

プロキシ協調型動画配信システムの実装と評価

谷口 義明[†] 若宮 直紀[†] 村田 正幸^{††}

[†] 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
^{††} 大阪大学 サイバーメディアセンター 〒560-0043 大阪府豊中市待兼山町 1-32
E-mail: †{y-tanigu,wakamiya}@ist.osaka-u.ac.jp, ††murata@cmc.osaka-u.ac.jp

あらまし WWW システムで広く用いられているプロキシ技術を適用することにより、実時間で応答性の高い動画配信システムを実現することができる。本稿では、ネットワーク内に配置された複数のプロキシが互いのキャッシュデータを共有することで、サービスの応答性を高め、プロキシに動画品質調整機能を持たせることにより、クライアント毎に異なる適切な品質の動画を配信することのできる MPEG-4 動画配信システムを実装、評価した。プロキシは、他のプロキシキャッシュサーバや動画サーバに蓄積された動画の品質、それらサーバとの伝搬遅延、転送速度を考慮して適切なサーバから動画を取得し、クライアントに提供する。提案システムを用いることで、再生のとぎれのない効果的な動画配信が行えることを示した。

キーワード 品質調整, 動画配信, プロキシキャッシング, QoS, MPEG-4

Implementation and Evaluation of Cooperative Proxy Caching System for Video Streaming Services

Yoshiaki TANIGUCHI[†], Naoki WAKAMIYA[†], and Masayuki MURATA^{††}

[†] Graduate School of Information Science and Technology, Osaka University
1-3 Machikaneyama, Toyonaka, Osaka 560-8531, Japan

^{††} Cybermedia Center, Osaka University
1-32 Machikaneyama, Toyonaka, Osaka 560-0043, Japan

E-mail: †{y-tanigu,wakamiya}@ist.osaka-u.ac.jp, ††murata@cmc.osaka-u.ac.jp

Abstract The proxy mechanism widely used in WWW systems offers low-delay and scalable delivery of data by means of a “proxy server”. By applying proxy mechanism to video streaming systems, high-quality and low-delay video distribution can be accomplished without imposing extra load on the system. We have proposed proxy caching mechanisms to accomplish the high-quality and highly-interactive video streaming services. In our proposed mechanisms, proxies communicate with each other, retrieve a missing video data from an appropriate server by taking into account transfer delay and offerable quality. In addition, the quality of cached video data can be adapted appropriately in the proxy to cope with the client-to-client heterogeneity, in terms of the available bandwidth, end-system performance, and user preferences on the perceived video quality. In this paper, to verify the practicality of our mechanisms, we implemented them on a real system for MPEG-4 video streaming services, and conducted experiments. Through evaluations, it was shown that our proxy caching system is effective in providing clients with a continuous and high-quality video streaming services.

Key words Quality Adjustment, Video Streaming Service, Proxy Caching, QoS, MPEG-4

1. はじめに

近年、コンピュータの高性能化やネットワークの広帯域化に伴い、ストリーミングサービスなどの動画データ配信を行うマルチメディアアプリケーションの利用が増加してきている。

しかしながら、現在のインターネットでは通信帯域、パケット配送遅延、パケット棄却に対する制御や保証のない、ベストエフォート型サービスしか提供されないため、高品質で応答性が高く、再生の途切れのない動画配信サービスを提供、享受することは困難である。

WWW システムでは、プロキシキャッシュサーバを用いて、サーバやサーバ近傍のネットワークの負荷を軽減するとともに、サーバとの通信で発生するパケット棄却や遅延を利用者から隠蔽し、応答時間の短縮を図っている。また、複数のプロキシキャッシュサーバが互いにキャッシュデータを共有し、協調型の制御を行う技術も実現されている [1]。動画配信システムにおいても、プロキシキャッシング技術を導入することにより、ネットワークに過剰な負荷を与えることなく低遅延で応答性の高いサービスを提供することができるように考えられる。しかしながら、動画配信ではクライアントが定期的にデータを受信、復号化、再生できるよう、ネットワークの輻輳状態に関わらず、動画データが再生時刻に間に合うよう順次配送しなければならない。また、同じコンテンツであっても、利用者の環境、すなわちネットワークへの接続回線の容量や動画サーバへの経路の輻輳状態、クライアントシステムの処理能力などの違いにより、クライアントに適した動画品質は様々に異なる。動画配信のためのプロキシキャッシング技術についてはすでにいくつかの検討がなされているが、利用者の環境を考慮していない、階層符号化技術を利用しているため様々な利用者環境に柔軟に対応できない、などの問題がある [2-5]。

