

携帯端末へのビデオマルチキャスト方式の設計と評価に関する研究

伊藤 実 奈良先端科学技術大学院大学情報科学研究科教授

1 はじめに

近年の無線ネットワーク技術の発展・普及に伴い、様々な携帯端末上でのビデオのストリーミング再生が実現可能となった。しかし、これら携帯端末の種類は多岐にわたる上、移動に伴いアクセスポイントおよび通信状況が変化する。そのため、TV放送のようにあるビデオコンテンツを多数の携帯端末にブロードキャストする際には、(1) 端末の画面サイズ、計算能力、バッテリー残量、通信速度に対する制約を考慮しつつ、各端末が受信するビデオの品質（要求品質）を決定できること、(2) ビデオの再生中、ユーザがいつでも要求品質を変更可能なこと、(3) 各ユーザの要求品質にできるだけ近い品質のビデオを配信可能なこと、(4) 端末が移動して接続アクセスポイントが変わっても、シームレスにビデオの再生が継続できること、(5) コンテンツ配信ネットワーク（CDN）の利用可能リソース量に対する制約を満たし、かつ、必要リソース量をできるだけ少なくすること、が重要となる。

異なる要求品質を持つ複数ノードへのビデオ配信方式に関し、既存手法では、MPEG-4 FGS[1]等の階層符号化に基づいた方式[2]が多い。しかし階層符号化では、階層ごとに独立したバッファ、デコーダが必要となるため、単一のストリームを受信・再生する場合に比べ多くの計算資源を消費する。また、ビデオの品質に関わるパラメータ（画像サイズ、フレームレート、ビットレート）の値を、複数同時に調整することが困難であるという欠点がある。

本報告では上記(1)-(5)を満足する方式として、新たにサービス合成に基づく方式を提案する。提案方式では、図1のように、サーバ、プロキシ、無線アクセスポイント(AP) から成る CDN が与えられるとする。各端末はいずれかの AP を経由してネットワークに接続し、端末が移動した際には最寄りの AP へ自動的に接続が切り替えられるものとする。提案手法では上記(1)、(3)を実現するため、プロキシ上でトランスコードサービスを実行し、ビデオの品質変換を行う。また各プロキシは、他のプロキシや携帯端末へのストリームの中継サービスを実行する。このとき、上記(5)を実現するため、CDN が消費するリソース量ができるだけ少なくなるように、各プロキシの入出力品質、および、サーバ、プロキシ、ユーザ端末間の接続パスを決定する。また、上記(2)を実現するため、各ユーザが希望する最新の要求品質に合わせて、定期的に配送経路を再構築するためのプロトコルを提案する。さらに上記(4)を実現するため、ある端末の接続 AP が切り替わった際には、切り替え後の AP の担当プロキシが、現在配信を行っている品質の中から要求品質に最も近いものを一時的に配信する。

シミュレーションによる評価により、提案方式で求めた配送経路が、ユーザの希望再生品質を満足しつつ、CDN の消費リソース量を少なく抑えられることを確認した。また、実環境上での評価により、提案手法がアルゴリズムの実行も含め、実用上問題ない程度の遅延時間でストリームの配信を開始できることを確認した。

2 関連研究

文献[4-6]では、本報告と同様に、異なるネットワーク環境下の多種多様な端末に対し、マルチメディアコンテンツを効率よく配送するための方式を提案している。しかし、文献[4]では、各プロキシで実行されるサービス数および種類は予め決められており、動的にサービスを生成するような環境は想定されていない。文献[5]では、ユーザの要求品質は予め与えられているものとしており、再生中の要求変更には対応していない。また、ユーザノードの移動も考慮していない。文献[6]では、プロキシ間を接続する大域的なマルチキャスト木と、ユーザノードから成る局所的なマルチキャスト木を構築し、それらを連結することでサービス配送パスを求めている。しかし、プロキシ間を接続するオーバーレイリンクのリソース消費量の効率は考慮されておらず、ユーザノードの移動も考慮されていない。一方、提案手法では、動画再生中の希望再生品質の変更や接続 AP の変更も考慮しつつ、CDN の消費リソース量を少なく抑えることができる。

