

動画品質を変更可能な分割放送型配信における帯域幅を考慮したスケジューリング手法の提案

出口 拓人¹ 後藤 佑介²

概要：近年、音声や映像といったデータの送受信で発生するトラフィックが全世界で急増しており、ネットワーク環境に適応した動画配信システムが求められている。動画配信方式の一つである放送型配信では、多くのクライアントに同じ動画データを同時に配信することでサーバの負荷を軽減でき、ネットワーク全体のトラフィックを削減できる。一方で、クライアントは、動画データの受信を要求してから再生が開始されるまでの間で待ち時間が発生する。この待ち時間を短縮するため、動画データを複数のセグメントに分割して複数のチャンネルで配信する分割放送型配信が提案されており、分割放送型配信においてデータ受信時の待ち時間を短縮するスケジューリング手法が多数提案されている。既存のスケジューリング手法では、配信する動画データの再生レートが同じ環境で配信スケジュールを作成していた。しかし、サーバは再生レートが異なる複数種類の動画を配信し、クライアントがネットワーク環境に基づいて動画品質を変更しながら再生する場合を考慮していない。本研究では、動画品質を変更可能な分割放送型配信において使用する帯域幅を考慮したスケジューリング手法を提案する。提案手法では低画質、中画質、高画質の3種類の動画データに対して、画質間の差分データを別のチャンネルで配信することで待ち時間を短縮する。評価では、データサイズ、帯域幅をそれぞれ変化させた場合における待ち時間の変化について、3種類の動画データをすべて配信する単純手法と比べて、提案手法は待ち時間を短縮することを確認した。

キーワード：スケジューリング、帯域幅、動画品質、分割放送型配信、待ち時間

1. はじめに

インターネットによる映像視聴方法の多様化 [1] にともない、ネットワーク環境の変化に適応した動画配信システムに対する注目が高まっている。動画データの配信方式として、オンデマンド型配信と放送型配信の2種類が挙げられる。オンデマンド型配信では、サーバはクライアントの受信要求に応じて帯域幅を確保して動画を配信する。このため、同じ時刻に動画を要求するクライアント数の増加にともない、サーバが使用する帯域幅は比例して増加する。一方、放送型配信では、サーバは一定の帯域幅で複数のクライアントに同じ動画データを繰り返して配信するが、クライアントは要求している動画データが配信されるまで待つ必要がある。

放送型配信で発生する待ち時間を短縮するため、動画データを複数のセグメントに分割し、複数のチャンネルで配

信する分割放送型配信が研究されてきた [2], [3]。分割放送型配信では、各チャンネルに帯域幅を割り当て、動画データの先頭部分となるセグメントをできるだけ多く配信することで、受信要求から再生開始までの待ち時間を短縮できる。

これまでの分割放送型配信では、再生中に動画データの品質を表す再生レートが変化する場合を考慮していなかった。そこで、動画配信技術の一つである Adaptive bitrate (ABR) [4], [5] によるストリーミング配信を想定したスケジューリングを考える。ABR は、通信状況に応じてクライアントが受信する動画品質を動的に切り替えできる。クライアントは、ABR を用いることで、再生を中断せずにできるだけ高品質の動画データを再生できる。

本研究では、再生中の動画品質の変更を考慮したスケジューリング手法を提案する。提案手法では、サーバは、クライアントが使用できる帯域幅を考慮して、再生レートが異なる複数種類の動画データを複数の部分に分割して複数のチャンネルで繰り返し配信することで、すべての動画データを受信する単純手法に比べて、動画品質の変更時に発生する待ち時間を短縮できる。

¹ 岡山大学大学院自然科学研究科
Graduate School of Natural Science and Technology,
Okayama University

