

複数動画を同期配信する分割放送型配信システム

井上 祐輔^{1,a)} 木村 明寛¹ 後藤 佑介¹

受付日 2016年5月22日, 採録日 2016年11月1日

概要: 音声や映像といったひと続きの動画を多くのクライアントが視聴する場合、放送型によるデータ配信が有効な手段である。特に、動画データを分割して複数の通信路を用いて繰り返し放送することでデータ受信時の待ち時間を短縮する分割放送型配信では、分割したデータを効率的に放送するためのスケジューリング手法が数多く提案されている。我々の研究グループでは、計算機ネットワーク環境でこれらのスケジューリング手法を導入可能な分割放送型配信システムを提案し、スケジューリング手法の性能評価を行っている。しかし、本システムでは、セグメントの配信契機を考慮した同期配信、および複数動画の同時再生の2つに対応しておらず、複数動画の分割放送型配信を考慮したスケジューリング手法の導入および評価ができなかった。本論文では、これら2つの問題に対処して複数動画を同期配信する分割放送型配信システムを実現する。評価では、サーバ用計算機、クライアント用計算機、および Dummynet を用いた計算機ネットワーク環境を構築し、スケジューリング手法を用いた実待ち時間の短縮効果を確認する。

キーワード: 分割放送型配信システム, スケジューリング, 待ち時間

Implementation and Evaluation of Division-based Broadcasting System for Multiple Videos

YUSUKE INOUE^{1,a)} AKIHIRO KIMURA¹ YUSUKE GOTOH¹

Received: May 22, 2016, Accepted: November 1, 2016

Abstract: Due to the recent popularization of IP multicasts, the continuous broadcasting of audio or video media data is attracting great attention. Although servers can concurrently deliver data to many clients, they have to wait until their data are broadcast. In division-based broadcasting, many researches have proposed scheduling methods to reduce waiting time. A division-based broadcasting system has also been proposed to evaluate these methods. However, this system does not consider several problems that are assumed in delivering multiple videos synchronously. In this paper, we implement a division-based broadcasting system for delivering multiple videos. In our proposed system, we evaluate the effectiveness of reducing both the waiting and interruption times with conventional scheduling methods in computer networks.

Keywords: division-based broadcasting system, scheduling, waiting time

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にともない、複数動画を同時に視聴する環境への注目が高まっている。たとえば、fv ソリューション [1] を利用した通信カラオケの店舗で、ユーザの要望に応じて部屋の映像を撮影し、テ

レビ画面上に表示するサービス [2] があげられる。複数のカラオケ部屋でそれぞれ撮影された複数の映像を各部屋のテレビ画面上で同時に表示することで、ユーザは1つのカラオケ部屋で他の部屋のユーザとカラオケを楽しんでいるような経験ができる。この例では、録画中の動画データを逐次配信するリアルタイム配信を用いている。

一方、録画完了後の動画データの蓄積型配信方式として、Video on Demand (以下、VoD) と放送型配信の2つがある。VoD では、サーバはクライアントの受信要求に応じて

¹ 岡山大学大学院自然科学研究科
Graduate School of Natural Science and Technology,
Okayama University, Okayama 700-8530, Japan

^{a)} p3jw26yg@s.okayama-u.ac.jp

帯域幅を確保して動画を配信する．このため、同じ時間帯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配信するが、クライアントは要求した動画データが配信されるまで待つ必要がある．この待ち時間を短縮するため、ひと続きの動画データを複数の部分に分割して複数の通信路で配信する分割放送型配信が提案されてきた．また、分割放送型配信において、データ受信時に発生する待ち時間を短縮するスケジューリング手法が数多く提案されており、我々はスケジューリング手法を導入可能な分割放送型配信システムを実現してきた．

これまでの研究で、複数動画を同時に視聴する場合を考慮したスケジューリング手法は提案されているが、計算機ネットワーク上での評価は行っていなかった．本論文では、Dummynet を用いた計算機ネットワーク環境を介して複数動画を同期配信する分割放送型配信システムを実現し、スケジューリング手法を用いた実待ち時間の短縮効果を確認する．