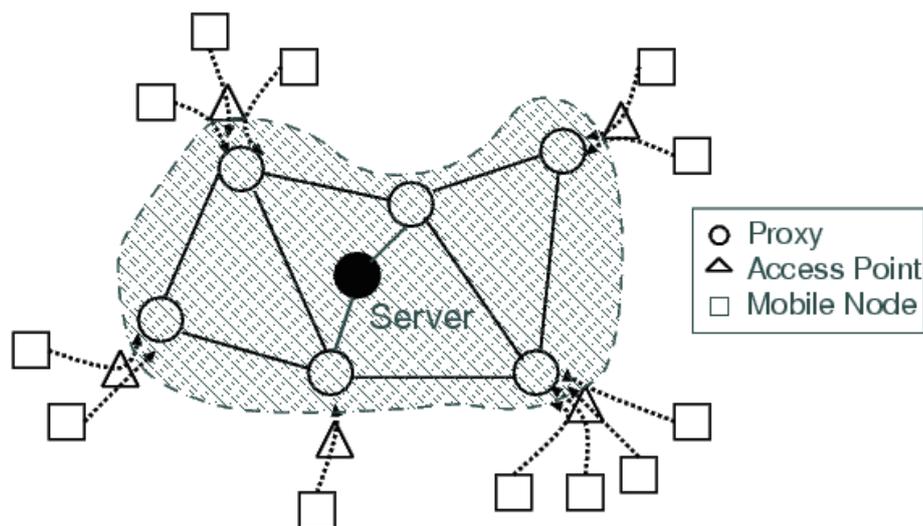


図 1: ネットワーク環境

3 概要説明

本章では、まず提案方式が対象とする環境と仮定を述べ、次に提案方式の概要を述べる。

3-1 対象とする環境と仮定

提案方式が対象とする環境を以下に示す。

(1) ビデオ配信サーバ：録画済または生放送のビデオの配信を行う。TV 放送と同様に、全ユーザは同時刻のビデオを試聴するものとし、ビデオオンデマンドは対象外とする。

(2) オーバーレイネットワーク：ビデオ配信サーバ、複数のプロキシ、複数の AP からなる CDN が与えられる (図 1)。ただし、プロキシ間、あるいは、サーバとプロキシ間、プロキシと AP 間の各オーバーレイリンク上では DiffServ[3]などの手法により、予め使用可能帯域が確保されているとする。プロキシと AP 間の帯域、AP とモバイルノード間の (AP 側から見た) 帯域は、無制限に使用可能と仮定する。これは、AP の増設が比較的安価かつ容易に行え、また、プロキシは対応する AP の近くに配置されていると考えられるためである。

(3) モバイルノード：画面サイズ、計算能力、最大伝送速度、バッテリー容量などが異なる多数の携帯端末 (PDA、携帯電話など) が存在する。各端末は、最寄りの AP を経由してプロキシと通信可能である。端末の位置に応じて通信可能な AP は一意に定まり、各 AP は端末の移動により AP が切り替わったことを瞬時に認識可能とする。

(4) 要求品質：配信サーバが送信するビデオの品質を最大として、システム側であらかじめ決められた数の品質 (最低品質から最高品質までを 5% 程度の刻み幅で均等分割) を用意する。各モバイルノードは用意された品質の中から、端末の計算能力、伝送速度、バッテリー量などの制約に基づいて品質を決定する。また、各モバイルノードはビデオの視聴中に再生品質を変更できるものとする。

(5) サービス：プロキシ上で実行されるサービスとして、トランスコードサービスおよび中継サービスの 2 種類を考える。各サービスの実行に必要な計算量および帯域は、サービスへの入出力ビデオ品質から算出可能である。

(6) プロキシ：各プロキシは資源の許す限り、サービスを複数個実行できる。プロキシごとに利用可能計算資源 (CPU パワー、メモリ量など) が決まっており、利用可能帯域はオーバーレイリンクの制限に従う。本報告では簡単のため、トランスコードサービスの実行に必要な計算資源として CPU パワーのみを扱う。また中継サービスに必要な CPU パワーは無視できるとする。

3-2 提案方式の概要

提案方式では、図 1 のような CDN、および各モバイルノードの要求品質が与えられたとき、配信サーバから全モバイルノードへのビデオ配送経路を求める。モバイルノードの要求品質は、端末の種類や性能に応じて様々であるため、配信サーバ上で全ての要求品質をあらかじめ用意することは、リソース (ディスク量や品質変換のための計算資源) 消費量の観点から望ましくない。提案手法では、CDN 上の複数のプロキシ上でトランスコードを分散実行することでこの問題に対応する。具体的には、配信サ