² 岡山大学学術研究院自然科学学域
Faculty of Natural Science and Technology, Okayama University

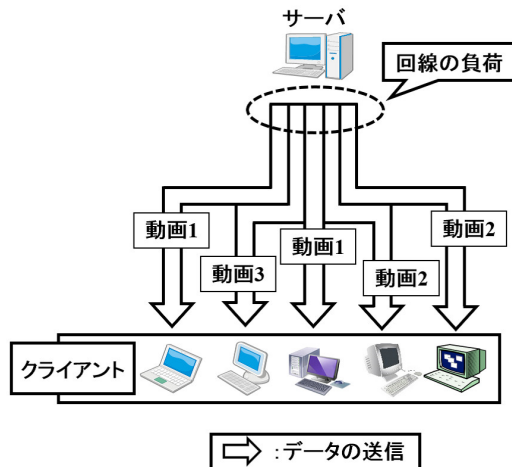


図 1 オンデマンド型配信によるデータ配信

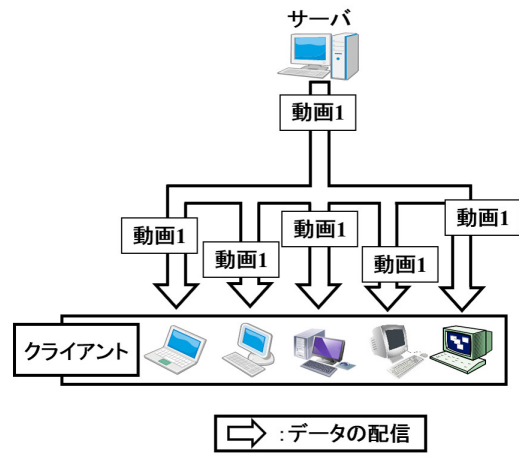


図 2 放送型配信によるデータ配信

2. 動画データの配信方式

2.1 概要

動画データの配信方式として、オンデマンド型配信および放送型配信の 2 種類が挙げられる。以下で、順番に説明する。

2.2 オンデマンド型配信

オンデマンド型配信は、NHK オンデマンド [6] をはじめとするインターネットテレビや YouTube [7] といった多数の動画を配信する動画共有サービスで用いられる配信方式である。オンデマンド型配信でサーバが動画データを配信する様子を図 1 に示す。オンデマンド型配信では、サーバはデータの受信を要求するクライアントごとにチャンネルを設定し、チャンネルの帯域幅を確保してデータを配信する。一方で、クライアントの数に比例して、サーバの処理負荷や通信に必要な帯域幅は増加する。通信に必要な帯域幅が使用できる帯域幅を上回ると、サーバにデータの受信を要求する新たなクライアントに対して帯域幅を確保できず、待ち時間は長大化する。

2.3 放送型配信

放送型配信は、多くのクライアントが同じ動画データを受信する場合に用いられる配信方式である。即時性が高いスポーツ番組やコンサート映像のストリーミング配信、および特定の番組を一つのチャンネルで繰り返し配信する CS デジタル放送 [8] といったサービスにおける利用を想定している。

放送型配信でサーバがデータを配信する様子を図 2 に示す。放送型配信は、マルチキャストを用いて、あらかじめ指定した複数のクライアントに同じ動画データをまとめて配信する方式である。サーバは一定の帯域幅でデータを配信でき、クライアントの数が増加しても配信サーバの処理

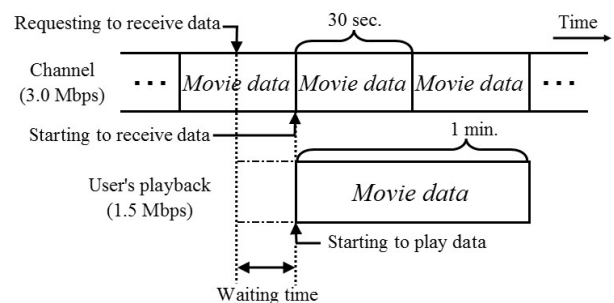


図 3 放送型配信における待ち時間発生の様子

負荷や使用する帯域幅の増加を抑制できる。一方で、クライアントは、データの受信を要求してから再生を開始するまでの間に待ち時間が発生する。

放送型配信において待ち時間が発生する様子を図 3 に示す。チャンネルの帯域幅を 3.0 Mbps、動画データの再生レートを 1.5 Mbps、再生時間を 1 分とする。サーバは複数のクライアントに同じデータを繰り返し配信する。図 3 の場合、クライアントが動画データの受信を開始する契機は動画データの先頭となるため、クライアントは動画データの受信を開始すると再生を開始できる。クライアントが動画データを受信するときの待ち時間は、サーバが動画データの先頭部分を配信した直後にクライアントが受信を要求した場合に最長となり、動画データ 1 周期分の $(60 \times 1.5) / 3.0 = 30$ 秒かかる。一方、動画データの先頭部分を配信する直前に受信を要求した場合に待ち時間は最短となり、0 秒となる。以上より、図 3 におけるクライアントの平均待ち時間は、 $(30 + 0) / 2 = 15$ 秒となる。

2.4 分割放送型配信

分割放送型配信は、放送型配信で発生する待ち時間を短縮するため、動画データを複数の部分（以下、セグメント）に分割して、最初のセグメントを頻繁に配信することで待ち時間を短縮する配信方式である。例えば、図 3 で

