

特別研究報告

題目

動画像品質調整機能を組み込んだ
プロキシキャッシングシステム

指導教官

宮原 秀夫 教授

報告者

笹部 昌弘

平成 13 年 2 月 22 日

大阪大学 基礎工学部 情報科学科

動画像品質調整機能を組み込んだプロキシキャッシングシステム

笹部 昌弘

内容梗概

コンピュータの高性能化やアプリケーションのマルチメディア化にともない、インターネットトラフィックにおける動画像データの占める割合は年々増加している。動画像トラフィックは定常的かつデータ量が多いため、ネットワークの輻輳を引き起こす。輻輳が発生するとデータ転送遅延が増大するため、マルチメディアアプリケーションの要求する実時間性や応答性が提供できなくなる。

WWW システムで今日広く用いられているプロキシ技術は、網内の中継システム(プロキシ)においてクライアントが過去に利用したデータを蓄積(キャッシュ)し、新たなデータ転送要求に対してサーバに代ってキャッシュデータを転送することにより、サーバからの伝送遅延を隠蔽し、高速なマルチメディアデータ配送を実現するものである。動画像通信においても、プロキシ技術を用いることにより、ネットワークに大きな負荷を与えることなくアプリケーションの実時間性や応答性を高めることができると考えられる。さらに、複数のクライアントがそれぞれ異なる品質の動画像を要求する動画像マルチキャスト通信において、プロキシで蓄積データを適切に品質調整することにより、ネットワークを流れる動画像トラフィックを減らすと同時に、高品質な動画像を配信することができる。また、サーバから動画像データを取得する場合に比べて、それぞれの動画像通信セッションの往復伝搬遅延が小さくなるため、近年活発に研究が行われている実時間通信のためのレート制御を適用した場合にもより高い制御効果が期待される。

本報告では、利用可能な帯域の範囲内で、ユーザの要求を考慮した高品質な動画像通信を実現することのできるプロキシキャッシングアルゴリズムを提案する。提案アルゴリズムでは、クライアントの利用可能な帯域が時間によって変化することを考慮し、動画像ストリームを複数のピクチャからなる固まりに分割し、蓄積、加工、転送、取得の単位とする。提案手法では、プロキシはクライアントからの要求にもとづいてサーバから動画像データを取得し、蓄積すると同時に品質調整を行った後、クライアントに配送する。帯域の有効利用とデータ転送遅延の短縮をはかるためキャッシュデータの品質がクライアントの要求を満足できる場合には動画像データの参照性を考慮してサーバ-プロキシ間の空き帯域を利用した動

画像データの先読みを行う．プロキシがサーバに要求する動画像データの品質決定法，データ配送に用いられない空き帯域の利用法，バッファが不足した際のデータ置き換え手法のそれぞれ異なる複数のアルゴリズムを提案し，必要となるバッファサイズ，動画像再生開始までの待ち時間，提供される動画像品質などについて比較評価を行う．シミュレーションにより，動画像品質の低下がある程度許される場合には，キャッシュバッファサイズが小さい場合でも動画像データの先読みとキャッシュデータの置き換えアルゴリズムを適切に組み合わせることにより，動画像再生開始までの待ち時間や動画像品質をそれほど劣化させることなく動画像配信サービスが実現できることを示す．

主な用語

品質調整，動画像通信，プロキシキャッシング，TCP-friendly，MPEG-2

目次

1	はじめに	7
2	プロキシキャッシュを用いた動画像ストリーミング配信システム	9
2.1	MPEG-2 動画像符号化方式	9
2.2	MPEG-2 動画像ストリーミング配信システムの概要	10
3	動画像品質調整機能を組み込んだプロキシキャッシングメカニズム	12
3.1	メカニズムの概要	12
3.2	クライアントの要求品質を考慮した動画像データ取得	15
3.3	性能評価	17
3.3.1	サーバ-プロキシ間で利用可能な帯域が大きい場合	19
3.3.2	サーバ-プロキシ間で利用可能な帯域が小さい場合	24
4	プロキシにおける動画像データ先読みメカニズム	28
4.1	動画像先読みのためのキャッシュ検索アルゴリズム	28
4.2	先読みする動画像データの品質	29
4.3	サーバにおける先読み要求処理メカニズム	29
4.4	性能評価	30
5	プロキシにおけるキャッシュデータ置き換えアルゴリズム	38
5.1	キャッシュデータの置き換えアルゴリズム	38
5.2	性能評価	38
5.2.1	利用可能なバッファサイズが大きい場合	39
5.2.2	利用可能なバッファサイズが比較的小さい場合	47
6	おわりに	54
	謝辞	55
	参考文献	56

目 次

1	GoP 構成例	9
2	プロキシキャッシュを用いた MPEG-2 動画像ストリーミング配信システム	10
3	動画像データの転送処理例	13
4	利用可能な帯域データ転送レート	14
5	量子化スケールと平均レートの関係	14
6	キャッシュヒットとキャッシュミス	15
7	表示間隔の揺らぎ	19
8	再生までの待ち時間	20
9	利用可能な帯域の変化 (サーバ-プロキシ間が広帯域な場合)	20
10	クライアントの参加時間と平均レート (サーバ-プロキシ間が広帯域な場合)	21
11	再生開始までの待ち時間 (サーバ-プロキシ間が広帯域な場合)	22
12	受信動画像品質に対する満足度 (サーバ-プロキシ間が広帯域な場合)	23
13	サーバ-プロキシ間の帯域利用率 (サーバ-プロキシ間が広帯域な場合)	23
14	キャッシュ内のデータ量の変化 (サーバ-プロキシ間が広帯域な場合)	24
15	利用可能な帯域の変化 (サーバ-プロキシ間が狭帯域な場合)	25
16	クライアントの参加時間 (サーバ-プロキシ間が狭帯域な場合)	25
17	再生開始までの待ち時間 (サーバ-プロキシ間が狭帯域な場合)	26
18	受信動画像品質に対する満足度 (サーバ-プロキシ間が狭帯域な場合)	26
19	サーバ-プロキシ間の帯域利用率 (サーバ-プロキシ間が狭帯域な場合)	27
20	キャッシュ内のデータ量の変化 (サーバ-プロキシ間が狭帯域な場合)	27
21	サーバにおける先読み要求処理メカニズム	29
22	再生開始までの待ち時間 (先読みあり, $\beta = 1$)	31
23	サーバ-プロキシ間の帯域利用率 (先読みあり, $\beta = 1$)	32
24	キャッシュ内のデータ量の変化 (先読みあり, $\beta = 1$)	33
25	再生開始までの待ち時間 (先読みあり, $\beta = 0.6$)	34
26	受信動画像品質に対する満足度 (先読みあり, $\beta = 0.6$)	35
27	サーバ-プロキシ間の帯域利用率 (先読みあり, $\beta = 0.6$)	36
28	キャッシュ内のデータ量の変化 (先読みあり, $\beta = 0.6$)	37
29	キャッシュ内に残すデータの優先度	39
30	再生までの待ち時間 (バッファサイズ 40Gbit, $\beta = 1$)	40
31	サーバ-プロキシ間の帯域利用率 (バッファサイズ 40Gbit, $\beta = 1$)	41
32	キャッシュ内のデータ量の変化 (バッファサイズ 40Gbit, $\beta = 1$)	42

33	再生までの待ち時間 (バッファサイズ 40Gbit , $\beta = 0.6$)	43
34	受信動画品質に対する満足度 (バッファサイズ 40Gbit , $\beta = 0.6$)	44
35	サーバ-プロキシ間の帯域利用率 (バッファサイズ 40Gbit , $\beta = 0.6$)	45
36	キャッシュ内のデータ量の変化 (バッファサイズ 40Gbit , $\beta = 0.6$)	46
37	再生までの待ち時間 (バッファサイズ 20Gbit , $\beta = 1$)	47
38	サーバ-プロキシ間の帯域利用率 (バッファサイズ 20Gbit , $\beta = 1$)	48
39	キャッシュ内のデータ量の変化 (バッファサイズ 20Gbit , $\beta = 1$)	49
40	再生までの待ち時間 (バッファサイズ 20Gbit , $\beta = 0.6$)	50
41	受信動画品質に対する満足度 (バッファサイズ 20Gbit , $\beta = 0.6$)	51
42	サーバ-プロキシ間の帯域利用率 (バッファサイズ 20Gbit , $\beta = 0.6$)	52
43	キャッシュ内のデータ量の変化 (バッファサイズ 20Gbit , $\beta = 0.6$)	53

表 目 次

1	キャッシュデータ管理リスト	13
2	サーバへ要求する動画像品質の決定アルゴリズム	18
3	シミュレーション条件	18
4	先読みメカニズムのためのキャッシュデータ管理リスト	28

1 はじめに

近年のコンピュータの高速化やアプリケーションのマルチメディア化，ネットワークの広帯域化にともない，ストリーミングサービスなどの動画像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，定期的かつ大量の動画像トラフィックによりネットワークの輻輳が発生し，既存のデータ通信アプリケーションの通信品質が劣化するだけでなく，データ転送遅延が増大するためマルチメディアアプリケーションの実時間性や応答性が満たせなくなることが問題となっている．

ネットワークの負荷を抑え，データ配送遅延を小さくすることでアプリケーションの応答性を高める手段として WWW (World Wide Web) システムで広く用いられているプロキシ技術がある．クライアントはサーバとの間に設置されたプロキシと呼ばれる代理サーバに対してデータ転送を要求し，プロキシはクライアントに代わってサーバから必要なデータを取得してクライアントに提供する．さらにプロキシは取得したデータをキャッシュバッファと呼ばれるプロキシ内記憶装置に蓄積 (キャッシュ) し，同じデータに対する新たな転送要求があった場合には，サーバからのデータ再取得を行うことなくサーバに代わってキャッシュデータをクライアントに転送する．プロキシにより，サーバ-クライアント間のトラフィックを減らすと同時に，サーバ負荷を軽減し，クライアントへの高速なデータ配信が可能となる．

動画像通信システムにおいても，プロキシ技術を適用することにより，ネットワークやサーバの負荷軽減，実時間で応答性の高い動画像データ配信サービスが実現できると考えられる．しかしながら，現在のプロキシ技術はテキストや静止画像といったファイルを単位とした蓄積を行うため，ひとつのファイルが数十ギガバイトにもなる動画像データにはそのまま適用できない．また，動画像配信サービスの場合には，それぞれのクライアントごとにネットワークへの接続形態やシステム性能，あるいはユーザの再生動画像に対する好みによって様々な異なる動画像品質への要求を扱わなければならない．したがって，たとえプロキシに大容量の記憶装置を備えたとしてもキャッシュから提供可能な動画像の種類や品質は非常に限られる．

そこで本報告では，ネットワークやサーバに負荷を与えることなく動画像配信サービスの応答性を高めると同時に，様々な異なるクライアントの要求品質を考慮した動画像データ配送を行うプロキシキャッシングシステムを提案する．動画像配信のためのプロキシではフレーム棄却，ローパス，再量子化といった動画像品質調整技術 [1] を利用し，クライアントの要求品質にもとづいてサーバから取得した，あるいはキャッシュ内に蓄積された動画像データの品質を適切に調整し，クライアントに提供する．提案システムではいくつかのピクチャをひとまとめとした単位で動画像データを取り扱うことにより，キャッシュバッファの効率的な利用を図る．クライアントからのサービス利用開始から再生開始までの時間を短くする

ことで応答性を高めると同時に高品質な動画像を提供可能な動画像配信サービスを実現するためには、サーバからどのような品質の動画像データを取得し、どのように蓄積し、クライアントに提供するかについて検討しなければならない。また、有限なキャッシュバッファを効率的に利用するためには、新たに取得した動画像データを蓄積するため適切なキャッシュデータを置き換えるアルゴリズムが必要となる。さらに、データ転送遅延を低く抑えるためには、キャッシュデータでクライアントの要求品質が満たされる場合などに生じるサーバ-プロキシ間の空き帯域を利用して、将来必要になるとと思われる動画像データを適切な品質であらかじめ先読みするのも有効であると考えられる。このような、動画像品質調整機能を組み込んだプロキシキャッシングシステムにおける様々な検討課題について、効果的な手法を提案し、それらをキャッシュバッファサイズ、動画像再生開始までの遅延、動画像品質などにもとづいて比較評価することにより適用範囲や有効性を示す。

本報告の構成は以下のとおりである。まず、第2章では、本報告で対象とするプロキシキャッシュを用いた動画像ストリーミング配信システムの概要について述べる。第3章では、本報告で提案した、動画像品質調整機能を組み込んだプロキシキャッシングメカニズムについて述べ、シミュレーションによりその基本性能を評価する。第4章では、提案手法の性能向上を図るため、プロキシにおける動画像データ先読みメカニズムを導入し、シミュレーションによりその有効性を評価する。続く、第5章では、キャッシュバッファサイズが有限な場合に対応できるようキャッシュデータの置き換えアルゴリズムを提案、評価する。最後に、第6章で本報告のまとめと今後の課題について述べる。

2 プロキシキャッシュを用いた動画像ストリーミング配信システム

本報告では、高品質な圧縮動画像データを生成する符号化手法として広く用いられている MPEG-2 符号化方式により実時間で、あるいはあらかじめ符号化された動画像データを扱う。そこで本章では、まず、動画像符号化方式 MPEG-2 について簡単に紹介し、プロキシキャッシュを用いた MPEG-2 動画像ストリーミング配信システムの概要について述べる。

2.1 MPEG-2 動画像符号化方式

MPEG-2 動画像符号化方式 [2] により生成されたデータは GoP (Group of Pictures) と呼ばれるそれぞれ符号化アルゴリズムの異なる 3 種類のピクチャの集合の繰り返しからなる (図 1)。単一ピクチャの情報のみを用いて符号化される I (Intra-coded) ピクチャは単独で符号化され、復号化が可能で GoP の先頭に位置する。GoP の残りのピクチャは、直前の I または P ピクチャとの間で前方向予測符号化を行なうことにより符号化データ量を減らす P (Predictive-coded) ピクチャと、前後の I または P フレームとの間で両方向予測符号化を行なう B (Bidirectionally predictive-coded) ピクチャによって構成されている。それぞれのピクチャは 16x16 画素の領域に分割され、これをさらに 4 つの輝度信号ブロック (8x8 画素) と 2 つの色差信号ブロック (8x8 画素) にわけ、6 つのブロックをまとめたものをマクロブロックと呼ぶ。それぞれのブロックは DCT (Discrete Cosine Transform) 処理され、マクロブロック単位に指定された量子化スケールと呼ばれるパラメータ値を用いて量子化される。