サーバからのビデオストリームを、プロキシ上でトランスコードし、そのストリームをモバイルノードへ配信、もしくは、下流のプロキシへ中継し、下流のプロキシでさらにトランスコードを行い、モバイルノードへ配信する。このとき、ある品質からより高い品質へのトランスコードを行うことは不可能であるため、配信経路上のトランスコードサービスの列は、徐々に品質が下がるように構築されなければならない。

CDN の利用可能資源（ネットワーク資源、計算資源）は限られているため、配送経路を求める際は、できるだけ資源の消費量が少なくなるような経路を求める。また、ネットワーク資源と計算資源のどちらのコストがより高いかは、様々な条件（機器の市場価格や CDN プロバイダのネットワーク構成）によって変化するため、どちらのコストがより高いかを指定できるようにし、両資源のコストのバランスを考慮した配送経路を求めるようにする。具体的には、計算資源とネットワーク資源のコスト比を $\alpha:1-\alpha$ とし、「コスト関数 = $\alpha \times$ 計算資源 + $(1-\alpha) \times$ ネットワーク資源」を最小化する。

4 ビデオ配信プロトコル

本章では、ビデオ配信プロトコルのうちビデオ配信開始手順と配信経路の再構築手順を示す。

4-1 ビデオ配信開始手順

Step 1: ビデオの配信開始時刻を t とする。各モバイルノード u は、 t 以前に要求品質 q_u を自分の属するプロキシ p に送信する。

Step 2: p は受信した要求をサーバ s に転送する。

Step 3: s は、受信した全要求品質から 5 章で述べるアルゴリズムを用いて配送経路を求める。 s は、求めた配送経路の情報を含むメッセージを、配送経路に沿って各プロキシに送信する。

Step 4: 各プロキシは、必要数のトランスコードサービスと中継サービスのインスタンスを起動する。

Step 5: 時刻 t になると、 s は配送経路に沿ってビデオを送信する。各プロキシは、各サービスの下流のプロキシに対し、トランスコードサービスおよび中継サービスの出力ストリームを送信する。

4-2 配送経路の再構築手順

配送経路の再構築時刻を t_r とする。ここで、 t_r はある時間間隔で与えられる。また、全モバイルノードは t_r を予め知っているものとする。時刻 $t_r - \delta$ が近づくと、各プロキシはモバイルノードから受信した最新の要求品質を s に送信する。ここで δ は、再構築後のストリームの配信を開始するための準備（全モバイルノードの要求品質の収集、再構築後の配送経路の計算とその配布、配送遅延を含むストリームの受信）に必要な時間である。再構築後の配送経路は、全モバイルノードの要求品質から、5 章で述べるアルゴリズムを用いて s が求める。

配送経路上で配信サーバから遠い位置にいたプロキシが、経路の再構築後に配信サーバに近い位置へ移動した場合、再構築前に比べてバッファリング遅延が減少し、ストリームの再生がその分途切れてしまう可能性がある。そこで、システムの動作前に予め想定される最大の経路長を計算しておき、全プロキシはその経路長分の移動の影響を吸収するのに十分なバッファを用意しておく。そして、経路の切り替えが完了するまでは、再構築前と後の両方の経路からストリームを受信・バッファリングすることで、再構築時にストリームが途切れることを防ぐ。

5 配送経路計算アルゴリズム

本章では、配信サーバから全モバイルノードへの配送経路を求めるアルゴリズムについて述べる。アルゴリズムにはオーバーレイネットワークのトポロジ情報、リソース量制約、および各モバイルノードの要求品質が入力として与えられる。コスト関数は計算資源消費量とネットワーク資源消費量の重み付き和であるが、これらの値の最小化はトレードオフの関係にある。この問題に対する最適解を得るためには、組合わせ最適化問題を解く必要があり、実用的な時間内に解を得ることは困難である。そこで本章では、この問題に対するヒューリスティックアルゴリズムの提案を行う。

提案アルゴリズムは、文献[7]で提案されている、シュタイナー木を計算するためのヒューリスティックアルゴリズムを拡張したものである。シュタイナー木とは、重み付きグラフ $G=(V, E, w)$ 上の頂点の集合 $S \subseteq V$ に対し、 S 上のスパニングツリーの中で、コストの和が最小のものをいう。