文献 [6] において我々は、高品質な MPEG-2 動画データを送切れなくクライアントに提供するためのプロキシキャッシングメカニズムを提案し、シミュレーションおよび簡単な実証実験を通してその有効性を示した。提案手法は、動画品質調整機能を有するプロキシキャッシュサーバが、ネットワークへの接続形態や輻輳状態、システム性能、再生動画に対する利用者の好みなどが様々に異なるそれぞれのクライアントに対し、キャッシュデータを適切に調整、提供することにより、高品質かつ低遅延な動画配信を実現する。また、キャッシュバッファや伝送帯域の有効利用を図るため、動画ストリームを一定時毎のブロックに分割し、ブロックを単位として取得、蓄積、加工、配信する。さらに、動画データの参照の順序性を利用した動画ブロックの先読み機構によりキャッシュミスによる遅延の増加を抑え、ブロック置き換え機構により有限容量のキャッシュバッファを効率良く利用する。我々はこれらの提案手法に基づき、文献 [7] において、一般に利用されている MPEG-4 動画配信を対象としたプロキシキャッシングシステムを実装し、既存の動画配信サービスへの適用可能性やその実用性、有効性を示した。

さらに効果的な制御を実現するため、文献 [8, 9] において我々は、WWW システムにおけるプロキシの協調技術、および近年注目されている P2P 型通信のモデルを応用し、動画品質調整機能を有する複数のプロキシキャッシュサーバが連携して動画配信サービスを提供するプロキシ協調型動画配信手法を提案し、シミュレーションによりその有効性を示した。しかしながら、これらの提案、シミュレーションではいくつかの前提、仮定が置かれており、実環境におけるシステムの適用性や実用性が十分に評価されていない。本稿では、MPEG-4 動画配信サービスを対象として、提案したプロキシ協調型動画配信システムを実装し、提案手法の検証を行う。その結果、提

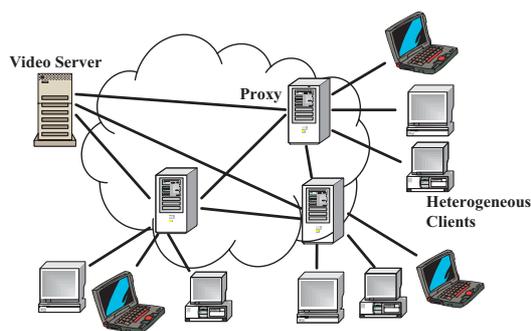


図 1 プロキシキャッシュを有する動画配信システム
Fig. 1 Video Streaming System with Proxies

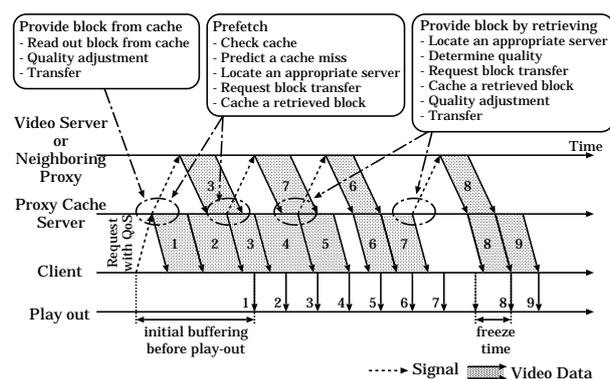


図 2 提案システムの基本動作
Fig. 2 Basic Behavior of Proposed System

案手法を用いることで、クライアントにおける動画再生のどぎれを引き起こすことなく、適切なサーバから動画ブロックを取得し、クライアントに提供する効果的な動画配信が行えることを示す。

以下、2 章では、プロキシ協調型動画配信手法の概要を、3 章で実システムへの実装方法を述べる。4 章で実装システムを用い、提案手法の実用性、有効性について検証する。最後に、5 章で本稿のまとめと今後の課題について述べる。

