

プロキシ協調型動画像配信システムの検討

若宮 直紀[†] 村田 正幸[†] 宮原 秀夫[†]

[†] 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 1-2-3

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

あらまし 動画像配信システムにおいては、それぞれのクライアントの処理能力や帯域などについて考慮しつつ、応答性の高いサービスを提供しなければならない。本稿では、様々に異なる品質の動画像を要求する複数のクライアントが存在する環境において、要求品質に応じた動画像を効率よく配信するためのプロキシ協調型動画像配信システムを提案し、動画像再生の途切れを抑えることができるなど、その有効性を示した。提案システムでは、ウェブシステムで広く用いられているプロキシキャッシング技術を動画像配信に応用し、ネットワーク内に複数配置されたプロキシキャッシュサーバが互いに連携、協調することにより、サービスの応答性を高めている。また、プロキシに動画像品質調整機能を持たせることにより、クライアントからの要求品質の多様性に対応する。

キーワード 動画像配信, プロキシキャッシュ, QoS, 動画像品質調整

Study on Cooperative Proxy Caching Mechanism for Video Streaming Services

Naoki WAKAMIYA[†], Masayuki MURATA[†], and Hideo MIYAHARA[†]

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

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

Abstract The proxy technique is originally developed for WWW system, but it is expected to be useful to improve the performance and quality of video streaming services. Furthermore, if each proxy is capable of adapting incoming or cached video data to user's demand, user's QoS requirements on video streams can be fully satisfied. For these purposes, proxies communicate with each other, retrieve a missing video block from an appropriate server by taking into account transfer delay and offerable quality, and adjust quality of the block to the client request. We show that our proposed mechanisms are effective in avoiding freezes during video playback and providing clients with high-quality and low-delay video streaming services.

Key words Video Streaming Service, Proxy Caching, QoS, Video Quality Adjustment

1. はじめに

ADSL 技術や光ファイバの導入による一般家庭のアクセス回線容量の増大を背景に、CDN (Content Distribution/Delivery Network) 技術にもとづく動画像配信サービスが注目を集めている。動画像配信システムにおいては、トラフィック量が多いため容易にネットワークの輻輳を引き起こす動画像データを、効率よく、高速に多数のクライアントに配送しなければならない。

WWW (World Wide Web) では、例えばドメインごとにプロキシキャッシュサーバと呼ばれる代理サーバを設置し、ドメイン内のクライアントの要求に応じて WWW サーバからデータを取得、提供すると同時にキャッシュバッファと呼ばれるバッファに蓄積する。新たなデータ転送要求に対し、キャッシュ内に一致するものがあればこれを提供することにより、サーバ-クライアント間のデータ転送遅延を隠蔽し、応答時間の短縮を図ることができる。さらに、プロキシをクライアントの近くに設置することにより、バックボーンネットワークを流れるトラフィックを抑えることができる。また、キャッシュバッファに該当するデータがある場合には、サーバはクライアントからのデータ転送要求を直接処理する必要がないため、サーバ負荷が軽減される [1]。

動画像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可能になると考えられる。しかしながら、現在 WWW で用いられているプロキシキャッシュサーバは、ファイルを単位としてデータを取得、蓄積、配信するため、一つの動画像ストリームが数ギガバイトに達する動画像配信には適さない。また、テキスト、静止画などと異なり、動画像の場合にはクライアントは定常的にデータを受信、復号化、再生するため、ネットワークの輻輳状態に関わらず、動画像を構成するピクチャを再生時刻に間に合うよう順次配送しなければならない。さらに、同じコンテンツであっても、クライアントシステムの性能や利用可能な帯域などの違いにより、クライアントの要求する、あるいは処理可能な動画像品質が様々に異なる。したがって、たとえプロキシが大容量のキャッシュバッファを有していても、必ずしも全てのクライアントの要求に応じた種別、品質の動画像を、高い応答性を保ちつつ、提供できるとは限らない。動画像配信のためのプロキシキャッシュ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 [2] ~ [4]、クライアントに提供される動画像の品質について考慮していない、階層符号化技術を利用しているため様々な要求品質に柔軟に対応できない、などの問題がある。

