

再生中の動画品質を変更可能な 分割放送型配信システムの提案

大石 貴之¹ 後藤 佑介¹

概要: 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に待ち時間が発生する。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システム *TeleCaS* では、スケジューリング手法を適用することで、待ち時間を効率的に短縮できる。しかし、サーバは、使用できる帯域幅の変化に応じて配信する動画のビットレートを動的に変更できないため、クライアントは、動画データの再生中に中断が発生する可能性がある。本研究では、再生中の動画品質を変更可能な分割放送型配信システムを提案する。提案システムでは、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ビットレートを変更可能なスケジューリング手法を作成し、*TeleCaS* 上で設計、実装する。評価では、帯域幅に応じた再生待ち時間、再生中断時間、および動画の平均ビットレートの変化を分析し、提案システムの有用性を確認する。

1. はじめに

近年、全世界のビデオトラフィックが急増 [1] しており、ネットワーク環境の変化に適応した動画配信システムが必要となっている。クライアントの要求に対して動画データをユニキャストで送信する Video On Demand (VOD) では、クライアント数が増加するとサーバが配信に必要な帯域幅は増加する。一方で、動画データを複数のクライアントにマルチキャストで配信する放送型配信では、クライアント数が増加しても、サーバは一定の帯域幅で配信できる。しかし、放送型配信では、クライアントの再生要求契機を考慮せずに動画を繰り返し配信するため、クライアントは受信要求から再生開始までの間で待ち時間が発生する。

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], [4], [5]。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。我々の研究グループでは、分割放送型配信のスケジューリング手法を適用可能なシステム *TeleCaS* [6]

を提案し、スケジューリング手法に対する評価を行っている。

分割放送型配信の問題点として、クライアントの再生中に、次に再生するセグメントを受信できていない場合、このセグメントの受信が完了するまでの間でデータの再生が中断する時間 (以下、再生中断時間) が発生する。再生中断時間を短縮するため、動画データの品質を表すビットレートを下げ、動画のデータサイズを減少することで、動画データの配信時間を短縮する方法が考えられる。しかし、動画データの品質を下げると、動画視聴におけるクライアントの満足度が低下する。

本研究では、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品質を変更可能なスケジューリング手法を提案し、分割放送型配信システム上で設計、実装する。提案システムでは、サーバは使用できる帯域幅をもとに、クライアントの再生中断が発生せず、かつ再生中の動画データの平均ビットレートができるだけ高くなるように、配信する各セグメントのビットレートを決定する。また、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る手法である Adaptive Bit Rate (以下、ABR) を用いることで、クライアントは再生を中断せずに、できるだけ高品質で動画データを再生できる。

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. 動画データの配信方式

2.1 オンデマンド型配信

オンデマンド型配信は、クライアントの要求を受信したサーバがユニキャストで動画データを送信する配信方式であり、YouTube [7] や GYAO! [8] に代表される Video On Demand (VOD) サービスで用いられる。オンデマンド型配信では、クライアントからの要求に応じて動画を送信するため、クライアントは、動画データの受信を要求した後、すぐに視聴を開始できる。しかし、すべてのクライアントに対してユニキャストで動画データを送信するため、動画データを要求するクライアント数が増加すると、サーバの処理負荷の増加により各クライアントに対する送信速度が低下し、動画再生中に中断時間が発生する可能性がある。

2.2 放送型配信

放送型配信は、サーバがクライアントの受信要求契機に関係なく、動画データをマルチキャストで配信する方式である。ひかり TV [9] に代表される IP 放送サービスやビデオ会議において、同時に多くのクライアントが同じ映像を視聴するサービスで用いられる。放送型配信では、すべてのクライアントに対してマルチキャストで動画データを配信するため、動画データを視聴するクライアント数が増加しても、サーバの処理負荷は増加しない。しかし、動画データの配信スケジュールはサーバ側で配信前に決定するため、クライアントは見たい動画データの配信スケジュールを考慮して再生を開始する必要がある。

2.3 Adaptive Bit Rate を用いた配信

Adaptive Bit Rate (ABR) による配信では、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送信する。ABR は、YouTube [7] や Gyao! [8] といった 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視聴が可能となる。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、視聴中に動画の品質を変更できる。

ABR による配信の様子を図 1 に示す。図 1 の例では、3 台のクライアント A, B, C がサーバに同じ動画の視聴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、720p (1280 × 720 pixel), 480p (854 × 480 pixel), および 240p (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送信する。