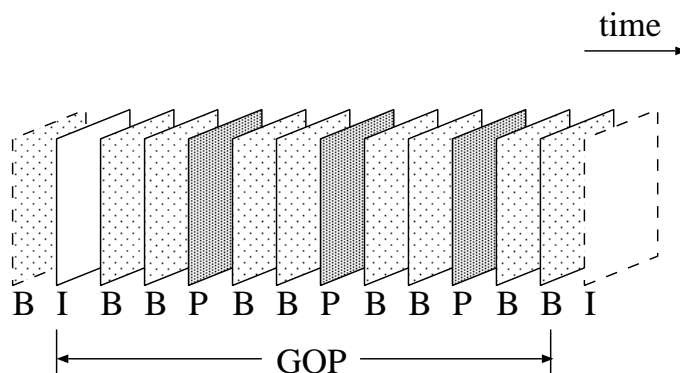


図 1: GoP 構成例

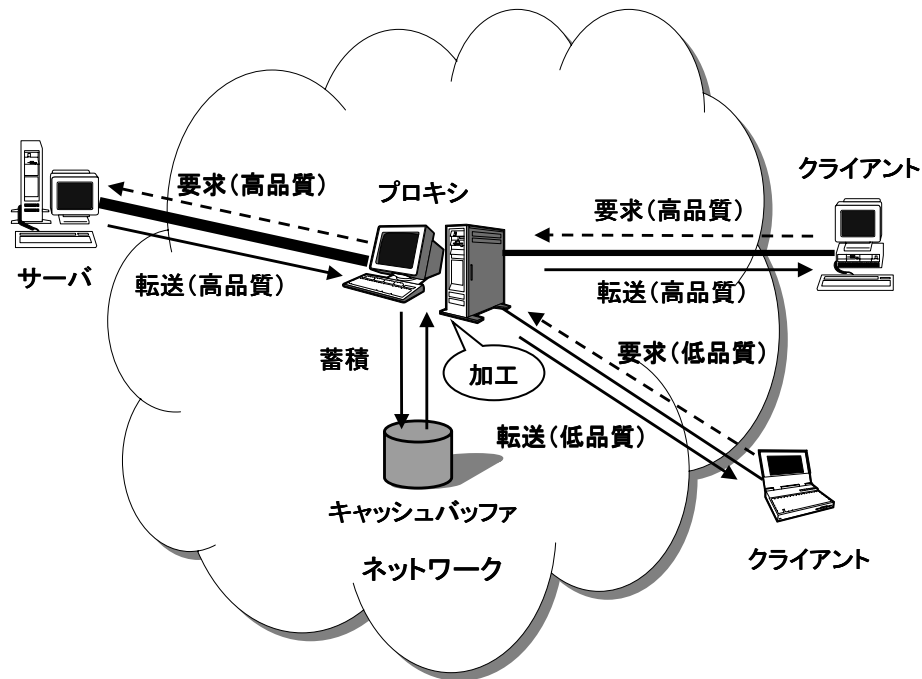


図 2: プロキシキャッシュを用いた MPEG-2 動画像ストリーミング配信システム

2.2 MPEG-2 動画像ストリーミング配信システムの概要

本報告では、図 2 に示すような一台の動画像サーバと複数のクライアントがプロキシを介して接続されたシステムでの動画像配信サービスを対象とする。クライアントはあらかじめ指定されたプロキシとの間に動画像通信セッションを確立することで動画像配信サービスの利用を開始し、プロキシに動画像データの転送を要求する。プロキシからクライアントに配信する動画像の品質は、クライアントシステムの性能や利用可能な帯域、あるいは再生動画像品質に対するユーザの好みなどによって決定されるが、本報告では比較的高性能なクライアントシステムと、動画像品質に対して寛容なユーザを仮定し、プロキシ-クライアント間で利用可能な帯域のみによってクライアントの要求品質が決定されるものとする。ただし、システム性能やユーザの好みは、提供する動画像品質の上限値と下限値としてそれぞれ与えることにより、本報告で提案するプロキシキャッシングメカニズムで取り扱うことができる。また、動画像配信サービスのためにサービス提供者やユーザがあらかじめ帯域を予約したり、動画像通信セッション開始時にネットワークが帯域を割り当てた場合には、プロキシ-クライアント間で利用可能な帯域はセッションを通じて固定、あるいは比較的長期にわたって安定するため、クライアントの要求品質を考慮した動画像ストリーミング配信システムの実現は容易である。

そこで本報告では、利用可能な帯域は下位層のレート制御によって決定されるデータ転送レートに従うものとする。マルチメディアデータ通信のためのレート制御手法としては MPEG-TFRC (TCP-Friendly Rate Control Protocol for MPEG video transfer) [3, 4] や TFRC (TCP Friendly Rate Control) [5] を用いる。これらは輻輳制御のないトランスポートプロトコルである UDP と、信頼性のあるデータ通信を提供する TCP との間の使用帯域に関する公平性を考慮したレート制御アルゴリズムであり、実時間マルチメディア通信に用いることでネットワークへのマルチメディアトラフィックの過度な流入を防ぐことができる。データ転送レートはパケット棄却の様子や RTT (Round Trip Time) の観測値にもとづいて決定されるため、ネットワークの負荷変動とともに上位アプリケーションの利用可能な帯域が大きく変化する。したがってクライアントに提供すべき動画データ品質も時間によって変動する。

プロキシは動画品質調整機能を有し、クライアントの要求品質を上回るキャッシュデータがあれば適切に加工しクライアントに転送する。なければプロキシ-クライアント間と同様にレート制御によって定められたサーバ-プロキシ間の利用可能な帯域やクライアントの要求品質などを考慮して、必要な動画データの品質を決定し、サーバに転送を要求する。サーバより取得した動画データはキャッシュに蓄積されると同時に必要に応じて加工された後、クライアントに転送される。次章以降では、このような動画ストリーミング配信システムを対象に高品質で効率のよい動画配信サービスを実現するための、動画品質調整機能を有するプロキシキャッシングのメカニズムやアルゴリズムについて検討する。

3 動画像品質調整機能を組み込んだプロキシキャッシングメカニズム

本章では、動画像品質調整機能を組み込んだプロキシキャッシングメカニズムの提案の概要とアルゴリズムを述べ、提案手法の理想特性を評価するため無限大のキャッシュバッファを有するプロキシを対象としたシミュレーションを行う。ただし、評価に際してはユーザは必ず動画像ストリームの先頭から最後まで順に動画像データを要求するものとし、すべてのクライアントが同一の動画像ストリームを鑑賞するものとする。

3.1 メカニズムの概要

提案メカニズムでは MPEG-2 動画像ストリームのデータ構造やキャッシュデータの再利用性などを考慮して GoP を単位として要求、転送、蓄積、加工を行なうものとする。クライアントは定期的に GoP 単位でデータ転送を要求する。要求の間隔は GoP あたりピクチャ数をフレームレートで除算することで求められる 1GoP 時間とする (図 3)。

プロキシは、クライアントからのデータ転送要求にもとづいて、レート制御アルゴリズムの定めるプロキシクライアント間の利用可能な帯域から、クライアントに転送する動画像データの品質を決定する。ただし、レート制御アルゴリズムの制御間隔はおよそ往復伝播遅延より定められネットワーク環境や負荷変動によって大きく異なるため、動画像データの転送要求の到着間隔とは一致しない。そこで、プロキシやサーバへのデータ転送要求到着時の TFRC のデータ転送レートを動画像通信に利用可能な帯域とみなす (図 4)。また、動画像品質は MPEG-2 の符号化パラメータである量子化スケールによって表されるものとし、図 5 に示すような量子化スケールと平均レートの関係から容易に決定可能であるものとする [6]。図 5 に示されるとおり量子化スケールが小さいほどデータサイズが大きく、品質が高い。

プロキシは扱う動画像ストリームごとに表 1 に示すようなキャッシュ内に蓄積された GoP のデータサイズと品質 (量子化スケール) の対応表を保持しており、これと先ほど決定された量子化スケール、および要求された GoP の番号とを比較する。キャッシュデータの量子化スケールがクライアントの要求品質を下回る場合はキャッシュヒット、上回る場合はキャッシュミスとなる。ただし、キャッシュにない GoP の量子化スケールは ∞ とする。キャッシュヒットの場合には、再量子化フィルタ [1] などを用いてキャッシュデータをクライアントの要求品質に合わせて加工し、クライアントに転送する。したがって、図 6(a) に示すとおり、キャッシュヒット時のデータ要求から受信完了までのデータ転送遅延はプロキシクライアント間の往復伝搬遅延 RTT_{pc} とデータ転送時間の和となる。ただしプロキシでの動画像品質調整などによる処理遅延はデータ転送時間や伝搬遅延と比較して非常に小さいものとし、ここでは考慮しない。

一方、キャッシュミスの場合には、すでにそのクライアントに対するサーバからの動画データ取得のためにサーバ-プロキシ間に確立されたセッションを用い、なければ新しくセッションを確立して、適切な品質の動画データをサーバから取得する。サーバから取得する動画データの品質は、クライアントの要求品質やキャッシュデータの再利用性、キャッシュバッファの容量などを考慮して決定するが、詳細なアルゴリズムについては次節で検討する。サーバから取得した動画データはそのまま蓄積し、クライアントの要求品質に合わせて加工、転送される。したがって、データ転送遅延はサーバ-プロキシ間とプロキシ-クライアント間往復伝播遅延の和 $RTT_{pc} + RTT_{sp}$ と、サーバ-プロキシ間とプロキシ-クライアント間のデータ転送時間のより大きい方を足しあわせた時間によって与えられる。ただし、プロキシからクライアントへのデータ転送順は GoP 番号にしたがうものとしており、どちらの場合についても以前の GoP データの転送完了がしていない場合には待ち合わせが生じる (図 3)。

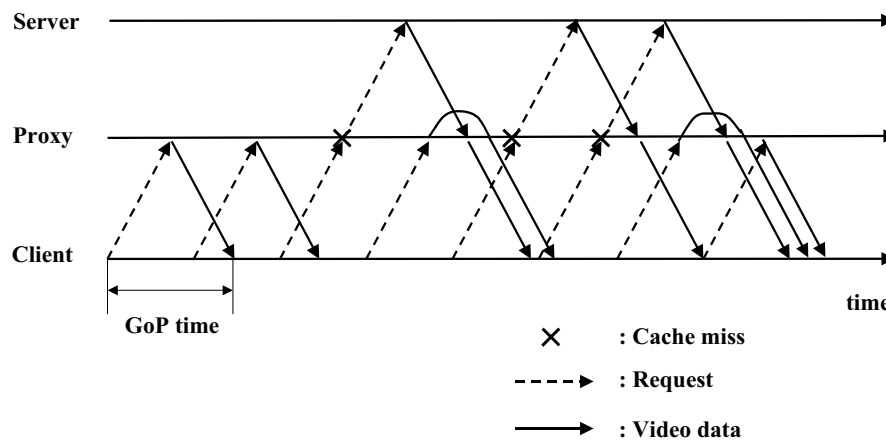


図 3: 動画データ転送処理例

GoP	サイズ	量子化スケール
1	a	A
2	b	C
3	0	∞
\vdots	\vdots	\vdots

表 1: キャッシュデータ管理リスト

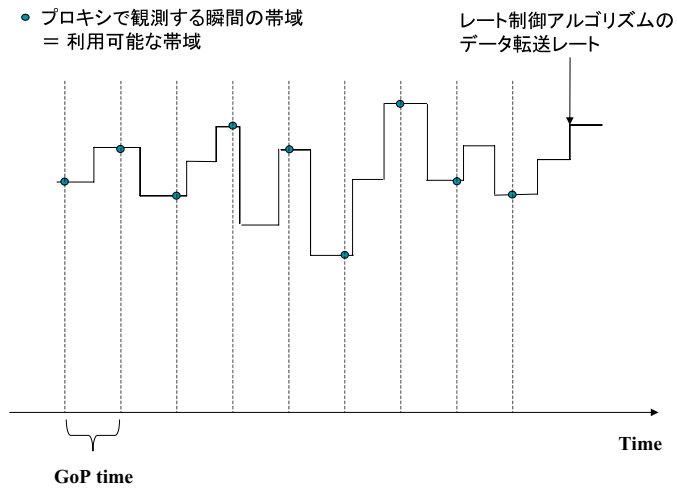


図 4: 利用可能な帯域データ転送レート

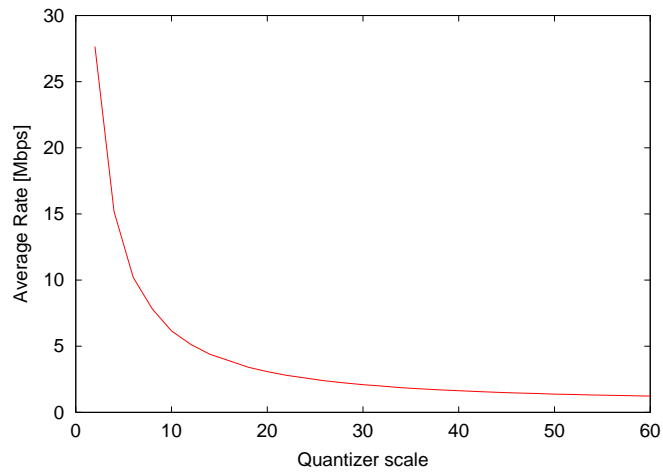


図 5: 量子化スケールと平均レートの関係

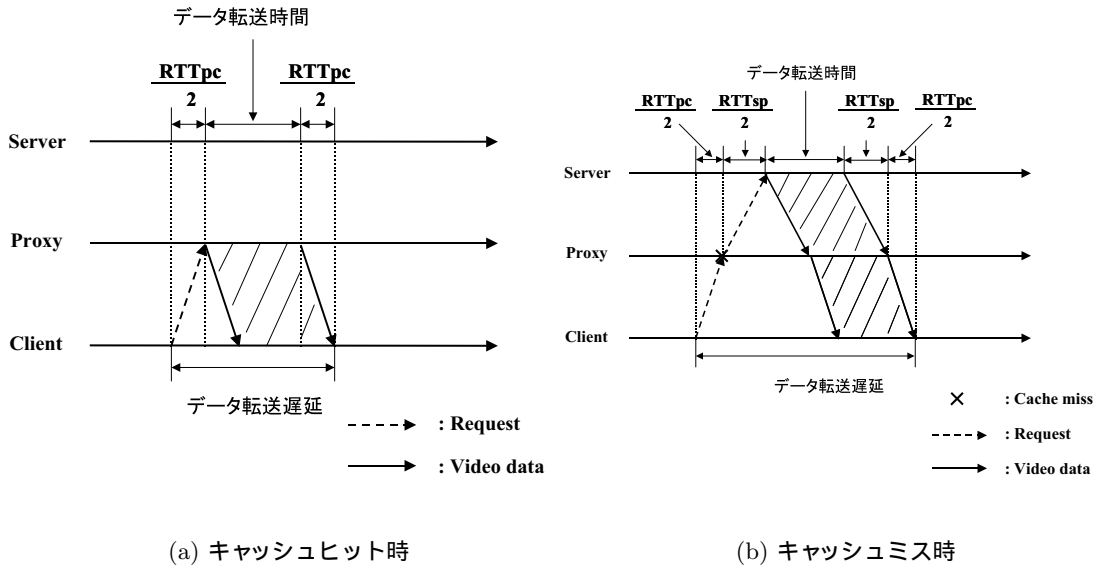


図 6: キャッシュヒットとキャッシュミス

3.2 クライアントの要求品質を考慮した動画データ取得

キャッシュミス発生時にプロキシがサーバに要求する動画データの品質は，取得したデータの再利用性や，キャッシュバッファの容量，サーバ-プロキシ間の帯域を考慮して決めなければならない．ただし，サーバからの動画データ取得に利用可能な帯域は，プロキシ-クライアント間の利用可能な帯域とは無関係に決定されるため，クライアントからの要求品質を満たすデータ取得に十分な帯域が利用可能であるとは限らない．

そこで，キャッシュデータやサーバから取得する動画データによりプロキシが提供可能な動画品質のクライアントの要求品質に対する比 α に基づいて制御を行う．

$$\alpha = \frac{\max(Q_{sp}(i, j), Q_{cache}(i, j))}{Q_{pc}(i, j)} \quad (1)$$

ここで， i はクライアント番号， j はクライアント i が要求している GoP の番号とする．またクライアント i に対してサーバ-プロキシ間に確立されたセッションで 1GoP 時間内に転送可能な GoP $_j$ の品質を $Q_{sp}(i, j)$ と表す． $Q_{sp}(i, j)$ はサーバ-プロキシ間で利用可能な帯域から決定される量子化スケールの逆数として与えられる．同様に，プロキシ-クライアント間で転送可能な最高の動画品質を $Q_{pc}(i, j)$ とし，これがクライアント i の GoP $_j$ に対する要求品質となる．また， $Q_{cache}(i, j)$ はキャッシュされた GoP $_j$ の品質を表す．キャッシュに該当する GoP データがない場合には量子化スケールは ∞ となるので，量子化スケールの逆

数として与えられる $Q_{cache}(i, j)$ は0となる．本節では，キャッシュデータが要求品質を満たせない場合 (キャッシュミス) について検討しているため， $Q_{cache}(i, j) < Q_{pc}(i, j)$ である．

プロキシの提供可能な動画データがクライアントの要求を満たせる場合 ($\alpha \geq 1$) にサーバに要求する動画品質の決定には，データの再利用性，キャッシュバッファ容量を考慮したいくつかの手法が考えられる．

後から同じ動画ストリームを見始めたユーザに対するデータ転送遅延を抑えるためには，できる限り高品質な動画データを取得，蓄積しておき，キャッシュミスの発生率を低く抑えるのがよいと考えられる．そこで，サーバ-プロキシ間の帯域を利用して転送することのできる最も高品質な動画データを要求する．この場合，サーバに要求する動画の品質 $Q_{req}(i, j)$ は，以下のようにして与えられる．

$$Q_{req}(i, j) = Q_{sp}(i, j) \quad (2)$$

しかしながら，この手法では，プロキシ-クライアント間に比べてサーバ-プロキシ間の利用可能な帯域が極端に広い場合には，不必要に高品質な動画データを取得，蓄積してしまうため，サーバ-プロキシ間の帯域やキャッシュバッファの有効利用が図れない．そこで， GoP_j に対して要求される品質を予測してサーバへの要求品質を決定する手法を提案する．クライアント i 自身とより過去の GoP を参照しているクライアントの集合を S とすると $Q_{req}(i, j)$ は以下の式で与えられる．

$$Q_{req}(i, j) = \min\left(\max_{k \in S, 0 \leq l \leq j} Q_{pc}(k, l), Q_{sp}(i, j)\right) \quad (3)$$