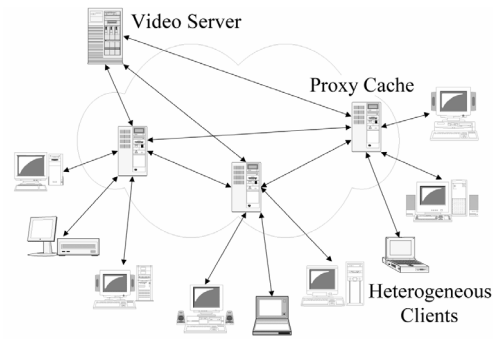


図 1 プロキシキャッシュを有する動画像配信システム

文献 [5] において我々は、様々な要求品質を考慮した動画像配信のためのプロキシキャッシングメカニズムを提案し、その有効性を示した。提案手法は、プロキシキャッシュサーバに動画像品質調整機能を持たせ、様々な要求品質に対して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像配信を実現している。また、キャッシュ管理、データ転送を効率よく行うため、動画像データを複数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、キャッシュミス発生時の待ち合わせ時間を低く抑えるための先読み機構についても検討している。本稿では、さらに効果的な制御を実現するため、ネットワーク内に配置された複数のプロキシキャッシュサーバが互いに連携することにより動画像配信サービスを提供するプロキシ協調型動画像配信システムを提案する。提案手法を用いることにより、ネットワーク負荷の軽減、サーバ負荷の分散、遅延の隠蔽、およびデータ可用性の向上を図れると同時に複数のクライアントからのさまざまな要求を満足することのできる動画像配信を実現できる。

2. システム概要

本稿では、図 1 に示すような、一台の動画像サーバ、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、および動画像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動画像配信システムを対象とする。それぞれの間の動画像データ転送は、動画像ストリームにつき 1 組ずつ設定される RTSP/TCP および RTP/UDP セッションを用いて行われる。動画像サーバはプロキシからの、プロキシはクライアントからのデータ転送要求に応じてストリームを分割して生成されるブロックごとに動画像データを送出する。

まず、クライアントは SDP (Session Description Protocol) などをおして動画像配信サービスの情報を取得する。次に、近隣のプロキシキャッシュサーバとの間に RTSP セッションを確立し、動画像に対

する要求品質やクライアントのシステム構成（CPU 処理能力，通信速度，バッファサイズなど）の情報をプロキシに通知するとともに，RTSP のPLAY メッセージにより動画ブロックの送信を要求する．以降，ブロックの受信を完了すると次のブロック転送要求を送出する．

クライアントからの要求にもとづき，プロキシはキャッシュバッファを検索し，該当するブロックが蓄積されており，かつ，要求品質を満たす場合（キャッシュヒット）には，これを適切に品質調整した後，送出手する．一方，キャッシュ内に適切なブロックがない場合（キャッシュミス）には，プロキシは適切な品質のブロックを，動画サーバや適切な近隣プロキシキャッシュサーバから取得し，キャッシュバッファに蓄積するとともに，必要に応じて品質を調整した後，クライアントに転送する．他のプロキシからのブロック転送要求に対し，動画サーバや近隣プロキシは自身の持つ動画ブロックに必要なに応じて品質調整を施した後，プロキシに提供する．

動画サーバが全てのクライアントの品質要求を考慮し，さまざまな動画データを生成，送出手しなければならない従来の動画配信システムとは異なり，提案システムでは，動画サーバは全ての要求のうち，最も高品質なものを上回る動画ストリームだけをあらかじめ用意しておけばよい．評価に際しては，文献 [6] などで MPEG-2 符号化動画向けに提案された動画品質調整技術を用いているが，本稿で提案するアルゴリズムや手法は他の符号化手法，品質調整技術，あるいは同種の機能を有する市販機器を用いた場合にも適用可能である．

クライアントは，ブロック転送遅れなどによる動画再生の途切れを防ぐため，サービス開始後数秒間は取得したブロックを先読みバッファに蓄積する．その後，バッファ内ブロックから順に復号，再生し，同時に将来必要となる動画ブロックをプロキシから取得，バッファに蓄積する．