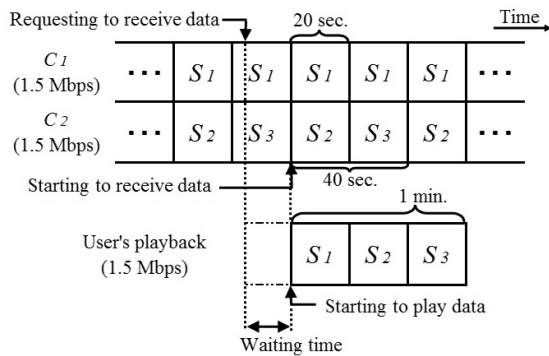


図 4 FB 法のスケジューリング例

用いたデータを 2 個のセグメント S_1, S_2 に等分割して、1.5 Mbps の 2 個のチャンネルで S_1, S_2 をそれぞれ繰り返し配信する場合、クライアントは受信した S_1, S_2 を途切れなく順番に再生できる。このとき、クライアントの待ち時間は S_1 の先頭が配信されるまでの時間であり、最長で $(60 \times 0.5 \times 1.5) / 1.5 = 30$ 秒、最短で 0 秒となり、平均の待ち時間は $(30 + 0) / 2 = 15$ 秒となる。分割放送型配信では、最初のセグメントのデータサイズを小さくすることで再生開始までの待ち時間をより短縮できるが、動画データの再生中にセグメント間で途切れが発生する。再生中に途切れないように途切れ時間分だけ再生開始を遅らせるため、全体の待ち時間は長大化する。そこで、動画データの再生中に途切れを発生させずに受信時の待ち時間を短縮するため、動画データの配信条件に応じてセグメントの分割比率を決定するスケジューリング手法が提案されてきた。

図 4 に、既存のスケジューリング手法である Fast Broadcasting (FB) 法 [9] による配信スケジュール例を示す。FB 法は、帯域幅を k 個のチャンネルに分け、動画データを $2^k - 1$ 個のセグメントに等分割した上で、 i 番目のチャンネル C_i に連続した $2^i - 1$ 個のセグメントをスケジューリングする手法である。図 4 では、チャンネルの帯域幅、動画データの再生レート、および再生時間は図 3 と同じとする。

図 4 の場合、サーバは使用できる帯域幅を 1.5 Mbps の 2 個のチャンネル C_1, C_2 に分割した上で、一続きの動画データを 2 個のセグメント S_1, S_2, S_3 に分割して、 C_1 で S_1, C_2 で S_2 と S_3 をそれぞれ繰り返し配信する。FB 法では、動画データを $2^k - 1$ 個のセグメントに等分割するため、 S_1, S_2, S_3 の分割比率は等しく、再生時間はそれぞれ 20 秒となる。このとき、待ち時間は S_1 の先頭が配信されるまでの時間となり、最長で $(20 \times 1.5) / 1.5 = 20$ 秒、最短で 0 秒となり、平均待ち時間は $(20 + 0) / 2 = 10$ 秒となる。データを分割しない場合、およびデータを 2 個のセグメントに等分割する場合の受信時間はどちらも 15 秒であり、スケジューリング手法を用いて分割することで、待ち時間を約 33.3 % 短縮できる。以上より、分割放送型配信においてセグメントの分割比率を決定するスケジューリング手法の導

入は重要である。

3. 関連研究

3.1 スケジューリング手法

HB (Harmonic Broadcasting) 法 [10] では、連続メディアデータを N 個のセグメント S_1, \dots, S_N に等分割する。さらに S_i ($i = 1, \dots, N$) を i 個のサブセグメント $S_{i,1}, \dots, S_{i,i}$ に等分割する。 N 個のチャンネル C_1, \dots, C_N を用い、 C_i で $S_{i,1}, \dots, S_{i,i}$ を繰り返して放送する。例えば、MPEG2 で符号化された 5.0 Mbps の 60 分の連続メディアデータを衛星デジタル放送を想定した 24 Mbps の帯域幅を用いて放送する場合、 $N = 67$ となり、67 個のチャンネルが必要になる。

AHB (Asynchronous Harmonic Broadcasting) 法 [11] では、MPEG2 の GOP (Group of Pictures) や MP3 のフレームといった再生の単位を考慮してスケジューリングすることで、HB 法に比べて平均待ち時間を短縮している。再生の単位ごとにデータを分割するため、使用するチャンネルの数は再生の単位と同じ数になる。例えば、衛星デジタル放送を想定した 24 Mbps の帯域幅を用いて、MPEG2 で符号化された 5.0 Mbps の 60 分の動画データを放送する場合、再生の単位を 0.6 秒とすると、再生単位の数は 6,000 個になり、使用するチャンネル数も 6,000 個と膨大になる。