式 (2) および式 (3) にもとづいて要求した動画の品質がクライアントの要求品質を上回る場合には，取得データをキャッシュに蓄積するとともに適切に品質を調整してクライアントに転送する．

また，キャッシュバッファの容量が非常に小さい場合には，より多くの GoP データを蓄積しておくことによりキャッシュデータの再利用性を高めることができると考えられるため，クライアント i からの要求品質どおりの動画データをサーバに要求する．

$$Q_{req}(i, j) = Q_{pc}(i, j) \quad (4)$$

一方，プロキシの提供可能な動画データがクライアントの要求品質を満たせない場合 ($\alpha < 1$) には，転送遅延が増大するが要求品質を満たす動画データを提供する手法と，要求品質を満たせないが $1GoP$ 時間内で転送可能な動画データを提供する手法が考えられる．そこで，ユーザの動画配信サービスに対する要求を表す指標として提供される動画品質の要求品質に対する比 β を導入する． β の値が0に近いほどユーザが高速な実時間性の

高いサービスを望んでいることを，1に近いほど利用可能な帯域の範囲内でできるだけ高品質な動画像を要求することをあらわす．

まず，キャッシュデータでクライアントの要求品質をほぼ満たせる場合 ($\beta \leq \frac{Q_{cache}(i,j)}{Q_{pc}(i,j)} \leq \alpha$) にはキャッシュデータを送り，サーバへの動画像データ転送要求は行わない．キャッシュデータの品質は低い帯域が比較的に広い場合 ($\frac{Q_{cache}(i,j)}{Q_{pc}(i,j)} < \beta \leq \frac{Q_{sp}(i,j)}{Q_{pc}(i,j)} = 1$) には，1GoP 時間内にサーバからプロキシに転送可能な最高品質の動画像データをサーバに要求する．

$$Q_{req}(i, j) = Q_{sp}(i, j) \quad (5)$$

また，プロキシの提供可能な動画像品質がクライアントの要求品質を十分満たせない場合 ($\alpha < \beta$) には，クライアントの要求を最低限満たせるだけの品質をサーバに要求する．ただし，この場合には動画像データ転送遅延が大きくなる．

$$Q_{req}(i, j) = \beta \cdot Q_{pc}(i, j) \quad (6)$$

3.3 性能評価

シミュレーションによりクライアントごとの動画像再生開始までの待ち時間，受信動画像品質に対する満足度，サーバ-プロキシ間の帯域利用率，およびキャッシュ内データ量の変化について提案手法の比較評価を行う．評価にはキャッシュミス時にサーバに要求する動画像品質の決定手法によって分類される3つのアルゴリズムを用いる(表2)．また，動画像品質調整機能を組み込んだプロキシキャッシュの有効性を評価するため，従来の品質調整のできないプロキシキャッシュとの比較を合わせて行う．シミュレーションの条件を表3に示す．

プロキシに対するクライアントからの動画像データの転送要求はGoP時間ごとに定期的に送信されるが，キャッシュヒットやミス，利用可能な帯域によって動画像データの到着間隔は大きく変動するため，受信データをそのまま再生するとピクチャの表示間隔に揺らぎが生じる(図7)．そのため，動画像ストリームの先頭から最後までデータ受信の待ち合わせなく滑らかに動画像を再生するためには，セッション開始時にある程度の受信データを蓄積してから再生を始めることにより遅延の揺らぎを吸収することが必要となる．クライアント*i*の再生開始までの待ち時間 $W(i)$ は，動画像全体を通した転送遅れの最大値で表される(図8)．

$$W(i) = \max_{1 \leq j \leq GoP_{end}} (T(i, j) - I(i, j)) \quad (7)$$

ただし， j はGoP番号， GoP_{end} は動画像ストリームの最後のGoP番号を表す．また， $T(i, j)$ はGoP $_j$ の到着時刻，および $I(i, j)$ はGoP $_j$ の理想的な到着時刻を表しており， $I(i, j) - I(i, j - 1) = 1GoP$ 時間， $I(i, 1) = T(i, 1)$ である．

	$\alpha \geq 1$	$\beta \leq \alpha < 1$	$\alpha < \beta$
アルゴリズム 1	式 (2)	式 (5)	式 (6)
アルゴリズム 2	式 (3)		
アルゴリズム 3	式 (4)		

表 2: サーバへ要求する動画像品質の決定アルゴリズム

シミュレーション時間	29000 秒
クライアントあたりの動画像データ再生時間	7200 秒
対象とする動画像データ	1 本
動画像データサイズ	8.6 ~ 194.5 Gbit
クライアント数	10
クライアントの到着間隔	平均 1800 秒の指数分布
サーバ-プロキシ間の往復伝播遅延	200 ミリ秒
プロキシ-クライアント間の往復伝播遅延	50 ミリ秒
キャッシュバッファサイズ	無限

表 3: シミュレーション条件

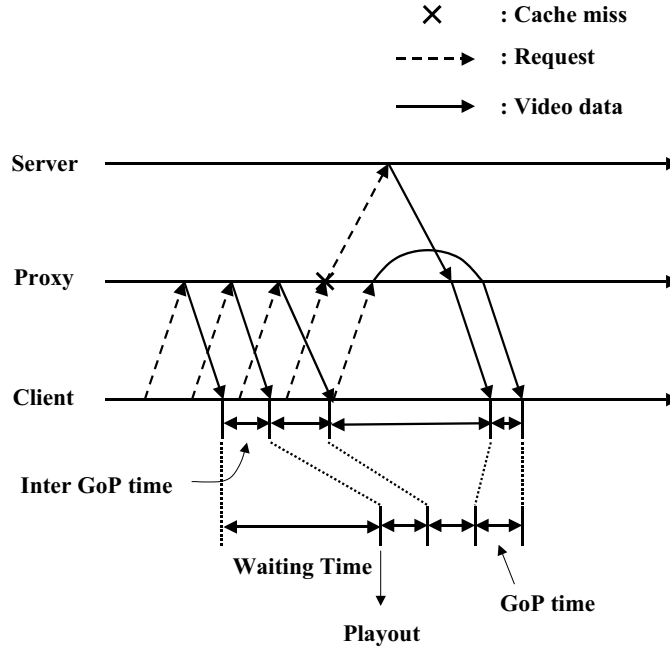


図 7: 表示間隔の揺らぎ

次に，クライアント i の受信動画品質に対する満足度 $S(i)$ を次式のように定義する．

$$S(i) = \frac{1}{GoP_{end}} \sum_{j=1}^{GoP_{end}} \frac{Q_{act}(i, j)}{Q_{pc}(i, j)} \quad (8)$$

ただし， $Q_{act}(i, j)$ はクライアント i に提供された GoP $_j$ の品質である．

キャッシュデータの再利用性とサーバ-プロキシ間のネットワークの負荷軽減の度合いを評価するため，サーバ-プロキシ間の帯域利用率 $U(i)$ を以下のように定義する．

$$U(i) = \frac{\text{クライアント } i \text{ に対するサーバ-プロキシ間の帯域利用時間}}{\text{クライアント } i \text{ の参加時間}} \quad (9)$$

3.3.1 サーバ-プロキシ間で利用可能な帯域が大きい場合

まず，図 9 に示すとおりプロキシ-クライアント間に比べてサーバ-プロキシ間で利用可能な帯域が大きく，サーバからのデータ転送が高速な場合について評価する．クライアントは動画通信サービスの利用開始順に番号が割りふられている (図 10)．なお，図 9 のデータ転送レートに関するデータはネットワークシミュレータ ns-2 [7] を用いたシミュレーションにより作成した．

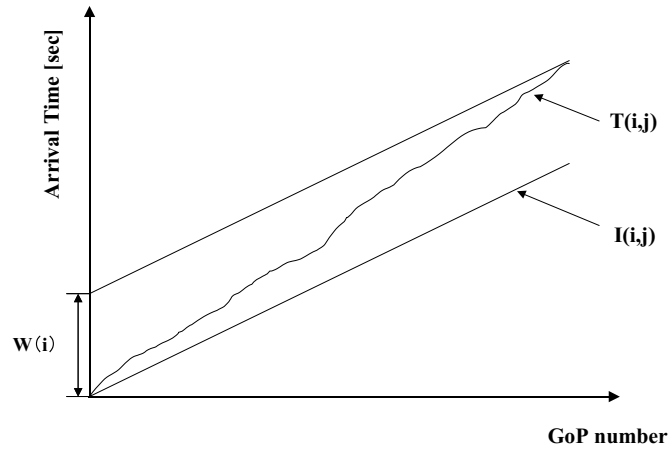
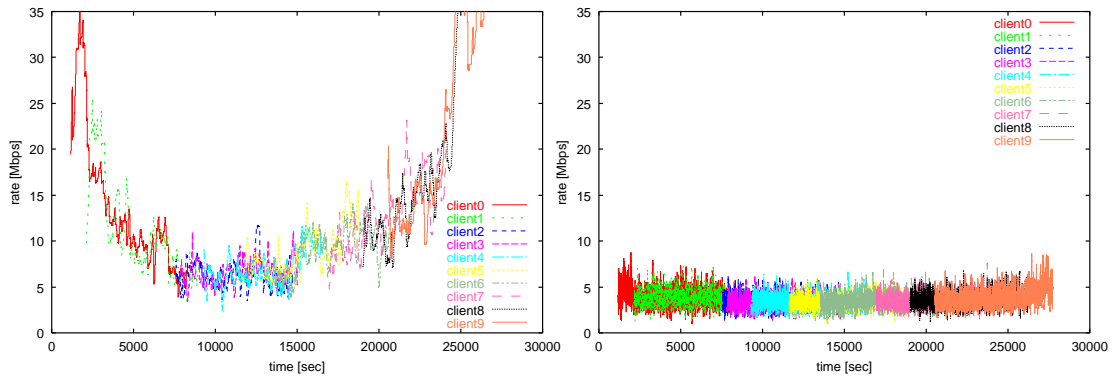


図 8: 再生までの待ち時間



(a) サーバ-プロキシ間

(b) プロキシ-クライアント間

図 9: 利用可能な帯域の変化 (サーバ-プロキシ間が広帯域な場合)

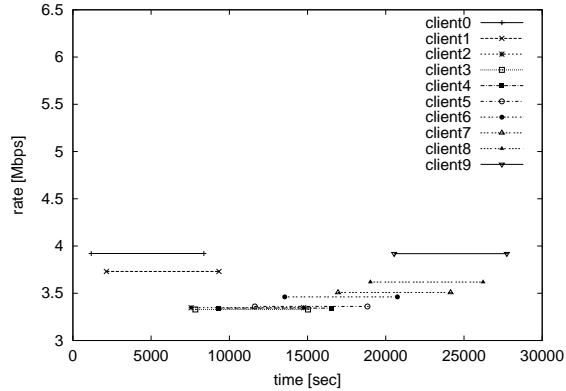


図 10: クライアントの参加時間と平均レート (サーバ-プロキシ間が広帯域な場合)

動画像品質調整機能を持たないプロキシキャッシング (以降では従来手法と呼ぶ) と提案アルゴリズムにおける再生開始までの待ち時間 $W(i)$ に関するシミュレーション結果を図 11 に示す．なお，図中では従来手法を “traditional” と表記する．また，クライアントの動画像配信サービスに対する要求 β の影響についてもあわせて検討するため，常に要求品質どおりの動画像データを求める $\beta = 1$ の場合と，ある程度の品質の劣化を許すことで高速なデータ転送を期待する $\beta = 0.6$ の場合について評価した．ただし，従来手法では要求品質どおりのキャッシュデータがある場合にのみキャッシュヒットとなり， β は 1 である．

図より， β や手法によらず 7 番目以降のクライアントについては，提案手法による待ち時間の改善効果があまりないことがわかる．サーバへの要求動画像品質の決定アルゴリズムによる性能差は，キャッシュミスの発生率とキャッシュミス時のデータ転送遅延による待ち時間の増加量となってあらわれる．しかしながら，本評価で用いた帯域データは，サーバ-プロキシ間の利用可能な帯域がプロキシ-クライアント間と比較して広いため，たとえキャッシュミスによるサーバからのデータ取得が頻繁に発生してもそれほど大きな遅延は生じず，むしろ比較的狭帯域なプロキシ-クライアント間の動画像データ転送時間が待ち時間 $W(i)$ に与える影響が大きい．画質に対する満足度を示す図 12 から明らかとなっており，プロキシ-クライアント間を転送される動画像データは $\beta = 1$ では要求どおり， $\beta = 0.6$ でもほぼ要求品質に等しいためアルゴリズム間のデータ転送時間も変わらず，結果として待ち時間 $W(i)$ に差が生じない．

一方，クライアント 1, 2 および 4 に対しては，比較的高品質な動画像データを取得，蓄積するアルゴリズム 1, 2 を用いる効果が高い．ただし，キャッシュ内にデータがなくすべての動画像データをサーバから取得する必要のあるクライアント 0 についてはアルゴリズム 1, 2 を用いることで待ち時間が増大する．提案したプロキシキャッシングメカニズムでは，

図 4 に示したとおりデータ転送中にアルゴリズムの想定する利用可能な帯域とレート制御の定めるデータ転送レートとの差が生じるため、データ転送時間が 1GoP 時間を超過する可能性がある。その結果、データ転送時間から導出されるサーバ-プロキシ間の帯域利用率も増加し (図 13)、待ち時間が大きくなる。

例えばレート制御結果を常に監視し、データ転送レートに変化が生じた場合には転送中の動画の品質を動的に再調整すれば 1GoP 時間内でデータ転送が完了し遅延の増加を抑えることができると考えられる。しかしながら、例えば本シミュレーション条件では GoP 時間が 1 秒であるのに対し、プロキシ-クライアント間の往復伝播遅延は 50 ミリ秒と短いため、データ転送中に平均 20 回の品質調整を行わなければならない。実際にはそのような品質調整は困難であるため、本報告ではプロキシキャッシングアルゴリズムを改良することにより遅延の低減をはかる。キャッシュデータの品質が低いアルゴリズム 3 では、サービスに対するユーザの要求を表す指標 β を下げることにより、一部のクライアントで待ち時間が改善されたが、帯域利用率 (図 13) やキャッシュ内データ量 (図 14) にほとんど変化はみられない。

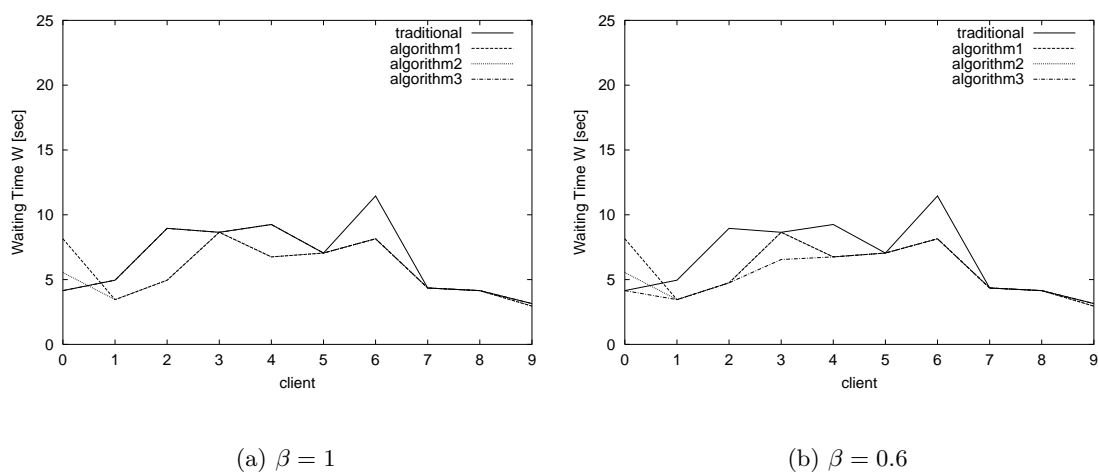
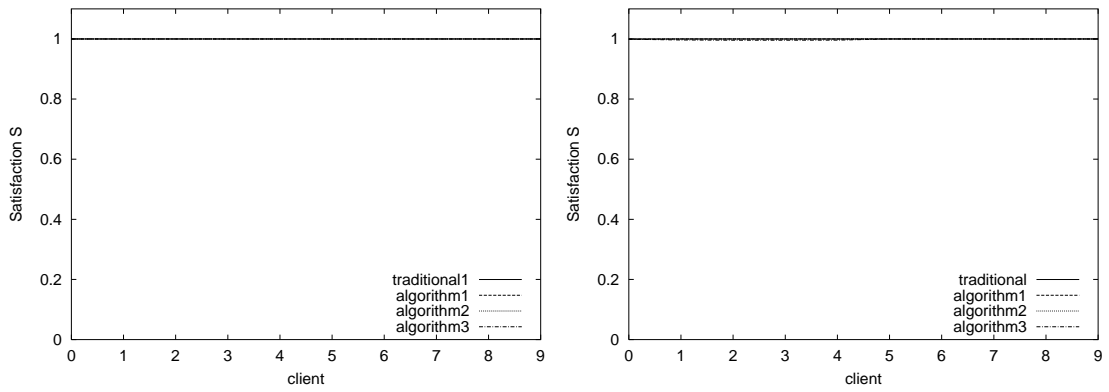