3. プロキシキャッシングメカニズム

本章では，動画品質調整可能なプロキシキャッシュサーバの協調による動画配信メカニズムについて述べる．簡単のため，単一ストリームの場合について取り扱うが，動画サーバが複数のストリームを提供する場合についても本章で述べるメカニズムを拡張することにより容易に対応可能である [4]．

3.1 キャッシュテーブルとリモートテーブル

プロキシは，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を管理するためのリモートテーブルを保持している．要求するデータの範囲をタイムスタンプによって指定する RTSP との親和性を考慮し，フレームやピクチャ，あるいは

MPEG 符号化における GoP (Group of Pictures) のようなピクチャの集合をブロックとする．動画ストリームはそれぞれ N 枚のピクチャからなる L 個のブロックから構成されているものとする．

キャッシュテーブルは，ブロック番号 i ，品質 $q(i)$ ，マーカ $M(i)$ からなる．大きい $q(i)$ ほど品質が高いことを， $q(i) = 0$ はブロック i がキャッシュ内不在であることを，それぞれ示す．プロキシ識別子の集合であるマーカ $M(i)$ は，QUERY メッセージによって更新され，マーカが空でないブロックは他のプロキシから要求される可能性があることを示す．

キャッシュミス発生時には，プロキシは，クライアントからの要求品質，動画ブロックのデータサイズ，クライアントの先読みバッファ内ブロック数，プロキシ-クライアント間のデータ転送時間，プロキシ-他のサーバ間のデータ転送時間などを考慮して，クライアントでの動画再生が途切れないうよう，適切な動画サーバや近隣プロキシキャッシュサーバから適切な品質の動画ブロックを取得しなければならない．そのため，プロキシは，動画サーバや近隣プロキシの提供可能なブロックとその品質，それらサーバとの伝搬遅延，転送速度などに関する情報を保持，管理する必要がある．

動画サーバ，近隣プロキシキャッシュサーバなどサーバ k の提供可能なブロックの品質に関する情報を保持するリモートテーブルは，片方向伝搬遅延の推定値 d_k^S ，転送速度の推定値 r_k^S ，ブロック i の品質 $O_k(i)$ からなり，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．伝搬遅延については *ping*，*echoping*，転送速度については *pathchar* や *pchar* などの計測手法を用いることにより推定することができる．ネットワークの効率的かつ公平な利用を目的に，動画ブロック転送に際して TCP-friendly の概念にもとづくレート制御手法 [7], [8] を用いている場合には，RTP，RTCP による RTT，スループットの推定値を利用することができる．

プロキシは，新しいクライアントからのデータ転送要求を受信すると，システム内で共通のマルチキャストアドレスを用いて QUERY メッセージを送出し，提供可能なブロックに関する情報を動画サーバや近隣プロキシに問い合わせる．問い合わせの対象となったブロックは，後述するブロック置き換えの対象から除外されるため，問い合わせウィンドウ I によりその範囲を制限する．クライアントが参照中のブロック，およびストリームの先頭のブロックからそれぞれ I 個のブロックを問い合わせの対象とする．また，クライアントが参照中のブロックから $P - 1$ 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新たに 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、マーカを設定、解除するとともに、提供可能なブロック品質 $q(i)$ をREPLY メッセージにより回答する。

3.2 ブロック取得アルゴリズム

クライアントは RTSP のPLAY メッセージを用いてブロックごとにプロキシに動画像データ転送を要求する。クライアント j のブロック i に対する要求品質を $q_j(i)$ とする。ブロック転送要求を受信したプロキシは、キャッシュテーブル、リモートテーブル、サーバ-プロキシ間と同様の手法で導出されるプロキシ-クライアント j 間の片方向伝搬遅延の推定値 d_j^C と転送速度の推定値 r_j^C 、PLAY メッセージを利用して通知される先読みバッファ内ブロック数 p_j 、および動画像配信サービスに対する要求を表す指標 β_j にもとづいて、ブロック i の提供方法を決定する。 β_j ($0 < \beta_j \leq 1$) は、セッション開始時にクライアントから通知され、配信されるブロック品質に対する要求の厳しさを表す。常に要求品質どおりのブロックを必要とするクライアントは $\beta_j = 1$ を選択することになるが、ブロック配信の遅れにより動画像再生が途切れる可能性がある。一方、動画像品質の劣化に寛容なクライアントは 0 に近い値を選択することにより、連続した動画像受信、再生を達成することができる。なお、プロキシは品質 q のブロック i のデータサイズ $s(i, q)$ を算出可能、あるいは既知であるものとする [9]。

