

異なる要求を持つユーザへのマルチオブジェクトビデオストリーミング

孫タオ, 玉井森彦, 安本慶一, 柴田直樹[†], 伊藤 実, 森 将豪[†]

奈良先端科学技術大学院大学情報科学研究科[†] 滋賀大学情報管理学科

概要 本稿では、品質に対して異なる要求を持つ多数のユーザに対し、マルチオブジェクトの MPEG-4 ビデオを、P2P ネットワークを介して効率良く配信する方法を提案する。提案手法では、各ユーザは、受信するビデオの品質に対する要求として、MPEG-4 ビデオにおける複数のオブジェクト間の相対的な重要度を、時間帯ごとに指定することが可能である。簡単な解析結果より、異種環境ユーザに対するビデオ配信において、提案方式を用いることで、階層化マルチキャスト方式より高いユーザ満足度が達成できることを確認した。

Multi-object Video Streaming for Users with Different Preferences

Tao Sun, Morihiko Tamai, Keiichi Yasumoto, Naoki Shibata[†], Minoru Ito and Masaaki Mori[†]

Grad. Sch. of Info. Sci., Nara Institute of Sci. and Tech. [†]Dept. of Info. Proc. and Man., Shiga Univ.

Abstract In this paper, a transmitted method of video stream based on multi-object for the different preferences of users in P2P network is proposed. In our proposed method, according to the request for playing quality, each user can specify relative importance among multi-objects in each time range. Experimental results show that the degree of satisfaction for the requested quality using our method is better than the layered multicast method for the different requests of users.

1 はじめに

一般に、インターネット上のビデオ配信システムにおいては、各ノード（ユーザ端末）における利用可能通信帯域の制約（配信されるデータと、制御用データの帯域の合計が各リンクの帯域を下回っていること）のもとで、各ノードの受信するビデオの品質が、要求した品質にできるだけ近いことが要求される。同一のビデオを異なる品質で複数ユーザに同時配信する方法として、マルチバージョン [1, 4], オンライントランスコード [5], 階層化マルチキャスト [3] などの方式が提案されている。

本稿では、オンライントランスコード方式に基づき、マルチオブジェクトの MPEG-4 ビデオを、異なる要求を持つ多数のユーザに対し、P2P ネットワーク上で効率良く配信する方法を提案する。提案手法では、各ユーザは、受信するビデオの品質（本稿では、ビットレートのみ対象とする）に対する要求として、MPEG-4 ビデオを構成する複数のオブジェクト間の相対的な重要度を、ユーザの好みより動的に指定することが可能である。各オブジェクトの品質は、受信する全オブジェクトの品質（ビットレート）の和が各ノードの利用可能通信帯域内に収まるように、指定した重要度に応じて比例配分することで決定する。

提案方式を実現する際には、以下の5つの条件を満たすことが望ましい。(1) 受信ノード数の増加に対応できること。(2) 各ノードの受信するビデオの品質が、要求した品質に近いこと。(3) 各ノードで実行される、配信システムの制御機構に使用される計算機資源が少ないこと。(4) 配信を新たに受ける場合に、リクエストを送ってから配信が開始されるまでの時間が短いこと。(5) ルートノードの配信サーバがデータを送信してから、リーフノードがそのデータを受信するまでの遅延が短いこと。

上記(1)-(5)を実現するため、コンテンツ送信者を根とする DAG (Direct Acyclic Graph) を作成し、DAG の各

ノード（コンテンツ受信者）が、受信したビデオストリームを実時間トランスコードし、下流のノード（より低い品質を希望する受信者）に転送することで、制約・要求が異なる複数のユーザへのビデオ配信を効率良く実現する。ここで、DAG はオブジェクト毎に作成する。また、各オブジェクトに対する重要度は、時間の経過に従い変化するため、各時間帯ごとの DAG を予め計算しておき、シームレスに切り替える。

解析結果より、異種環境ユーザに対するビデオ配信において、提案方式が、階層化マルチキャスト方式より高いユーザ満足度を達成できることを確認した。

2 関連研究

複数ユーザの異なる要求品質を同時に満足するためのビデオ配信手法としてマルチバージョン法、オンライントランスコード法、スケラブルビデオコーディング法等が提案されている。

マルチバージョン法では同じビデオコンテンツの異なるビットレートのデータをサーバに貯蔵し、ユーザの要求品質から、適切なビットレートのデータを選んで配信する [1, 4]。この方式では、1章で挙げた基準のうち、(1) は比較的良好に満たされている。一方、(2) を満たすためにはルートノードに用意するデータの数を増やす必要があり、(2) と (3) の実現はトレードオフになる。一方、(4) と (5) の条件は満たされている。