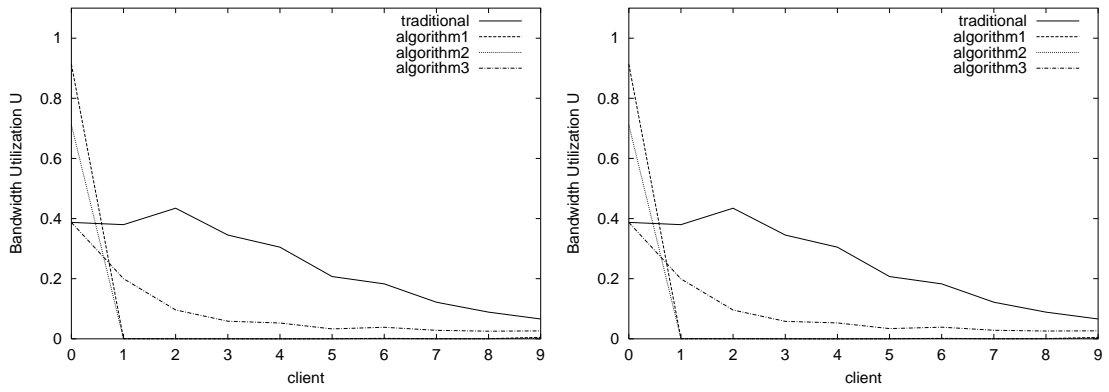
図 11: 再生開始までの待ち時間 (サーバ-プロキシ間が広帯域な場合)



(a) $\beta = 1$

(b) $\beta = 0.6$

図 12: 受信動画像品質に対する満足度 (サーバ-プロキシ間が広帯域な場合)



(a) $\beta = 1$

(b) $\beta = 0.6$

図 13: サーバ-プロキシ間の帯域利用率 (サーバ-プロキシ間が広帯域な場合)

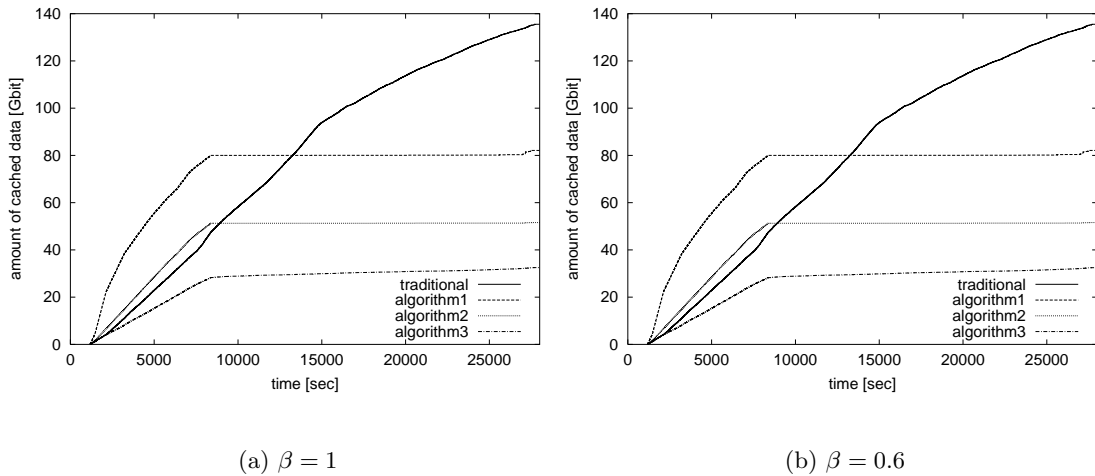
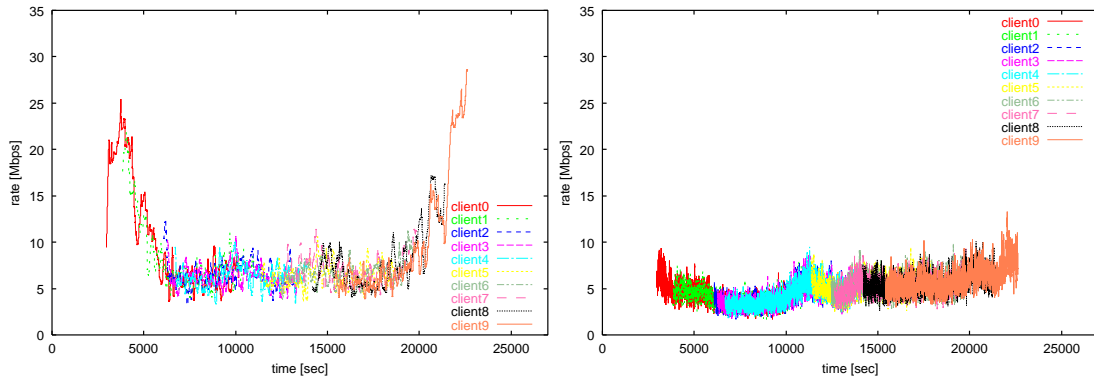


図 14: キャッシュ内のデータ量の変化 (サーバ-プロキシ間が広帯域な場合)

3.3.2 サーバ-プロキシ間で利用可能な帯域が小さい場合

次に，図 15 に示すようにプロキシ-クライアント間に比べてサーバ-プロキシ間で利用可能な帯域がそれほど小さくなく，サーバ-プロキシ間のデータ転送時間が待ち時間に大きな影響を与える場合に関する評価を行う。

再生開始までの待ち時間 $W(i)$ に関するシミュレーション結果を図 17 に示す．図より，従来手法ではサーバ-プロキシ間，プロキシ-クライアント間の利用可能な帯域がそれほど変わらないクライアント 5 の待ち時間が増大するのに対し，特にアルゴリズム 1 と 2 では他のクライアントと比較して性能が劣化していない．これはサーバ-プロキシ間の帯域利用率を問わず図 19 から明らかとなっており，サーバ-プロキシ間の帯域が比較的広いクライアント 0 に対して高品質な動画データを取得，蓄積しており，以降のクライアントに対するデータ再取得があまり発生しないためである．一方，キャッシュミス時にクライアントの要求品質どおりの動画をサーバから取得するアルゴリズム 3 では，先行するクライアントよりも高い品質の動画データを要求するクライアント 5 でのサーバ-プロキシ間のデータ転送が発生し，他のアルゴリズムと比較して待ち時間が大きくなる．また，受信動画品質の低下をある程度許容できる場合には， β を小さくすることによりアルゴリズムによらず再生開始までの待ち時間を大幅に小さくできるが，図 18 に示すとおり受信動画品質は若干低くなる．ただし， β を小さくしてもキャッシュデータ量はそれほど変わらない (図 20) ．



(a) サーバ-プロキシ間

(b) プロキシ-クライアント間

図 15: 利用可能な帯域の変化 (サーバ-プロキシ間が狭帯域な場合)

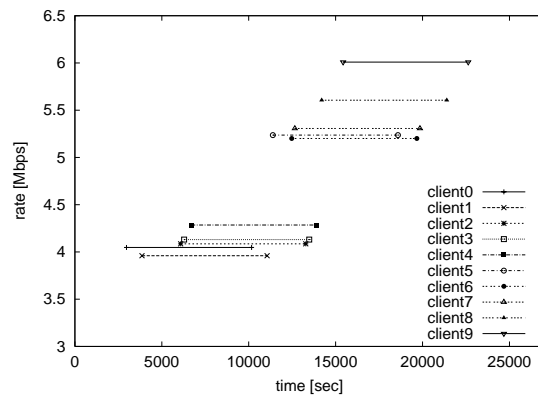
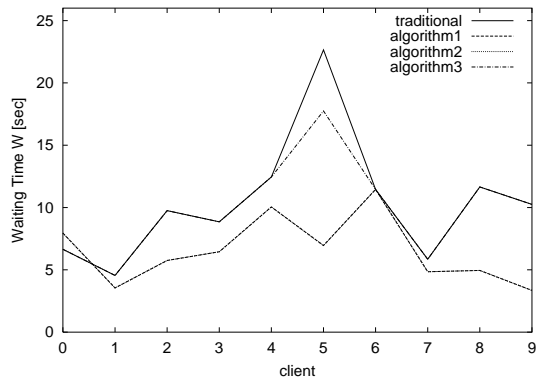
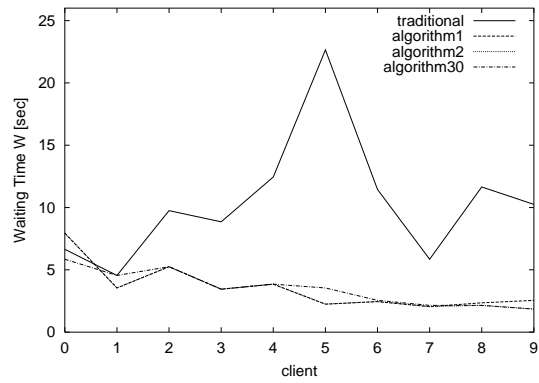


図 16: クライアントの参加時間 (サーバ-プロキシ間が狭帯域な場合)

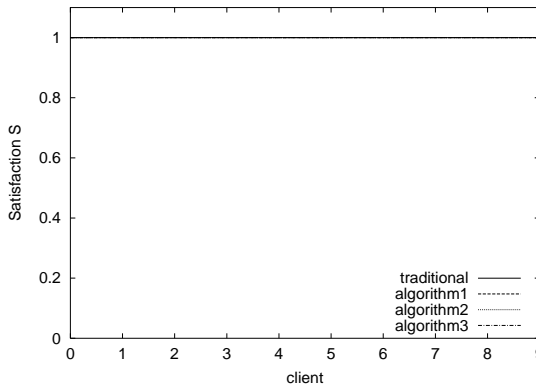


(a) $\beta = 1$

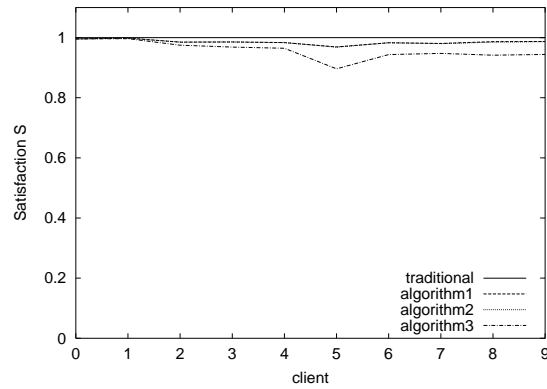


(b) $\beta = 0.6$

図 17: 再生開始までの待ち時間 (サーバ-プロキシ間が狭帯域な場合)



(a) $\beta = 1$



(b) $\beta = 0.6$

図 18: 受信動画品質に対する満足度 (サーバ-プロキシ間が狭帯域な場合)

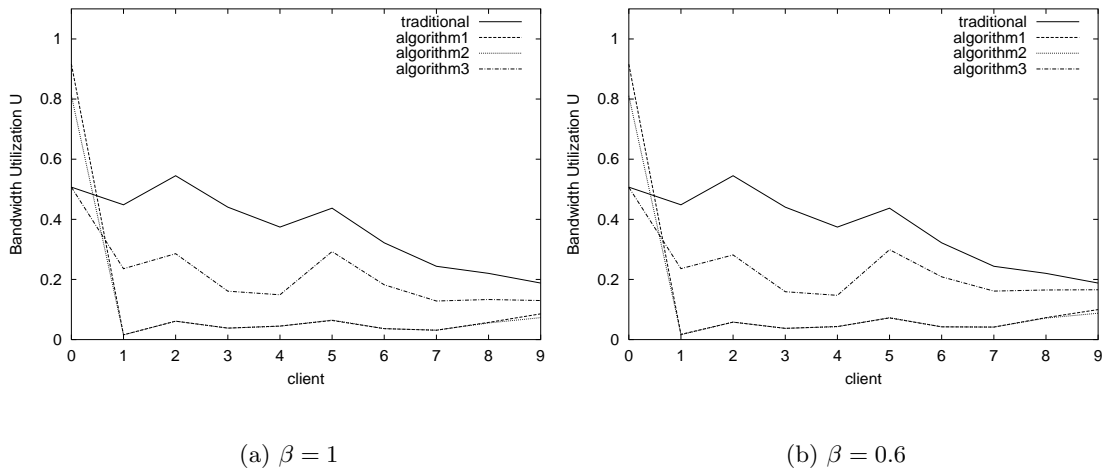


図 19: サーバ-プロキシ間の帯域利用率 (サーバ-プロキシ間が狭帯域な場合)

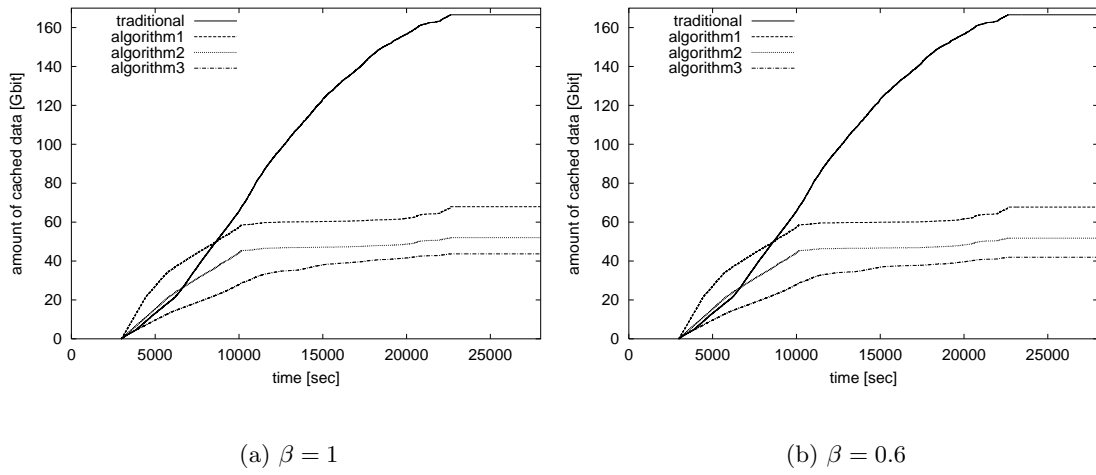


図 20: キャッシュ内のデータ量の変化 (サーバ-プロキシ間が狭帯域な場合)

4 プロキシにおける動画像データ先読みメカニズム

3章で述べたキャッシングメカニズムでは、クライアントからの動画像データの転送要求がプロキシに届いた時にキャッシュデータの検索を行なうため、キャッシュミス発生時にはサーバへのデータ転送要求、およびそれに続く実際のデータ転送といった遅延が生じる。そこで、クライアントが近い将来必要とする動画像データをあらかじめキャッシュに準備しておく先読みメカニズムを導入する。先読みメカニズムを有するプロキシは、クライアントからの動画像データ転送要求を受信すると、要求された GoP と、続くいくつかの GoP のためのキャッシュ検索を行なう。後続の GoP についてキャッシュ内に十分な品質の動画像データがない場合には、適切な品質の動画像データをあらかじめサーバから取得、蓄積することにより、将来キャッシュミスが発生するのを防ぐ。