パケット欠損を考慮したスケジューリング手法として、Synchronized Double Buffering Broadcasting [12]、および Data Packet Broadcasting [13] が挙げられる。これらのスケジューリング手法で、パケットが欠損する可能性が高いセグメントの配信回数を増やすことで、欠損したパケットを再び受信するまでの間で発生する待ち時間の長大化を抑える。しかし、配信回数が増えたセグメントの受信において、パケット欠損が連続的に発生した場合、待ち時間は短くならない。また、実際の動画配信サービスで配信中にパケット欠損が生じた場合、オンデマンド型配信で欠損したパケットを再送する方法、もしくは欠損前後のパケットを用いて欠損した部分を補完する方法が考えられる。

また、ライブ配信を考慮したスケジューリング手法として、The Scalable Binomial Broadcasting [14] が挙げられる。The Scalable Binomial Broadcasting では、配信が終了する予定時刻をもとに、FB 法や HB 法といった既存のスケジューリング手法を用いて配信スケジュールを作成する。配信が終了する予定時刻を過ぎる場合、視聴中のクライアントの再生が途切れないように配信スケジュールを更新する。

3.2 分割放送型配信システム

これまで提案されてきたスケジューリング手法の多くは、計算機を用いたシミュレーション環境を想定している [15]。このため、配信スケジュールの動的更新を行う

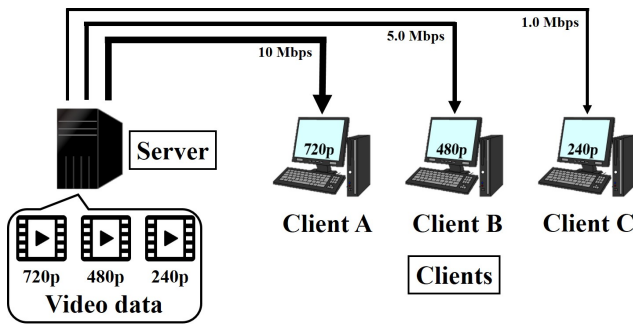


図 5 ABR を用いた動画配信の例

スケジューリング手法 [16] を用いる場合、システムの負荷やパケット欠損による動画再生への影響、および実際のネットワーク環境においてスケジューリング手法を用いた場合に発生する影響を考慮していない。筆者らの研究グループでは、実際の計算機ネットワーク環境でスケジューリング手法の導入、および評価が可能な分割放送型配信システム Telecommunication and BroadCasting System (*TeleCaS*) [17] を提案してきた。

TeleCaS における課題として、配信する動画データを動的に変更できない点が挙げられる。分割放送型配信において、サーバで使用できる帯域幅が不足すると、動画の再生を中断する可能性がある。そこで、サーバが使用できる帯域幅に応じて配信する動画データの再生レートを変更することで、再生中断の発生を抑制する必要がある。しかし、*TeleCaS* ではあらかじめ設定したスケジュールで配信するため、サーバで使用できる帯域幅が変化した場合、サーバは変化後の帯域幅に対応した再生レートの動画をクライアントに配信できない。

4. 動画品質の変更を考慮した配信方式

4.1 Adaptive bitrate によるストリーミング配信

Adaptive bitrate (ABR) [4], [5] によるストリーミング配信において、サーバは、動画配信中に各クライアントの通信状況に応じた再生品質の動画データを配信する。ABR は、YouTube [7] や Gyaol! [18] といった Video on Demand (VOD) サービスで利用されている。サーバは、ABR を用いることで、通信状況が良い場合は高品質の動画、通信状況が悪い場合は低品質の動画をそれぞれ配信し、クライアントは再生中断の発生を抑えた動画再生が可能となる。また、サーバは、ABR で配信する動画を数秒ごとのデータに分割して保存することで、クライアントは、通信状況の変化に応じて、再生中に動画の品質を変更できる。

ABR による配信の様子を図 5 に示す。図 5 の例では、3 台のクライアント A, B, C がサーバに同じ品質となる動画の再生をそれぞれ要求する。サーバは、動画の品質が異なるデータとして、720 p (1280 × 720 pixel), 480 p (854 × 480 pixel), および 240 p (426 × 240 pixel) の 3 種

類を保存しており、クライアントとの通信速度に応じて、最適な品質となる動画ファイルを選択して配信する。

4.2 ABR の利用例

ABR の利用例として、HTTP LIVE Streaming (HLS) [19] や MPEG-Dynamic Adaptive Streaming over HTTP (MPEG-DASH) [20] が挙げられる。MPEG-DASH による配信では、サーバは、再生レートが異なる複数の動画を複数のセグメントに等分割したデータ、およびこれらの動画のデータサイズや URL を記述したマニフェストファイルを用いる。はじめに、クライアントはマニフェストファイルを受信して、サーバが配信する動画の分割数や再生レートに関する情報を取得する。次に、クライアントは、マニフェストファイルの情報や通信速度に基づいて算出したセグメントをサーバに要求する。

