社団法人　情報処理学会　研究報告
IPSJ SIG Technical Report

2005－MBL－35（11）
2005－ITS －23（11）
2005／11／17

多様な要求品質を持つ移動端末ユーザへのリソース効率の良いビデオ配信方式

山岡 修一　孫 タオ　玉井 森彦　安本 慶一　柴田 直樹† 伊藤 実

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

あらまし

本稿では，異なる要求品質を持つ多数無線移動端末ユーザへのビデオ同時配信手法を提案する．提案手法では，携帯端末の制約を満たし，かつ，システム全体での使用リソース量が出来るだけ少なくなるようなプロキシ間の配送経路を構築する．各プロキシは複数のトランスコードサービスと中継サービスを実行可能であり，ビデオ配信サーバからのオリジナルビデオをそれらのサービスを使用して様々な品質に変換し，配送経路に沿ってユーザへの中継を行う．配送経路はオーバーレイネットワーク上の物理リンクで消費される帯域とトランスコードに必要な計算資源の和を出来るだけ少なくするように構築される．提案手法では，ユーザの移動により近隣のアクセスポイントが変化してもシームレスな再生が可能であり，また，要求品質の動的な変更も可能である．シミュレーションにより，提案手法が有用であることを示した．

Resource-Aware Video Multicast to Multiple Wireless Mobile Users with Different Quality Requirements

Shuichi Yamaoka　Tao Sun　Morihiko Tamai　Keiichi Yasumoto　Naoki Shibata† Minoru Ito

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

Abstract

In this paper, we propose a method to deliver video for multiple wireless mobile users with different quality requirements. In the proposed method, we assume that there are several proxies and wireless access points in the network. Overlay links are established between these nodes, and certain amounts of bandwidths are reserved in advance. Each proxy is capable of executing multiple transcoding services and forwarding services. The original video sent from the server is transcoded into various quality by these services, and delivered to user nodes with the required quality along the service delivery paths. In this paper, we propose an algorithm to construct the service delivery paths which minimize the weighted sum of computation power for transcoding and the bandwidth consumed on physical links on the overlay network. The proposed method is capable of handling user nodes moving to a range of another access point. Also, users of the proposed system can change quality requirements anytime. Through experiments with simulations, we show the usefulness of our method.

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，ノート PC，PDA，携帯電話など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り，それぞれ画面サイズ，処理能力，バッテリ容量，通信速度が異なる上，移動に伴いインターネットへの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) 画面サイズ，CPU 性能，バッテリ残量，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（以降，要求品質と呼ぶ）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) 携帯端末が移動して近隣の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，が重要となる．また，サーバから多数の端末にビデオをユニキャスト配信するためには多くのリソース（サーバの CPU パワーとネットワーク帯域）が必要であるため，(5) コンテンツ配信ネットワーク（CDN）で利用可能なリソース量の制約を満たし，かつ，必要リソース量をできるだけ少なくすること，を実現することも重要となる．

異なった要求品質を持つ複数ノードへの既存のビデオ配信方式では，MPEG-4 FGS[1] などの階層符号化に基づいたもの [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メモリや計算量が必

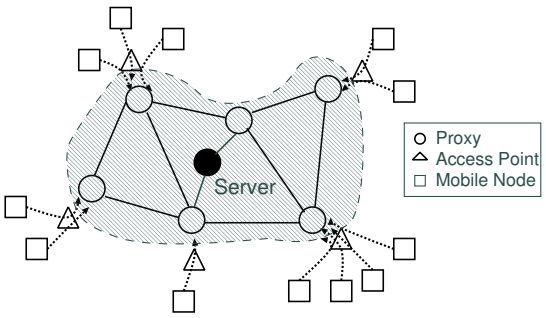


図 1: ネットワーク環境

要となる．また，ビデオの品質に関わるパラメタ（画像サイズ，フレームレート，ビットレート）の内，どれか一つの値しか調整できないという欠点がある．

本稿では上記 (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．提案方式の想定環境は以下のとおりである．(i) CDN として，図 1 のように，複数のプロキシおよびビデオ配信サーバから成るオーバレイネットワークが与えられるとする．ここでプロキシ間のオーバレイリンクは，DiffServ[3] などを用いてある容量の帯域があらかじめ確保されていると仮定する．(ii) 各プロキシは利用可能なリソースの範囲内で，複数のトランスコードサービスと中継サービスを実行できる．(iii) 各プロキシには最大一つの無線アクセスポイント（AP）が接続される．(iv) 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な，各プロキシの入出力品質，および，ビデオ配信サーバ，プロキシ，ユーザ端末間の接続パスを決定する．また，上記 (1) に関しては，我々が文献 [4] で提案している省電力ビデオ配信手法を用いて実現する．この手法では，各ビデオセグメントの再生品質（すなわち，画像サイズ，フレームレート，ビットレートの組）は，(a) ユーザの設定項目，および，(b) 端末の制約条件 から自動的に決定される．ここで，ユーザの設定項目は，ビデオの再生希望時間，ビデオセグメント間の相対的な重要度，各ビデオセグメントの希望再生特性（画面サイズとフレームレートのどちらを重視するかを示す値）であり，端末の制約条件は，バッテリ残量，通信帯域，計算能力，画面サイズである．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，定期的にサービス合成パス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，一時的に最も希望品質に近いものを受信させる．これにより，AP の切り替えが発生した場合でも，ストリームを途切れることなく再生可能となる．

シミュレーションによる評価により，提案方式によって求めたサービス合成パスが，ユーザの希望再生品質を満足しつつ CDN で消費するリソース量をできるだけ少なく抑えられることを確認した．