2. プロキシ協調型動画配信システムの概要

本章では、我々の提案する、プロキシ協調型動画配信システムの概要を述べる。なお、提案手法の詳細については、文献 [8, 9] を参照されたい。

2.1 基本動作

提案システムは、図 1 に示すような、一台の動画サーバ、処理能力や利用可能な帯域などの異なる複数のクライアント、および動画品質調整可能な複数のプロキシキャッシュサーバからなる。それぞれの間の動画データ転送は、動画ストリームにつき 1 組ずつ設定される動画セッションを用いて行われる。

図 2 に提案システムの基本動作を示す。まず、クライアントは近隣のプロキシキャッシュサーバに動画の配信を要求する。要求メッセージには、望む動画の識別子や再生の範囲とともに、受信動画に対する最低許容品質および最大許容品質が含まれる。プロキシキャッシュサーバは、キャッシュバッファや伝

送帯域の有効利用を図るため、動画像ストリームを一定時間毎のブロックに分割、ブロックを単位として取得、蓄積、品質調整する。動画像配信要求を受け取ったプロキシキャッシュサーバは、要求された動画像ストリームの先頭の動画像ブロックから順に、適切な提供手法を決定し、クライアントに提供する。遅延のゆらぎによる再生の途切れを防ぐため、クライアントは受信したブロックを一時バッファリングした後、順次再生する。

2.2 キャッシュテーブルとリモートテーブル

プロキシキャッシュサーバは、自身のキャッシュバッファの状態を管理するためのキャッシュテーブルと、他のサーバの提供可能なブロックとその品質、およびそれらとの間の伝搬遅延の推定値、転送速度の推定値を管理するためのリモートテーブルを保持している。キャッシュテーブルは、ブロック番号と品質、マーカからなる。リモートテーブルはサーバ間の QUERY、REPLY メッセージのやり取りによって更新される。

プロキシは、新しくクライアントからの動画像配信要求を受信すると、全サーバに QUERY メッセージを送出し、クライアントへ提供中のブロックおよびストリームの先頭のブロックからそれぞれ I 個のブロックに関するキャッシュ情報を問い合わせる。 I を問い合わせウィンドウと呼び、問い合わせ範囲を制限するために用いる。また、プロキシは、クライアントへ提供中のブロックから $P - 1$ 番目のブロックが問い合わせウィンドウの最後尾に達したときにも同様に QUERY メッセージを送出する。ここで、 P は先読みウィンドウと呼ばれ、プロキシにおけるブロック先読み用いられる。QUERY メッセージを受信したサーバは、提供可能なブロック品質を REPLY メッセージにより回答する。また、キャッシュテーブルの問い合わせ対象のブロックをプロキシ識別子でマークするとともに、他のブロックのマークを解除する。プロキシは、受信した REPLY メッセージによりリモートテーブルを更新する。

2.3 ブロックの提供手法

プロキシキャッシュサーバは、動画像ブロックの大きさ、他のサーバおよびクライアントへの伝搬遅延や転送速度などを推定し、ブロックの転送が再生に間に合うかどうかを判断する。プロキシは、ブロックごとに、(1) 蓄積しているブロックを提供する場合、(2) 他のサーバから受信中のブロックを提供する場合、(3) 他のサーバに転送要求中のブロックを提供する場合、および (4) 新たに適切なサーバから動画像ブロックを取得、提供する場合のそれぞれについて、クライアントの動画像再生に間に合うように転送可能かどうかを調べる。最も高い品質の動画像ブロックを最も速く転送できる提供手法を用いるが、いずれの提供手法でもクライアントの最低許容品質を満たせない場合は、最も速く転送可能な提供手法で動画像ブロックを転送する。

2.4 ブロックの先読み手法

さらに効果的な制御を行うため、プロキシは、帯域の空きを利用して近い将来キャッシュミスを起こす可能性のあるブロックをあらかじめ取得する。クライアントへのブロック転送に際し、プロキシは、他のサーバに動画像ブロック転送を要求していなければ、続く P 個のブロックについてキャッシュでの蓄積状態を調べる。クライアントの最低許容品質を下回るブロック

のうち、クライアントに転送中のブロックに最も近いものを先読みする。ブロックの品質および取得先サーバはキャッシュミス発生時のブロック取得と同様にして適切に決定される。