5.1 節では、シュタイナー木を計算し、これをもとに配送経路を生成する基本的なアルゴリズムについて述べる。次に、使用計算資源と使用ネットワーク資源をそれぞれ単独で最小化する 2 つのアルゴリズムについて述べ、最後に 5.2 節で、これらのアルゴリズムを一般化した、ハイブリッドアルゴリズム

について述べる。

5-1 シュタイナー木を用いた配送経路の構築

プロキシ p および p' について、 p' が p から直接ビデオを受信しているとき、 p を p' の親プロキシと呼び、 p' を p の子プロキシと呼ぶ。サーバ s からビデオを直接受信しているプロキシを根プロキシと呼ぶ。子プロキシを持たないプロキシを葉プロキシと呼ぶ。また、親プロキシが転送するビデオの品質が、子プロキシのビデオの要求品質を満たすとき、 p から p' への転送条件を満たすという。

以下で述べる各アルゴリズムでは、シュタイナー木の計算における各リンク上のコストとして物理ホップ数を考える。配送経路としてホップ数の小さいリンクを優先的に選択することで、物理ネットワークの帯域消費量を小さく抑えられるからである。各アルゴリズムは、シュタイナー木に基づく配送経路を決定した後、転送条件を満たすように、以下の手順により経路上の各プロキシの受信品質を調整する。

Step 1: 葉プロキシ p は、自身の受信品質 q_p を含むメッセージ r_p を親プロキシ p' に送信する。

Step 2: p' はすべてのメッセージを受信後、各受信品質 q_p と自身の受信品質 $q_{p'}$ を比較し、上位品質のものがあれば、 $q_{p'}$ をその品質に調整する。次に、 p' は親プロキシに要求品質 $q_{p'}$ を含むメッセージ $r_{p'}$ を送信する。

Step 3: 上記 Step 2 の操作を根プロキシまで繰り返す。

Step 4: 根プロキシは s にメッセージを送信する。

(1) 計算資源最小化アルゴリズム

計算資源消費量を最小化する場合（コスト関数における $\alpha = 1$ の場合）、各要求品質に対し、起動するトランスコードサービス数を1つにする必要がある。そのため、あるモバイルノード u の要求品質 q_u へのトランスコードサービスが、 u の接続プロキシ p 以外のプロキシ p' 上で実行されている場合は、 q_u のビデオを p' から p へ中継する。具体的には、以下の手順により配送経路を構築する。

Step 1: プロキシの集合を、利用可能計算資源の降順でソートする。ここで、 $SP = (sp_1, \dots, sp_{np})$ はソートされたリストを示す。

Step 2: モバイルノードからの要求品質の集合を必要な計算パワーの昇順でソートする。ここで、 $QR = (qr_1, \dots, qr_{mq})$ はソートされたリストを示す。

Step 3: sp_1 から順に、そのプロキシの計算資源の範囲内ですべての QR の要素を割り当てる。

Step 4: 同様に、 sp_j ($j \geq 2$) について残りの QR の要素を割り当て、 QR が空集合になるまで繰り返す。

Step 5: QR の要素を割り当てられたプロキシ間で、シュタイナー木を構築する。また、各品質から、それを必要とする他のプロキシの間においてもシュタイナー木を構築する。

Step 6: 5.1 節のアルゴリズムを使用して各プロキシの受信品質を調整する。

(2) ネットワーク資源最小化アルゴリズム

ネットワーク資源消費量を最小にする場合（コスト関数における $\alpha = 0$ の場合）、各要求品質に対するトランスコードサービスの起動数は、その品質を要求するモバイルノードが所属するプロキシ数となる。すなわち、各プロキシは、自分が担当する全てのモバイルノードへのトランスコードを実行する。具体的には、以下の手順により配送経路を構築する。