4.1 動画像先読みのためのキャッシュ検索アルゴリズム

あらかじめキャッシュ内に動画像データを準備しておくことにより、キャッシュミスを防ぐためには、キャッシュ内に適切な品質の動画像データがすでに蓄積されているかどうかを調べ、必要に応じてサーバから動画像データを取得しなければならない。ただし、無駄な先読みをなくしてサーバ-プロキシ間の帯域の有効利用をはかるためには、取得しようとする GoP データに対するデータ転送要求がすでに他のクライアントによってなされていないか調べる必要がある。そこで表 1 に示したキャッシュデータの管理リストを表 4 のように拡張する。取得中の量子化スケールの欄にはサーバから受信中の GoP データの品質が書き込まれ、データ受信完了と同時にキャッシュデータの量子化スケールを表す欄へ上書きされ、 ∞ に書き換えられる。クライアント i が要求している GoP の番号を j 、先読み検索の対象とする GoP 数を P とすると、 GoP_k ($j+1 \leq k \leq j+P$) についてキャッシュデータ品質 $Q_{\text{cache}}(i, k)$ と受信中のデータ品質 $Q_{\text{rec}}(i, k)$ のいずれもキャッシュミスを起こさない最低限度の動画像品質 $\beta \cdot Q_{pc}(i, j)$ を下回る最も手前の GoP についてサーバに転送を要求する。

GoP	サイズ	量子化スケール	受信中の量子化スケール
1	a	A	B
2	b	C	∞
3	0	∞	D
\vdots	\vdots	\vdots	\vdots

表 4: 先読みメカニズムのためのキャッシュデータ管理リスト

4.2 先読みする動画データ品質

先読みによりサーバに要求する動画データの品質は、キャッシュミス時の動画品質決定アルゴリズムにしたがう。サーバ-プロキシ間にクライアント i のためのセッションがあれば新たにセッションを確立して先読みを用いるが、すでにセッションがある場合にはキャッシュミスによる GoP_j のためのデータ取得に用いられない帯域を利用して先読みを行なう。したがって、サーバ-プロキシ間で利用可能な帯域のすべてを用いてデータ取得を行なうアルゴリズム 1 ではキャッシュミスが発生すると先読みができない。

4.3 サーバにおける先読み要求処理メカニズム

サーバ-プロキシ間で設定されたセッションを用いて効率よく動画データを転送するため、サーバでは図 21 に示すような動画データ転送要求の優先権付き処理を行なう。キャッシュミスによる転送要求と、先読みによる転送要求を優先権の異なる待ち行列で管理し、先読みが重要性、緊急性の高い動画データ転送に影響を与えないようにする。先読みのためのキャッシュ検索はクライアントからのデータ転送要求ごとに行なわれるため、データ受信が開始されない限り、同じ GoP に対する先読みが繰り返し要求される可能性がある。そこで、先読み用の待ち行列のサイズを 1 とし、常に新しい先読み要求によって上書きされるようにすることで、無駄な先読みを行なわないようにする。

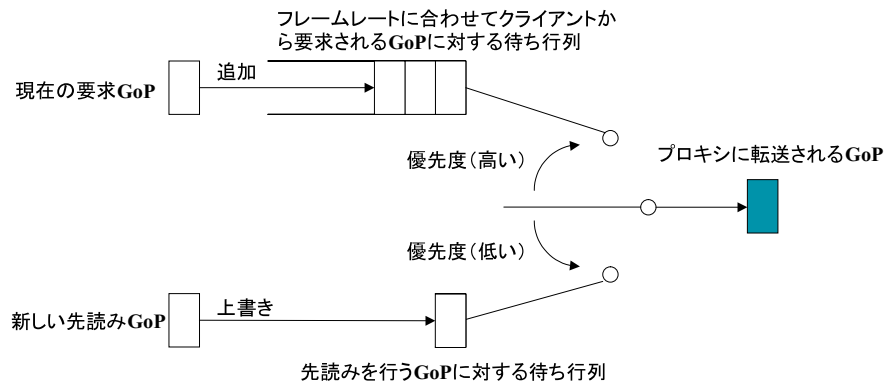


図 21: サーバにおける先読み要求処理メカニズム

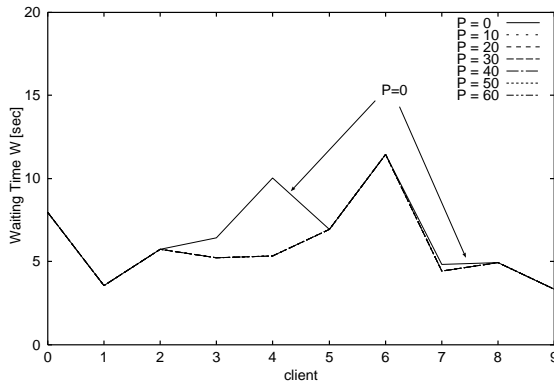
4.4 性能評価

クライアントごとの再生開始までの待ち時間 $W(i)$ ，受信動画像品質に対する満足度 $S(i)$ ，サーバ-プロキシ間の帯域利用率 $U(i)$ ，およびキャッシュ内データ量の変化にもとづいて，先読みメカニズムの性能を比較評価する．ただし $\beta = 1$ の場合の満足度はすべて等しいためここでは検討しない．また，クライアントの要求 GoP と先読み GoP で共通の動画像品質決定アルゴリズムを用いる．

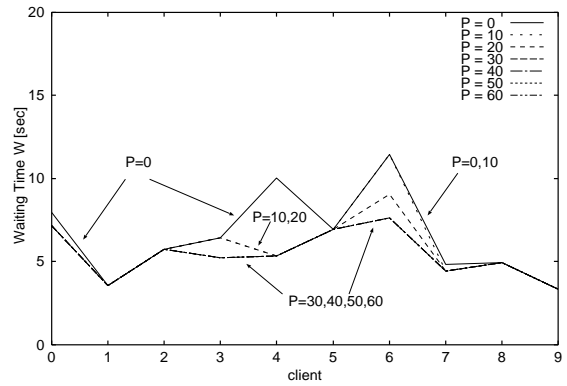
ただし，3.3 節で示されたとおり，サーバ-プロキシ間で利用可能な帯域が比較的大きい場合には，プロキシ-クライアント間のデータ転送時間に再生までの待ち時間が大きく影響されるため，動画像品質調整機能を組み込むことによる効果はそれほど高くない．そこで，以降の評価ではサーバ-プロキシ間の帯域が小さい場合 (図 15) についてのみ評価を行なうものとする．

先読み範囲 P を 0 (先読みを行わない) から 60 GoP まで変化させた場合のミュレーション結果を図 22 ~ 28 に示す．図 22 より， $\beta = 1$ の場合にはどのアルゴリズムについても先読みにより再生開始までの待ち時間を減らせ，アルゴリズム 2 の方がアルゴリズム 1 よりも先読みの効果が高い．アルゴリズム 3 ではキャッシュミスによる動画像データ取得の際にもサーバ-プロキシ間の帯域に空きが生じるため，図 23 の帯域利用率が高いことから明らかのようにより多くの先読みがなされているが，キャッシュデータ品質がそれほど高くないため待ち時間があまり改善されない．

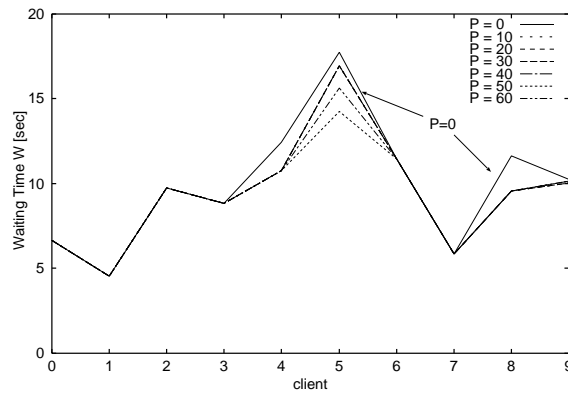
一方，図 25 ~ 27 に示されるとおりいずれのアルゴリズム，指標においても $\beta = 0.6$ の場合には先読みの効果は低い．また，図 27 に示される帯域利用率もあまり増加していないことから先読みによるデータ取得もあまりない．ただし，アルゴリズム 2 では先読みによるクライアント 0 の待ち時間の増減が生じている．待ち時間減少は，プロキシ-クライアント間と比べてサーバ-プロキシ間の帯域が大きく，高品質な動画像データを先読み，蓄積できるためである．しかしながら，セッションの後半では利用可能な帯域が減少するため，キャッシュデータの品質が低下する．その結果，より高品質な動画像データを要求するクライアント 5 でキャッシュミスが多く発生し，データ転送により待ち時間が大きくなる．



(a) アルゴリズム 1

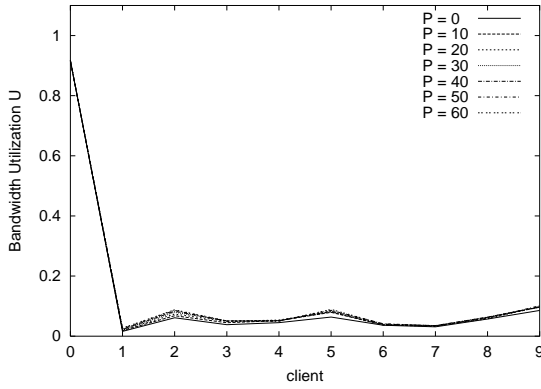


(b) アルゴリズム 2

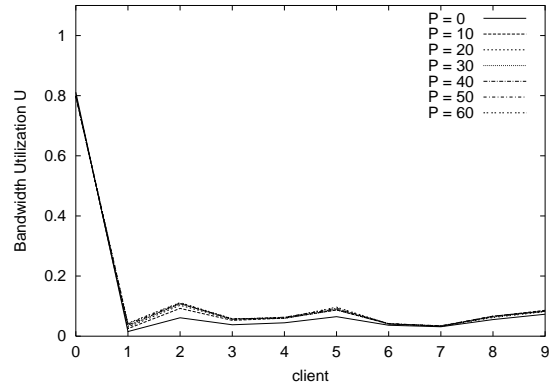


(c) アルゴリズム 3

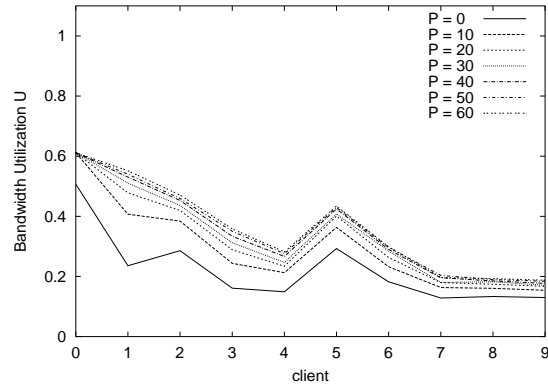
図 22: 再生開始までの待ち時間 (先読みあり, $\beta = 1$)



(a) アルゴリズム 1

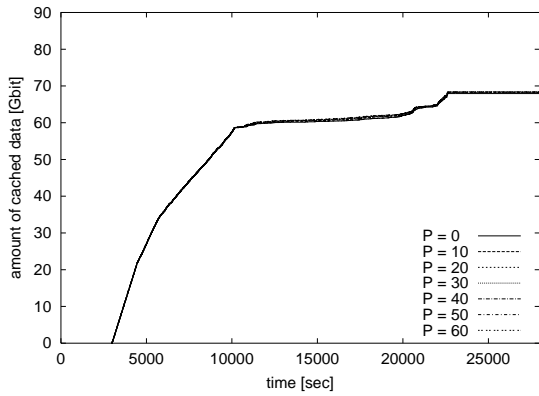


(b) アルゴリズム 2

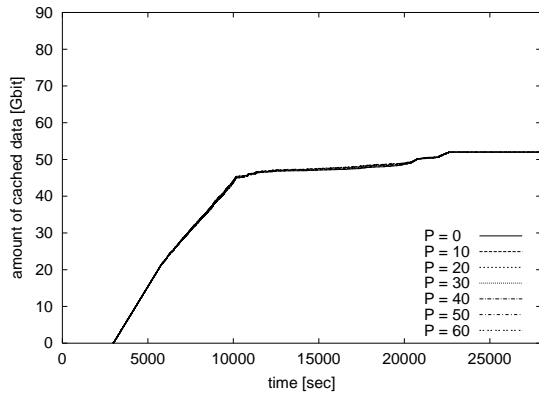


(c) アルゴリズム 3

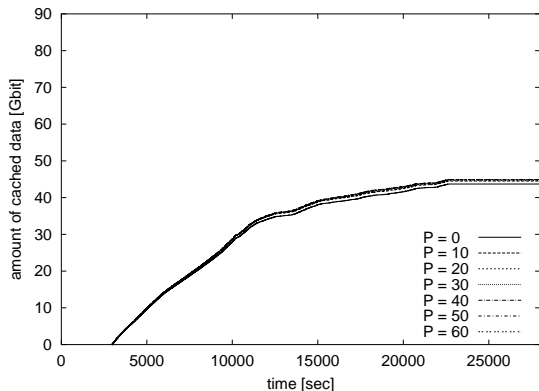
図 23: サーバ-プロキシ間の帯域利用率 (先読みあり, $\beta = 1$)



(a) アルゴリズム 1

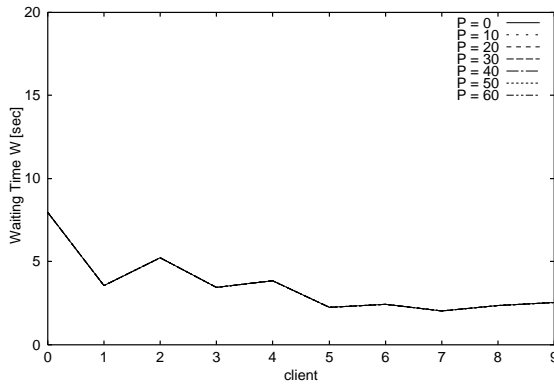


(b) アルゴリズム 2

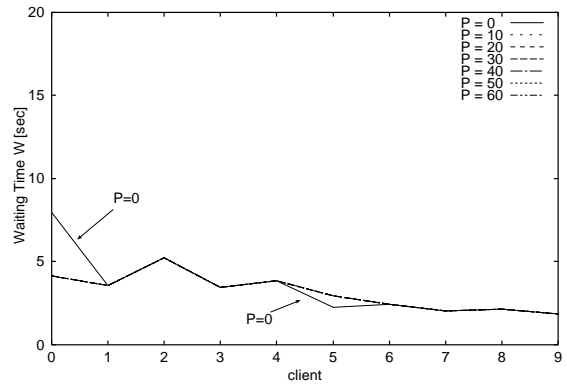


(c) アルゴリズム 3

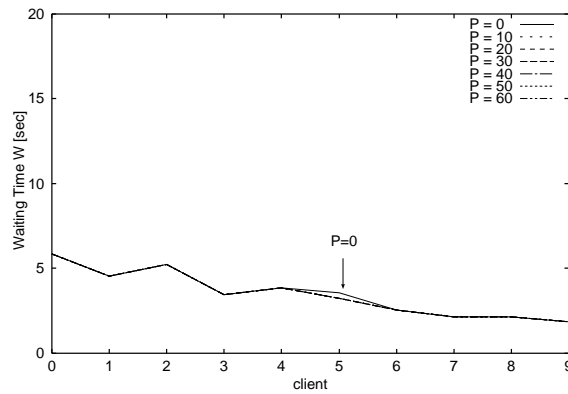
図 24: キャッシュ内のデータ量の変化 (先読みあり, $\beta = 1$)