4.3 分割放送型配信における動画品質の変更に対する課題

ABR におけるセグメントの選択手法として、サーバとクライアントとの間で使用できる帯域幅をもとに動画の再生レートを選択する手法、およびクライアントのバッファ占有率をもとに選択する手法 [21] が挙げられる。しかし、従来の分割放送型配信では、サーバがクライアントの要求に基づいて配信スケジュールを変更できないため、クライアントが動画の再生レートを制御する手法を適用できない。

5. 提案手法

5.1 概要

動画品質を変更可能な分割放送型配信において帯域幅を考慮したスケジューリング手法を提案する。提案手法では、帯域幅を考慮して動画の分割比率を決定することで、品質変更時の待ち時間を短縮する。動画品質の変更方法として、サーバは、低画質の動画データとともに、低画質と中画質の差分、および中画質と高画質の差分となる動画データをそれぞれ作成する。次に、これらの動画データを複数のセグメントに分割し、複数のチャンネルで繰り返し放送する。

5.2 想定環境

本研究で想定する分割放送型配信の環境について、以下に箇条書きで示す。

- 放送帯域に上限がある。
- 連続する動画データは受信開始と同時に再生できず、分割された 1 個のセグメントを受信しなければ再生を開始できない。
- クライアントはすべてのチャンネルから同時に受信できる。
- クライアントがデータの再生を開始すると、最後まで途切れずに再生できる。
- クライアントはバッファをもち、受信したデータを再

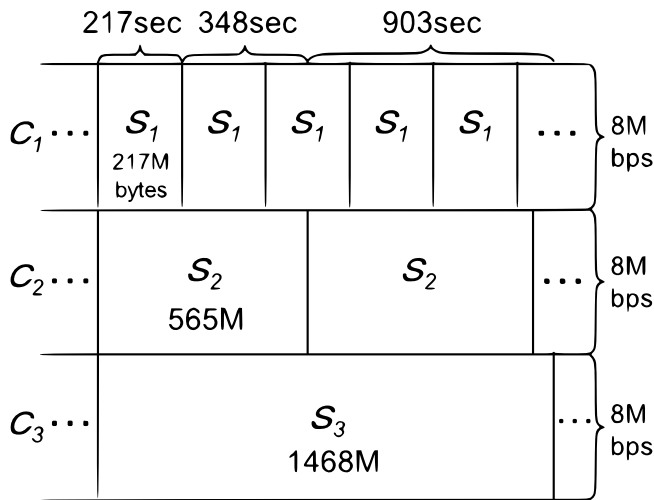


図 6 低画質の動画データを用いたスケジューリング例

生している間も放送されているデータを受信し、バッファに保存できる。

5.3 スケジューリング手順

提案手法のスケジューリング手順を以下に示す。 r を再生レート， B を使用できる帯域幅，使用するチャンネル数を n ，および動画データの再生時間を D とする。

- (1) 低画質の動画データを構成する n 個のセグメント S_i ($i = 1, \dots, n$) に対して，動画データのサイズを a_i ，および分割数 n を用いて， S_i のデータサイズ a_i を以下の式で算出する。

$$a_i = \frac{BD}{n} \times \frac{(1 + \frac{B}{nr})^{i-1}}{(1 + \frac{B}{nr})^n - 1}.$$

- (2) 各チャンネルで使用できる帯域幅がすべて等しい場合，各チャンネルの帯域幅である b_i ($i = 1, \dots, n$) を以下の式で算出する。

$$b_1 = b_2 = \dots = b_n = \frac{B}{N}.$$

- (3) C_i で S_i を繰り返し配信するようにスケジューリングする。
(4) 低画質と中画質の差分データである M_n ，および中画質と高画質の差分データである L_n を各チャンネルでそれぞれスケジューリングする。

5.4 導入例

図 6 に，低画質の動画データを提案手法で配信する場合のスケジューリング例を示す。使用できる帯域幅は 24 Mbps，再生レートは 5.0 Mbps，分割数は 3，再生時間は 60 分とする。図 6 より，低画質の動画データの受信における待ち時間は，約 217 秒となる。