サーバやプロキシにおいて、キャッシュミスによるブロック転送要求は FIFO 待ち行列によって管理され、順次処理されるが、先読み要求はサイズ 1 のバッファによって管理され、未処理の要求は新しい要求によって上書きされる。また、先読み要求は FIFO 待ち行列に要求の無い場合のみ処理される。したがって、先読み要求と同じブロックに対する、同じかそれ以上の品質のブロック転送要求が FIFO 待ち行列に入力された場合には、先読み要求はキャンセルされる。

2.5 ブロックの置き換え手法

取得したブロックを蓄積するだけの空きがキャッシュバッファに無い場合、プロキシはキャッシュ内ブロックの置き換えを行う。ただし、利用者は、通常、動画像ストリームの先頭から最後まで順に視聴を続けると考えられるため、問い合わせウィンドウと先読みウィンドウ内のブロック、マークが空でないブロック、およびストリームの先頭に位置する I 個のブロックは置き換えの対象としない。残りのブロックのうち、ストリームの最後尾に近いものから順に置き換えの対象とする。

プロキシは、まず、優先度の最も低いブロックに対して品質調整を施すことにより、新たに取得したブロックを蓄積するための空きを作る。品質調整により十分な空きが生まれない場合は、そのブロックを棄却する。新たに取得したブロックが蓄積可能になるまで、優先度の低いブロックから順に品質調整、棄却を繰り返す。さらに空き容量が必要な場合には、それぞれのクライアントの問い合わせウィンドウ内のブロックのうち、最後の P ブロックについても置き換えを検討し、さらに置き換えが必要になる場合には、新たに取得したブロックを棄却する。

3. プロキシ協調型動画像配信システムの実装

提案手法では、クライアントやサーバ、プロキシ間の情報共有機構、ネットワーク状態の推定などについて、いくつかの前提、仮定を置いており、既存の動画像配信サービスに適用するためにはいくつかの問題がある。本章では、提案手法を実システムに実装する上で問題となる点を明らかにするとともに、実システムへの実装方法を述べる。

3.1 システムモデル

動画像サーバとして Darwin Streaming Server [10] を、クライアントアプリケーションとしては QuickTime Player [11] および RealOne Player [12] を対象とする。また、動画像フォーマットには MPEG-4 を用いる。サーバ/プロキシ間、プロキシ/プロキシ間、プロキシ/クライアント間それぞれにおいて、動画像配信制御には RTSP/TCP セッションを、動画像ブロック転送には映像ブロック、音声ブロック別に RTP/UDP セッションおよび RTCP/UDP セッションを用いる [13]。

実装システムの概要を図 3 に示す。実線の矢印は動画像データの流れを、破線の矢印は制御情報の流れを表している。それぞれの四角は、プロキシ内で生成されるプロセスを表している。本システムは、文献 [7] において我々が実装したプロキ

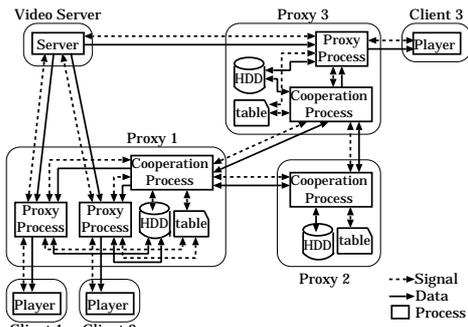


図 3 実装システムの概要

Fig. 3 Modules Constituting Our System

シキャッシングシステムを拡張することで実現する。プロキシ機能を持つプロセス *Proxy Process* はクライアント毎に生成され、*Table* および *HDD* で管理されるキャッシュデータを共有し、クライアントに適切な品質の動画ブロックを提供する。また、*Cooperation Process* は、近隣プロキシとの動画ブロック転送、動画配信制御、帯域や伝搬遅延測定を管理する。なお、動画サーバは、動画品質調整機能を有さず、動画の符号化レートで動画ブロックを転送するため、1 組の動画セッションだけで複数のブロック転送要求を同時に処理することはできない。そのため、実装システムでは、プロキシと動画サーバとの RTSP/TCP セッションおよび RTP/UDP セッションは、プロキシに接続されているクライアントごとに設定される。