(a) アルゴリズム 1

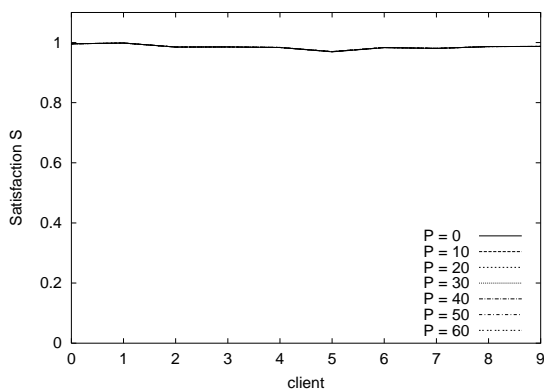


(b) アルゴリズム 2

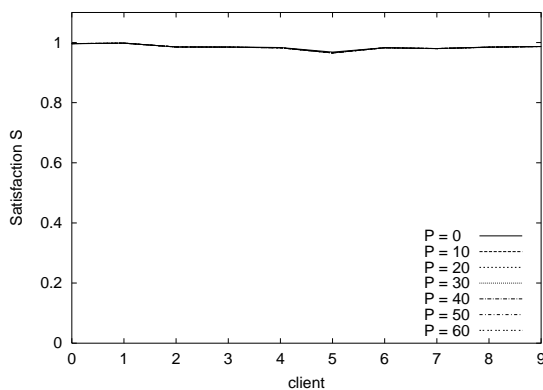


(c) アルゴリズム 3

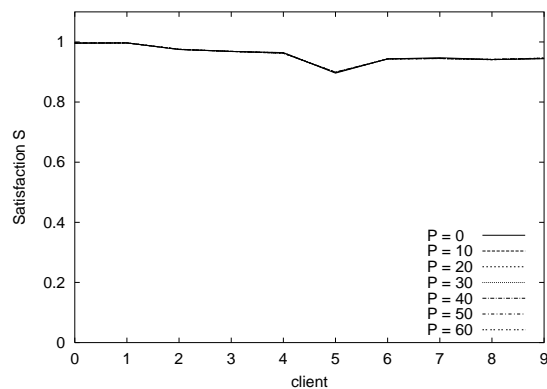
図 25: 再生開始までの待ち時間 (先読みあり, $\beta = 0.6$)



(a) アルゴリズム 1

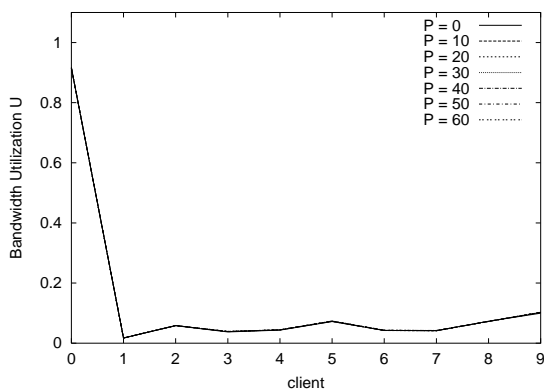


(b) アルゴリズム 2

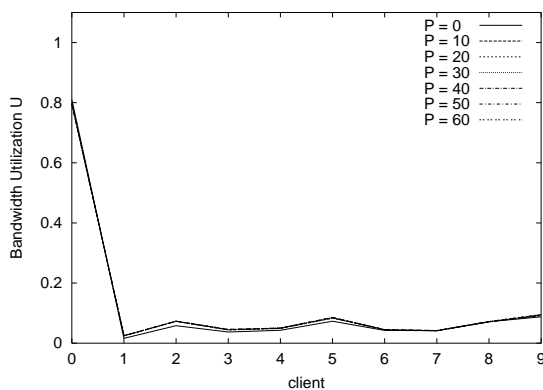


(c) アルゴリズム 3

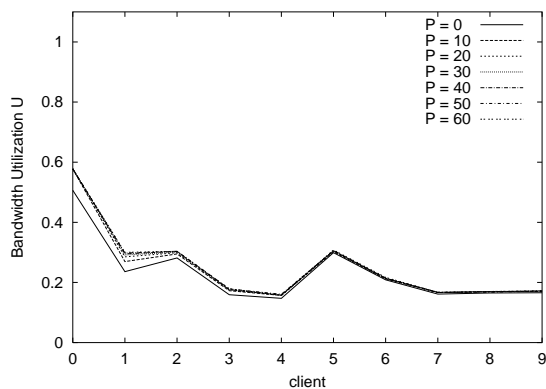
図 26: 受信動画像品質に対する満足度 (先読みあり, $\beta = 0.6$)



(a) アルゴリズム 1

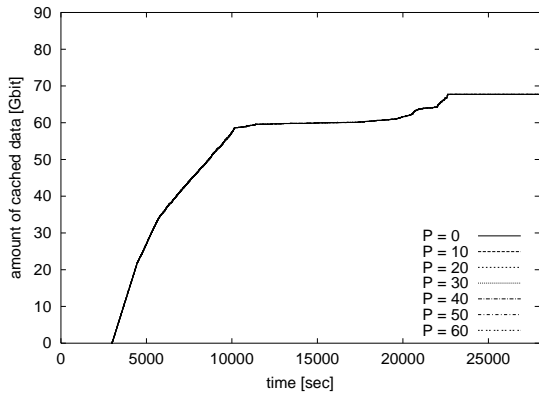


(b) アルゴリズム 2

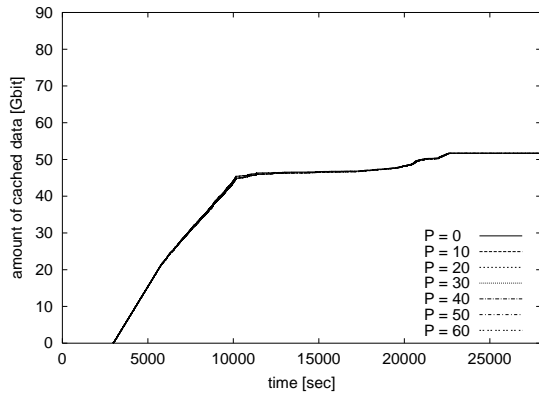


(c) アルゴリズム 3

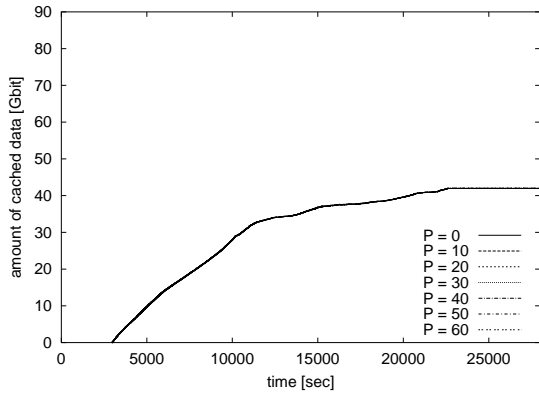
図 27: サーバ-プロキシ間の帯域利用率 (先読みあり, $\beta = 0.6$)



(a) アルゴリズム 1



(b) アルゴリズム 2



(c) アルゴリズム 3

図 28: キャッシュ内のデータ量の変化 (先読みあり, $\beta = 0.6$)

5 プロキシにおけるキャッシュデータ置き換えアルゴリズム

これまで、提案手法の理想特性を明らかにするため、無限大のキャッシュバッファを有するプロキシキャッシュシステムを対象とした評価を行ってきた。多くの動画ストリームを同時に扱わなければならないプロキシでは動画ストリームあたりの利用可能なバッファサイズが限られることを考慮し、本章では有限なキャッシュバッファを有するプロキシにおけるキャッシングメカニズムについて検討する。

5.1 キャッシュデータの置き換えアルゴリズム

バッファサイズ有限のプロキシでは、キャッシュミスや先読みによりサーバから取得した動画データを蓄積するのに十分な空きがキャッシュバッファにないという問題が生じる。そのため、本節ではデータサイズや品質、参照性を考慮したキャッシュデータの置き換えアルゴリズムを提案する。

提案アルゴリズムでは、クライアントごとにキャッシュデータの管理領域を定義し、動画データの参照性を考慮してキャッシュに残す優先度を定める (図 29)。クライアントごとの管理領域は、クライアントがデータ転送を要求している GoP から次のクライアントの管理領域開始位置までとする。また、動画ストリームの先頭は必ず管理領域とする。ストリームの先頭に近い管理領域ほど優先度が高く、また管理領域内ではクライアントが要求中の GoP と先読み範囲内の GoP が最も優先度が高い。

サーバからのデータ取得に際し、キャッシュバッファに十分な空きがない場合には、優先度を考慮して置き換えるキャッシュデータを決定する。ただしクライアント間の公平性を考慮し、領域が大きく、参照性の低い GoP を多く含む管理領域から順に処理する。管理領域の大きさが同じ場合には動画ストリームの後ろに位置する優先度の低い管理領域を選択する。まず管理領域内の最も優先度の低い GoP に品質調整処理を施すことによりデータサイズを小さくする。ただし、キャッシュデータの再利用性を考慮し、式 (3) により導出される品質まで下げても十分な空きが生まれない場合には、GoP データを棄却する。新たに取得した動画データを蓄積可能になるまで、管理領域の選択、置き換え処理を行なう GoP の決定、品質調整によるデータサイズ削減、および棄却の手順を繰り返す。図 29 に GoP の処理順の例を数字で表す。

5.2 性能評価

シミュレーションによりクライアントごとの動画再生開始までの待ち時間 $W(i)$ 、受信動画品質に対する満足度 $S(i)$ 、サーバ-プロキシ間の帯域利用率 $U(i)$ 、およびキャッシュ

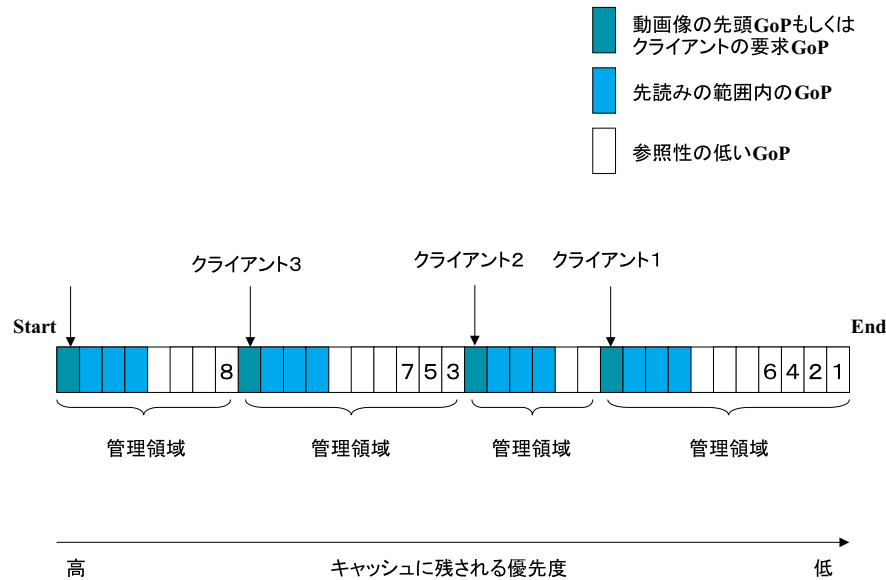


図 29: キャッシュ内に残すデータの優先度

内データ量の変化について提案手法の比較評価を行う。

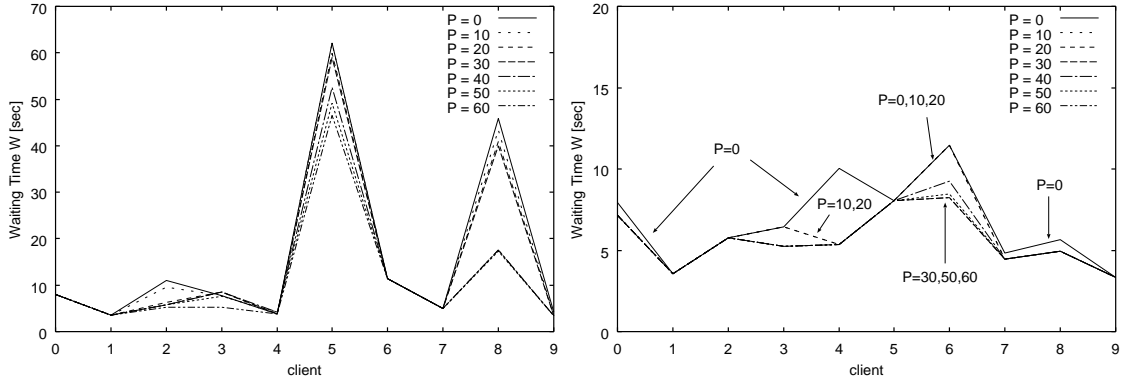
5.2.1 利用可能なバッファサイズが大きい場合

利用可能なバッファサイズを 40Gbit とし、先読みの範囲を変化させた場合のシミュレーション結果を図 30 ~ 36 に示す。

$\beta = 1$ の場合には、アルゴリズム 2, 3 において先読み、置き換えメカニズムを組み合わせることにより、著しい性能劣化を引き起こすことなく、バッファサイズを減らせることがわかる。図 31(a) より、アルゴリズム 1 ではクライアント 1 によるデータ転送はほとんどない。したがって、図 32(a) に示されるキャッシュ内データ量の増加はクライアント 0 によるデータ取得によるものであり、クライアント 2 がセッションを開始するまでのキャッシュバッファはクライアント 0 の取得した高品質な動画像データで占められている。さらに、5.1 節で述べたとおり、動画像ストリームの先頭部分は非常に優先度が高いため、置き換えの対象となりにくく、キャッシュに残り続ける。そのためバッファが有効に利用されず、キャッシュミスによりデータ転送遅延が増大し、待ち時間が大きくなる。また、クライアント 4 までとクライアント 5 のサービス開始時刻が離れているため (図 16)、クライアント 5 の管理領域は大きく、優先度が低い。したがってクライアント 5 が将来参照する GoP が置き換えの対象となりやすくなりキャッシュミスが多く発生するが、利用可能なサーバ-プロキシ間の帯

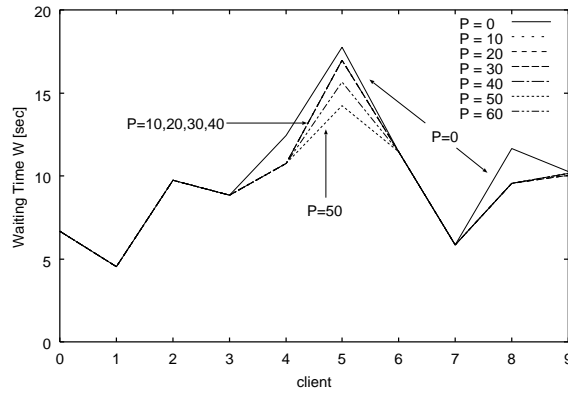
域が比較的狭いため (図 15(a)), 転送遅延が増大する .

一方, $\beta = 0.6$ の場合には, 図 27 と図 35 を比較してアルゴリズム 1, 2 でのサーバ-プロキシ間の帯域利用率が増加していることから, キャッシュミスの発生頻度が無限バッファの場合に比べて大きくなっていると考えられる . ただし, $\beta = 1$ の場合と比べて性能劣化の度は小さく, 画質もそれほど低下していない .