Step 1: モバイルノードを1つ以上担当する全プロキシ間でシュタイナー木を構築する。

Step 2: 5.1 節のアルゴリズムを使用して各プロキシの受信品質を調整する。

5-2 ハイブリッドアルゴリズム

要求品質 q へのトランスコードを行うプロキシの数を NP_q と表記する。コスト関数において、 $\alpha = 1$ の場合はすべての q について $NP_q = 1$ であり、一方、 $\alpha = 0$ の場合は $NP_q = (q$ を要求するモバイルノードが所属するプロキシ数) となる。全モバイルノードからの要求品質の集合の内、 NP_q の最大値を NP_{max} とする。ハイブリッドアルゴリズムでは、 NP_q の値を 1 から NP_{max} の範囲で変化させてコスト関数の値を計算し、値が最小となる解を選択する。これにより、計算資源のコストが大きい場合には、計算資源最小化アルゴリズムに近い解が得られ、ネットワーク資源のコストが大きい場合には、ネットワーク資源最小化アルゴリズムに近い解が得られる。また、両コストがバランスしている場合には、両アルゴリズムの中間的な解が得られることが期待される。なお、 NP_q はすべての要求品質について同じ値を使用する。アルゴリズムでは、Step 1 から Step 4 まで各 i について 1 から NP_{max} まで繰り返し、コスト関数の値が最小となる解をそれらの中から選択する。

Step 1: 各品質ベクトル q に対して、 $NP_q = (q$ を要求するモバイルノードが所属するプロキシ数) とする。ここで、プロキシ x で品質 q を要求しているモバイルノードの数を $N_{q,x}$ とし、 q へのトランスコードを行うプロキシの集合を PS_q とする。 PS_q の要素として、 i 個のプロキシが $N_{q,x}$ の降順で選択

される。

Step 2: 上記 Step 1 で選択されたプロキシの集合を接続するシュタイナー木を計算する。

Step 3: 各品質 q に対して、シュタイナー木を計算する。 $i < NP_{max}$ のとき、 i 個のプロキシが同時にトランスコードを行う。 その際、ネットワーク資源を削減するため、シュタイナー木を物理ホップ数を基準として計算し、木に沿って複数のモバイルノードに同じ品質のビデオを配信する。

6 アルゴリズムの具体例

図 2 に上記 3 つのアルゴリズムの具体例を示す。 図において、プロキシ x が親プロキシから受信すべき品質、リンクを通して中継されるビデオの品質ベクトルをそれぞれ q_{max_x}, q_{str} とする。

図 2(a) は計算資源最小化アルゴリズムが適用された場合の具体例である。 図のように、モバイルノード u_1, \dots, u_6 と要求品質 150, 200, 300, 400 (簡単のためビットレートについてのみ考える) が与えられた場合を考える。 このアルゴリズムでは、各品質について 1 つのトランスコードサービスが 1 つのプロキシで実行される。 そのため、トランスコードサービス T_1, T_2, T_3, T_4 がそれぞれプロキシ p_1, p_2, p_3, p_2 で実行されている。 例えば、 u_3 についてみると、要求品質 300 に対するトランスコードサービス T_1 は u_3 が直接接続している p_1 に存在するため、 p_1 から直接ビデオを受信できる。 一方、 u_4 が要求する品質 200 に対するトランスコードサービス T_2 は p_2 に存在するため、 p_3, p_1 を経由して中継される。 このアルゴリズムでは複数のビデオストリームが各オーバーレイリンクで中継される可能性がある。

図 2(b) はネットワーク資源最小化アルゴリズムの具体例である。 このアルゴリズムでは、各プロキシは自分に直接接続しているすべてのモバイルノードに対しトランスコードサービスを実行する。 例えば、 u_1, u_2 は p_3 に直接接続しているため、 p_3 はトランスコードサービス T_2, T_3 を実行し、中継を行う。 このアルゴリズムでは、ただ一つのビデオストリームが各オーバーレイリンクを通して中継される。

ハイブリッドアルゴリズムは上記 2 つを同時に含む一般形であり、コスト関数で表現されている計算資源消費量とネットワーク資源消費量の重み付き和を最小にしようとするものである。