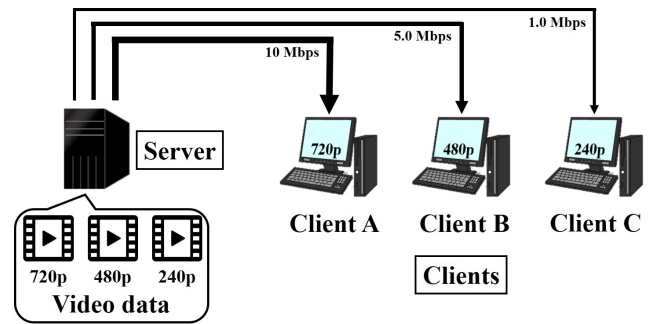


図 1 ABR を用いた動画配信の例

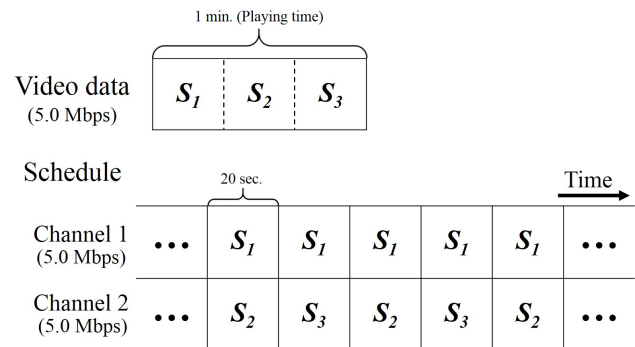


図 2 FB 法による配信スケジュールの例

3. 関連研究

3.1 分割放送型配信

放送型配信で VOD に近いシステムを実現する手法として、分割放送型配信が提案されている。分割放送型配信では、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配信する。また、動画データを繰り返し配信するため、クライアントは一定時間待つことで、動画を最初から再生できる。この再生開始までにかかる時間（以下、再生待ち時間）を短縮するため、動画の分割比率や配信順序を決定するスケジューリング手法がいくつか提案されている [2], [3], [4], [5]。

分割放送型配信について、スケジューリング手法である Fast Broadcasting (FB) 法 [3] を用いて説明する。FB 法では、帯域幅を k 個のチャンネルに均等に分け、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 番目のチャンネルで連続した 2^{i-1} 個のセグメントを繰り返し配信する。

FB 法による配信スケジュールの例を図 2 に示す。サーバは動画のビットレートと等しい帯域幅の 2 つのチャンネルを使用し、各 20 秒のセグメントに 3 分割された合計 1 分間の動画を配信する。各チャンネルでは、1 個のセグメントをセグメントの再生時間と等しい 20 秒で配信する。これにより、クライアントは各セグメントの再生を終了するまでに次に再生するセグメントの受信を完了でき、再生中に中断が発生することなく動画を視聴できる。しかし、サーバで使用できる帯域幅が不足し、FB 法による配信が必要

な帯域幅を割り当てられない場合、クライアントはセグメントの再生終了までの間に次のセグメントの受信が終了せず、再生中断時間が発生する可能性がある。

3.2 Adaptive Bit Rate

Adaptive Bit Rate (ABR) の実現例として、HTTP LIVE Streaming (HLS) [10] や MPEG-Dynamic Adaptive Streaming over HTTP (MPEG-DASH) [11] が挙げられる。MPEG-DASH による配信では、サーバは、ビット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ビット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と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。

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ビット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 [12] が挙げられる。しかし、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しないため、クライアントが動画のビットレートを制御する手法を適用できない。

4. 分割放送型配信システム

4.1 TeleCaS

分割放送型配信におけるスケジューリング手法を実際のネットワーク環境で評価するため、分割放送型配信システム Telecommunication and BroadCasting System (*TeleCaS*) [6] が提案されている。*TeleCaS* は、分割放送型配信で配信するチャンネル数、各チャンネルの帯域幅、および各セグメントのデータサイズと配信順序を自由に設定でき、多くのスケジューリング手法を利用できる。

4.2 TeleCaS における課題

TeleCaS では 2 つの課題が挙げられる。一つ目は、現状の *TeleCaS* においてサーバが配信する動画データのファイル形式が Flash Video であり、再生時に Adobe Flash Player [13] を用いる点である。Flash Video は、Flash Player を用いて Web ブラウザ上で再生でき、多くの動画配信サービスで利用されている。しかし、近年はセキュリティ上の脆弱性により Flash Player の利用は推奨されおらず、Adobe Systems による更新および配布が終了する予定である [14]。このため、*TeleCaS* で配信する動画のファイル形式を変更する必要がある。

二つ目は、配信する動画データを動的に変更できない点である。3.1 節で述べたように、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、動画が再生中に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ビット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を行うため、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ビットレートの動画をクライアントに配信できない。

5. 設計

5.1 概要

本研究では、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ビットレートを変更可能な分割放送型配信システムを提案する。提案システムでは、4.2 節で述べた *TeleCaS* における課題を解決する機能を設計、実装する。また、提案システムに導入可能なスケジューリング手法を提案する。

5.2 MP4 形式の動画配信

分割放送型配信で用いる動画データのファイル形式について、これまでの *TeleCaS* で使用した Flash Video から、MP4 形式に変更する。MP4 は、国際標準化機構 (ISO) により標準化されたファイルフォーマット [15] であり、動画配信で広く利用されている形式である。

5.3 配信動画におけるビットレートの動的変更

本研究では、再生中断の発生を抑えるため、サーバが使用できる帯域幅に応じて最適なビットレートの動画を配信する。また、異なるビットレートの動画ごとに作成するセグメントについて、1 種類の動画データとしてクライアントが受信することで、再生時にビットレートを変更できる機能を実現する。