3.2 RTSP による動画配信制御

図 4 に、本システムの動作の概要を示す。まず、クライアントは RTSP OPTIONS, DESCRIBE, SETUP メッセージを順次プロキシに転送することで、映像および音声ストリーム配信のためのセッションを確立する。*Proxy Process* は、受信した RTSP メッセージを動画サーバに中継することにより、動画サーバ/プロキシ間のセッションを確立する。動画サーバ/プロキシ/クライアント間それぞれのセッションの確立後、クライアントは RTSP PLAY メッセージをプロキシに送り、動画ストリームの転送を要求する。提案システムでは、クライアントは、動画の要求と同時に受信動画に対する許容品質をプロキシに通知するが、一般的なクライアントアプリケーションには通知機能がない。そのため、実装システムでは、クライアントの要求品質を考慮しない。クライアントには、再生に間に合う方法のうち最も高品質、かつ高速に転送可能な方法でブロックが転送される。

プロキシは、RTSP TEARDOWN メッセージをクライアントから受信すると、動画サーバにメッセージを転送し、対応する各セッションを切断する。

3.3 MPEG-4 動画のブロック分割

MPEG-4 動画は、フレームに相当する VOP (Video Object Plane) を単位として RTP パケット化されることを考慮し [14]、実装システムでは、1 ブロックの大きさを 300 VOP としている。例えば、30 フレーム毎秒の動画ストリームの

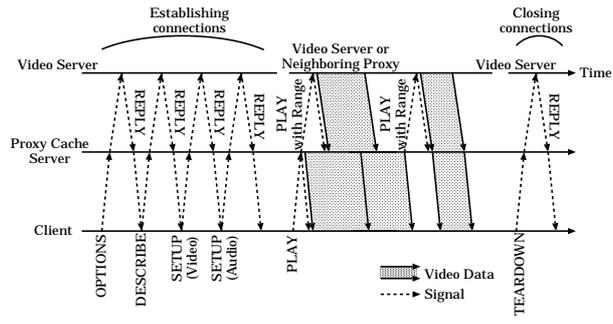


図 4 実装システムの基本動作

Fig. 4 Basic Behavior of Implemented System

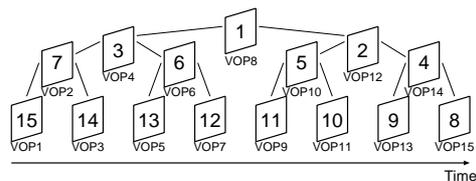


図 5 フレーム棄却順

Fig. 5 Frame Discarding Order

場合、1 ブロックの大きさは 10 秒となり、先頭から 3 番目のブロックを取得するには RTSP PLAY メッセージを用いて Range: 20-30 とすればよい。なお、動画ブロックは、映像ブロックと、音声ブロックに分割して扱われる。

3.4 動画品質調整

実装システムでは、映像ブロックの符号化レートを動画ブロックの品質とする。近隣プロキシに動画ブロックの転送を要求する際には、RTSP PLAY メッセージの Bandwidth フィールドを用いて取得する動画ブロックの品質を指定する。なお、映像データに比べ、音声データは比較的サイズが小さいため、音声ブロックは品質調整の対象としない。映像ブロックの品質調整には、フレーム棄却フィルタを用いる。フレーム棄却フィルタは適当な数のフレームを棄却することで、動画品質およびデータ量を調整する。再生動画の動きのなめらかさは失われるが、ローパスフィルタや再量子化フィルタに比べて、単純で処理負荷が小さい [15]。

MPEG-4 において VOP を棄却する際は、VOP 間の依存関係を考慮する必要がある。VOP の並びが固定されている場合には、所望のレートを達成するために間引く VOP の位置や枚数をあらかじめ決定し、VOP を順次棄却していくことが可能である [15]。しかしながら、一般的にはより高品質な符号化動画を得るために動画の特性にあわせて適宜 VOP の種別が選択されるため、以下の手順に従って VOP の棄却順を決定する。

まず、1 秒分の VOP を蓄積する。次に、バランスよく VOP を棄却して品質調整後の再生をなめらかにするため、優先度を設定する。例えば、15 フレーム毎秒の動画の場合には、中央に位置する 8 番目の VOP を根とした二分木を作成し、根から順に幅優先で優先度を高くしていく。図 5 に、15 フレーム毎秒の動画ストリームにおける VOP の優先度順の例を示す。図中、それぞれの四角形は VOP を表し、四角形の中の数字は