オンライントランスコード法は、ユーザの要求品質をエンコードのパラメータとして、中間ノードでオリジナルビデオデータをその場でトランスコードして配信する方法である [5]。この方式では、ユーザ数の増加に伴い、複数の異なる要求品質に対して複数の中間ノードが必要になり、エンコード、デコードのため計算量もコストも増える。したがって、(1),(2),(3) の実現はトレードオフになる。(4) は比較的良好に満たされている。(5) に関しては、トランスコード回数

に比例する遅延が不可避となる。

階層化マルチキャスト方式では、複数ユーザの異なる要求品質に同時に対応するため、階層符号化を用いてエンコードされたビデオストリームを、複数のマルチキャストストリームによって配信する。受信ノードは、自分の要求するビットレートを超えない範囲で、できるだけ多くの層を受信する。受信ノードの要求するビットレートは任意の値をとる一方、層の数はあらかじめ決められた個数しか存在しない。したがって、ユーザの要求した品質の数が層の数を超える場合、原理的に全てのユーザの要求を満たすことは出来ない。ユーザ数の増加に伴い、層の数を増加させると、それに伴ってエンコード、デコードに必要な計算量も増加する。文献 [3] は動的に各層の符号化レートを変更することで、論文中で決められた評価尺度に基づきユーザ全体の満足度を最適化する方法を提案している。したがって、1章で挙げた基準のうち、(1),(2),(3) はトレードオフとなる。一方、条件 (4),(5) はよく満たされている。

3 対象とするシステム・ネットワーク環境

本稿では、インターネットに接続された PC をノードとした、オーバーレイネットワーク上でのマルチオブジェクトビデオの配信を考える。ビデオに含まれるオブジェクトのうち、どれが重要であり、どれがそうでないかは、一般にユーザ毎に異なる。本手法では、各ユーザがあらかじめ指定した、オブジェクト毎、時間帯毎の重要度と、オブジェクトのオリジナルのビットレートに従って、各ノードで使用可能な下り帯域をオブジェクト毎に配分する。配信システムの制御の一部として、各ノードでトランスコードを行う。後述の通り、各ノードで配信システムの制御機構に使用される計算機資源のうち大部分はトランスコードに費やされる。本方式では、各ノードで同時にトランスコードを行うことのできる最大数(トランスコード次数)があらかじめ分かっているとす。各ノードにおける上り帯域と下り帯域は別々に確保されていると仮定する。ノード間のリンクの情報(帯域、リンク次数制限)は、*trace* などのツールを用いてあらかじめ検出する。送受信ノードの総数が 1000 程度で正常に動作する配信システムの構築を目指す。データの遅延時間は総ノード数の対数オーダーに収まることを目指す。文献 [3] で定義された満足度の評価基準において、ユーザの満足度を最大化することを目指す。

4 提案方式

4.1 概要

提案方式では、オブジェクト毎に、配信ノードを根とする DAG を作成し、各ノードが受信したオブジェクトのビデオストリームをリアルタイムトランスコードし、この DAG に沿って、下流のノード(より低い品質を希望する受信者)に転送していくことで、制約・要求が異なる複数のユーザへのビデオ配信を効率良く実現する。この DAG を以降トランスコード DAG と呼ぶ。

ノードの故障が起きた場合、他のノードに与える影響や、Join、Leave 時のコストを低減するため、提案手法では、オブジェクトの再生品質(ビットレート)ごとに、ノード全ての集合を分割する。各部分集合をレイヤと呼ぶ。同じレ

イヤに属するノードは、そのレイヤに対応するオブジェクトについて同じ品質のビデオストリームを受信する。各レイヤには、代表ノードと呼ばれるノードが一つ存在し、各レイヤに含まれるノードからランダムに選択される。提案手法では、トランスコード DAG の全情報を、DAG の各ノードに持たせるため、オブジェクト毎にレイヤ数が 100 以下であることを想定している。

トランスコード回数と遅延時間を短くするため、トランスコード DAG がなるべく完全二分木に近くなるようにする。ノード数が k の場合、トランスコード DAG を完全二分木にできれば、最大トランスコード回数は $\log_2 k$ となり、最大遅延時間もおよそ $O(\log k)$ となる。提案手法では、他の制限も考慮した上で、トランスコード DAG を構成する。

トランスコード数を減少させるため、オブジェクト毎に品質ズレ許容範囲(次節で定義する)を設定し、ある 2 つのノードの、品質要求のズレがこの範囲以内であるとき、その 2 つのノードを同じレイヤに配置しても良い(すなわち、対応するオブジェクトについて同じ品質のストリームを受信しても良い)ものとして扱う。ここで、品質ズレ範囲は重要度が高くなるほど狭くする¹。