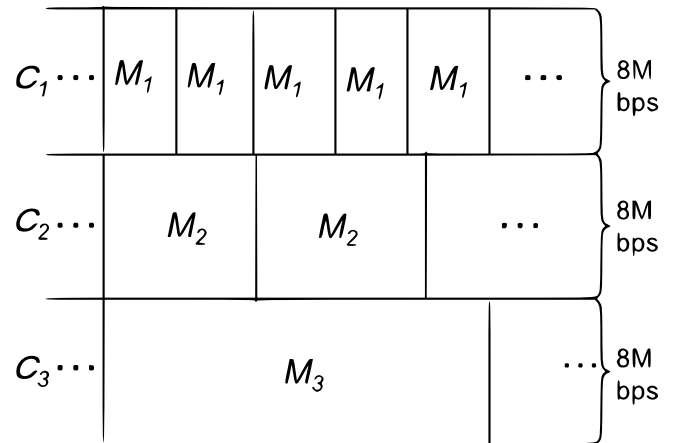


図 7 低画質と中画質の差分データを用いたスケジューリング例

次に，低画質と中画質の差分データを用いたスケジューリング例を図 7 に示す。クライアントは，再生中に低画質から中画質へ動画品質を変更する場合，図 6 に示すスケジューリングで低画質の動画データを受信しながら，図 7 に示すスケジューリングで低画質と中画質の差分データを受信することで，中画質の動画データを再生できる。同様に，クライアントは，中画質と高画質の差分データを受信することで，高画質の動画データを再生できる。

6. 評価

6.1 概要

計算機を用いたシミュレーションで，提案手法の性能を評価する。評価では，はじめに，使用できる帯域幅に応じた待ち時間の変化を求める。次に，動画のデータサイズに応じた待ち時間の変化を求める。そして，最後にはチャンネル数に応じた待ち時間の変化を求める。評価に用いるスケジューリング手法は，提案手法，および差分データを利用しない手法（以下，単純手法）の 2 種類である。動画の品質変更の契機について，低画質の動画データにおいて最初のセグメントを受信した時点で，低画質から中画質に変更する。同様に，中画質の動画データにおいて最初のセグメントを受信した時点で，中画質から高画質に変更する。

6.2 使用できる帯域幅

使用できる帯域幅に応じた待ち時間の変化について評価する。評価結果を図 8 に示す。横軸は使用できる帯域幅，縦軸は待ち時間である。動画データの再生時間は 3,600 秒，再生レートは低画質，中画質，高画質の 3 種類であり，それぞれ 3.0 Mbps， 5.0 Mbps， 7.0 Mbps である。各画質の動画データの分割数は 3 とする。