5.4 スケジューリング手法

5.4.1 概要

本研究では、配信する動画データのビットレートを変更可能なスケジューリング手法を提案する。ビットレートの変更方法として、はじめに、サーバはビットレートが異なる複数種類の動画データを設定する。次に、これらの動画データを複数のセグメントにそれぞれ分割し、チャンネルごとに異なるビットレートの動画を配信するためにスケジューリングを行う。最後に、サーバは、使用できる帯域幅に応じて複数のチャンネルを選択してセグメントを配信する。

本研究では、上記の手順をもとに、ビットレートを変更するスケジューリング手法を 2 種類提案し、評価する。それぞれの手法について、以下で順番に説明する。

5.4.2 F-SHB 法

Fast Broadcasting based Synchronous Harmonic Broadcasting (以下, F-SHB 法) は, FB 法 [3] に基づいて, 再生中にビットレートを変更できるように改良したスケジューリング手法である.

F-SHB 法のスケジューリング手順について説明する. サーバが設定した n 個の動画について, ビットレートが低い順番に V_1, \dots, V_n とし, V_i ($1 \leq i \leq n$) に対応するビットレートを r_i とする.

はじめに, k ($k \geq 1$) 個のチャンネルを用いて, FB 法のスケジューリングと同様に, V_i を $2^k - 1$ 個のセグメント $S_i^1, \dots, S_i^{2^k-1}$ に分割し, 各チャンネルにスケジューリングする. このとき, V_i を配信する k 個のチャンネルについて, 配信するセグメント数が少ないチャンネルから順番に, C_1^i, \dots, C_k^i とする. また, 配信するセグメントの個数が等しいチャンネルの集合として, H_m ($1 \leq m \leq k$) を作成する. このとき, $H_m = (C_m^1, \dots, C_m^n)$ とする.

次に, 再生中断ができるだけ発生しない配信において, ビットレートが一番高い動画を求める. V_i を FB 法で配信するために必要な帯域幅の合計として, $B_i = kr_i$ とする. このとき, サーバが使用できる帯域幅 B_{total} で配信でき, かつビットレートが一番高い動画を V_j とすると, j は式 (1) で表される. 一方で, $i = \phi$ の場合, サーバは, 再生中断が発生しないように配信できる動画が無い場合, 使用できる帯域幅 B_{total} を用いて, ビットレートが一番低い動画を FB 法で配信する.

$$j = \max_i \{i \mid B_i \leq B_{total}, 1 \leq i \leq n\} \quad (1)$$

$j \neq n$ である場合, V_j を配信予定のチャンネルは, 配信する動画を V_j から V_{j+1} に変更するため, サーバは $r_{j+1} - r_j$ の帯域幅が追加が必要となる. また, FB 法で V_j を配信する場合, サーバは $B_{total} - B_j$ の帯域幅を追加で利用できる. このため, V_{j+1} に変更できるチャンネル数 M は, 式 (2) で求められる.

$$M = \lfloor \frac{B_{total} - B_j}{r_{j+1} - r_j} \rfloor \quad (2)$$

さらに, チャンネルの集合ごとに, サーバが配信するチャンネルを一つずつ選択する. 式 (1), 式 (2) で求めた j, M を用いて, 配信する k 個のチャンネルを $(C_1^j, \dots, C_{k-M}^j, C_{k-M+1}^{j+1}, \dots, C_k^{j+1})$ とする. このとき, 動画データ全体のビットレートの平均を高くするため, 配信するセグメント数が多いチャンネルを優先して, V_{j+1} を配信するチャンネルに変更する.

最後に, 配信する各チャンネルに帯域幅を割り当てる. H_m から選択したチャンネルに割り当てる帯域幅を b_m とする. 各チャンネルのセグメントの配信時間を同じにするため, 各チャンネルの帯域幅は, 式 (3) で求められる.

C_1^1 (1.0 Mbps)	...	S_1^1	S_1^1	S_1^1	S_1^1	S_1^1	S_1^1	...
C_2^1 (2.0 Mbps)	...	S_2^1	S_2^1	S_2^1	S_2^1	S_2^1	S_2^1	...
C_3^1 (3.0 Mbps)	...	S_3^1	S_3^1	S_3^1	S_3^1	S_3^1	S_3^1	...
C_1^2 (1.0 Mbps)	...	S_1^2	S_1^3	S_1^2	S_1^3	S_1^2	S_1^3	...
C_2^2 (2.0 Mbps)	...	S_2^2	S_2^3	S_2^2	S_2^3	S_2^2	S_2^3	...
C_3^2 (3.0 Mbps)	...	S_3^2	S_3^3	S_3^2	S_3^3	S_3^2	S_3^3	...