以下では、新規にビデオ配信を希望するノードのトランスコード DAG への加入手続きの概要を述べる。ビデオの配信を新たに希望するノード u は、(1) オブジェクト毎、時間帯毎の重要度、および、オブジェクトのオリジナルのビットレートに従って、使用可能下り帯域をオブジェクト毎に配分し、各オブジェクトに対する要求品質を定める。次に、(2) 既にビデオの受信を行っているノード² の一つ (u_A とする) に *join* メッセージを送信し、 u_A は自分の保持する各オブジェクトに関する配信情報テーブル(トランスコード DAG、各レイヤの品質、各レイヤの代表ノードのアドレスを格納する)を返す。ノード u は、(3) 受信した配信情報テーブルを用いて、オブジェクト毎に、自分の品質ズレ許容範囲内に入る既存レイヤが存在するかを調べる。(4a) そのようなレイヤが存在すれば、ストリーム配信要求をそのレイヤの代表ノードに送信する。品質ズレ許容範囲内に入る既存レイヤが複数存在する場合は、ストリーム配信要求を全てのレイヤの代表ノードに同時に送信する。一方、(4b) 品質ズレ許容範囲内に入るレイヤが存在しないときは、要求品質と同じ品質となる新規のレイヤを作成し、そのレイヤへの加入を試みる。(5) 既存レイヤへの加入、新規レイヤの作成のいずれにおいても、ストリーム配信要求を受信した各既存レイヤの代表ノードが、自分の持つレイヤ内の全ノードに関する次数制約および残り上り帯域の情報に基づき、 u の親ノードを探索する。(6a) u の親ノードが見付かれば、 u は既存レイヤまたは新規レイヤに加入する。(6b) u の親ノードが見付からない場合は、加入可能な最大の品質を持つ既存レイヤへ加入する。

以下では、提案方式で使用するいくつかの概念について定義した後、上記の加入手続きについて、より詳細に述べる。

¹一般に、重要度が高いほど、要求品質通りに受信したいと考えられ、逆に重要度が低い場合は、品質のズレはあまり問題にならないと考えられる。

²そのようなノードの発見は、ロビーサーバや、Chord[2] など、既存の方法を使う。

4.2 定義

ビデオ

オリジナルのビデオが、 m 個のカットから構成されており、それぞれの(連続する)時間範囲を T_1, T_2, \dots, T_m とする。ビデオに含まれるオブジェクトの数は n であり、それらのリストを $O = (O_1, O_2, \dots, O_n)$ とする。各時間範囲において、各オブジェクトのビットレートは同じであると仮定し、時間範囲 T_i における、オブジェクト O_j のビットレートを $O_j.bitrate_i$ と表記する。

ノード

あるノード u について、 u の上り、下り帯域、ノードのリンク次数制限、トランスコード次数制限をそれぞれ $u.upper_bw$, $u.lower_bw$, $u.n.transcode$, $u.n.link$ と表記する。オブジェクト O_1, O_2, \dots, O_n それぞれについて、現在ノード u が受信しているビットレートを $u.cur_br_1, u.cur_br_2, \dots, u.cur_br_n$ と表記する。オーバーレイネットワーク上で、ノード P がノード Q から直接ビデオデータを受信しているとき、 Q を P の親ノードとよび、また、 P を Q の子ノードとよぶ。

品質に対する要求

各オブジェクト毎に、各ユーザが重要度を 0 以上の整数で指定する。ノード(ユーザ) u の時間帯 T_i におけるオブジェクト O_j に対する重要度を $u.importance_{i,j}$ と表記する。なお、各ユーザ u は、重要度 $\bigcup_{i=1}^m \bigcup_{j=1}^n \{u.importance_{i,j}\}$ を予め与えると仮定する³。

ユーザ u の下り帯域 $u.lower_bw$ を、各オブジェクトの重要度 $u.importance_{i,j}$ により、次式に従って配分する。

$$u.lower_bw = \sum_{j=1}^n u.demand_{i,j} \times u.importance_{i,j} \quad (1)$$

上記の式により決定されるビットレート $u.demand_{i,j}$ は、 u の時間帯 T_i におけるオブジェクト O_j に対する要求品質である。

レイヤ

ノード全ての集合を S とするとき、時間範囲 T_i 内のオブジェクト O_j に関する S の分割(レイヤ全ての集合)を $L_{j,i}$ と表記する。レイヤ l_k ($l_k \in L_{j,i}$, $1 \leq k \leq |L_{j,i}|$) 内のノードが受信しているストリームの品質(レイヤ品質とよぶ)を $l_k.bitrate$ と表記する。各送受信ノード u は、オブジェクト O_j 毎にテーブル $u.Table_j = (\Gamma_j, u.rep_j, (l_1.bitrate, \dots, l_{|L_{j,i}|}.bitrate))$ を持つ。ここで、 Γ_j は、前述のトランスコード DAG である。これは、各レイヤをノードとする DAG であり、他のレイヤを介さずオブジェクトのビデオデータの送受信が行われる時に限り、対応するノード間にリンクが存在する。 $u.rep_j$ はレイヤの代表ノードのアドレスである。代表ノードは、自分の属するレイヤ内の全てのノードの残り上り帯域、リンク次数、トランスコード次数、子ノードへの送信ビットレートを知っているものとする。