プロキシがクライアント j に提供可能なブロック i の品質 $q_j^P(i)$ は以下の式で与えられる。

$$q_j^P(i) = \min(q(i), q_j^{Pmax}(i)) \quad (1)$$

ここで、 $q_j^{Pmax}(i)$ はクライアント j の先読みバッファを枯渇させることなく転送できるブロックの最高品質を表し、以下の式により求められる。

$$q_j^{Pmax}(i) = \max(q | 2d_j^C + \frac{s(i, q)}{r_j^C} \leq \frac{p_j N}{F} - \Delta) \quad (2)$$

Δ は、遅延の揺らぎや推定誤差による影響を抑えるための緩衝時間である。セッション開始時はクライアントの先読みバッファにはブロックが蓄積されておらず $q_j^P(i) = 0$ となるため、十分な数のブロックを先読みするまでは適当な先読みバッファ内ブロック数 p_j をプロキシに通知する。プロキシは、ブロック転送要求 $q_j(i)$ を受信すると、まず、キャッシュテーブルを検索し、キャッシュ内ブロックが要求品質を満たす場合(キャッシュヒット)、すなわち、 $q_j^P(i) \geq \beta_j q_j(i)$ が成立する場合には、要求品質に応じて動画像品質調整を行ったのち、品質 $\hat{q}_j(i) = \min(q_j(i), q_j^P(i))$ のブロックをクライアントに提供する。

キャッシュミス発生時には、まず他のサーバに転送要求中のブロックでクライアント j の要求を満足できるかどうか調べ、不可能な場合には新たに適

切なサーバにブロック転送要求を送る。そのため、プロキシは他のサーバ k に対して送出したブロック転送要求のリスト $Q_k = \{q_n^{Qk}\}$ を保持している。 q_n^{Qk} は、サーバ k に送出された n 番目の転送要求によるブロック b_n^k の要求品質を表す。リスト Q_k は、他のサーバに対して転送要求を送出、あるいはブロック受信を完了するごとに更新される。プロキシは、 m 番目 ($m \leq n$) の要求がブロック i に対するものであり ($b_m = i$)、その品質が要求を上回り ($q_m^{Qk} \geq \beta_j q_j(i)$)、かつブロック取得、転送が再生間に合う ($\frac{\sum_{n=1}^m s(b_n^k, q_n^{Qk})}{r_k^S} \leq \frac{p_j N}{F} - \Delta$) 場合には、品質 $\hat{q}_j(i) = \min(q_j(i), q_m^{Qk})$ のブロックを提供する。

ブロックの取得が間に合わない、あるいは他のサーバに要求中のブロックの品質ではクライアントの要求品質を満足することができない場合には、新たに適切なサーバを選択し、動画像ブロックの転送を要求しなければならない。サーバ k がプロキシを介してクライアント j に提供可能なブロック i の品質 $q_{k,j}^S(i)$ は、次式で導出される。

$$q_{k,j}^S(i) = \min(O_k(i), q_{k,j}^{Smax}(i)) \quad (3)$$

ここで、クライアント j の先読みバッファを枯渇させることなく提供可能なブロックの最高品質 $q_{k,j}^{Smax}(i)$ は、以下の式で与えられる。

$$q_{k,j}^{Smax}(i) = \max(q | 2d_j^C + \max(\frac{\sum_n s(b_n, q_n^{Qk})}{r_k^S}, 2d_k^S) + \max(\frac{s(i, \hat{q})}{r_j^C}, \frac{s(i, q)}{r_k^S}) \leq \frac{p_j N}{F} - \Delta) \quad (4)$$