本論文は、以下のように構成される．2章で動画データの配信方式について述べ、3章で複数動画の分割放送型配信を説明する．4章で分割放送型配信システム *TeleCaS* における課題、5章で実現方法について述べ、6章で評価を行う．最後に、7章で本論文をまとめる．

2. 動画データの配信方式

動画データの配信方式として、VoD と放送型配信の2つがあげられる．以下で、順番に説明する．

2.1 VoD

VoD は、NHK オンデマンド [3] をはじめとするインターネットテレビや YouTube [4] 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．クライアントがサーバに動画データの受信を要求すると、サーバは帯域幅を確保し、通信路（以下、チャンネル）を用いて動画データをクライアントに送信する．サーバは、クライアントの受信要求に応じて動画データを送信するため、クライアントは要求した動画を即座に受信して再生を開始できる．しかし、クライアントが動画データの受信を要求するたびにサーバは帯域幅を確保する必要があるため、サーバが使用する帯域幅は動画を要求するクライアント数に比例して増加し、サーバの処理負荷は大きくなる．

2.2 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．利用例として、即時性の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1つのチャンネルで繰り返し

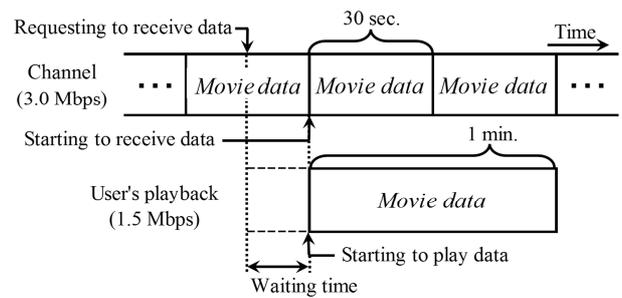


図 1 放送型配信における待ち時間発生の様子

Fig. 1 Waiting time in broadcasting system.

配信する CS デジタル放送 [5] があげられる．サーバは一定の帯域幅で同じ動画データを繰り返し配信し、クライアントは所望の契機で動画データを受信して再生する．サーバは、マルチキャストやブロードキャストといった通信方式を用いてクライアントに放送型で配信することで、VoD に比べてサーバの処理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、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 1 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．図 1 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であり、クライアントは動画データの先頭部分を受信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、サーバが動画データの先頭部分を配信する直前にクライアントが受信を要求した場合に待ち時間は最短で 0 秒となり、平均待ち時間は $(30 + 0) / 2 = 15$ 秒となる．

3. 複数動画の分割放送型配信

3.1 分割放送型配信

分割放送型配信は、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．たとえば、図 1 で用いた動画データを 2 つのセグメント S_1, S_2 に等分割し、1.5 Mbps の 2 つのチャンネル C_1, C_2 を用いて C_1 で S_1, C_2 で S_2 を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、動画データの受信を要求してから 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

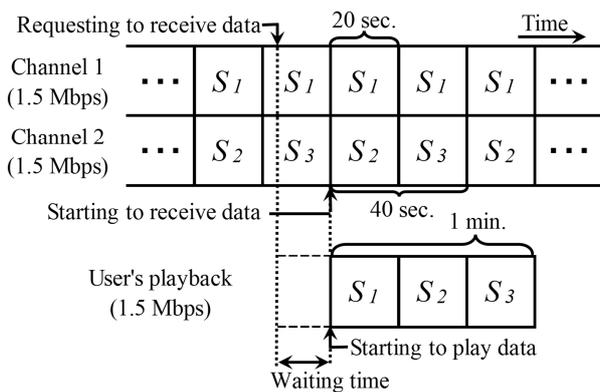


図 2 分割放送型配信における配信スケジュール例 (FB 法)
 Fig. 2 Example of broadcasting schedule in FB scheme.