トランスコード DAG のリンクには、通常リンクとショートカットリンクの二つの属性がある。トランスコード DAG から全てのショートカットリンクを取り除くと、木となる。この木の上で、レイヤ l_A がレイヤ l_B と直接リンクを持ち、

³全ての重要度、全てのオブジェクトについて、ユーザが予め重要度を与えることは煩雑だと考えられるが、コンテンツプロバイダが推奨設定を幾つか用意し、ユーザがその中から選択するなどの方法で、ある程度緩和できると考える。

l_A が l_B よりルートノードから遠いとき、 l_B を l_A の親レイヤとよび、 l_A を l_B の子レイヤとよぶ。

品質ズレ許容範囲

ノード u における時間範囲 T_i 内でのオブジェクト O_j の品質ズレ許容範囲 $u.drift_{i,j}$ は次式(2)で定義する。

$$u.drift_{i,j} = \frac{O_j.bitrate_i}{u.importance_i} u.\beta \quad (2)$$

ここで、 $u.\beta$ はノード u のユーザが指定した 0 から 1 までの定数である。

4.3 トランスコード DAG への加入手続き

本節では、ノード u が、要求品質 $u.demand_{i,j}$ 、品質ズレ許容範囲 $u.drift_{i,j}$ を持つオブジェクト O_j に対するストリームの配信要求を送信した場合の、親ノードの決定方法について述べる。以下では、ノード w がノード v の子ノードであるとき、 $v \rightarrow w$ と表記する。また、ノード v の残り上り帯域を $v.remain_upper_bw$ と表記する。

親ノードの決定方法は、以下の式が成り立つか否かで大きく二つに分かれる。

$$\begin{aligned} \exists k (u.demand_{i,j}(1 - u.drift_{i,j}) &\leq l_k.bitrate \\ &\leq u.demand_{i,j}(1 + u.drift_{i,j})) \end{aligned} \quad (3)$$

(3) 式が成り立つ場合、 u は既存のレイヤ l_k s.t. $(u.demand_{i,j}(1 - u.drift_{i,j}) \leq l_k.bitrate \leq u.demand_{i,j}(1 + u.drift_{i,j}))$ に加入する。一方、(3) 式が成り立たない場合、レイヤ品質が $u.demand_{i,j}$ である新たなレイヤを作成し、そのレイヤに加入する。以下では、それぞれの場合についての処理をより詳細に説明する。

(3) 式が成り立つ場合

親ノードの探索および既存レイヤへの加入を以下の手順で行う。

Step.1 集合 $\{k \mid u.demand_{i,j}(1 - u.drift_{i,j}) \leq l_k.bitrate \leq u.demand_{i,j}(1 + u.drift_{i,j})\}$ の各要素に対し、 l_k の親及び祖先レイヤの代表ノードへストリーム配信要求(要求品質は $l_k.bitrate$)を送信する。

Step.2 ストリーム配信要求を受信した代表ノードは、自分の保持するレイヤ内の全てのノードの情報に基づいて、ストリーム配信要求を満足できるノードを探索する(あるノードがストリーム配信要求を満足するための条件は後述する)。要求を満足できるノードが見付ければ、そのノードのアドレスをストリーム配信要求元に送信する。自分の属するレイヤ内にストリーム配信要求を満足できるノードが存在しない場合は、当該レイヤにおいて探索が失敗したことをストリーム配信要求元に知らせる。

Step.3 要求品質 $l_k.bitrate$ の親ノード v が見付ければ⁴、 u は v の子ノードとなり、レイヤ l_k に加入する。

Step.4 親レイヤが存在しない場合は、配信要求を落して加入可能なレイヤに加入する。

あるノード v が u のストリーム配信要求を満足するための条件を以下に示す。

⁴親ノードが複数ノード見付かった場合には、その中からランダムに選択する。

条件 1 v はリンク次数制限およびトランスコード次数制限を満足し、かつ、

条件 2a $l_k.bitrate \leq v.remaining_upper_bw$ が成り立つ。または、

条件 2b $l_k.bitrate \leq v.remaining_upper_bw + \sum_{w \in W} w.cur_br_j$ 、かつ、 $\sum_{w \in W} w.cur_br_j \leq v.remaining_upper_bw$ が成り立つような、 v の子ノードの部分集合 $W \subseteq \{x | v \rightarrow x, x.cur_br_j < l_k.bitrate\}$ が存在する。