1.1 関連研究

文献 [5, 6, 7] では本稿と同様に，多種多様な端末および異なるネットワーク環境下における多数のユーザに対し，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 [5] では，各プロキシで実行されるサービスコンポーネントの数および種類はあらかじめ決められており，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 [6] では，ユーザの要求品質はあらかじめ与えられているものとしており，動画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 [7] はプロキシ間を接続するマルチキャスト木と，ユーザノードから成るローカルマルチキャスト木をリソース消費量が最小となるように構築し，それらを連結することで効果的なサービス配送パスを計算するアルゴリズムを提案している．この手法では，プロキシを結ぶ各マルチキャスト木は配送経路においてより物理ホップ数が少なく，より利用可能帯域が大きいプロキシにパスを切り替える．しかし，プロキシ間のオーバーレイリンクにおける複数の配送経路でのリソース消費量の最適化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．本稿で提案する提案手法では，これら既存手法とは異なり，動画再生中における希望再生品質の変更や接続 AP の変更も考慮しつつ，CDN における消費リソース量の少ないサービス合成パスを求めることができる．

2 問題定義

本章では，まず提案手法が対象とする環境と仮定について述べ，次に問題の形式的な定義を与える．

2.1 対象とする環境と仮定

ビデオ配信サーバ，オーバーレイネットワーク，モバイルノード，サービス，プロキシに関し，以下のような環境を仮定する．

（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．携帯端末ユーザは，ビデオ配信サーバの送信するビデオと同じか，もしくはそれ以下の品質のビデオを要求する．テレビ放送と同様に，全てのユーザは同じ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，複数の無線アクセスポイント（AP）からなるオーバレイネットワークが与えられる（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 間を接続する各オーバレイリンク上では DiffServ [3] などの手法により，あらかじめ使用可能帯域が確保されているものとする．また，各プロキシに接続されている AP の数は，便宜上，高々一つとする（複数の AP を接続する場合，伝送速度が大きな一つの AP と考える）．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，無線 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（同一 LAN セグメントなど）に配置され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，処理能力，最大伝送速度などが異なる多数の携帯端末 (ラップトップ PC, PDA，携帯電話など) が存在する．各端末は，現在位置をカバーする無線範囲に対応する 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対応する AP は一意に定まり，各 AP は，端末の移動時に AP が切り替わったことを瞬時に認識可能とする．また，伝送速度は電波状況により刻々と変化し，モバイルノード間での通信は行わないものとする．

（4） 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．それぞれのサービスの実行に必要な計算量およびネットワーク帯域は，サービスへの入出力ビデオ品質または入出力ビットレートから算出可能である．

（5） プロキシ：各プロキシは資源の許す限り，複数のサービスを実行できる．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レイリンクの制限に従う．本稿では簡単のため，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ものとする．

2.2 問題の形式的定義

2.2.1 諸定義

オーバーレイネットワーク

ビデオ配信サーバを s とする．プロキシの集合を $P = \{p_1, p_2, ..., p_{np}\}$ とする．モバイルノードの集合を $U = \{u_1, u_2, ..., u_{nu}\}$ とする．モバイルノード $u \in U$ が，ある AP を経由して，その AP に接続しているプロキシ $p \in P$ と通信可能なとき，u，p 間のオーバーレイリンクを (u, p) と表記する．ある時刻でのプロキシとモバイルノード間のオーバレイリンクの集合を W とする．W の要素は，モバイルノードの移動に伴い変化する．また，$\{s\} \cup P$ の要素間のオーバレイリンクの集合を F とする．全てのノードの集合を $V = P \cup U \cup \{s\}$，全てのオーバレイリンクの集合を $E = W \cup F$ とし，オーバーレイネットワークを無向グラフ $G = (V, E)$ で表す．各プロキシ $p_i \in P$ の利用可能計算資源を $c\_avail(p_i)$ とする．$p_i \in P$，$p_j \in \{s\} \cup P$ に対し，オーバーレイリンク (p_i, p_j) の利用可能帯域を $b\_avail(p_i, p_j)$，物理ネットワーク上でのホップ数を $hop(p_i, p_j)$ とする．前節で述べたように，各オーバレイリンク $w \in W$ の利用可能帯域はモバイルノード u ごとに最大値 $bmax_u$ が決まっており，受信可能伝送速度 bw_u は電波状況などにより $bmax_u$ より小さい範囲で変化するとする．オーバーレイネットワークの構成例を図 2 (a) に示す．

トランスコードサービスの実行に必要な計算資源

ビデオの品質は，画像サイズ，フレームレート，ビットレートの 3 つの値から成るとする．品質 q に対し，画像サイズ，フレームレート，ビットレートの各値を $q.s$，$q.f$，$q.b$ と表記する．以下では q を品質ベクトルとよび，$q = (q.s, q.f, q.b)$ と表記する．品質ベクトルが q のビデオから q' のビデオへのトランスコードに必要な計算資源は，品質ベクトルが q のビデオのデコードに必要な計算資源と，品質ベクトルが q' のビデオのエンコードに必要な計算資源との和によって表されるとする（本稿では，単一のコーデックを使用するものと仮定）．また，デコード，エンコードに必要な計算資源は，文献 [4] に基づき，単位時間当りに処理すべき画素数に比例すると仮定する．以上より，τ_d，τ_e をある定数値としたとき，品質ベクトル q のビデオのデコードに必要な計算資源，および，品質ベクトル q' のビデオのエンコードに必要な計算資源はそれぞれ次式で表される．