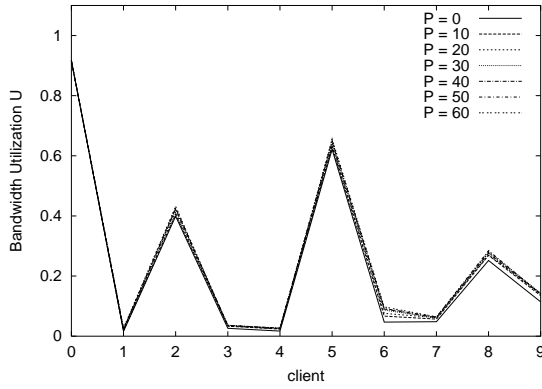
(a) アルゴリズム 1

(b) アルゴリズム 2

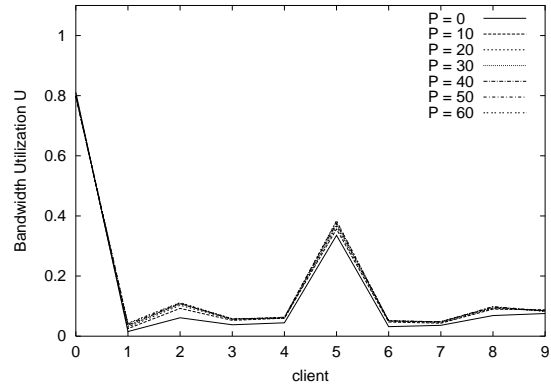


(c) アルゴリズム 3

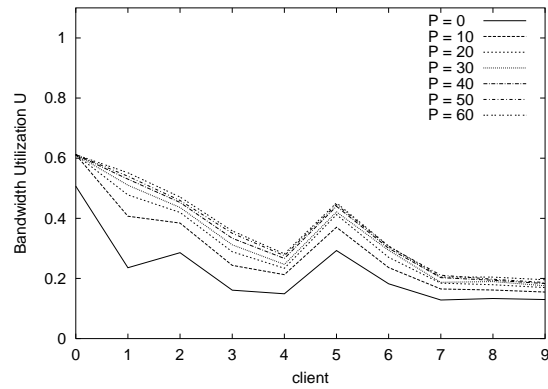
図 30: 再生までの待ち時間 (バッファサイズ 40Gbit , $\beta = 1$)



(a) アルゴリズム 1

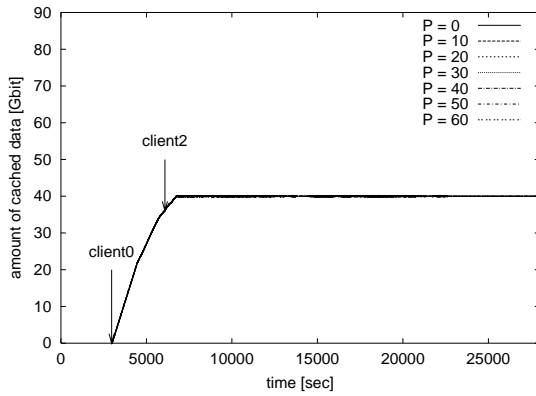


(b) アルゴリズム 2

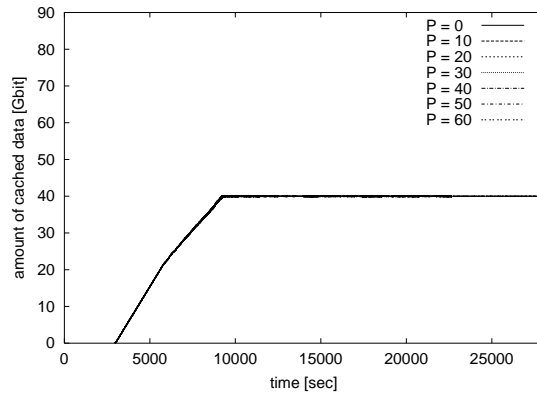


(c) アルゴリズム 3

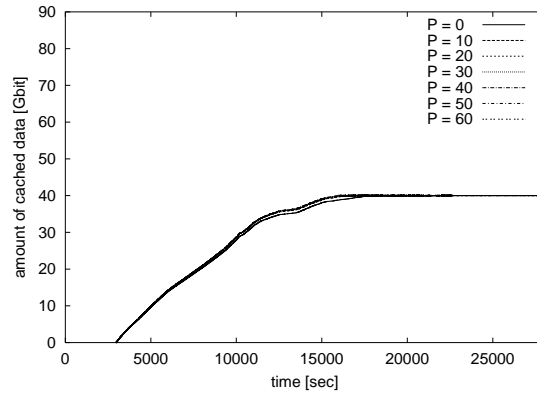
図 31: サーバ-プロキシ間の帯域利用率 (バッファサイズ 40Gbit , $\beta = 1$)



(a) アルゴリズム 1

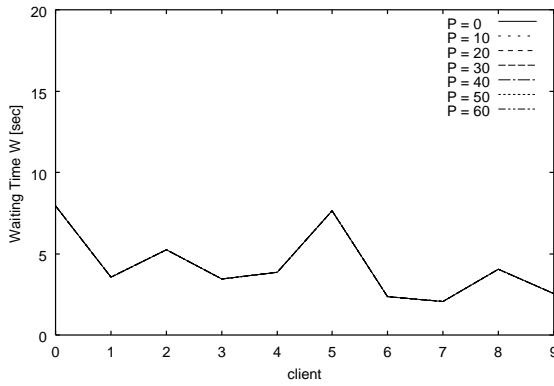


(b) アルゴリズム 2

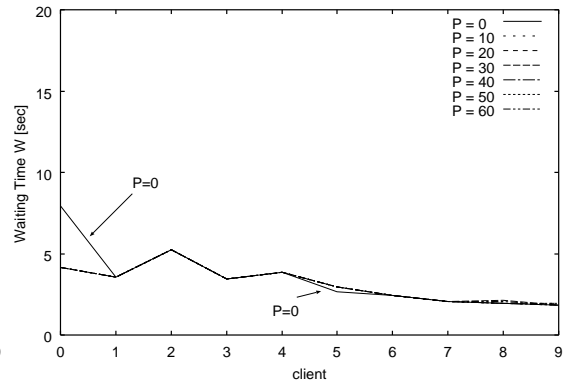


(c) アルゴリズム 3

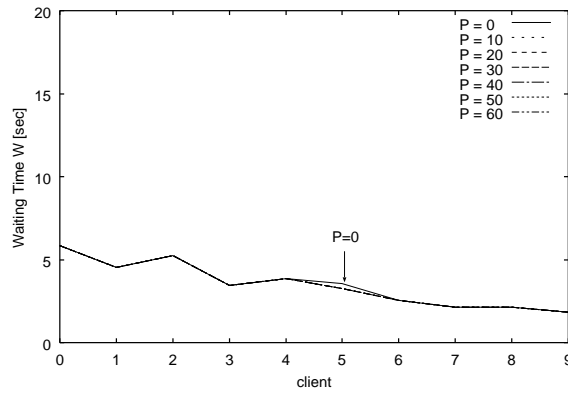
図 32: キャッシュ内のデータ量の変化 (バッファサイズ 40Gbit , $\beta = 1$)



(a) アルゴリズム 1

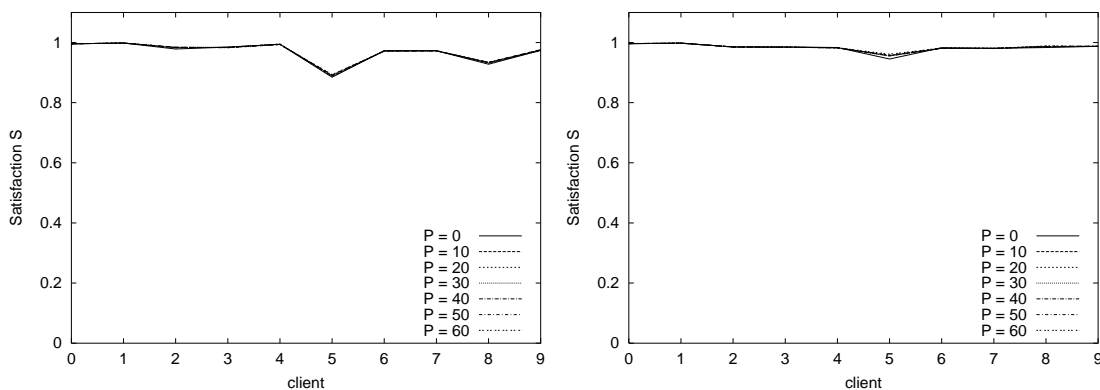


(b) アルゴリズム 2



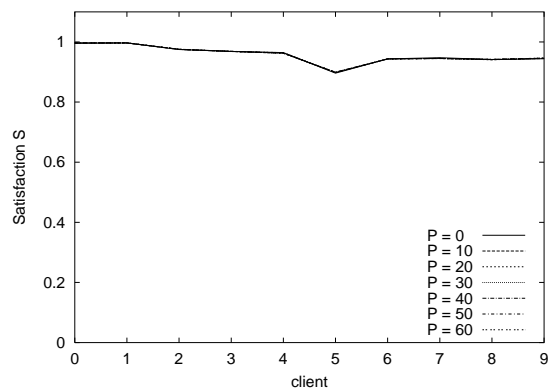
(c) アルゴリズム 3

図 33: 再生までの待ち時間 (バッファサイズ 40Gbit , $\beta = 0.6$)



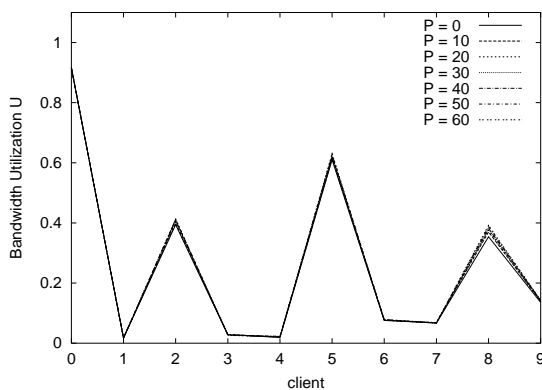
(a) アルゴリズム 1

(b) アルゴリズム 2

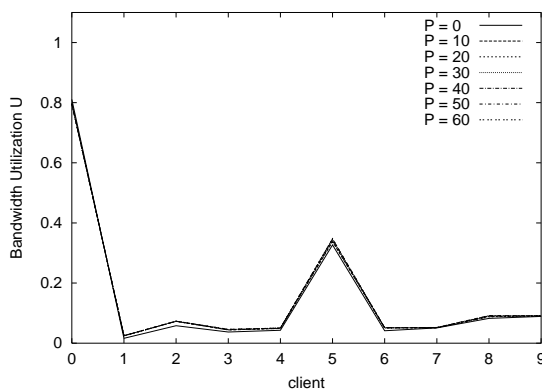


(c) アルゴリズム 3

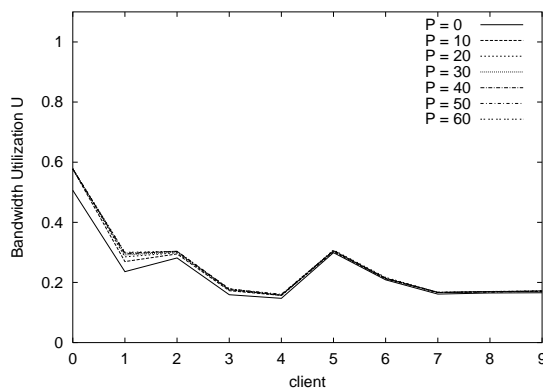
図 34: 受信動画像品質に対する満足度 (バッファサイズ 40Gbit , $\beta = 0.6$)



(a) アルゴリズム 1

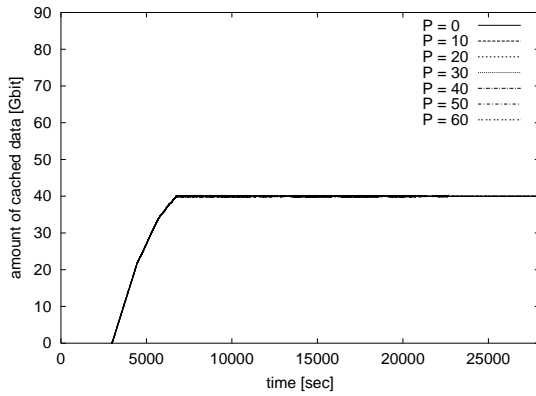


(b) アルゴリズム 2

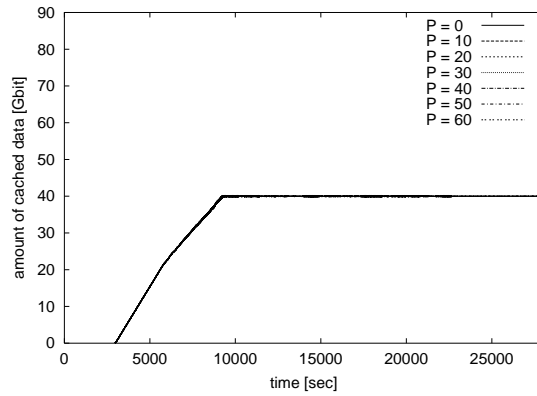


(c) アルゴリズム 3

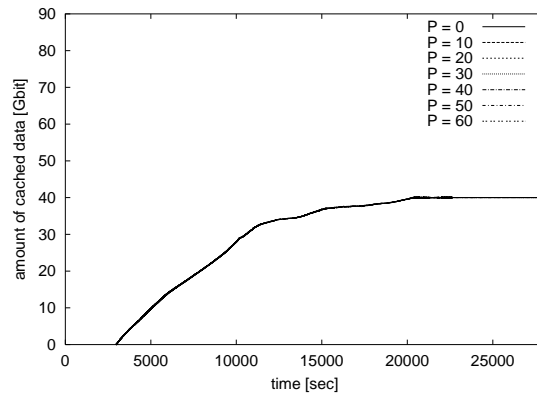
図 35: サーバ-プロキシ間の帯域利用率 (バッファサイズ 40Gbit , $\beta = 0.6$)



(a) アルゴリズム 1



(b) アルゴリズム 2



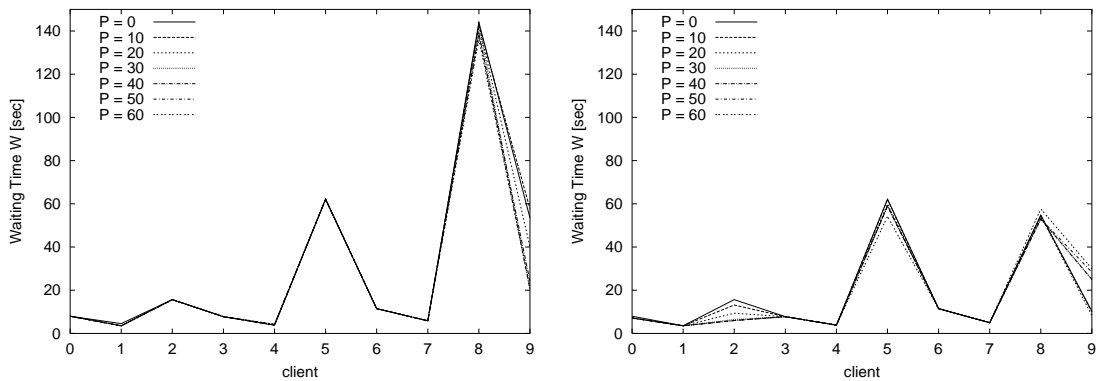
(c) アルゴリズム 3

図 36: キャッシュ内のデータ量の変化 (バッファサイズ 40Gbit , $\beta = 0.6$)

5.2.2 利用可能なバッファサイズが比較的小さい場合

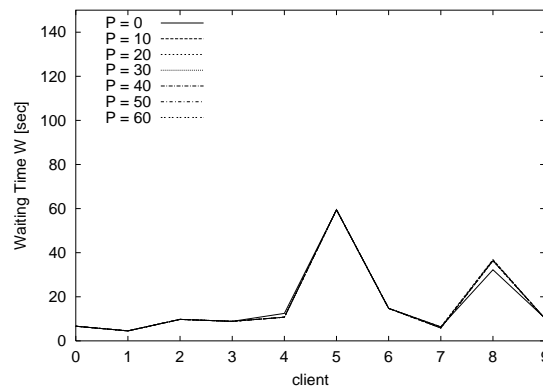
利用可能なバッファサイズを 20Gbit とした場合のシミュレーション結果を図 37~43 に示す。

$\beta = 1$ の場合には、すべての提案アルゴリズムにおいて性能が劣化しており、特にクライアント 5, 8 の待ち時間が極端に大きい。また、サーバに対する要求品質の高いアルゴリズムほど性能劣化の度合いが大きく、これはデータサイズが大きいことによる転送時間の増加と、キャッシュ内に蓄積できる GoP 数が少なくなることによるキャッシュミス発生率の増大によるものである。したがって β を低くすることによりキャッシュデータの再利用性を高めることの効果は高く、またキャッシュデータの品質調整範囲が広がるためより高いサイズ圧縮効果が得られ、キャッシュバッファを有効利用できる。