図 3 F-SHB 法における配信スケジュール例

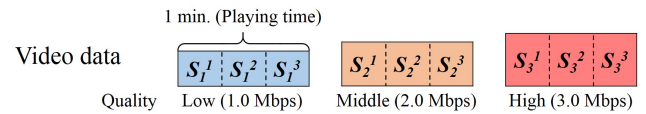


図 4 サーバが分割配信する動画データの構成

$$b_m = \begin{cases} \lfloor B_{total} \frac{B_j}{(k-M)B_j + MB_{j+1}} \rfloor & (1 \leq m \leq k-M) \\ \lfloor B_{total} \frac{B_{j+1}}{(k-M)B_j + MB_{j+1}} \rfloor & (k-M < m \leq k) \end{cases} \quad (3)$$

F-SHB 法における配信スケジュールの例を図 3 に示す. サーバは, 図 4 に示すように, ビットレートが 1.0 Mbps, 2.0 Mbps, および 3.0 Mbps となる 3 種類の動画データ V_1, V_2, V_3 をそれぞれもつ. サーバは, 各動画を 20 秒ずつの 3 個のセグメントに分割して配信する. また, これらの動画データを配信する 6 チャンネルを 3 チャンネルずつ 2 種類の集合に分け, サーバが使用できる帯域幅に応じて, 集合ごとに 1 チャンネルを選択する. 例えば, 図 3 の配信スケジュールにおいて, サーバは, 使用できる帯域幅が 3.0 Mbps の場合, C_1^1 で 1.0 Mbps, C_2^2 で 2.0 Mbps を用いて配信する. また, 使用できる帯域幅が 5.5 Mbps の場合, 式 (3) に基づき, C_1^2 で $5.5 \times \frac{2+3}{2} = 2.2$ Mbps, C_3^3 で $5.5 \times \frac{2.0+3.0}{3.0} = 3.3$ Mbps を用いて配信する.

5.5 F-AHB 法

Fast Broadcasting based Asynchronous Harmonic Broadcasting (以下, F-AHB 法) は, FB 法に基づいて, 先頭のセグメントの配信周期ができるだけ短くなるようにスケジューリングする手法である.

F-AHB 法では, はじめは F-SHB 法と同様に, k 個のチャンネルを作成する. この後, 先頭以外のセグメントを配信する $k-1$ 個のチャンネルに必要最低限の帯域幅を割り当てた上で, 先頭のセグメントを配信するチャンネルに残りの帯域幅をすべて割り当てる. また, 先頭のセグメントの再生終了時に次のセグメントを途切れなく再生開始するために必要な帯域幅は, 動画のビットレートの 2 倍となる. このた

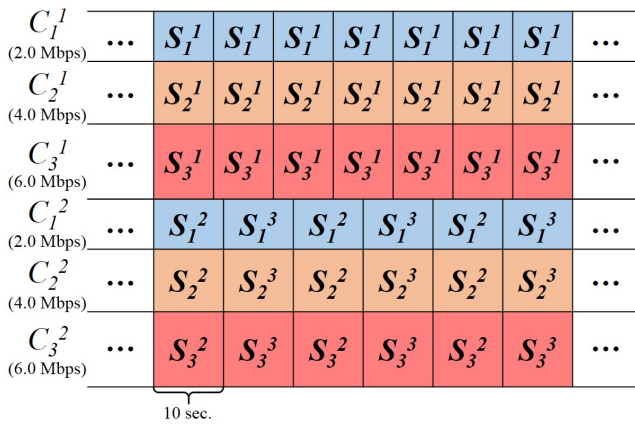


図 5 F-AHB 法における配信スケジュール例

め、 i 番目の動画を配信するために必要な帯域幅の合計は、 $B_i = 2kr_i$ となる。以上の条件で、式 (1)、式 (2) を用いて、配信するチャンネルを決定する。各チャンネルの帯域幅は、式 (4) で求められる。

$$b_m = \begin{cases} B_{total} - \{(k - M - 1) \frac{B_j}{k} + M \frac{B_{j+1}}{k}\} & (m = 1) \\ \frac{B_j}{k} & (1 < m \leq k - M) \\ \frac{B_{j+1}}{k} & (k - M < m \leq k) \end{cases} \quad (4)$$

F-AHB 法における配信スケジュールの例を図 5 に示す。サーバが配信する動画データは、図 4 で述べた 3 種類とする。動画の先頭部分のセグメントを配信しないチャンネルでは、セグメントを再生時間の半分である 10 秒で配信する。一方で、動画の先頭部分のセグメントを配信するチャンネルでは、セグメントを 10 秒以下で配信する。例えば、図 5 の配信スケジュールにおいて、サーバは、使用できる帯域が 11 Mbps の場合、式 (4) に基づき、 C_2^1 で 5.0 Mbps、 C_3^1 で 6.0 Mbps を用いて配信する。このとき、動画の先頭部分のセグメントは 8.0 秒で配信され、再生待ち時間を短縮できる。