VOP の優先度を示す．なお，数字の小さいものほど優先度が低く，棄却されやすい．B-VOP のうち優先度の低い VOP から順に棄却することにより，利用可能な帯域で送信可能なよう動画データ量を調整する．全ての B-VOP を削除しても，動画レートが利用可能な帯域を上回る場合は，P-VOP を同様に順次棄却していく．ただし，再生動画の著しい品質劣化を防ぐため，I-VOP の棄却は行わない．

3.5 プロキシ間の情報交換

提案システムでは，QUERY，REPLY メッセージにより，動画サーバおよび近隣プロキシの提供可能なブロックに関する情報を取得し，キャッシュテーブルおよびリモートテーブルを更新する．動画サーバは動画ブロック全体を蓄積しているため，実装システムでは，近隣プロキシに対してのみ QUERY メッセージを送出する．なお，QUERY，REPLY メッセージは，それぞれ RTSP GET_PARAMETER および RTSP REPLY メッセージで実現される．

プロキシ間の動画データ転送には，ヘッダを拡張した RTSP PLAY メッセージを用い，キャッシュミスによる動画ブロック転送要求と先読み要求を区別する．ただし，動画サーバはこの拡張ヘッダを適切に処理できないため，動画サーバに対しては先読みメッセージを送信しない．

3.6 TFRC を利用した帯域推定

実装システムでは，映像ブロック配信に用いられる RTCP セッションからフィードバック情報を得て，RTT から伝搬遅延を，RTT とパケット棄却率から転送速度を推定する．転送速度の推定には TFRC [16] を用いる．

TFRC とは「同一ネットワークパス上における，非 TCP コネクションの得るスループットと TCP の得るスループットが等しい」という TCP-friendly の概念に基づき，帯域使用において TCP と公平なマルチメディア通信を実現するレート制御アルゴリズムである．TFRC では，RTT やロスイベント率といったネットワークの負荷状態をあらわすフィードバック情報に基づき，同一パス上の TCP セッションの得るスループットを推定し，マルチメディアデータの送出レートを決定する．

4. 実装システムの実験評価

通信状態に応じた動画品質調整を行うプロキシの有効性，実用性についてはすでに評価しており [7]，本稿では，特にプロキシの協調動作について検証するため，実験環境を以下のように設定した．

実験システムの構成を図 6 に示す．1 台のクライアントがルータを介してプロキシ 1 に接続されており，プロキシ 1 はルータを介してプロキシ 2 と動画サーバに接続されている．ネットワークエミュレータ NIST Net [17] により，それぞれの接続の伝搬遅延と通信容量を図に示すように設定した．実験には，平均 1 Mbps の符号化レートで MPEG-4 VBR 符号化された 320×240 画素，30 フレーム毎秒の映像ストリームと 96 kbps の音声ストリームが多重化された 100 秒，約 14 MByte の動画ストリーム 1 本を用いる．3.2 節で述べたように，実装システムでは，1 ブロックの大きさを 300 VOP としている