$$r_{decode}(q) = \tau_d (q.s \times q.f) \quad (1)$$

$$r_{encode}(q') = \tau_e (q'.s \times q'.f) \quad (2)$$

オーバーレイネットワークの制約

ビデオ配信サーバから各モバイルノードへのストリームの配送は，ビデオ配信サーバとモバイルノード間を接続するプロキシの列，および，各プロキシへの入出力品質ベクトルを決定することで実現される．このとき，各配送経路上では，各プロキシの利用可能計算資源と各オーバレイリンクの利用可能帯域に対する制約が満たされる必要がある．図 2 (a) のオーバーレイネットワークにおいて，$u_1, u_2, u_3, u_4, u_5, u_6, u_7$ をモバイルノードとし，それぞれの要求品質が $q_1, q_2, q_2, q_3, q_3, q_4, q_5$ であるときのストリーム配送の実現例を図 2 (b) に示す（ただし，$q_i.s \le q_j.s \wedge q_i.f \le q_j.f \wedge q_i.b \le q_j.b$, $i > j$）．

各ノード $v \in V$ に対し，v が受信する全ストリームの品質ベクトルの集合を $R(v)$ とする．ただし特別な場合として，$v = s$，$v = u$ のとき，それぞれ $R(s) = \{q_{orig}\}$，$R(u) = \{q_u\}$ とする（q_{orig} はサーバ s が保持するビデオの品質，q_u はモバイルノード $u \in U$ の要求品質）．例えば，図 2 (b) では，$R(p_5) = \{q_1, q_4\}$，$R(u_1) = \{q_1\}$，$R(s) = \{q_0\}$ となる．

ノード $v \in P \cup U$ において，あるノード v' から v へ品質ベクトル $q' \in R(v)$ のビデオの送信が行われるとき，v が受信するビデオの品質ベクトル q は，条件 $q.s \le q'.s \wedge q.f \le q'.f \wedge q.b \le q'.b$ を満たす必要がある．また，このときのモバイルノード v，v' の関係を**転送関係**とよび，$(v', q') \to (v, q)$ と表記する．

転送関係 $(v', q') \to (v, q)$ において，$q' \neq q$ ならば，v' 上で品質ベクトル q' のビデオから q のビデオへのトランスコードサービスが実行される．一方，$q' = q$ ならば，v' 上で v へのストリームの中継サービスが実行される．プロキシ p_i で実行されるトランスコードサービスへの入力品質ベクトルの集合を $D(p_i) = \{q \mid (p_i, q) \to (v, q'),\ v \in P \cup U,\ q \neq q'\}$，出力品質ベクトルの集合を $E(p_i) = \{q' \mid (p_i, q) \to (v, q'),\ v \in P \cup U,\ q' \neq q\}$ と表記する．

計算資源消費量と帯域消費量

プロキシ p_i の計算資源消費量を $c\_cons(p_i)$，オーバーレイリンク $\{p_i, p_j\}$ 上の帯域消費量を $b\_cons(p_i, p_j)$ と表記する．$c\_cons(p_i)$，$b\_cons(p_i, p_j)$ をそれぞれ以下のように定義する．

$$c\_cons(p_i) = \sum_{q \in D(p_i)} r_{decode}(q) + \sum_{q' \in E(p_i)} r_{encode}(q')$$

$$b\_cons(p_i, p_j) = \sum_{brate \in \{q'.b \mid (p_i, q) \to (p_j, q')\}} brate + \sum_{brate \in \{q'.b \mid (p_j, q) \to (p_i, q')\}} brate$$

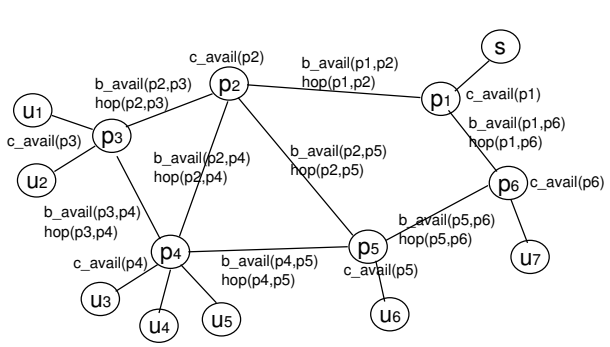
2.2.2 問題定義

オーバーレイネットワーク G，ビデオ配信サーバの送信するビデオの品質ベクトル q_{orig}，各ユーザノード $u_i \in U$ の要求品質ベクトル q_{u_i} が与えられたとき，各プロキシ p_i の受信品質ベクトルの集合 $R(p_i)$，および，全ての転送関係 $(v', q') \to (v, q)$ $(v' \in \{s\} \cup P, v \in P \cup U)$ を求める．ただし，以下に示す制約条件 (3)–(6) を全て満たすものとする．