条件 2a が成り立つ場合、 v の残り上り帯域に、 u の要求を満足できるだけの余裕があるため、 v は u の親ノードとなることができる。また、条件 2b が成り立つ場合、 v の残り上り帯域には、 u の要求を満足できるだけの余裕がないが、 v のいくつかの子ノードを u の子ノードへ変更することにより、 v が u の要求を満足できるだけの帯域を確保することが可能となる。このとき、 v は u の親ノードとなり、かつ、 W の全ての要素が、 v の子ノードから u の子ノードへと変更される。例えば、あるノード u ($u.remaining_upper_bw = 200kbps$) が、レイヤ l_k ($l_k.bitrate = 300kbps$) へ加入すべく、要求品質 $300kbps$ をレイヤ品質 $400kbps$ のレイヤの代表ノードへ送信したとする。代表ノードは、条件 1 および 2b を満足するノードとして、子ノード w ($w.cur_br_j = 100kbps$) を持つノード v ($v.remaining_upper_bw = 200kbps$) を発見し、そのアドレスを u に送信したとする。このとき、 v は u の親ノードとなり、 w は v の子ノードから u の子ノードへと変更される。

全ての代表ノードにおいて u の親ノードの探索が失敗した場合、レイヤ品質が $u.demand_{i,j}(1-u.drift_{i,j})$ 未満で、親ノードが確保可能なレイヤの内、できるだけ高品質のレイヤへ加入する。このようなレイヤは、上記の探索過程において、条件 1 および 2 を満足できる最大の品質がいくらであるのかを、代表ノードが探索失敗の報告とともに返信しておくことで、新たな探索を行うことなく発見できる。例えば、合計 5 つのレイヤ l_1 ($l_1.bitrate = 100kbps$)、 l_2 ($l_2.bitrate = 200kbps$)、 l_3 ($l_3.bitrate = 300kbps$)、 l_4 ($l_4.bitrate = 400kbps$)、 l_5 ($l_5.bitrate = 500kbps$) がそれぞれ親子関係で存在し、あるノード u がレイヤ l_3 へ加入すべく、レイヤ l_3 の親及び祖先レイヤとなる l_4 、 l_5 の代表ノードへストリーム配送要求を送信し、親ノードを探索したとする。そして、親ノードの探索は、 l_4 、 l_5 いずれにおいても失敗し、 l_4 、 l_5 の代表ノードは、探索失敗の報告とともに、条件 1 および 2 を満足できる最大の品質として、それぞれ $150kbps$ 、 $250kbps$ を返信したとする。この情報に基づき、 u はレイヤ l_3 の子レイヤ l_1 、 l_2 から加入可能な最も高い品質であるレイヤ l_2 を選んで加入する。

全ての既存レイヤに加入できない場合、葉レイヤを探索し、親ノードとして確保可能な葉ノードの内、もっとも高品質なレイヤを作成できる葉ノードを親ノードとし、新しいレイヤを作成し加入する。

(3) 式が成り立たない場合

Step.1 一方、ノード u の要求する品質の品質ズレ許容範囲内にレイヤが存在しないとき、レイヤ品質が $u.b_{i,j}$ である新しいレイヤを作成し、そのレイヤに加入する。トランスコード DAG 上の、新しいレイヤを作成する場所は、次のようにして決める。ノード u のストリーム配送要求よりも高いレイヤ品質を持つ全てのレイヤに対して、要求に近い品質を持つレイヤから順に子レイヤの数が 2 未満である

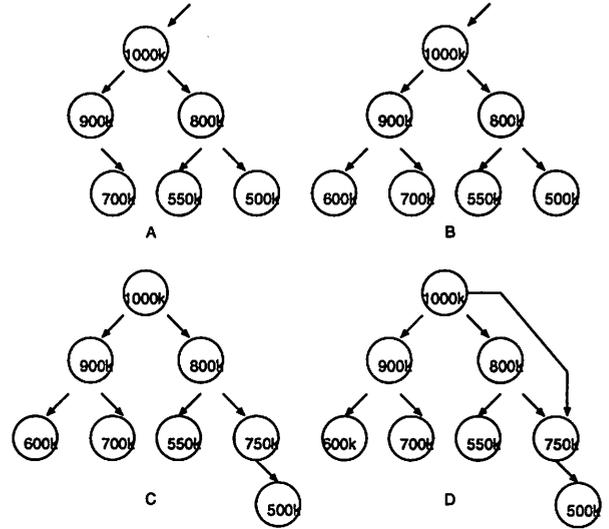


図 1: ダグにレイヤを追加する例

レイヤがあるか調べる。もしあれば、そのレイヤの子レイヤとして、新しいレイヤを作成する。ない場合は、要求されたビットレート品質がもっとも近いレイヤ L_M を探し、 L_M の親の子として新しく作成するレイヤを接続し、そのレイヤの子として L_M を根とする部分 DAG を接続する。新しいレイヤの残り帯域が子レイヤ全ての帯域より少ない場合は、子レイヤから親レイヤにショートカットを作成する。