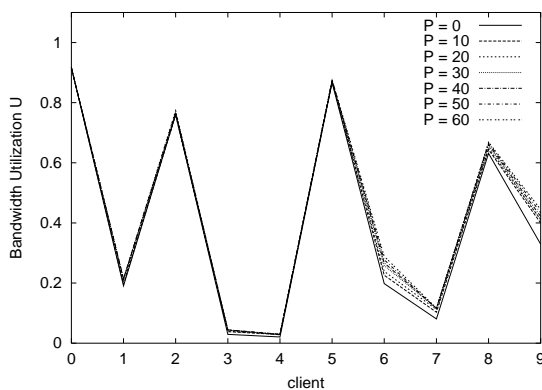
(a) アルゴリズム 1

(b) アルゴリズム 2

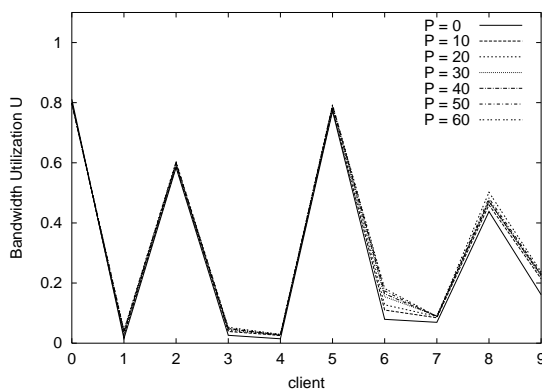


(c) アルゴリズム 3

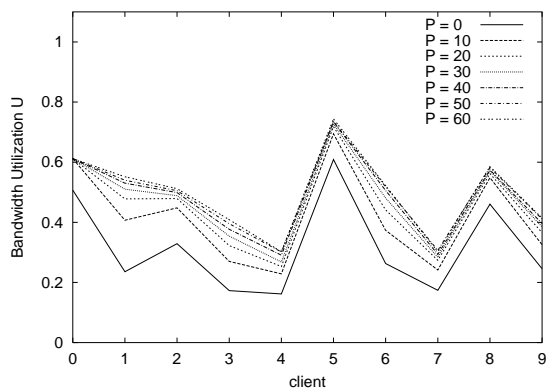
図 37: 再生までの待ち時間 (バッファサイズ 20Gbit , $\beta = 1$)



(a) アルゴリズム 1

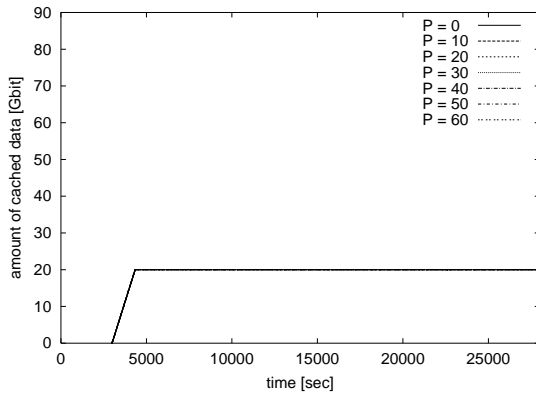


(b) アルゴリズム 2

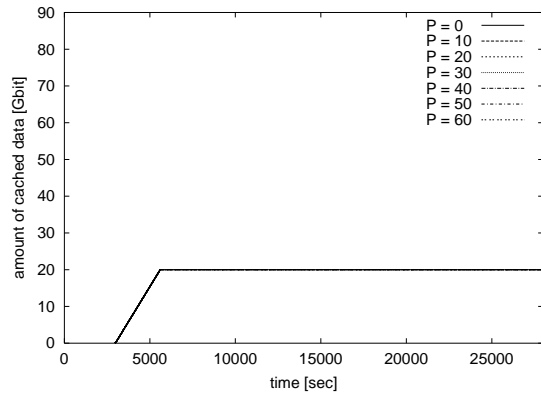


(c) アルゴリズム 3

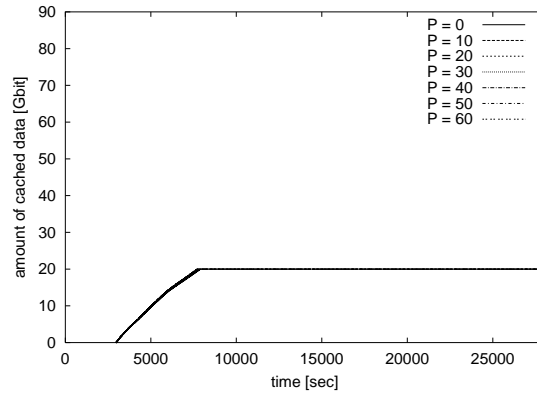
図 38: サーバ-プロキシ間の帯域利用率 (バッファサイズ 20Gbit , $\beta = 1$)



(a) アルゴリズム 1

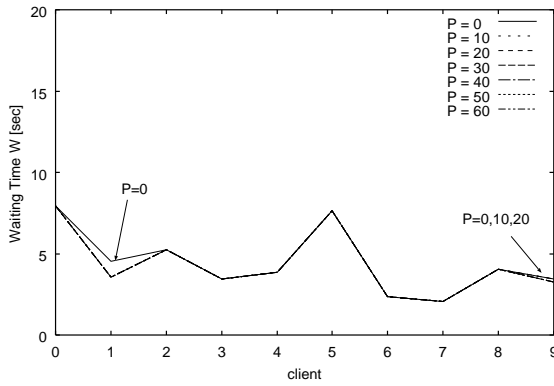


(b) アルゴリズム 2

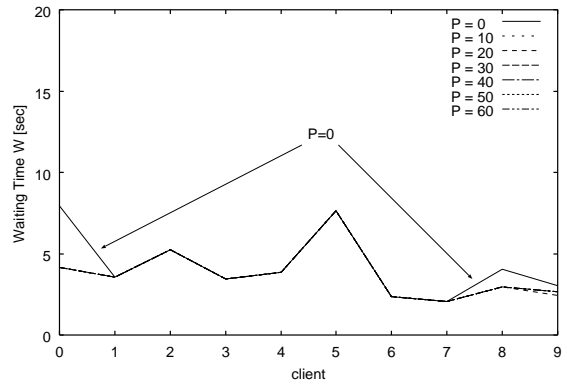


(c) アルゴリズム 3

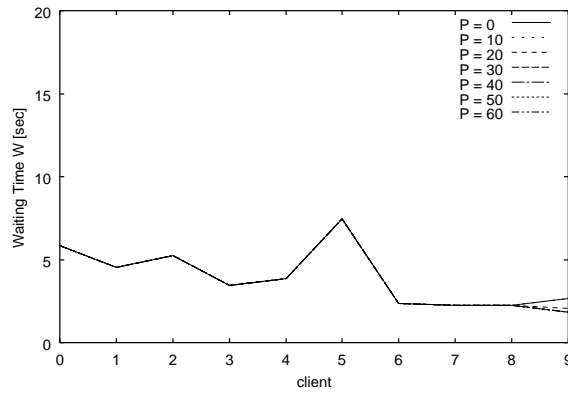
図 39: キャッシュ内のデータ量の変化 (バッファサイズ 20Gbit , $\beta = 1$)



(a) アルゴリズム 1

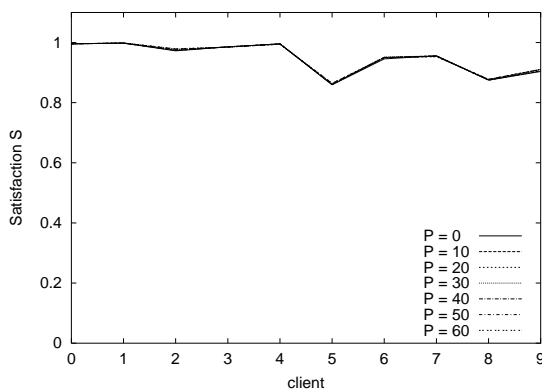


(b) アルゴリズム 2

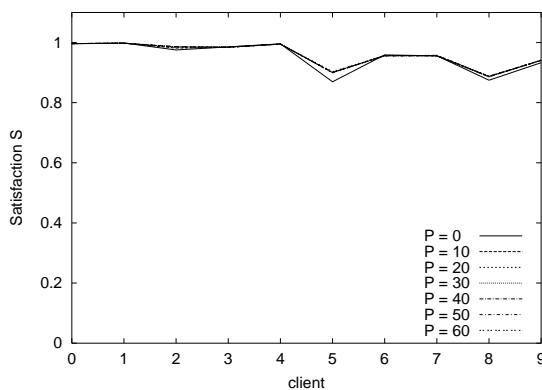


(c) アルゴリズム 3

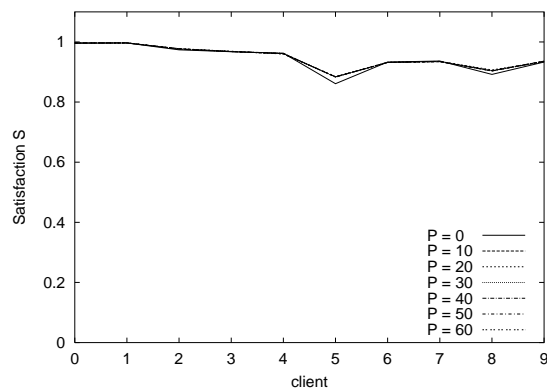
図 40: 再生までの待ち時間 (バッファサイズ 20Gbit , $\beta = 0.6$)



(a) アルゴリズム 1

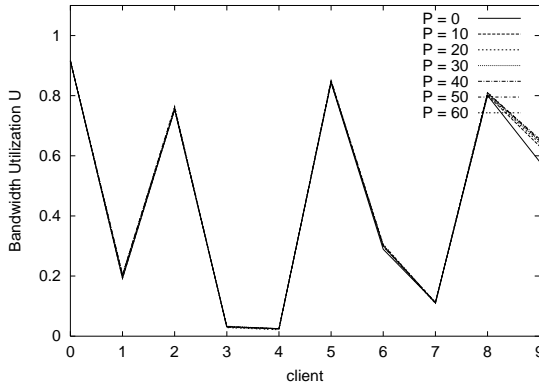


(b) アルゴリズム 2

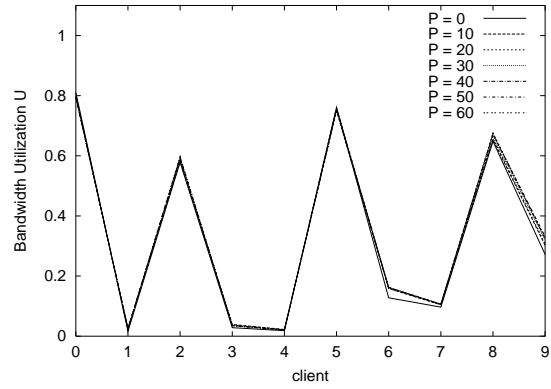


(c) アルゴリズム 3

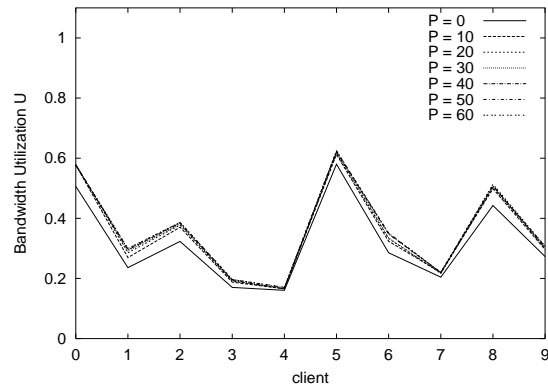
図 41: 受信動画像品質に対する満足度 (バッファサイズ 20Gbit , $\beta = 0.6$)



(a) アルゴリズム 1

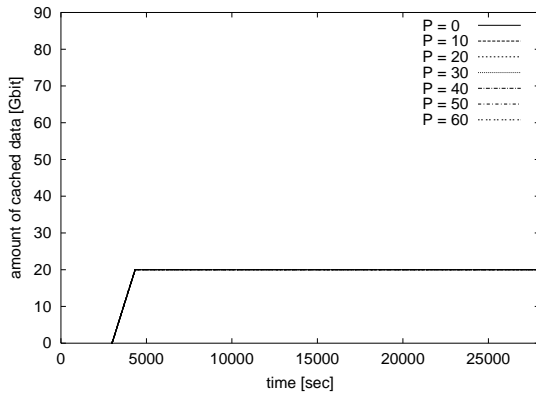


(b) アルゴリズム 2

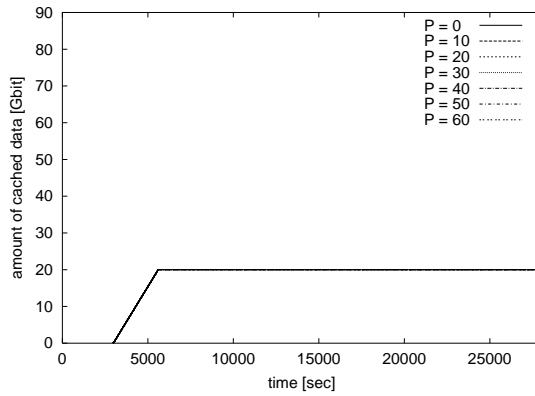


(c) アルゴリズム 3

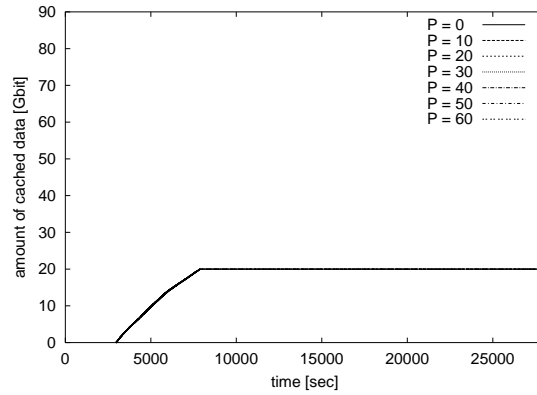
図 42: サーバ-プロキシ間の帯域利用率 (バッファサイズ 20Gbit , $\beta = 0.6$)



(a) アルゴリズム 1



(b) アルゴリズム 2



(c) アルゴリズム 3

図 43: キャッシュ内のデータ量の変化 (バッファサイズ 20Gbit , $\beta = 0.6$)

6 おわりに

本報告では、インターネットにおける動画像ストリーミングサービスにおいて、クライアントへの応答性の向上やネットワークの負荷軽減をはかるための、動画像品質調整機能を組み込んだプロキシキャッシングメカニズムを提案した。複数の異なるキャッシングアルゴリズムを提案し、シミュレーションにより、プロキシにおいてネットワーク環境やキャッシュバッファ容量などを考慮した効果的な制御を行なう提案手法の有効性を示した。特に、バッファサイズが小さい場合にも、配送される動画像データの品質に対する制約をゆるやかにすることで、待ち時間や動画像品質に関する性能をそれほど劣化させることなく、効率のよい動画像配信サービスをクライアントに提供できることが明らかとなった。ただし、動画像品質に対する制約が厳しい場合にも性能劣化の度合の少ない制御アルゴリズムについて今後検討する必要がある。また、本報告ではユーザは動画像ストリームを先頭から最後まで順に見ることを前提としたが、早送り、巻き戻しといった操作についても考慮しなければならない。さらに、複数の動画像ストリームを扱うプロキシではキャッシュバッファを共有するストリーム間での公平性を考慮する必要がある。

謝辞

本報告を終えるにあたり，御指導，御教授を頂いた宮原秀夫教授に深く感謝致します．また，本報告において終始直接御指導頂いた村田正幸教授，若宮直紀講師に深く感謝致します．

並びに日頃から適切な助言を頂いた大阪大学サイバーメディアセンターの下條真司教授，大阪府立看護大学の菅野雅嗣助教授，大阪市立大学の藤川和利講師，大阪大学サイバメディアセンター馬場健一助教授，神戸商船大学の鎌原淳三講師，大阪大学サイバーメディアセンターの大崎博之助手，長谷川剛助手，奈良先端科学技術大学院大学の奥田剛助手，大阪市立大学の阿多信吾助手，経済学部の荒川伸一助手，国際公共政策研究科の植田和憲助手に心から感謝致します．

最後に，御協力頂いた宮原研究室および村田研究室の皆様にご心からお礼申し上げます．

参考文献

- [1] N. Yeadon, F. García, D. Hutchinson, and D. Shepherd, “Filters: QoS support mechanisms for multipeer communications,” *IEEE Journal on Selected Areas in Communications*, vol. 14, pp. 1245–1262, September 1996.
- [2] ISO/IEC DIS 13818-2, “MPEG-2 video,” *ISO standard*, 1994.
- [3] Naoki Wakamiya, Masayuki Murata, and Hideo Miyahara, “TCP-friendly video transfer,” in *Proceedings of SPIE International Symposium on Information Technologies 2000*, November 2000.
- [4] Masaki Miyabayashi, Naoki Wakamiya, Masayuki Murata, and Hideo Miyahara, “Implementation of video transfer with TCP-friendly rate control,” in *Proceedings of International Technical Conference on Circuits/Systems, Computers and Communications 2000*, vol. 1, pp. 117–120, July 2000.
- [5] 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-00-03*, March 2000.
- [6] 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.
- [7] “UCB/LBNL/VINT Network Simulator - ns (version 2).” available at <http://www-mash.cs.berkeley.edu/ns/>.