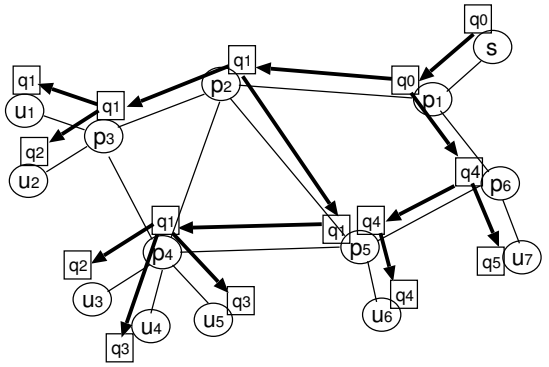
$$\text{for each } (v', q') \to (v, q),\quad q.s \le q'.s \wedge q.f \le q'.f \wedge q.b \le q'.b \quad (3)$$

$$\text{for each } u_i \in U,\quad \exists(s, p_{j1}) \exists(p_{j1}, p_{j2}) ... \exists(p_{jk}, u_i)\ s.t.\quad (s, q_{orig}) \to (p_{j1}, q_1) \wedge (p_{j1}, q_1) \to (p_{j2}, q_2) \wedge \ldots \wedge (p_{jk}, q_{jk}) \to (u_i, q_{u_i}) \quad (4)$$

$$\text{for each } p_i \in P,\ c\_cons(p_i) \le c\_avail(p_i) \quad (5)$$



(a)



(b)

図 2: オーバーレイネットワークの構成例

$$\text{for each } (p_i, p_j) \in F,\ b\_cons(p_i, p_j) \leq b\_avail(p_i, p_j) \tag{6}$$

制約条件 (3) は，各転送関係 $(v', q') \rightarrow (v, q)$ において，v' は v よりも同等以上の品質のビデオを受信していなければならないことを表す．制約条件 (4) は，サーバ s から各モバイルノード $u \in U$ に対し，s と u を接続するオーバーレイリンクの列が存在しなければならないことを表す．制約条件 (5)，(6) は，各プロキシ $p_i \in P$ で消費する計算資源と各オーバーレイリンク (p_i, p_j)（または，(s, p_i)）で消費する帯域が，あらかじめ定められた制限を越えてはならないことを表す．

一般に，上記の制約を満足する解は複数個存在するが，オーバーレイネットワークの利用効率の観点から，各プロキシで消費される計算資源，および，物理ネットワーク上での消費帯域ができるだけ少なくなるような解を求めることが望ましい．そこで，次の目的関数の値を最小化する解を求める．

$$\text{Min } (\alpha \sum_{p \in P} c\_cons(p) + (1-\alpha) \sum_{\{p_i, p_j\} \in F} b\_cons(p_i, p_j) \times hop(p_i, p_j)) \tag{7}$$

式 (7) の第一項は，プロキシによって消費される計算資源の総和を表し，第二項は，プロキシ間を接続するオーバーレイリンク上で消費される，(物理ホップ数を考慮に入れた) 帯域消費量の総和を表す．また α は，計算資源と帯域のどちらがよりコストとして高いかを設定するのに使用する定数値である．

3 システムの動作

本章では，提案手法におけるシステム全体の動作の詳細について述べる．3.1 節では，各モバイルノードにおけるバッテリ残量などを考慮した要求品質の決定方法について述べる．3.2 節では，同程度の要求品質をグループ化し，要求品質の総数を減らすための前処理について述べる．3.3 節では，提案方式が使用するノード間の通信プロトコルについて述べる．

3.1 バッテリ残量に基づいた要求品質の決定

モバイルノード $u \in U$ の利用可能帯域を bw_u，計算能力を cp_u，画面サイズを ds_u，バッテリ残量を en_u とする．ノード u の要求するビデオの品質ベクトルは以下の制約条件を満たさなければならない．

$$q_u.b \leq bw_u$$
$$q_u.s \leq ds_u$$
$$\tau_d \times (q_u.s \times q_u.f) \leq cp_u$$

さらに，モバイルノード u のバッテリ残量と希望視聴時間 T_u を考慮して品質ベクトルを求める．我々は文献 [4] において，T_u，en_u，および端末固有の電力消費特性から，品質ベクトルを求める方法を提案している．ビデオが生放送ではなく録画済のものであり，ビデオの各シーンの内容があらかじめ分かっている場合には，各シーンを自動ラベリングツール [8] などを用いてビデオセグメントに分類することができる．この場合には，ビデオセグメントごとに，そのセグメントの相対的な重要度と，希望再生特性（画面サイズとフレームレートのどちらを重視するかを示す値）をユーザから入力してもらうことで，ビデオセグメントごとに異なった品質で再生可能である．以下ではビデオセグメントごとの再生品質の決定方法について，詳細に述べる．

あるビデオのカテゴリ（シーンの種類）の集合を $C = \{c_1, ..., c_m\}$ とする．ユーザは各カテゴリ $c_i \in C$ に対する重要度 p_i を指定する（p_i は 1 以上の整数）．バッテリ残量 en_u を各カテゴリ c_i の長さ T_i と重要度 p_i の積で比例配分したもの，すなわち $\frac{en_u \cdot p_i \cdot T_i}{\sum_{j=1}^{m} p_j \cdot T_j}$ が，カテゴリ c_i の再生に使用されるバッテリ量として決定される．

また同じ消費電力でも，画像サイズ，フレームレート，ビットレートの割合により，特性の異なる再生品質を実現できる．そこで，カテゴリ c_i における再生品質の希望特性を，画像サイズとフレームレートの比 $q_u.s/q_{orig}.s : q_u.f/q_{orig}.f = x : y$ としてユーザから受け取る（ここで，q_{orig} は配送元におけるビデオの品質ベクトル，x, y は 1 以上の整数）．カテゴリ c_i が利用可能なバッテリ量，および c_i の希望再生特性から，文献 [4] の方法を用いてカテゴリ c_i に属するビデオセグメントの再生品質を決定できる．