図 8 より，使用できる帯域幅の変化に応じた提案手法の待ち時間は，単純手法に比べて短くなる。提案手法では，低画質と中画質，および中画質と高画質の差分データを受

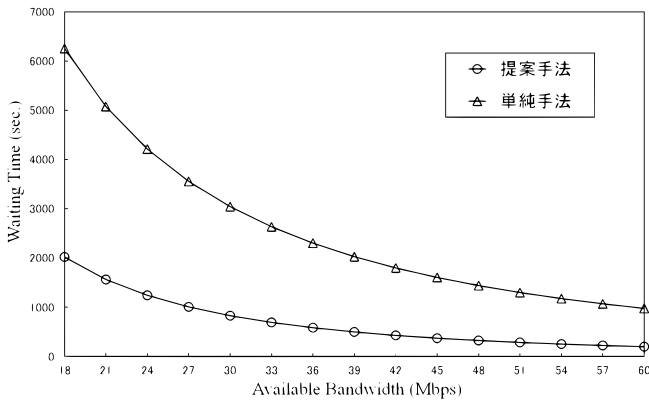


図 8 使用できる帯域幅に対する待ち時間の変化

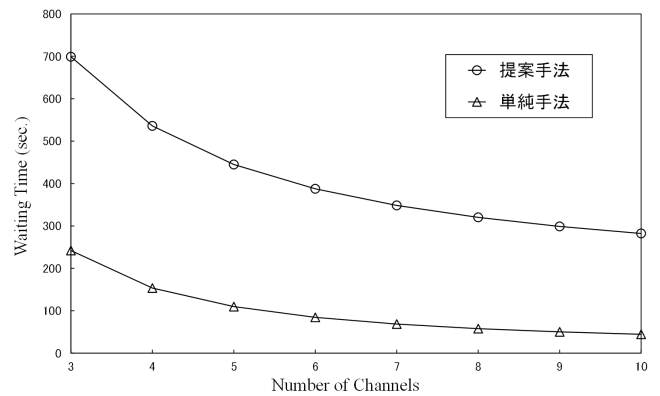


図 10 チャンネル数に対する待ち時間の変化

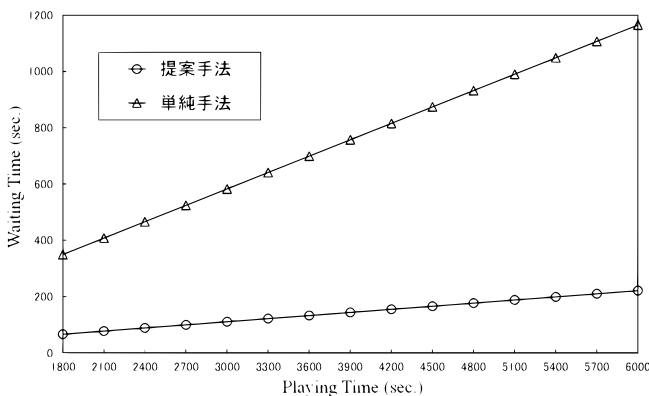


図 9 動画の再生時間に対する待ち時間の変化

信することで、3種類の動画データをすべて受信する単純手法に比べて動画データの受信時間を短縮できる。例えば、使用できる帯域幅が 36 Mbps の場合、提案手法における待ち時間は約 584.5 秒、既存手法で約 2302.4 秒となり、提案手法は単純手法に比べて待ち時間を約 74.6 % 短縮できる。

6.3 動画の再生時間

動画の再生時間の変化に応じた待ち時間の変化について評価する。評価結果を図 9 に示す。横軸は動画の再生時間、縦軸は待ち時間である。使用する帯域幅は 72 Mbps、チャンネル数は 3 とする。再生レートは低画質、中画質、高画質の 3 種類であり、それぞれ 3.0 Mbps, 5.0 Mbps, 7.0 Mbps である。各画質の動画データの分割数は 3 とする。

図 9 より、動画の再生時間が長くなると、両方の手法で待ち時間は長大化する。一方で、提案手法と単純手法との間で待ち時間の差は大きくなる。提案手法では、画質間の差分データを配信するため、動画の再生時間が長い場合でも待ち時間の長大化を抑える。例えば、再生時間が 3,600 秒の場合における待ち時間は、提案手法で約 132.6 秒、単純手法で約 699.0 秒となり、提案手法は単純手法に比べて待ち時間を約 81.0 % 短縮できる。

6.4 チャンネル数に応じた待ち時間

チャンネル数の変化に応じた待ち時間の変化について評価する。評価結果を図 10 に示す。横軸はチャンネル数、縦軸は 3 種類の動画データの受信で発生する待ち時間の平均である。使用する帯域幅は 72 Mbps、動画の再生時間は 3,600 秒とする。再生レートは低画質、中画質、高画質の 3 種類であり、それぞれ 3.0 Mbps, 5.0 Mbps, 7.0 Mbps である。各画質における動画データの分割数は、チャンネル数と同じとなる。

図 10 より、提案手法および単純手法について、チャンネル数の増加にともない待ち時間は短くなる。チャンネル数が増加すると、 C_1 で配信する各画質の動画データのうち最初のセグメントのデータサイズが小さくなり、受信時間が短くなるため、待ち時間は短縮する。また、提案手法では中画質および高画質について動画データの差分のみを配信することで、単純手法に比べて待ち時間を短縮できる。例えば、チャンネル数が 6 の場合における待ち時間は、提案手法で約 84.5 秒、単純手法では約 387.6 秒となり、提案手法は単純手法に比べて待ち時間を 78.2 % 短縮できる。

7. おわりに

本研究では、再生中の動画品質の変更を考慮したスケジューリング手法を提案した。提案手法では、サーバは、クライアントが使用できる帯域幅を考慮して、再生レートが異なる複数種類の動画データを複数の部分に分割して複数のチャンネルで繰り返し配信することで、動画品質の変更時に発生する待ち時間を短縮する。評価では、提案手法と単純手法に対して、使用できる帯域幅および動画の再生時間の変化に応じた待ち時間を比較した。評価の結果、提案手法の待ち時間は、単純手法に比べて短縮できることを確認した。

今後の予定として、提案手法を分割放送型配信システム *TeleCaS* に導入し、スケジューリングの有用性を評価する。

謝辞 本研究は、文部科学省科学研究費補助金（基盤研

究 (B) 課題番号: 21H03429, 22H03587), 日本学術振興会二国間交流事業 (課題番号: JPJSBP120229932), および (公財) 日揮・実吉奨学会の研究助成によるものである。ここに記して謝意を表す。