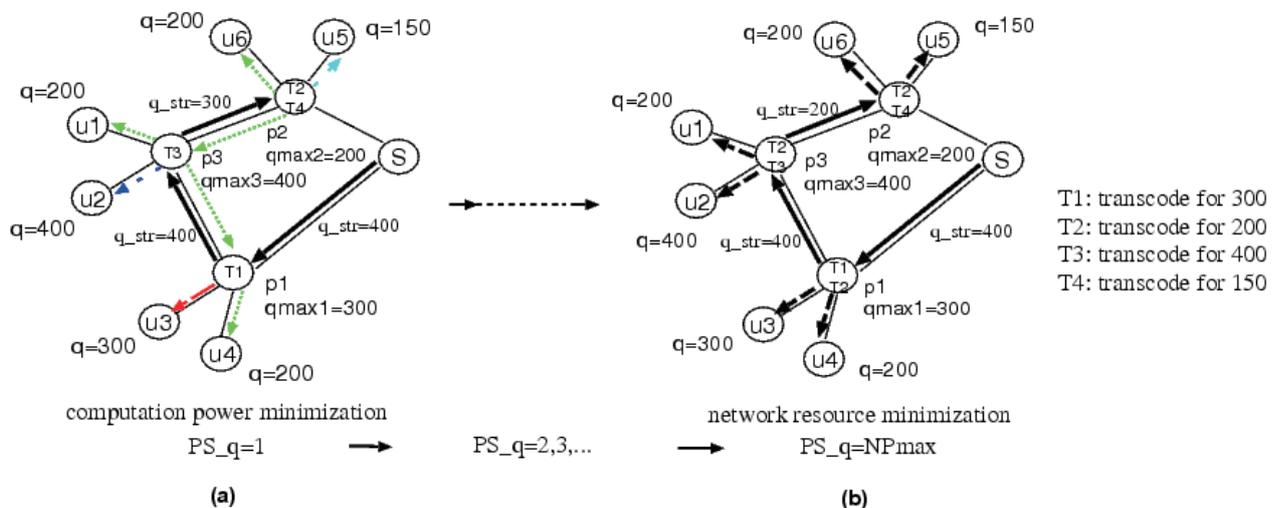


図 2: ハイブリッド手法の具体例

7 性能評価

提案手法の有効性を検証するため、シミュレーション、および PlanetLab[8]を使用した広域分散ネットワーク上での評価を行った。シミュレーションでは、ハイブリッドアルゴリズムの有効性を検証するため、当アルゴリズムにより達成されるコストを求め、計算資源最小化、およびネットワーク資源最小化アルゴリズムとの比較を行った。また PlanetLab を使用した評価では、ストリーム配送開始までの遅延時間が実用上十分少ないことを検証するため、配信サーバがモバイルノードの要求品質を受信してから配信が開始されるまでの経過時間を測定した。

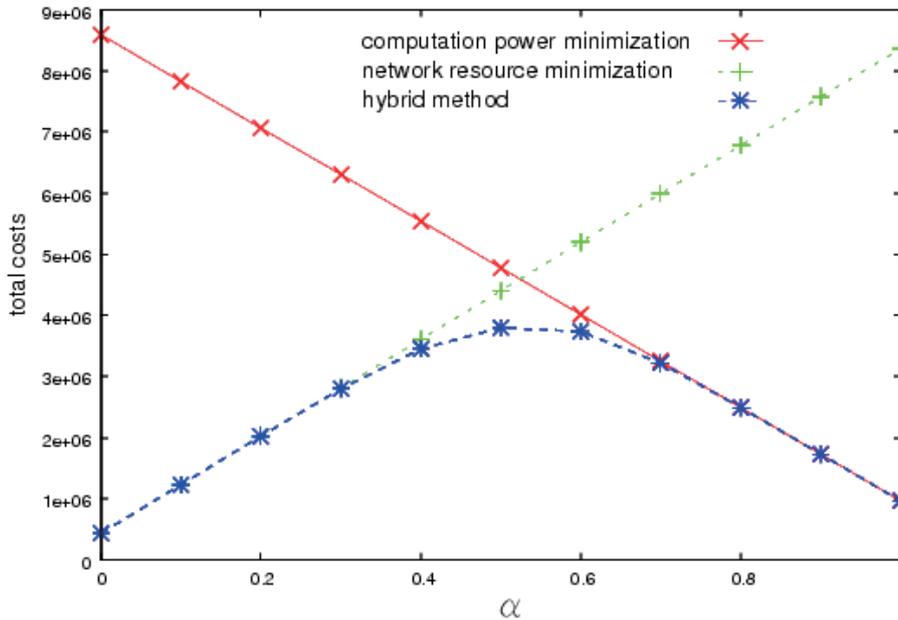


図3: 各アルゴリズムの α の変化に対するトータルコスト

7-1 シミュレーションによる評価

シミュレーションの設定を以下に示す。プロキシ間のオーバーレイネットワークとして、GT-ITM[9]のLocality modelを用いてプロキシ数 50 のトポロジを生成した。プロキシ間の各リンクに対し、物理ホップ数を区間 [1: 10] 上の一様乱数で決定した。各モバイルノードの要求品質は、最大品質 640×480 画素、30 フレーム毎秒、最低品質 80×60 画素、5 フレーム毎秒の範囲から一様乱数で決定した。

