292, Jan. 2002.
- [2] J. Smith, K. Calvert, S. Murphy, H. Orman, and L. Peterson, "Activating networks: A progress report," IEEE Computer, vol.32, no.4, pp.32-41, April 1999.
- [3] K. Calvert, "Architectural framework for active networks," RFC Draft, July 1999.
- [4] H. Akamine, K. Nakada, N. Wakamiya, M. Murata, and H. Miyahara, "Implementation and evaluation of video filtering mechanisms for real-time multicast," Technical Report of the IEICE (NS 2001-50), pp.13-18, June 2001.
- [5] T. Yamada, N. Wakamiya, M. Murata, and H. Miyahara, "Implementation and evaluation of video-quality adjustment for heterogeneous video multicast," Proceedings of The 8th Asia-Pacific Conference on Communications (APCC 2002), Bandung, pp.454-457, Sept. 2002.
- [6] N. Shalaby, L. Peterson, A. Bavier, Y. Gottlieb, S. Karlin, A. Nakao, X. Qie, T. Spalink, and M. Wawrzoniak, "Extensible Routers for Active Networks," DARPA Active Networks Conference and Exposition (DANCE), San Francisco, CA, pp.92-116, May 2002.
- [7] D.S. Alexander, B. Braden, C.A. Gunter, A.W. Jackson, A.D. Keromytis, G.J. Minden, and D. Wetherall, "Active network encapsulation protocol (ANEP)," RFC Draft, July 1997.
- [8] "Active network transport system (ANTS)." available at <http://www.sds.lcs.mit.edu/activeware/>.
- [9] "Netscript." available at <http://www.cs.columbia.edu/dcc/netscript>.
- [10] "Active networks assigned numbers authority (ANANA) type ID registry." available at <http://www.isi.edu/~braden/anana/info/typeid.txt>.
- [11] Intel Corporation, Intel IXP1200 Network Processor Family Hardware Reference Manual, Dec. 2001.
- [12] Intel Corporation, Intel IXP1200 Network Processor Family Programmer's Reference Manual, Dec. 2001.
- [13] T. Spalink, S. Karlin, and L. Peterson, "Evaluating Network Processors in IP Forwarding," Tech. Rep. TR-626-00, Department of Computer Science, Princeton University, Nov. 2000.