動画像サーバ、近隣プロキシのうち、提供可能な品質が最も高く、高速に転送可能なものに対してブロック転送要求を送出するが、要求品質 $q_{k,j}(i)$ は、そのブロックを利用する可能性のある全てのクライアント m の要求品質の最大値 $\max_{l_m \leq i} q_m(l_m)$ 以下とする。

近隣サーバのいずれも時間内に十分な品質のブロックを提供できない場合には最低許容品質 $\beta_j q_j(i)$ のブロックを最も速く転送可能なサーバから取得する。クライアント j におけるブロック i の再生までの待ち合わせ時間 $f_j(i)$ は次式で導出される。

$$f_j(i) = 2d_j^C + \max(\frac{\sum_n s(b_n, q_n^{Qk})}{r_k^S}, 2d_k^S) + \frac{s(i, \beta_j q_j(i))}{\min(r_j^C, r_k^C)} - \frac{p_j N}{F} + \Delta \quad (5)$$

3.3 ブロック先読みアルゴリズム

転送遅れによる待ち合わせを引き起こさないよう、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。ブロック i に対する転送要求をクライアントから受信すると、プロキシはブロック $i+1$ から $i+P$ についてもキャッシュテーブルと他のサーバへ

の要求リスト Q_k を検索し、最低許容品質 $\beta_j q_j(i)$ と比較する。 P は先読みウィンドウと呼び、先読みの範囲を指定する。将来のキャッシュミスが予想されるブロックについてはこれを先読みするが、不十分な品質のブロックが複数ある場合にはブロック i に最も近いものを先読み対象とする。ただし、動画サーバやプロキシでは、キャッシュミスによるブロック転送要求と先読み要求とは区別して扱われる。前者は 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 の待ち行列によって管理され、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 待ち行列に要求のない場合にのみ処理される。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。

サーバ k がクライアント j に提供可能なブロック m の品質 $q_{k,j}^R(m)$ は、次式によって求められる。

$$q_{k,j}^R(m) = \min(O_k(m), q_{k,j}^{Rmax}(m)) \quad (6)$$

クライアントの先読みバッファを考慮した提供可能な最高品質 $q_{k,j}^{Rmax}(m)$ は、次式から与えられる。

$$q_{k,j}^{Rmax}(m) = \max(q|d_q \leq \frac{(p_j + m - i)N}{F} - \Delta) \quad (7)$$

ブロック転送時間 d_q は、ブロック転送を要求するサーバや、ブロック i の提供方法などによって異なるが、紙幅の都合により導出法については省略する。

3.4 ブロック置き換えアルゴリズム

プロキシの備えるキャッシュバッファ容量は有限のため、新たに取得したブロックを蓄積するため、キャッシュ内ブロックを品質調整、棄却しなければならない場合がある。ユーザは、通常、動画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 [10]、近い将来に必要とされる、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 $M(i)$ 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。

まず、プロキシは置き換え対象ブロック n に動画品質調整を施し、データサイズを減らすことにより、キャッシュバッファに空きを作る。ただし、キャッシュミスを防ぐため、ブロック $l_m \leq n$ を参照しているクライアント m の要求品質を考慮し、 $\max_{l_m < n} \beta_m q_m(l_m)$ まで品質調整した後、空きが不十分な場合にはブロックをキャッシュから取り除く。新たに取得したブロックの蓄積に十分な空き容量ができるまで、置き換え対象ブロックを順に決定し、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

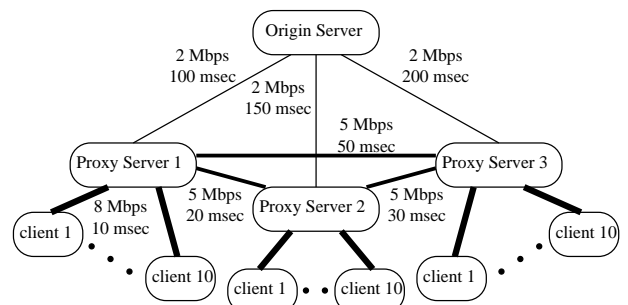


図 2 シミュレーションモデル

ンドウ内ブロックのうち、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。

4. シミュレーション評価

本章では、図 2 に示すそれぞれ 10 台のクライアントが接続された 3 台のプロキシを含むシステムについて、シミュレーションにより提案手法の性能を評価する。それぞれのセッションで利用可能な帯域、片方向伝搬遅延は図のとおりであり、シミュレーションをとおして固定とした。