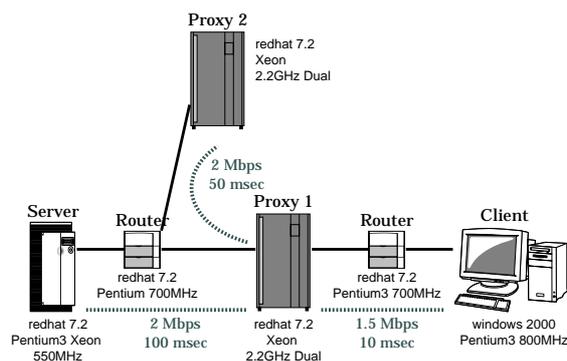


図 6 実験システム構成図

Fig. 6 Configuration of Experimental System

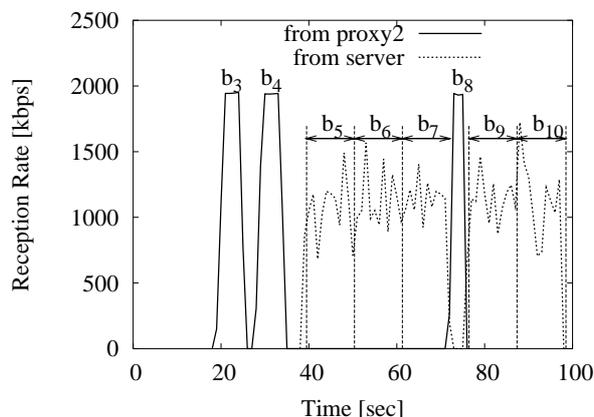


図 7 プロキシ 1 の受信レート

Fig. 7 Reception Rate Variation at Proxy1

ため，動画ストリームは 10 個のブロック b_1, b_2, \dots, b_{10} に分割される．動画サーバには動画全体が蓄積されており，プロキシ 1 には原画像と同じ 1Mbps の動画ブロック b_1, b_2 と 700kbps に品質調整された動画ブロック b_3, b_4 が，プロキシ 2 には 1Mbps の動画ブロック $b_1 \sim b_4$ と 700kps に品質調整された動画ブロック $b_5 \sim b_{10}$ がそれぞれ蓄積されている．クライアントは一時停止や早送りなどの操作無く，動画全体を鑑賞する．なお，転送速度に応じたブロック取得先決定アルゴリズムの実用性を評価するため，ここでは TFRC による帯域推定は行わず，プロキシは NIST Net で設定された帯域を用いるものとした．

プロキシ 1 とクライアントそれぞれの動画データの受信レートの変化の様子を，図 7 および図 8 に示す．計測にはクライアント，プロキシ 1 上で動作する tcpdump を用い，1 秒毎の総データ受信量から受信レートを算出した．なお，クライアントが OPTIONS メッセージをプロキシに転送する時刻を 0 とした．図より，プロキシ 1 は自身の持つ動画ブロックと，サーバおよびプロキシ 2 から取得した動画ブロックを用いて，クライアントに動画を配信できていることがわかる．

個々のブロックについて，プロキシ 1 は最適な提供方法を定める．動画ブロック b_1, b_2 に関しては，自身の持つ動画ブロックが最も品質が高く，最も速くクライアントに提供可能であるので，これをクライアントに配信している．このことによ

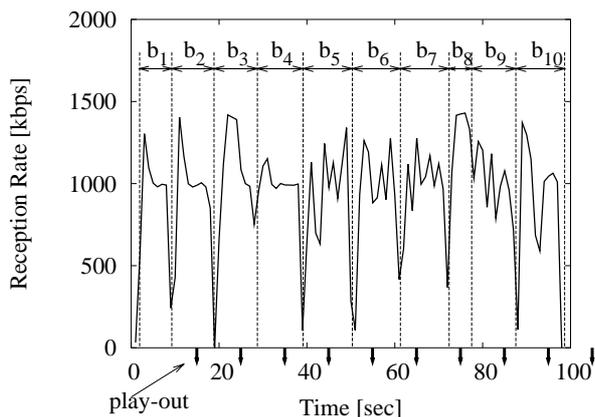


図 8 クライアントの受信レート

Fig. 8 Reception Rate Variation at Client

り、動画取得に余裕が生じ、動画ブロック b_3, b_4 に関しては、プロキシ 2 より、より高品質な動画ブロックを取得し、クライアントに提供している。さらに、動画ブロック b_5 から b_7 については、動画サーバからより高品質なブロックを取得し、再生に間に合っクライアントに提供している。しかしながら、動画サーバは符号化レートで動画を転送するため、ブロック取得に先立つ PLAY メッセージのやりとりのための遅延が累積する。また、動画サーバは、要求された動画ブロックの先頭が I-VOP でない場合には、直前の I-VOP から配信を行うため、遅延が増大する。そのため、動画ブロック b_8 については、プロキシ 1 は動画サーバからでなく、低品質であるが再生に間に合っクライアントに提供することのできる 700kbps の動画ブロックをプロキシ 2 から取得している。以上より、提案システムでは、利用可能な帯域および再生時刻に応じて、適切なサーバからブロックを取得し、クライアントに提供できていることがわかる。また、プロキシ 2 においても同様にクライアントが接続されている場合には、互いに動画ブロックを取得、提供しあう、協調制御が実現できる。