(1) 各手法で達成されるコスト

各手法の性能を比較するため、提案した3つのアルゴリズムで達成されるコストを求めた。ノード数は 3,000 とした。本実験では、各アルゴリズムで達成されるコストの比較が目的であるため、利用可能計算資源、利用可能帯域は十分であると仮定した。実験結果を図3に示す。

図3より、計算資源最小化、およびネットワーク資源最小化アルゴリズムは、それぞれ $\alpha = 1.0$, $\alpha = 0.0$ において最も低いコストを達成できることが分かる。またハイブリッドアルゴリズムは、 α の値に関わらず、他の手法と同等、もしくはより低いコストを達成できており、特に $\alpha = 0.5$ 付近で他の手法よりもコストを低く抑えられることが分かる。

(2) アルゴリズムの計算時間

各アルゴリズムの計算時間を評価するため、配送経路の探索を完了するまでにかかる時間を計測した。実験では、Pentium 4 2.4 GHz, 1GB の RAM を搭載した PC を使用した。実験結果を表1に示す。

表1: 配送経路の計算に要する時間 (秒)

ユーザノード数	100	500	1000	2000	3000
計算資源最小	0.03	0.11	0.15	0.37	0.63
ネットワーク資源最小	0.03	0.11	0.20	0.51	0.71
ハイブリッド	0.12	1.03	2.43	6.03	9.77

表1より、計算資源最小化、およびネットワーク資源最小化アルゴリズムの計算時間はほぼ同等であることが分かる。また、ハイブリッドアルゴリズムは他の2つのアルゴリズムよりも長時間を要するが、ノード数が 3,000 程度までならば実用上問題ない程度の遅延時間で計算を完了できることが分かる。各手法で達成されるコストを考慮すると、計算資源最小化、およびネットワーク資源最小化アルゴリズムは、 α がそれぞれ 0.0 ~ 0.3, 0.7 (1.0 付近の時に使用すれば、コストを低く抑え、かつ配送経路を高

速に計算可能であることが分かる。

7-2 プロトタイプの実装と評価

提案システムのプロトタイプを実装し、PlanetLab を用いて実験を行った。ストリーム配信開始時の遅延時間を求めるため、ビデオ配信サーバがモバイルノードの要求を受信してから、ストリームの配信が開始されるまでにかかる時間を測定した。ストリーム配信開始時の遅延時間は、(a) サーバにおける配送経路計算に必要な時間、(b) 各プロキシ間でのコネクション確立に必要な時間、(c) サーバがストリームの配信を開始後、葉プロキシがバッファリングを終えるまでの時間、の総和となる。上記(a)に関しては7.1節で求めたので、ここでは (b), (c) について測定を行う。

提案システムにおいて、配送経路はサーバから順に木に沿って構築されるため、木の高さに応じた遅延が生じることが考えられる。そのため、実験では任意6カ所の PlanetLab ホストを直列に接続し、各リンク上での遅延時間を測定した。実験では、まず、各ホストのペアに対し配送経路情報の送信後、コネクションの確立を終えるまでの時間 (Resp_Time) を測定した。次に、各ホストのペアに対し、ホスト間でストリームの送受信を行った。その際、受信側の入力バッファのアンダーフローが生じないために最低限必要となるバッファリング時間 (Buff_Time) を測定した。実験結果を表2に示す。

表2：ストリーム配信開始時に必要な時間（ミリ秒）

Path	Resp_Time	Buff_Time
米 ⇒ 米	252.4	1000
米 ⇒ 米	398.4	1000
米 ⇒ 日	674.2	2000
日 ⇒ 韓	213.2	1000
韓 ⇒ 豪	1631.2	3000
全体(米 ⇒ 豪)	3081.2	8000

表2より、オーバーレイリンク上でのホップ数が5の場合、サーバが配送経路情報を送信してから、末端のプロキシがコネクションの確立を終えるまでに必要な時間は約3秒、末端のプロキシがバッファリングを終えるまでの時間が約8秒であり、全体で11秒の遅延時間が生じることが分かる。提案手法において、ノード数3,000としたときにアルゴリズムが出力する配送経路長の最大値は10ホップであった。この場合、約2倍の遅延時間(22秒)が必要となる。以上より、ノード数が3,000程度の比較的大きなシステムにおいても、実用上問題ない程度の遅延時間でストリームの配信を開始できることが分かる。