動画ストリームは 90 分、フレームレート 30 fps で再生され、ブロック長は 1 秒とする ($L = 5400$, $N = 30$)。ブロックは 10 段階の品質調整が可能なものとし、そのサイズは $\forall i s(i, q) = \frac{q^N}{F} \times 10^6$ ビットで与えられる。ウィンドウサイズはそれぞれ $P = 10$, $I = 20$ ブロックとする。緩衝時間 Δ は 2 秒、動画再生開始までの先読み時間は 4 秒とする。サービス開始直後の先読み中の要求品質 $q_j(i)$ を 1 とし、再生開始後は、それぞれ 10% の確率で要求品質を 1 ずつ上げ下げする。また、要求の厳しさを表す指標 β_j を 0.6 とする。クライアントのセッション開始の間隔は平均 30 分の指数分布で与えた。なお、全ての試行間でクライアントのサービス要求開始時刻および要求品質の変化を同一にした。

評価に際しては、キャッシュミス発生時に常に動画サーバからブロックを取得し、かつ先読みを行わない “Independent w/o Prefetch”，協調しないが先読みを行う “Independent c/w Prefetch”，協調するが先読みを行わない “Cooperative w/o Prefetch” と先読みを行う提案手法 “Cooperative c/w Prefetch” の 4 方式について比較した。ただし、全てのプロキシに同じ手法を導入するものとする。以降ではプロキシ 2 に着目し、評価結果を示す。

図 3 および図 4 に、プロキシのキャッシュバッファサイズが無敵大の場合の、クライアントごとの待ち合わせ時間 $f_j(i)$ の平均値と、要求品質に対する受信ブロック品質の比の平均値を示す。いずれの図においても、先読みのあるなしに関わらず動画サーバからのみブロック取得を行う手法では、ほぼ同じ

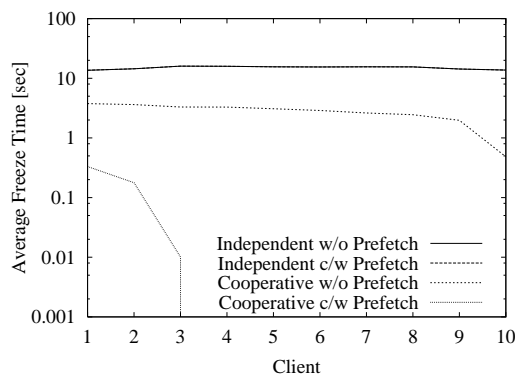


図3 平均待ち合わせ時間 (バッファ無限大)

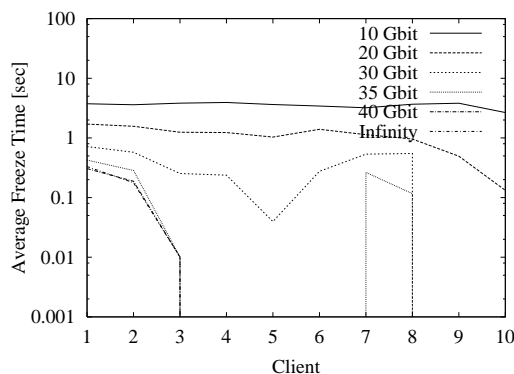


図5 平均待ち合わせ時間比較

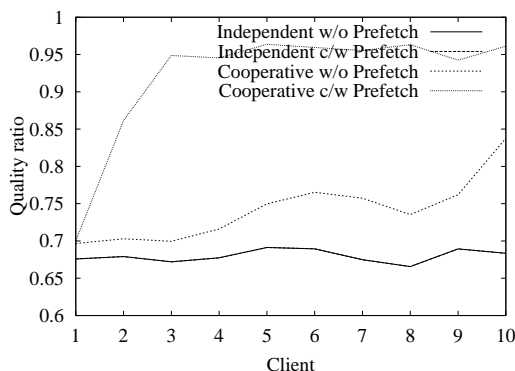


図4 平均要求品質充足度 (バッファ無限大)