例として，3 つのカテゴリ $\{shoot, play, other\}$ に分類されたビデオセグメントから成る，サッカーのビデオを考える．*shoot* は高品質で, *play* は中程度の品質で，*other* は低品質で再生し，また，各カテゴリの再生特性として，*shoot* は画像サイズ，フレームレートともに重要であり，*play* はフレームレートがより重要であり，*other* は画像サイズがより重要であると指定したい場合は，以下の表のようになる．

カテゴリ	重要度	$\frac{q_u.s}{q_{orig}.s}$	$\frac{q_u.f}{q_{orig}.f}$	長さ
shoot	4	1	1	10min
play	2	1	2	35min
other	1	2	1	15min

この場合，$shoot$ に割り当てられるバッテリ量は $\frac{en_u \cdot 4 \cdot 10}{4 \cdot 10 + 2 \cdot 35 + 1 \cdot 15} = \frac{8}{25} en_u$ となり，他のセグメントに比べてより高い品質で再生される．

録画済のビデオに関しては，上記の方法で各ビデオセグメントの再生品質が求められる．一方，生放送に関しては，ビデオセグメントへの分類や各セグメントの長さをあらかじめ知ることができないため，上記の方法をそのまま用いることはできない．そこで生放送に関しては，ユーザは，再生中の任意の時点で何段階か（例えば，高，中，低）の品質から選択できるようにする．中程度の品質が選択された場合は，現在のバッテリ残量と残りの再生希望時間から再生品質を決定する．また，高品質，低品質が選択された場合は，中程度の品質が消費する電力量から，あらかじめ決められた割合（例えば，20%）を増減させた値を使用可能バッテリ量とすることで再生品質を決定する．この場合，次回の品質変更時もしくは一定時間経過後に中程度の品質で消費される電力量を再計算し，再生品質の更新を行う．

3.2 要求品質のグループ化

同程度の要求品質に対し，それぞれ異なった品質としてトランスコードを行うことは，計算資源の利用効率の観点から望ましくない．そこで，同程度の要求品質をグループ化して同一品質とすることで，要求品質の総数を減少させる．具体的には，(1) モバイルノード u の要求品質 $q_u.s, q_u.f, q_u.b$ に対して，品質許容範囲 r を設定する．r の具体的な値は，ユーザの満足度に関する制約から決定される．例えば，ユーザの満足度を 0.95 とした場合，品質許容範囲 r は $1 - 0.95 = 0.05$ となる．(2) 要求品質の集合を S とする．(3) 各モバイルノード u について，要求品質の集合 S_u を求める．ここで S_u は，S の要素 $q_{u'} = (q_{u'}.s, q_{u'}.f, q_{u'}.b)$ の内，$(1-r) \cdot q_u.s \leq q_{u'}.s \leq q_u.s \wedge (1-r) \cdot q_u.f \leq q_{u'}.f \leq q_u.f \wedge (1-r) \cdot q_u.b \leq q_{u'}.b \wedge q_u.b$ を満たすものの集合である．(4) ステップ (3) で求めた S_u の集合の内，要素数が最大のものをグループ化し，S_u の要素を S から除く．また，グループ化後の品質は，S_u の要素の内，最も品質の低いものとする．(5) ステップ (3) と (4) を S が空集合になるまで繰り返す．

3.3 ビデオ配信プロトコル

ビデオ配信プロトコルは，ビデオ配信開始手順，配信経路の再構築手順，モバイルノードの加入・離脱手順，モバイルノードの移動による AP 間のハンドオフ手順から成る．

3.3.1 ビデオ配信開始手順

（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，3.1 節の方法で求めた要求品質 q_u を自分の属する AP に接続するプロキシ p に送信する．

（2） 各プロキシ p は，受信した要求をビデオ配信サーバ s に送信する．

（3） s は，受信した全要求品質から 3.2 節で述べた方法で品質のグループ化を行う．また，各プロキシ $p \in P$ が実行するトランスコードサービスの出力品質の集合 $E(p)$ を決定する．各 $p \in P$ において，$E(p)$ に属するどの要素に対してもトランスコードを可能とするため，$E(p)$ の要素の内，画像サイズの最大値，フレームレートの最大値をそれぞれ $q_p.s$，$q_p.f$ とするとき，p が上流のプロキシから受信するビデオの品質を $q_p.s, q_p.f, q_p.b$ と同じか，またはより高いものする．ここで，ビットレート $q_p.b$ の値は，$q_p.s$，$q_p.f$ の値から文献 [4] で述べられている方法を用いて決定する．

（4） s は，上で決定した各 $p \in P$ の受信品質から，4 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．

（5） メッセージを受け取ったプロキシは，必要な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．

（6） 時刻 t になると，サーバ 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．

3.3.2 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．t_r は，録画済みビデオの場合はビデオセグメントの境界とし，生放送の場合は，ある時間間隔で与えられるものとする．また，すべてのモバイルノードは 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，4 章で述べるアルゴリズムを用いて s が求める．各プロキシは再構築前の配送経路を維持したまま，再構築後のストリームの受信を開始しておき，t_r の到来と同時に新たな配送経路に切り替える．

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するため，ストリームの再生がそれに伴った時間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あらかじ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．

3.3.3 モバイルノードの加入・離脱手順

ビデオの配信開始時刻以降に新たにビデオの受信を希望するモバイルノード u_{new} は，3.1 節で述べた方法を用いて要求品質 $q_{u_{new}}$ を決定し，現在の AP に属するプロキシ p へ，$q_{u_{new}}$ の情報を含む $join$ メッセージを送信する．p は，現在トランスコード中のビデオの中から，$q_{u_{new}}$ より低い品質の中で最も近いものを選び，それを一時的に q_{new} へ中継する．次回の経路の再構築時に，品質許容範囲を考慮した再生品質に更新される．