8 おわりに

本報告では、異なる要求品質を持つ多数の無線移動端末ユーザへのビデオマルチキャスト手法を提案した。提案方式の特徴は次のとおりである。ユーザに対する利点： 端末のバッテリー残量、計算能力、通信速度、画面サイズ、などの制約に基づいてビデオの要求品質を決定でき、ビデオ再生中の要求品質の変更や、AP間のハンドオフに対しても、ビデオを途切れることなく配信可能である。サービスプロバイダに対する利点： 利用するCDN上の、サーバ、プロキシ、AP、それらの間のオーバーレイリンクの構成を与え、各プロキシの利用可能計算資源、各オーバーレイリンクの利用可能帯域を指定するだけで、CDNの消費リソース量を抑えた配送経路を求めることができる。

シミュレーションによる評価から、 α が0.5程度の場合において、ハイブリッド手法が妥当な計算時間で帯域と計算資源のコストのトレードオフを考慮しつつ、リソース消費量の少ない配送経路を求められることを確認した。また、PlanetLabを用いた評価により、実用上問題ない程度の遅延時間でストリームの配信を開始できることを確認した。

ハイブリッドアルゴリズムでは、品質ごとに実行するトランスコードサービスの数を全ての品質に対し同時に変更し、その中で目的関数を最小化している。探索空間を若干広げることで、さらに資源消費量を減らす方法を考案することは今後の課題である。

【参考文献】

- [1] Hayder M. Radha, Mihaela van der Schaar, and Yingwei Chen, "The MPEG-4 Fine-Grained-Scalable video coding method for multimedia streaming over IP," IEEE Trans. on Multimedia, 3, 1, pp.53-68, 2001.
- [2] Brett J. Vickers, Celio Albuquerque, and Tatsuya Suda, "Source-Adaptive Multilayered Multicast Algorithms for Real-Time Video Distribution," IEEE/ACM Trans. on Networking, 8, 6, pp.720-733, 2000.
- [3] Yoram Bernet, James Binder, Steven Blake, Mark Carlson, Brian E. Carpenter, Srinivasan Keshav, Elwyn Davies, Borje Ohlman, Dinesh Verma, Zheng Wang, and Walter Weiss, "A framework for differentiated services", IETF working draft, 1999.
- [4] Jingwen Jin, and Klara Nahrstedt, "QoS service routing in one-to-one and one-to-many scenarios in next-generation service-oriented networks," Proc. of the 23rd IEEE Int'l. Performance Computing and Communications Conf. (IPCCC 2004), 2004.
- [5] Jin Liang, and Klara Nahrstedt, "Service Composition for Advanced Multimedia Applications," 12th Annual Multimedia Computing and Networking (MMCN 2005), 2005.
- [6] Jingwen Jin, Klara Nahrstedt, "On Exploring Performance Optimizations in Web Service Composition," Proc. of ACM/IFIP/USENIX Int'l. Middleware Conf. (Middleware 2004), 2004.
- [7] L. Kou, George Markowsky, and Leonard Berman, "A fast algorithm for Steiner trees," Acta Informatica, 15, pp.141-145, 1981.
- [8] "PlanetLab," <http://www.planet-lab.org>
- [9] Ellen Zegura, Kenneth Calvert, Samrat Bhattacharjee, "How to Model an Internetwork," 15th Annual Joint Conf. of the IEEE Computer and Communications Societies (INFOCOM 1996), 1996.

〈発表資料〉

題 名	掲載誌・学会名等	発表年月
An Energy-Aware Video Streaming System for Portable Computing Devices	7th International Conference on Mobile Data Management (MDM'06), Demo session	2006年5月
A Delivery Method for Compound Video Playback in Wireless Network	Proc. of 2006 International Workshop Parallel and Distributed Multimedia Computing, LNCS4331, pp.803-812	2006年12月
MANET 上での効率良いビデオ配信を目的とした準最適マルチキャスト配送木の分散構築法	情報処理学会論文誌, Vol.48, No.2, pp.483-493	2007年2月
A Method for Distributed Computation of Semi-Optimal Multicast Tree in MANET	IEEE Wireless Communications and Networking Conference (WCNC 2007)	2007年3月