たとえば、図 1 の A に示すようなレイヤ DAG にノード u_1 が $600k$ の配送要求を送信した場合、まず、配送要求の $600k$ より高い品質のレイヤに対して上り帯域の空きを探す。レイヤ L_{900k} の子レイヤは L_{700k} だけであるので、図 1 の B に示すように新しいレイヤ L_{600k} ($u_1 \in L_{600k}$) はレイヤ L_{900k} の子レイヤにする。そして、ノード u_2 が $750k$ の配信を要求した場合、 u_2 の品質ズレ許容範囲内にレイヤは存在せず、 $750k$ より高いレイヤの空きもないので、図 1 の C に示すように $750k$ より高く、かつ、品質が一番近いレイヤである L_{800k} を L_{700k} の親レイヤとする。レイヤ L_{800k} に属する二つの子レイヤから品質の低い子レイヤ L_{500k} を選んで置き換える。この際、上流のレイヤ L_{1000k} からデータを受信する必要があるとき、図 1 の D に示すようにレイヤ L_{750k} とレイヤ L_{1000k} 間でショートカットを追加する。**Step.2** (3) 式が成り立つ場合と同様の方法でノード u が新しく作成されたレイヤに加入できるか判定する。加入できない場合には、Step.1 の操作をキャンセルし、(3) 式が成り立つ場合で述べたように全ての代表ノードにおいて u の親ノードの探索が失敗した場合の処理に従って品質を落して加入できるレイヤを探す。加入できる場合は、DAG の変更部分の差分を全てのレイヤの代表ノードに送る。

以上のアルゴリズムでは、上流レイヤの残った上り帯域に基づいて配信レイヤを選択するが、オーバーレイネットワークの性質上、各ノードの上りと下り帯域制限のほかにも帯域のボトルネックが発生する場合がある。この場合、品質要求を落して配信レイヤを探し直す。

4.4 トランスコード DAG からの離脱手続き

ビデオの配信を受けているあるノード u がビデオの配信中止を希望する場合、 u がトランスコード DAG から離脱する前に u の子ノードが新たに配信を受けるノードを確定す

必要がある。まず、 u は u の親ノードに *leave* メッセージを送り、 u の子ノードに *rejoin* メッセージを送る。 u は、 u の全ての子ノードから *rejoin.complete* メッセージを受け取るのを待ち、その後全てのコネクションを切断する。 u の親ノードは *leave* メッセージを受け取った以降、 u がいないものとして、受け取った *join* メッセージとその処理を行う。*rejoin* メッセージを受け取ったノードは、現在のレイヤに留まったまま、新たに配信を受けられるノードを、加入手続きと同様の方法で探す。もしあれば、以後そのノードから配信を受けることとし、親ノードに *rejoin.complete* メッセージを送信する。現在のレイヤに留まったまま、新たに配信を受けられるノードが無ければ、全ての子ノードに *rejoin* メッセージを送り、その後加入時と同じ方法で、配信を受けられるレイヤを探す。新たに配信を受けるノードが決まれば、*rejoin* メッセージを送ったノードに *rejoin.complete* メッセージを送信する。

4.5 カット切り替え時の DAG 再構築

カットが切り替わる時、各オブジェクトに対する要求品質も切り替わる。提案手法では、カット切り替えの一定時間前に次のカット用のトランスコード DAG の構成を始める。トランスコード DAG の下流に位置するノードほど、より大きな遅延があるので、カットの切り替わりはトランスコード DAG の *root* から *leaf* の順になる。もっとも早く切り替わりの時間を迎える *root* ノードが新しいカット用のトランスコード DAG を最初に作る。この時、新しいカット用のトランスコード DAG には *root* ノード (配信ノード) のみが登録されている。下流の各レイヤは、カットの切り替わりの一定時間前になると、自分の親ノードに新しいカットのための *join* メッセージを送信し、前述の加入アルゴリズムに従って新しい DAG に Join する。全てのノードが新しい DAG に Join が終了してからカットが切り替わるように、時間的余裕を持って新しい DAG の構築が行われる。

5 評価

本章では、ユーザ間での異なる品質要求を満足するためのビデオ配信手法として一般的に研究されている階層化マルチキャストとの比較により、提案方式の有効性を検証する。なお本章では、ビデオは単一のオブジェクトから成るものとする。

以下では、提案手法が階層化マルチキャストに比べ、どの程度各ユーザの品質要求を満足できるかを検証する。ユーザ i の満足度 $0 \leq S_i \leq 1$ を文献 [3] と同様に、次のように定義する。

$$S_i = 1 - \frac{|E_i - R_i|}{E_i} \quad (4)$$

ここで、 E_i はユーザ i が要求する品質 (ビットレートで表す) を、 R_i はユーザ i が実際に受信できたビットレートを表す。ユーザ数を N としたときの、階層化マルチキャストおよび提案手法における各ユーザの満足度の平均値 ($1/N \sum S_i$) を計算機シミュレーションによる実験により求めた。実験は以下のような設定で行った。 $N = 1000$ とし、各ユーザの末端のリンクの利用可能帯域を以下の 3 種類からランダムに選択する: (i) 上り 128kbps, 下り 64kbps, (ii) 上り 1.5Mbps, 下り 512kbps, (iii) 上り 10Mbps, 下り 10Mbps。また、各ユーザの末端のリンク以外の利用可能帯域は、十分満足されているものとする。各ユーザの要求品質 E_i は、32kbps を最低値、3Mbps を最大値として、利用可能下り帯域を満