6. 実装

提案した *TeleCaS* のシステム構成を図 6 に示す。提案システムでは、サーバは、Fragmented MP4 (fMP4) 形式でビットレートが異なる複数の動画ファイルを用いる。fMP4 は、動画データを分割した複数のセグメントをファイル内で保存する MP4 ファイルの形式であり、HLS [10] や MPEG-DASH [11] で利用される。fMP4 を配信に利用することで、ビットレートが異なる動画ファイルのセグメントを組み合わせ再生できる。また、サーバは、スケジュールファイルで設定したチャンネルを配信にすべて利用せず、スケジューリング手法に基づいて決定したチャンネルのみを用いて、User Datagram Protocol (UDP) でマルチキャストする。クライアントは、受信が完了した動画データのセ

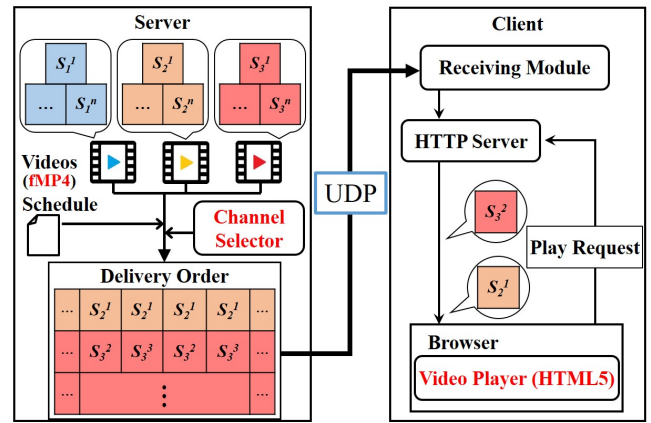


図 6 提案した *TeleCaS* のシステム構成

表 1 計算機の性能

Server	CPU	Intel®Pentium(R) CPU G4400 (3.30 GHz) × 2
	Memory	7.7 GiBytes
	OS	Ubuntu 18.04.1 LTS
	NIC	RTL8111/8168/8411
Client	CPU	Intel®Pentium(R) CPU G4400 (3.30 GHz) × 2
	Memory	7.7 GiBytes
	OS	Ubuntu 18.04.1 LTS
	NIC	RTL8111/8168/8411

表 2 評価に用いる動画データの品質

Quality	Image Quality	Bit Rate
High	720p	665 kbps
Middle	480p	1687 kbps
Low	240p	2970 kbps

グメントを HTTP サーバモジュールに送り、HTTP サーバモジュールはセグメントをブラウザ上の Video Player に送信する。このとき、ブラウザ上で動作可能な HTML5 で実装したプレイヤーに、World Wide Web Consortium (W3C) が標準化した Media Source Extensions [16] を用いることで、複数のビットレートによるセグメントを組み合わせ再生を実現する。

7. 評価

7.1 評価環境

TeleCaS を導入したサーバ計算機とクライアント計算機を Gigabit Ethernet で有線接続し、評価を行った。評価に用いた計算機の性能を表 1 に示す。また、サーバ計算機では、高画質、中画質、および低画質の 3 種類の動画データを用いる。3 種類の動画データの品質を表 2 に示す。

7.2 想定環境

評価における想定環境について箇条書きで以下に記す。

- サーバは、ビットレートが異なる 3 種類の動画データのセグメントを配信する。

- クライアントは動画の再生を開始すると、最後まで再生する。
- クライアントは、サーバで使用できる帯域幅に応じて動画のビットレートを再生中に変更できる。
- クライアントの再生中は、サーバで使用できる帯域幅は変動しない。
- クライアントは、配信する動画の受信で十分な帯域幅を使用できる。
- サーバとクライアントとの間の通信路でパケットロスが発生しない。

7.3 ビットレートの影響

配信動画のビットレートを変更可能な F-SHB 法と F-AHB 法、および各動画のビットレートを変更せずに配信する FB 法 [3] の 3 種類を用いて評価する。評価項目は、サーバで使用できる帯域幅に対する平均再生待ち時間、平均再生中断時間、および動画の平均ビットレートの 3 種類である。また、クライアントは、受信が完了したセグメントのみ再生できるため、再生待ち時間は、クライアントが受信を要求してから先頭セグメントの受信が完了するまでの時間とする。サーバは、1 分間の動画を 2 チャンネルで配信する。

はじめに、帯域幅の変化に応じた平均再生待ち時間を図 7 に示す。横軸はサーバが使用できる帯域幅、縦軸はクライアント計算機で再生待ち時間を 5 回測定した平均値である。図 7 より、F-SHB 法および F-AHB 法について、使用できる帯域幅が 1.0 Mbps および 2.0 Mbps の場合、再生待ち時間は FB (240p) に近い値となる。一方で、使用できる帯域幅が 13 Mbps から 15 Mbps の場合、再生待ち時間は FB (720p) に近い値となる。また、F-AHB 法における再生待ち時間は、F-SHB 法と比べて短くなる。

FB (720p) と FB (240p) では、サーバで使用できる帯域幅の増加に応じて、再生待ち時間の差は小さくなる。FB (240p) では、使用できる帯域幅が 9.0 Mbps を上回ると、再生待ち時間はほとんど変化しない。同様に、FB (480p) では、使用できる帯域幅が 11 Mbps を上回ると、再生待ち時間はほとんど変化しない。