図 8 の下部に、クライアントにおける動画ブロックの再生のタイミングを示す。なお、本実験で用いたクライアントアプリケーションは、遅延のゆらぎによる再生のとぎれを防ぐため、3 秒間のバッファリングの後、動画の再生を開始する。図中には、クライアントが最初の VOP を受信した時刻の 3 秒後から、動画ブロックの再生時間である 10 秒毎を、再生時刻として示している。図に示されるように、すべての動画ブロックが再生に間に合うタイミングで到着しており、とぎれのない動画再生を実現している。

5. おわりに

本稿では、ネットワーク内に配置された複数のプロキシが互いのキャッシュデータを共有しサービスの応答性を高めるプロキシ協調型動画配信システムの実装を行い、提案手法を用いることで、効果的な動画配信が行えることを示した。ただし、他のサーバやクライアントアプリケーションが混在する大規模なネットワークでの実証実験や、利用者が停止、早送りといっ

た操作を行う場合については、引続き評価を行う必要がある。

謝 辞

本研究の一部は通信・放送機構創造的情報通信技術研究開発推進制度によっている。ここに記して謝意を表す。

文 献

- [1] "Squid." available at <http://www.squid-cache.org/>.
- [2] R. Rejaie and J. Kangasharju, "Mocha: A quality adaptive multimedia proxy cache for internet streaming," in *Proceedings of NOSSDAV 2001*, June 2001.
- [3] K. Wu, P. S. Yu, and J. L. Wolf, "Segment-based proxy caching of multimedia streams," in *Proceedings of the 10th International WWW Conference*, pp. 36–44, May 2001.
- [4] B. Wang, S. Sen, M. Adler, and D. Towsley, "Optimal proxy cache allocation for efficient streaming media distribution," in *Proceedings of IEEE INFOCOM 2002*, June 2002.
- [5] M. Reisslein, F. Hartanto, and K. W. Ross, "Interactive video streaming with proxy servers," *Information Sciences: An International Journal*, Dec. 2001.
- [6] M. Sasabe, Y. Taniguchi, N. Wakamiya, M. Murata, and H. Miyahara, "Proxy caching mechanisms with quality adjustment for video streaming services," *IEICE Transactions on Communications Special Issue on Content Delivery Networks*, vol. E86-B, pp. 1849–1858, June 2003.
- [7] Y. Taniguchi, A. Ueoka, N. Wakamiya, M. Murata, and F. Noda, "Implementation and evaluation of proxy caching system for MPEG-4 video streaming with quality adjustment mechanism," in *Proceedings of The 5th AEARU Workshop on Web Technology*, pp. 27–34, Oct. 2003.
- [8] N. Wakamiya, M. Murata, and H. Miyahara, "Video streaming systems with cooperative caching mechanisms," in *Proceedings of SPIE International Symposium ITCOM 2002*, pp. 305–314, July 2002.
- [9] N. Wakamiya, M. Murata, and H. Miyahara, "On proxy-caching mechanisms for cooperative video streaming in heterogeneous environment," in *Proceedings of IFIP/IEEE International Conference on Management of Multimedia Networks and Services 2002*, pp. 127–139, Oct. 2002.
- [10] "Darwin Streaming Server." available at <http://developer.apple.com/darwin/>.
- [11] "QuickTime Player." available at <http://www.apple.com/quicktime/>.
- [12] "RealOne Player." available at <http://www.real.com/>.
- [13] Internet Streaming Media Alliance, "Internet streaming media alliance implementation specification version 1.0," Aug. 2001.
- [14] Y. Kikuchi, T. Nomura, S. Fukunaga, Y. Matsui, and H. Kimata, "RTP payload format for MPEG-4 audio/visual streams," *Internet Request for Comments 3016*, Nov. 2000.
- [15] H. Akamine, K. Nakada, N. Wakamiya, M. Murata, and H. Miyahara, "Implementation and evaluation of video filtering mechanisms for real-time multicast," in *Technical Report of IEICE (NS2001-50)*, pp. 13–18, June 2001.
- [16] M. Handley, S. Floyd, J. Padhye, and J. Widmer, "TCP Friendly Rate Control (TFRC): Protocol specification," *Internet Request for Comments 3448*, Jan. 2003.
- [17] "NIST Net." available at <http://snad.ncsl.nist.gov/itg/nistnet/>.