足する範囲内で一様分布により決定する。

階層化マルチキャストでは、IP マルチキャストによる利用を想定し、各ノードはストリームの受信のみを行うものとする。階層化マルチキャストにおける満足度は、階層符号化における各層の符号化レートの割り当て方に大きく依存することが予備実験により分かった。そこで、予備実験の中でも高い満足度を達成した以下の 2 種類の手法を用いて、各層の符号化レートを決定し、階層数の増加に伴う満足度の平均値の変化を測定した。

手法 1 基本層の符号化レート B_0 を 32kbps とし、 i 番目の拡張層 ($i = 1, 2, \dots, L$) の符号化レートを $B_i = a^{i-1} B_0$ とする。ここで a は、次式によって決定する: $\sum_{i=1}^L a^{i-1} B_0 = 10\text{Mbps}$ 。

手法 2 満足度の平均値を評価関数として、焼きなまし法 (SA 法) により各層の符号化レートを決定する (反復回数は 10000 回)。

手法 1 は各層の符号化レートをあらかじめ決められた値に割り当てる手法であり、実現が容易である一方、ユーザの要求品質を全く考慮していない。そこで文献 [3] のように、ユーザの要求に応じて動的に各層の符号化レートを決定する手法が提案されており、静的に割り当てる手法に比べ、高い満足度を達成することが報告されている。そこで手法 2 では、動的に符号化レートを決定する手法と同等の満足度を達成するため、焼きなまし法を用いて各層の符号化レートの最適化を行った。

提案手法における品質許容範囲は、 $E_i(1 \pm 0.05)$ とした。また、次数制約として、最大リンク次数を 10、最大トランスコード次数を 1 から 10 まで変化させ、満足度の平均値の変化を測定した。実験結果を図 2 に示す。なお、試行回数は 100 回とした。

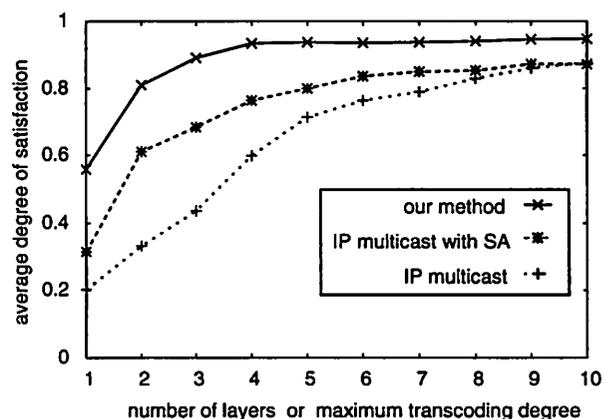


図 2: 階層化マルチキャスト、提案手法におけるユーザの満足度の平均値

図 2 より、最大トランスコード次数が 3 以上であれば、階層数が 10 までの階層化マルチキャストと比較して、提案手法は大きな満足度を達成できることが分かる。

次に、提案手法における、ユーザ数の増加に対するトランスコード回数および遅延時間への影響を検証するため、ノード数の増加に対するトランスコード木の長さの変化を測定した。実験結果を図 3 に示す。なお、各ノードの最大トランスコード次数は 3 に設定した。

図 3 より、ノード数の増加に対するトランスコード木の長さは、ほぼ $\log N$ に比例することが分かる。従って、ルー

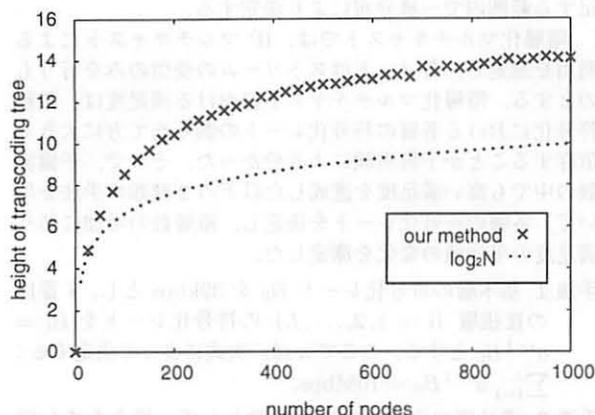


図 3: 提案手法におけるノード数の増加に対するトランスコード木の長さの変化

トノードからのストリームが各ノードへ配信されるまでの遅延時間およびトランスコード回数は、ノード数の増加に対して十分に小さく抑えられることが分かる。

次に、トランスコード DAG への加入アルゴリズムにおける、加入に必要な遅延時間について調べた。実験 1 と同じパラメータを設定し、ノード間リンクの遅延を $[10ms, 160ms]$ の間で乱数により決定した。トランスコードの遅延と動画データのバッファリング遅延の合計を $2000ms$ に設定し、ノード数の増加に対するトランスコード DAG への加入時間を測定した。実験結果を図 4 に示す。