次に、帯域幅の変化に応じた平均再生中断時間を図 8 に示す。横軸はサーバが使用できる帯域幅、縦軸はクライアント計算機で動画再生中に発生する中断時間を 5 回測定した平均値である。図 8 より、F-SHB 法および F-AHB 法では、使用できる帯域幅が 1.0 Mbps の場合において、約 2.0 秒の再生中断が発生する。一方で、FB (720p) において帯域幅が 1.0 Mbps から 5.0 Mbps の場合、および FB (480p) において帯域幅が 1.0 Mbps、2.0 Mbps の場合で、再生中断時間がそれぞれ発生する。例えば、帯域幅が 1.0 Mbps の場合、FB (720p) では約 89.0 秒、FB (480p) では約 38.1 秒となる。

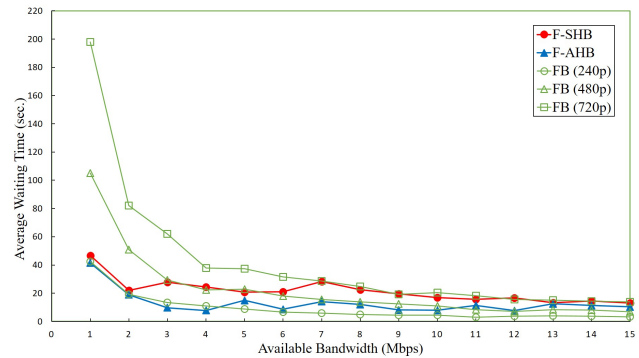


図 7 手法ごとの帯域幅に対する平均再生待ち時間

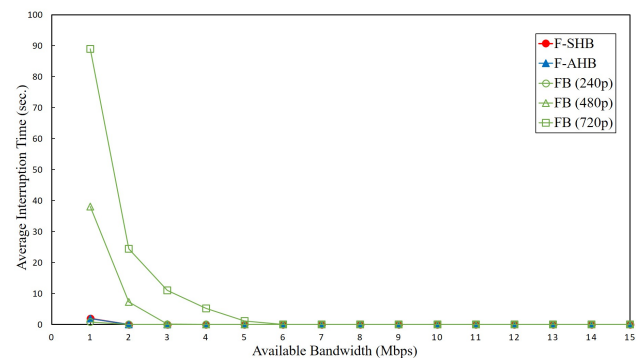


図 8 手法ごとの帯域幅に対する平均再生中断時間

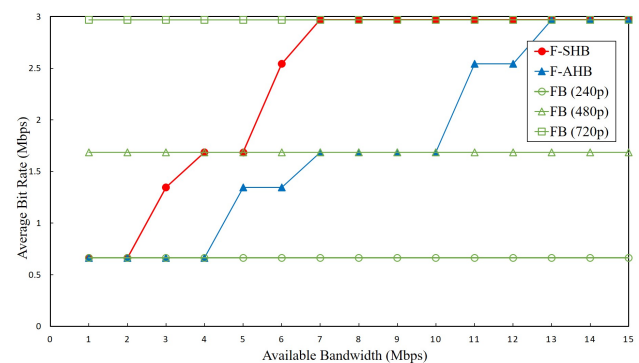


図 9 手法ごとの帯域幅に対する動画データの平均ビットレート

最後に、帯域幅の変化に応じた動画データの平均ビットレートを図 9 に示す。横軸はサーバが使用できる帯域幅、縦軸はクライアント計算機で再生される動画データの平均ビットレートである。図 9 より、帯域幅が小さい場合において、F-SHB 法におけるビットレートは、F-AHB 法に比べて高い。また、F-SHB 法において、使用できる帯域幅が 7.0 Mbps 以上の場合、ビットレートは約 3.0 Mbps と一番高い。一方で、F-AHB 法において、使用できる帯域幅が 13 Mbps 以上の場合、ビットレートは約 3.0 Mbps と一番高い。