ビデオの受信を辞めたいノード u_{leave} は，好きなときに離脱することが可能である．その際，u_{leave} と同じ品質でビデオを受信している他のモバイルノードが存在しない場合は，p はその品質へのトランスコードサービスを終了する．これにより，p の上流のプロキシからの受信品質が変わる可能性がある．そのような場合の対処法については，4 章で述べる．

3.3.4 AP 間のハンドオフ手順

モバイルノード u_{move} が現在の AP から別の AP へ移動した際，移動後の AP に属するプロキシ p は，u_{move} の要求品質と p が上流のプロキシから受信している品質 q_p との比較を行う．もし $q_{u_{move}}.s \geq q_p.s \vee q_{u_{move}}.f \geq q_p.f \vee q_{u_{move}}.b \geq q_p.b$ であれば，u_{move} の要求品質へのトランスコードサービスを起動することが不可能であるため，p は一時的に品質 q_p のビデオを u_{move} に送信し，次回の経路の再構築時に新たな品質に更新する．また，$q_{u_{move}}.s \leq q_p.s \wedge q_{u_{move}}.f \leq q_p.f \wedge q_{u_{move}}.b \leq q_p.b$ である場合，以下のいずれかが実行さ

れる．p が現在トランスコードしている品質の集合 $E(p)$ について，$q_{move} \in E(p)$ であれば，既存のストリームを u_{move} に送信する．そうでない場合，計算資源に余裕があれば，新たにトランスコードサービスを立ち上げ，品質 $q_{u_{move}}$ のビデオを送信する．計算資源に余裕がない場合，集合 $E(p)$ から $q_{u_{move}}$ より低い品質の中で最も近いものを選び送信する．

プロキシ p からビデオを受信しているノード u の，サーバ s からの配送遅延を D_p とする．ハンドオフ後の接続プロキシを p' とするとき，もし $D_p > D_{p'}$ ならば，$D_p - D_{p'}$ 分のバッファリング遅延が生じるため，その間ビデオの再生が途切れることが考えられる．この問題は，再構築時に生じる同様の問題を回避するために，各プロキシがバッファリングしているデータを利用することで回避する．

モバイルノード u の移動先のプロキシの AP にモバイルノードが存在していない場合，そのプロキシは現在ビデオの受信を行っていないため，u へのビデオの配信が遅れることが考えられる．この問題には以下のように対処する．

各プロキシ p について，自分の属する AP に隣接する AP の担当プロキシの集合を $NB(p)$ と表記する．4 節のアルゴリズムを使用して配送経路を計算する際，$R(p) \neq \emptyset$ であるプロキシ p に対し，$R(p') = \emptyset$ となる $p' \in NB(p)$ が存在する場合，そのような全ての p' に対し，$R(p') = max(R(p))$ とする．これにより，u の移動先のプロキシは，少なくとも一つのストリームを受信中であることを保証する．

4 配送経路計算アルゴリズム

本章では，2 章に定義した目的関数をできるだけ小さくするような配送経路を求めるための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，および各プロキシ $p \in P$ が上流プロキシから受信するビデオの品質 $q_p = (q_p.s, q_p.f, q_p.b)$ が入力として与えられる．ここで q_p は，p に接続しているモバイルノードの要求品質の内，最高品質のビデオの品質ベクトルである．配送経路の計算はサーバ s が行い，作成した配送経路情報の各プロキシへの配布は 3.3.1 節で述べた手順で行う．目的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 2 つの値の最小化はトレードオフの関係にある．計算資源消費量を最小化するためには，実行されるトランスコードサービスの数を最小にすればよい．しかし，同一品質のビデオを要求するモバイルノードが複数のプロキシに分散している場合，ストリームの中継を行うために多くのネットワーク資源を消費する．一方，ネットワーク資源消費量を最小化するためには，同一品質のビデオを生成するトランスコードサービスを複数のプロキシ上で実行するため，多くの計算資源を消費する．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 [9] で提案されている最小スパニングツリー（シュタイナー木）を計算する既存のヒューリスティックなアルゴリズムを拡張したものである．4.1 節では，文献 [9] の手法を利用して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，4.2 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．

4.1 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．

本手法では，まず文献 [9] の手法を使用して，オーバーレイリンクのホップ数の合計が最小となるような，プロキシ間を接続するシュタイナー木を計算する．次に，s とプロキシ間でストリームを中継する全ての経路が式 (3) と (4) の制約を満たす必要があるので，経路上の各プロキシが受信する品質を調整する．これは，次の 4 つのステップから成る．

Step1. 葉プロキシ p は自身の品質要求 q_p(p に接続しているモバイルノードの要求品質の最大値) を含むメッセージ r_p を親プロキシ p' に送信する．

Step2. p' はすべての子プロキシからメッセージを受信後，各受信品質 q_p と自身の品質要求 $q_{p'}$ を比較し，$q_p.s \geq q_{p'}.s \vee q_p.f \geq q_{p'}.f \vee q_p.b \geq q_{p'}.b$ であれば $q_{p'}$ を $q_{p'} = (max(q_{p'}.s, q_p.s), max(q_{p'}.f, q_p.f), max(q_{p'}.b, q_p.b))$ 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．