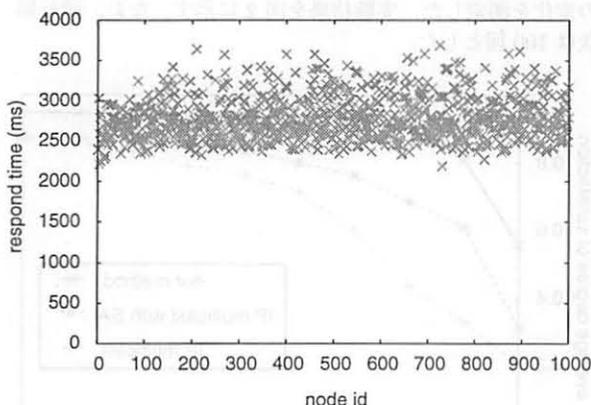


図 4: トランスコード DAG への加入時間

図 4 より、トランスコード DAG への加入のための時間はノード数の増加に対してほぼ一定であり、加入の時期によらず、 $4000ms$ 以内に抑えられることが分かった。

最後に、カット切り替え時の DAG の再構築について実験した。ユーザの次のカットに対する品質要求は異なる想定して、要求品質、上り、下り帯域をランダムで再割当した (複数オブジェクトビデオの中の 1 つのオブジェクトに対する実験とし、カット切り替えにより次のオブジェクトに対する品質要求と使える帯域は変わると考えられるため)。ノード間リンクの遅延はそのままにした。提案手法における、100 回のカットの切り替えに対する新しい DAG への加入のための時間を測定した。実験結果から、もっとも遅いノードが加入するときにかかる時間は $5000ms$ 以内であることが分かった。従って、カットの切り替わりの $5000ms$

前に、次のカットのための Join message を送れば、カットが切り替わっても二つのオブジェクトを連続再生することが可能になる。なお、カットの切り替えの頻度に対する遅延時間への影響はないことを確認した。

1 章で述べた評価基準に沿って、提案アルゴリズムを評価すると、以下ようになる。(1) については、1000 ノード程度まで対応できる。(2) については、上述の実験結果によって示したとおり、10 層までの階層化マルチキャスト方式よりも高い満足度を達成できた。また、オブジェクト毎に品質を指定することができるようになった。(3) については、Join については十分小さいといえるが、Leave に関しては評価が終わっていない。トランスコードにかかる計算機資源に関しては、トランスコード次数を設定することで対応できる。(4) については、十分少ないといえる。(5) については、ノード数の log に比例し、また、トランスコードの遅延に比例するだけの遅延が発生する。

6 おわりに

本稿では、P2P オーバーレイ環境でのマルチオブジェクトビデオストリームの再生において、異なる要求を持つ多数のユーザに対し、各ピアの帯域、次数などの制約を満たし、かつ、各オブジェクトの相対的な重要度、時間帯ごとの重要度を指定可能なビデオ配信方式を提案した。

実験により、トランスコード次数が 3 以上の場合、階層化マルチキャストに対し、各ユーザの満足度の平均値を向上できること、また、ユーザ数の増加に対するトランスコード回数および遅延時間を十分小さく抑えられることを確認した。

提案手法では、セッション内の葉ノードを除く全てのピアがトランスコードを行うが、各ノードは単一の親ノードのみからデータを受信している。従って、リンクもしくはノードの故障が起こった時、配信が一時的に停止してしまうことが考えられる。この問題について、信頼性を向上させるため、トランスコード DAG の親子のノード間、すなわち、親子のレイヤ間にメッシュ上のオーバーレイリンクを構築し、FEC による並列伝送を行う手法を検討している。

参考文献

- [1] G. Conklin, G. Greenbaum, K. Lillevold, and A. Lippman. Video Coding for Streaming Media Delivery on the Internet. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(3), March 2001
- [2] Ion Stoica, Robert Morris, David Liben-Nowell, David R. Karger, M. Frans Kaashoek, Frank Dabek, Hari Balakrishnan, A Scalable Peer-to-peer Lookup Protocol for Internet Applications", *IEEE/ACM Transactions on Networking*, Vol. 11, No. 1, pp. 17-32, February 2003.
- [3] J. Liu, B. Li, and Y.-Q. Zhang, "An End-to-End Adaptation Protocol for Layered Video Multicast Using Optimal Rate Allocation," *IEEE Transactions on Multimedia*, February 2004.
- [4] K. Nichols, S. Blake, F. Baker, and D.L. Black, "Definition of the DiRerentiated Services Field (DS Field) in the IPv4 and IPv6 Headers," *IETF RFC 2474*, Dec. 1998.
- [5] S. Jacobs and A. Eleftheriadis. Streaming Video using Dynamic Rate Shaping and TCP Flow Control. *Visual Communication and Image Representation Journal*, January 1998. (invited paper).