7.4 配信チャンネル数の影響

F-SHB 法および F-AHB 法において、配信チャンネル数の変化に応じた平均再生待ち時間、平均再生中断時間、およ

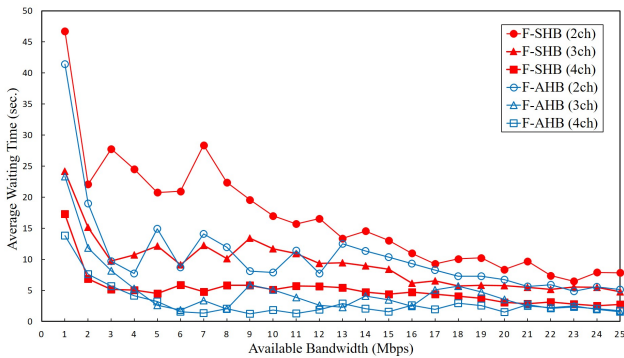


図 10 配信チャンネル数ごとの帯域幅に対する平均再生待ち時間

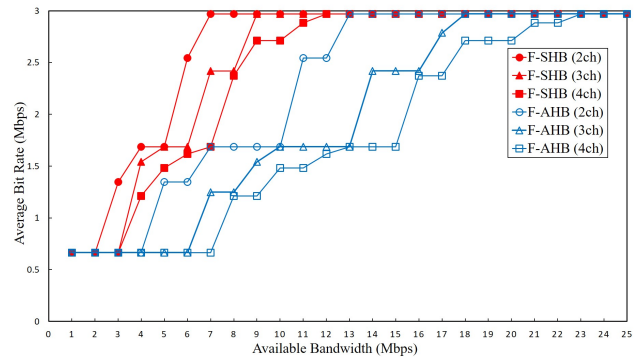


図 12 配信チャンネル数ごとの帯域幅に対する動画データの平均ビットレート

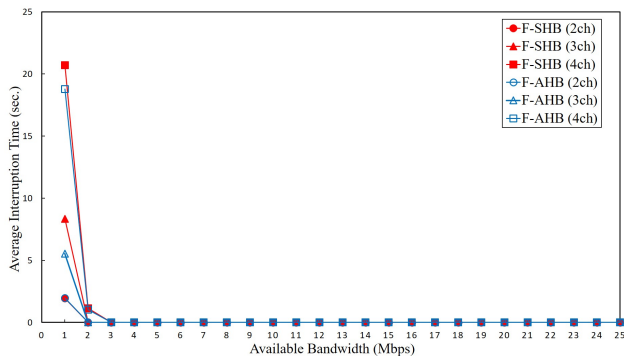


図 11 配信チャンネル数ごとの帯域幅に対する平均再生中断時間

び動画の平均ビットレートの3種類を評価する。また、比較項目として、サーバが1分間の動画を配信するために用いるチャンネルの数が2, 3, 4の3種類の場合について、手法ごとにそれぞれ設定する。

はじめに、使用できる帯域幅に応じた平均再生待ち時間を図 10 に示す。横軸はサーバが使用できる帯域幅、縦軸はクライアント計算機で再生待ち時間を5回測定した平均値である。図 10 より、配信チャンネル数が等しい場合、F-AHB 法の再生待ち時間は、F-SHB 法に比べて短い。また、両方の手法において、配信チャンネル数が多いほど再生待ち時間は短くなる。例えば、帯域幅が 25 Mbps の場合、F-AHB 法で配信チャンネル数が 4 の場合の再生待ち時間は約 2.7 秒であり、一番短い。

次に、使用できる帯域幅に応じた平均再生中断時間を図 11 に示す。横軸はサーバが使用できる帯域幅、縦軸はクライアント計算機で動画再生中に発生する中断時間を5回測定した平均値である。図 11 より、両方の手法において、配信チャンネル数が 2 および 3 の場合、帯域幅が 1.0 Mbps のときに再生中断が発生する。また、配信チャンネルが 4 の場合、帯域幅が 1.0 Mbps および 2.0 Mbps のときに再生中断が発生する。

最後に、帯域幅に応じた動画データの平均ビットレートを図 12 に示す。横軸はサーバが使用できる帯域幅、縦軸はクライアント計算機で再生される動画データの平均ビットレートである。図 12 より、両方の手法において、チャ

ネル数が少ないほど帯域幅に応じた動画のビットレートが高い。また、F-SHB 法で帯域幅に応じた動画データのビットレートは、F-AHB 法と比べて高い。F-SHB 法および F-AHB 法においてビットレートがもっとも低い場合はチャンネル数が 4 であり、F-AHB 法でチャンネル数が 2 の場合に比べて高い。また、F-SHB 法でチャンネル数が 2 の場合において、使用できる帯域幅が 7.0 Mbps 以上のときにビットレートが一番高い。一方で、F-AHB 法でチャンネル数が 4 の場合において、使用できる帯域幅が 23 Mbps 以上のときにビットレートが一番高い。

8. 考察

8.1 ビットレートの影響

7.3 節の評価結果より、帯域幅が 1.0 Mbps から 3.0 Mbps の場合、FB (720p) および FB (480p) では再生待ち時間が長くなり、再生中断時間が発生した。一方で、帯域幅が 12 Mbps から 15 Mbps の場合、FB (480p) および FB (240p) では、使用できる帯域幅に関係なく再生待ち時間は短縮しなかった。このため、使用できる帯域幅が大きい場合、サーバはビットレートがより高い動画データを配信できる。また、F-SHB 法および F-AHB 法の両方について、使用できる帯域幅が大きいほど動画データの平均ビットレートは高くなった。

F-SHB 法と F-AHB 法の比較について、F-AHB 法において使用できる帯域幅に応じた平均再生待ち時間は、F-SHB 法に比べて短縮した。これは、再生を開始するために必要となる先頭のセグメントを配信するチャンネルに帯域幅が多く割り当てられているためである。また、F-SHB 法において、使用できる帯域幅に応じた動画データの平均ビットレートは、F-AHB 法に比べて高かった。これは、F-SHB 法において各チャンネルで必要となる帯域幅が、F-AHB 法に比べて半分となり、ビットレートがより高い動画データを配信するチャンネルを利用できるためである。