結果となっている。これは、サーバ-プロキシ間の帯域がキャッシュミスによるブロック転送にほぼ占有されるためである。図3より、プロキシ間協調制御を行うことにより待ち合わせ時間を短縮できることが分かる。特に、先読み機構を導入することにより、クライアント4以降では待ち合わせを行うことなく、スムーズな動画再生が実現できている。また、図4に示されているとおり、提案手法を用いることでほぼ要求どおりの品質で動画データが提供される。

ブロック置き換えアルゴリズムの有効性を評価するため、図5にキャッシュバッファ容量を10, 20, 30, 35, 40ギガビットとした場合の結果を示す。なお、図3, 4における最大バッファ内データ量は約49ギガビットである。図に示されるとおり、それほど大きな性能劣化を引き起こすことなく必要バッファ容量を3割程度削減できる。ただし、クライアント7, 8においてはブロック取得にともなう動画再生の途切れが発生している。

5. おわりに

本稿では、置かれる環境により様々に異なる品質の動画データを要求する複数のクライアントが存在する動画配信サービスにおいて、個々の要求を考慮しながら効率よく動画を配信するため、動画品質調整可能なプロキシを利用した、プロキシ協調型動画配信システムを提案し、シミュレーションによりその有効性を評価した。提案システムは、ブ

ロックごとの品質調整可能な動画データを扱う動画配信サービスに適用可能であるが、2. や3. で述べたように、プロキシ間の情報共有機構、ネットワーク状態の推定機構などについて、いくつかの前提、仮定をおいている。したがって、提案システムの実装をとおして、適用性、実用性などについて検証することが必要である。

文 献

- [1] G. Barish and K. Obraczka, "World Wide Web caching: Trends and techniques," *USC Technical Report 99-713*, 1999.
- [2] R. Rejaie, H. Yu, M. Handley, and D. Estrin, "Multimedia proxy caching mechanism for quality adaptive streaming applications in the Internet," in *Proceedings of IEEE INFOCOM 2000*, March 2000.
- [3] M. Andrews and K. Munagala, "Online algorithms for caching multimedia streams," in *Proceedings of ESA 2000*, pp. 64-75, September 2000.
- [4] M. Reisslein, F. Hartanto, and K. W. Ross, "Interactive video streaming with proxy servers," in *Proceedings of IMCN*, pp. 588-591, February 2000.
- [5] M. Sasabe, N. Wakamiya, M. Murata, and H. Miyahara, "Proxy caching mechanisms with video quality adjustment," in *Proceedings of SPIE IT-Com*, pp. 276-284, August 2001.
- [6] 赤嶺エクトル, 中田和久, 若宮直紀, 村田正幸, 宮原秀夫, "実時間動画マルチキャストのためのフィルタリング手法の実装と評価," 電子情報通信学会技術研究報告 (NS2001-50), pp. 13-18, June 2001.
- [7] S. Floyd, M. Handley, J. Padhye, and J. Widmer, "Equation-based congestion control for unicast applications: the extended version," *International Computer Science Institute technical report TR-00-003*, March 2000.
- [8] M. Miyabayashi, N. Wakamiya, M. Murata, and H. Miyahara, "MPEG-TFRCP: Video transfer with TCP-friendly rate control protocol," in *Proceedings of IEEE ICC 2001*, pp. 137-141, June 2001.
- [9] K. Fukuda, N. Wakamiya, M. Murata, and H. Miyahara, "QoS mapping between user's preference and bandwidth control for video transport," in *Proceedings of IFIP IWQoS '97*, pp. 291-302, May 1997.
- [10] S. Sen, J. Rexford, and Don Towsley, "Proxy prefix caching for multimedia streams," in *Proceedings of IEEE INFOCOM '99*, pp. 1310-1319, March 1999.