Step3. Step2 の操作をメッセージが根プロキシに届くまで繰り返す．

Step4. 最後に，根プロキシはサーバ s にメッセージを送信する．

4.1.1 計算資源最小化アルゴリズム

計算資源消費量を最小にする場合 (目的関数 (7) において $\alpha = 1$ の場合)，各品質ベクトル q に対して起動するトランスコードサービス数を 1 つだけにする必要がある．その際，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 が接続するプロキシ p 以外のプロキシ p' で実行されている場合，図 3(a) のように品質 q_u のビデオを p' から p へ中継しなければならない．

また，ある品質のビデオを複数の品質へトランスコードする際には，それぞれの品質に対してデコード・エンコードを行うのではなく，1 回のデコード後それぞれの品質へエンコードすることで，デコードのための計算資源を削減することが可能となる．

そのため，本手法ではモバイルノードの要求品質の集合をプロキシに割り当てるとき，デコードにより消費する計算資源を削減するため，出来るだけ少ない数のプロキシを使用する．しかし，これは組み合わせ最適化問題であるため，以下のヒューリスティックアルゴリズムを使用して計算する．

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

（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 (品質 q へ処理するのに必要な計算パワーを $r_{encode}(q)$ とする) の昇順でソートする．ここで，$QR = (qr_1, ..., qr_{nu})$ はソートされたリストを示す．

（3） sp_1 から順に $c\_avail(sp_1) > r_{decode}(q_{orig}) + \sum_{i=1}^{j} r_{encode}(qr_i)$ を満足するよう，出来るだけ多くの QR の要素を割り当てる．

（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．

（5） 各プロキシの受信品質 q_p を 3.3 節で述べた方法で計算する．

（6） 4.1 節のアルゴリズムを使用してシュタイナー木を計算し，各プロキシ p の最大受信品質 q_p を調整する．

4.1.2 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 (目的関数 (7) において $\alpha = 0$ の場合)，ある要求品質に対しトランスコードサービスを起動する数は，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である．つまり，各プロキシは自分に接続している全てのモバイルノードへのトランスコードを担当する．そのため，プロキシ間で同じ品質を共有するといったストリームの中継は起こらない．各プロキシが受信する品質は，3.3.1 節に述べたように自プロキシ内の要求品質の集合の内，最大の画像サイズと最大のフレームレートである．そのため，プロキシは受信したビデオを自プロキシ内の全ての品質にトランスコードすることが可能である．

4.2 ハイブリッドアルゴリズム

品質 q へトランスコードを行うプロキシの数を NP_q とする．式 (7) の目的関数において，$\alpha = 1$ の場合（計算資源最小）はすべての q について $NP_q = 1$ である．また，$\alpha = 0$ の場合（ネットワーク資源最小）は $NP_q = |\{p|q \in E(p) \wedge p \in P\}|$ となる．全てのモバイルノードからの要求品質の集合の内，最大の NP_q の値を NP_{max} とする．

式 (7) の目的関数を最小化する問題は組み合わせ最適化問題である．そのため，提案手法では NP_q の値を 1 から NP_{max} の範囲で変化させて目的関数の値を計算し，その中で値が最小となる解を選択する．ここで，NP_q はすべての要求品質について同じ値を使用する．

提案するアルゴリズムを以下に示す．Step2 から Step 4 まで各 i について 1 から NP_{max} まで繰り返し，目的関数の値が最小となる解をそれらの中から選択する．

Step1. 各品質ベクトル q に対して，$NP_q = |\{p|q \in E(p) \wedge p \in P\}|$ を計算する．まず，すべてのモバイルノードの要求品質について，3.2 節に述べた方法で要求品質のグループ化を行う．ここで，プロキシ $x \in P$ で品質 q を要求しているモバイルノードの数を $N_{q,x}$，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 P から選択される．

Step2. プロキシ x でのモバイルノードの要求品質の最大値 $qmax_x$ を計算する．$qmax_x$ は $qmax_x = max(E(x))$ により計算される．

Step3. プロキシ間のシュタイナー木を計算する．オーバーレイネットワーク G と $qmax_x$ を用いて，4.1 節に述べた方法で，プロキシ間のシュタイナー木を計算する．

Step4. 各品質 q に対して，シュタイナー木を計算する．i が NP_{max} より小さいとき，i 個のプロキシが同時にトランスコードを行い，プロキシ間で同品質のストリームを共有する．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．

4.2.1 具体例

図 3 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，リンクを通って中継されるビデオの品質ベクトルをそれぞれ，$qmax_x, q_{str}$ とする．

図 3(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 $u_1, ..., u_6$ 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一つのプロキシで実行される．そのため，トランコードサービス $T1, T2, T3, T4$ 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，$u3$ 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 $T1$ は $u3$ 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，$u4$ が要求する品質 200 に対するトランスコードサービス $T2$ は p_2 に存在するため，$p3, p1$ 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．

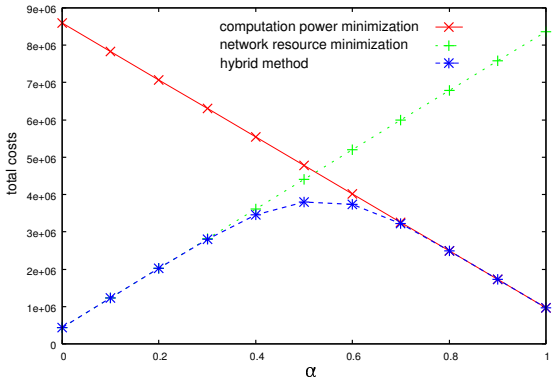


図 4: 各アルゴリズムの α の変化に対するトータルコスト

図 3(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 $T2, T3$ 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って中継される．

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，目的関数 (7) 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するものである．