8.2 チャネル数の影響

7.4節の評価結果より、チャネル数が多いほど再生待ち時間が短くなる。これは、チャネル数が多いほど動画の分割数が多くなり、セグメントあたりのデータサイズが小さくなることで、先頭のセグメントの受信にかかる時間が短くなるためである。また、帯域幅が小さい場合において、チャネル数が多いほど再生中断時間が長大化した。配信に利用するチャネル数が増加すると、再生中断が発生しない配信で必要となる帯域幅の合計が増加し、低いビットレートの動画配信において再生中断が発生する。

F-SHB法でチャネル数が2の場合、帯域幅が7.0 Mbpsを上回ると、チャネル数に関係なく、動画データのビットレートが一番高い。一方で、帯域幅が7.0 Mbps以上において、再生待ち時間はほとんど短縮しない。また、F-AHB法においてチャネル数が2で利用できる帯域幅が13 Mbpsの場合におけるビットレートは、先ほど述べたF-SHB法においてチャネル数が2で利用できる帯域幅が7.0 Mbpsの場合と等しくなる。一方で、F-SHB法における再生待ち時間は、F-AHB法に比べて長大化する。このため、利用できる帯域幅が大きい場合、F-AHB法による配信が良いことが分かる。

9. おわりに

本研究では、サーバが利用できる帯域幅に応じて配信する動画データのビットレートを変更可能な分割放送型配信システムを提案した。提案システムでは、利用できる帯域幅が小さい場合は低いビットレート、大きい場合は高いビットレートの動画データのセグメントをそれぞれ配信することで、クライアントは動画をできるだけ中断せずに再生できる。評価では、2種類の提案手法と既存のスケジューリング手法について、利用できる帯域幅および配信チャネル数の変化に応じて、再生待ち時間、再生中断時間、および配信動画のビットレートの3項目で比較した。評価の結果、提案手法では、利用できる帯域幅が小さい場合における再生待ち時間は既存手法に比べて短縮した。また、利用できる帯域幅が大きい場合、既存手法に比べて、より高いビットレートの動画データを配信できる。

今後の予定として、FB法以外のスケジューリング手法に基づくビットレートを変更可能な手法の提案、およびブラウザ上においてMP4ファイルのセグメントをより細かく分割したサブセグメント単位で再生が可能なビデオプレイヤーの実装を行う。

謝辞 本研究は、文部科学省科学研究費補助金（基盤研究（C））（課題番号：18K11265）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] Cisco: Cisco Visual Networking Index: Forecast and Trends, 2017–2022 (online), available from < <https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.html> > (accessed 2019-02-04).
- [2] S. Viswanathan and T. Imilelinski: Pyramid Broadcasting for Video on Demand Service, SPIE Multimedia Computing and Networking Conf., pp.66-77 (1995).
- [3] L. Juhn and L. Tseng: Fast Data Broadcasting and Receiving Scheme for Popular Video Service, IEEE Trans. Broadcasting, Vol.44, No.1, pp.100-105 (1998).
- [4] Y. Zhao, D.L. Eager, M.K. Vernon: Scalable On-Demand Streaming of Nonlinear Media, IEEE/ACM Trans. Networking, Vol.15, No.5, pp.1149-1162 (2007).
- [5] H. Feng, Z. Chen, H. Liu: Design and Optimization of VoD Schemes with Client Caching in Wireless Multicast Networks, IEEE Trans. Vehicular Technology, Vol.67, No.1, pp.765-780 (2018).
- [6] 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).
- [7] YouTube: Google (online), available from < <https://www.youtube.com/> > (accessed 2019-02-04).
- [8] GYAO!: ヤフー (online), available from < <https://gyao.yahoo.co.jp/> > (accessed 2019-02-04).
- [9] ひかり TV: NTT ぷらら (online), available from < <https://www.hikaritv.net/> > (accessed 2019-02-04).
- [10] HTTP Live Streaming: IETF (online), available from < <https://tools.ietf.org/html/rfc8216> > (accessed 2019-02-04).
- [11] 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 2019-02-04).
- [12] K. Spiteri, R. Uргаonkar, R.K. Sitaraman: BOLA: Near-Optimal Bitrate Adaptation for Online Videos, IEEE INFOCOM (2016).
- [13] Adobe Flash Player: Adobe Systems (online), available from < <https://www.adobe.com/jp/products/flashplayer.html> > (accessed 2019-02-04).
- [14] Flash & The Future of Interactive Content: Adobe Systems (online), available from < <https://theblog.adobe.com/adobe-flash-update/> > (accessed 2019-02-04).
- [15] Information Technology - Coding of Audio-visual Objects - Part 14: MP4 File Format: ISO (online), available from < <https://www.iso.org/standard/75929.html> > (accessed 2019-02-04).
- [16] Media Source Extensions: W3C (online), available from < <http://w3c.github.io/media-source/> > (accessed 2019-02-04).