参考文献

- [1] 総務省: 情報通信白書令和 3 年版 (オンライン), 入手先<<https://www.soumu.go.jp/johotsusintokei/whitepaper/ja/r03/>> (参照 2022-02-01).
- [2] Gotoh Y., Yoshihisa T., Taniguchi H. and Kanazawa M.: A Scheduling Method for Heterogeneous Clients on Media Data Broadcasting, International Journal of Pervasive Computing and Communications (IJPCC), Vol.9, Issue.2, pp.98-114 (2013).
- [3] 後藤佑介, 義久智樹, 金澤正憲: 帯域幅を考慮した選択型コンテンツの放送型配信における待ち時間短縮手法, 情報処理学会論文誌, Vol.49, No.5, pp.1692-1701 (2008).
- [4] RFC8216 - HTTP Live Streaming: IETF (online), available from <<https://www.ietf.org/rfc/rfc8216.txt>> (accessed 2022-02-01).
- [5] Information Technology - Dynamic Adaptive Streaming over HTTP (DASH) - Part 1: Media Presentation Description and Segment Formats: ISO (online), available from <<https://www.iso.org/standard/75485.html>> (accessed 2022-02-01).
- [6] NHK: NHK オンデマンド, NHK (オンライン), 入手先<<https://www.nhk-ondemand.jp/>> (参照 2022-02-01).
- [7] YouTube: YouTube, Youtube (オンライン), 入手先<<https://www.youtube.com>> (参照 2022-02-01).
- [8] (一社) 衛星放送協会: 衛星放送協会のしくみ, 衛星放送協会 (オンライン), 入手先<<https://www.eiseihoso.org/guide/howto.html>> (参照 2022-02-01).
- [9] Juhn, L. and Tseng, L.: Fast Data Broadcasting and Receiving Scheme for Popular Video Service, IEEE Trans. on Broadcasting, Vol.44, No.1, pp.100-105 (1998).
- [10] Janakiraman, R. and Waldvogel M.: Fuzzycast: Efficient Video-on-Demand over Multicast, Proceedings of IEEE INFOCOM, pp.920-929 (2002).
- [11] Yoshihisa, T., Tsukamoto, M., and Nishio, S.: A Broadcasting Scheme for Continuous Media Data with Restrictions in Data Division, Proceedings of IPSJ Int. Conf. on Mobile Computing and Ubiquitous Networking (ICMU '05), pp. 90-95 (2005).
- [12] 安里諒, 青木輝勝, 沼澤潤二: 受信エラー耐性を有する分割放送型配信手法に関する一検討, 電子情報通信学会技術研究報告, Vol.109, No.449, pp.31-36 (2010).
- [13] 後藤佑介, 谷口秀夫: 連続メディアデータの放送型配信における誤り耐性を考慮したスケジューリング手法, 情報処理学会研究報告, Vol.2012-DPS-152, No.24, pp.1-8 (2012).
- [14] Yang, Z.Y., Chen, Y.M. and Tseng, L.M.: A Seamless Broadcasting Scheme with Live Video Support, International Journal of Digital Multimedia Broadcasting, Vol.2012, pp.1-8 (2012).
- [15] Wang, X., Cai, G., and Men, J.: Wrap Harmonic Broadcasting and Receiving Scheme for Popular Video Service, IEEE Trans. on Broadcasting, Vol.66, Issue 1, pp.78-87 (2020).
- [16] Kanamoto S. and Gotoh Y.: A Division-based Broadcasting System Considering Dynamic Updates of Delivery Schedule, Proceedings of 17th International Conference on Advances in Mobile Computing and Multimedia (MoMM 2019), pp.175-184 (2019).
- [17] 木村明寛, 後藤佑介, 谷口秀夫: 動画データを分割配信するシステムの実現と評価, 電子情報通信学会論文誌 B, Vol.J96-B, No.10, pp.1217-1225 (2013).
- [18] GYAO!: ヤフー (online), available from <<https://gyao.yahoo.co.jp/>> (accessed 2022-02-01).
- [19] HTTP Live Streaming: IETF (online), available from <<https://tools.ietf.org/html/rfc8216>> (accessed 2022-02-01).
- [20] Information Technology - Dynamic Adaptive Streaming over HTTP (DASH) - Part 1: Media Presentation Description and Segment Formats: ISO (online), available from <<https://www.iso.org/standard/65274.html>> (accessed 2022-02-01).
- [21] Spiteri K., Urgaonkar R., and Sitaraman R.K.: BOLA: Near-Optimal Bitrate Adaptation for Online Videos, IEEE/ACM Trans. on Networking, Vol.28, Issue 4, pp.1698-1711 (2020).