5 性能評価

提案手法の有効性を検証するため，計算資源最小化アルゴリズム，ネットワーク資源最小化アルゴリズム，ハイブリッドアルゴリズムのそれぞれで達成されるコストをシミュレーションにより求めた．

シミュレーションの設定は次の通りである．プロキシ間のオーバーレイネットワークとして，GT-ITM[10] の Locality model を用いてプロキシ数 50 のトポロジを生成した．また，モバイルノード数は 3000 とした．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．τ_d，τ_e はそれぞれ $\tau_d = 0.00028$，$\tau_e = 5 \times \tau_d$ とした．各モバイルノードの要求品質は，最大品質 640×480 pixel, 30fps ，最低品質 80×60 pixel，5 fps の範囲から一様乱数で決定し，品質許容範囲の 20%以内に収まるようにグループ化した．本実験では，各アルゴリズムのコスト値の比較を行うのが目的であるため，各プロキシの利用可能計算資源，プロキシ間のリンクの利用可能帯域は十分であると仮定した．各アルゴリズムに対し，α を 0.0 から 1.0 まで変化させたときに達成されるコストを測定した．実験結果を図 4 に示す．

図 4 より，ハイブリッドアルゴリズムは $\alpha = 0.5$ 付近で他の 2 つのアルゴリズムよりもコストを低く抑えられることが分かる．また，計算資源最小化アルゴリズム，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．

次に，各アルゴリズムの計算時間を評価するため，品質のグループ化，および配送経路の探索を完了するまでにかかる時間を，ノード数を 100 から 3000 まで変化させて計測した．実験では，CPU として Athlon 64 3400+，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 より，計算資源最小化アルゴリズム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

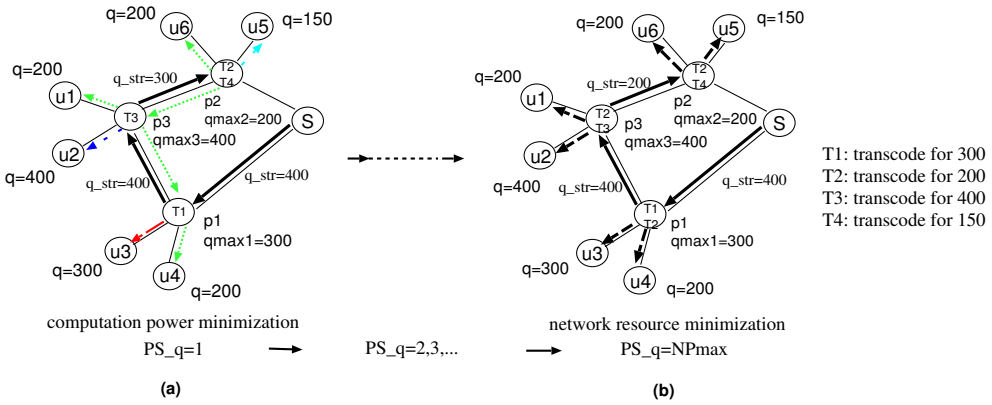


図 3: ハイブリッド手法の具体例

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
計算資源最小アルゴリズム	0.016	0.12	0.43	1.68	3.91
ネットワーク資源最小アルゴリズム	0.023	0.13	0.43	1.67	3.91
ハイブリッド手法	0.076	1.34	3.18	9.99	18.7

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，ユーザノード数が 3000 程度までならば実用上十分な時間で計算を完了できることが分かる．

6 おわりに

本稿では，CDN 上でのサービス合成手法に基づいた，異なる要求品質を持つ多数の携帯無線端末へのリソース効率のよいビデオ配信方式を提案した．提案方式の特徴は次のとおりである．(1) ユーザに対する利点：ユーザは，端末のバッテリ量，計算能力，利用可能帯域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 の移動に対しても，ビデオを途切れることなく再生可能である．(2) サービスプロバイダに対する利点：サービスプロバイダは，利用する CDN 上の，ビデオ配信サーバ，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 の使用リソース量をできるだけ少なくするようなビデオ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実験から，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経路を求められることを確認した．

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, Vol. 3, No. 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, Vol. 8, No. 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 <draft-ietf-diffservframework-02.txt>*, 1999.

[4] Morihiko Tamai, Tao Sun, Keiichi Yasumoto, Naoki Shibata and Minoru Ito, "Energy-aware Video Streaming with QoS Control for Portable Computing Device," *Proc. of the 14th ACM Int'l. Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV 2004), pp.68-73*, 2004.

[5] 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.

[6] Jin Liang, and Klara Nahrstedt, "Service Composition for Advanced Multimedia Applications," *12th Annual Multimedia Computing and Networking (MMCN 2005)*, 2005.

[7] Jingwen Jin, Klara Nahrstedt, "On Exploring Performance Optimizations in Web Service Composition," *Proc. of ACM/IFIP/USENIX Int'l. Middleware Conf. (Middleware 2004)*, 2004.

[8] Ching-Yung Lin, Belle L. Tseng, John R. Smith, "IBM MPEG-7 Annotation Tool," http://www.alphaworks.ibm.com/tech/videoannex

[9] L. Kou, George Markowsky, and Leonard Berman. "A fast algorithm for Steiner trees," *Acta Informatica, 15:141–145*, 1981.

[10] 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.

[11] Sumit Roy, Michele Covell, John Ankcorn, and Susie Wee, "A System Architecture for Managing Mobile Streaming Media Services," *Int'l. Workshop on Mobile Distributed Computing (MDC 2003)*, 2003.