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可能性は高くなる。データ受信時に再生中の途切れの発生が見込まれる場合、途切れ時間の分だけ動画データの再生開始を遅らせる必要があり、動画データの受信要求から再生開始まで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が発生しないようにしたうえで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 [6], [7], [8] が提案されてきた。

図 2 に、既存のスケジューリング手法である Fast Data Broadcasting and Receiving Scheme (以下、FB 法) [9] による配信スケジュール例を示す。FB 法は、サーバが使用できる帯域幅を 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うえで、 i 番目のチャンネル C_i に連続した 2^{i-1} 個のセグメントを配置する手法である。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 1 と同じとする。

図 2 の場合、ひと続きの動画データを 3 つのセグメント S_1, S_2, S_3 に分割して、1.5 Mbps の 2 つのチャンネル C_1, C_2 を用いて C_1 で S_1, C_2 で S_2, S_3 を繰り返して配信する。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、セグメントの再生時間は 20 秒となる。このとき、待ち時間は動画データの受信を要求してから S_1 の先頭が配信されるまでの時間であるため、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 つに等分割する場合の平均待ち時間はどちらも 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。

3.2 複数動画の同期配信を考慮したスケジューリング手法

これまでの分割放送型配信では、サーバはスケジューリング手法を用いて配信することで、クライアントの動画再生までの待ち時間を効率的に短縮していた。スケジューリング手法では、サーバが使用できる帯域幅、動画の再生単位、およびクライアントの受信能力といった配信条件をもとに、各セグメントをどのタイムスロットでどのチャンネルから配信するかを決定する。分割放送型で複数動画を同期配信する場合、まとめて配信するすべての動画データについて待ち時間を短縮する必要があり、スケジューリングは複雑になる。以下で、複数動画の同期配信を考慮した既存のスケジューリング手法である Basic multiple-video broadcasting scheme (以下、MV-B 法) [10]、および Multiple-video broadcasting scheme with repairing (以下、MV-R 法) [10] の 2 つを説明する。

3.3 複数動画の同期配信における想定環境

複数動画を同期配信するためのスケジューリング手法である MV-B 法、および MV-R 法の想定環境を以下に示す。

- (1) 各チャンネルの帯域幅は、再生レートと等しい。
- (2) 各チャンネルは、複数の動画データのセグメントを組み合わせてスケジューリングできる。
- (3) 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。

3.4 MV-B 法

MV-B [10] 法は、複数動画の分割放送型配信において待ち時間を短縮するスケジューリング手法である。動画数が m のとき、 m 番目の動画の第 n セグメントは $S_{(m,n)}$ と表す。また、 j 番目のタイムスロット T_j までに i 番目の動画の第 j セグメント $S_{(i,j)}$ を放送しなければならないという条件があり、この条件をスケジューリング条件と呼ぶ。たとえば、セグメント $S_{(i,j)}$ は、 T_j までに最低 1 つのチャンネルにスケジューリングされる必要がある。このとき、スケジューリング手順は以下ようになる。

- (1) サーバは、各動画で j 番目のセグメントである $S_{(1,j)}, \dots, S_{(m,j)}$ を $f(j) = \lceil m/j \rceil$ 番目以降のチャンネルに順番にスケジューリングする。
- (2) サーバは、チャンネル $h_{(i,j)}$ でセグメント $S_{(i,j)}$ を放送する。 $h_{(i,j)}$ は、以下の式で求められる。

$$h_{(i,j)} = \sum_{k=1}^{j-1} f(k) + \lceil i/j \rceil \quad (1 \leq i \leq m, j \geq 1)$$

上記のスケジューリング手順に基づき、MV-B 法でスケジューリングすると、セグメントがチャンネルにスケジューリングされない時間帯 (以下、アイドル時間) が発生する。動画数を 5、チャンネル数を 15 とした場合の配信スケジュール例を図 3 に示す。図 3 の場合、*idle* と表示されたタイム

Time slot	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_{10}	...
Channel 1	S(1,1)	S(1,1)	S(1,1)	S(1,1)	S(1,1)	S(1,1)	S(1,1)	S(1,1)	S(1,1)	S(1,1)	...
Channel 2	S(2,1)	S(2,1)	S(2,1)	S(2,1)	S(2,1)	S(2,1)	S(2,1)	S(2,1)	S(2,1)	S(2,1)	...
Channel 3	S(3,1)	S(3,1)	S(3,1)	S(3,1)	S(3,1)	S(3,1)	S(3,1)	S(3,1)	S(3,1)	S(3,1)	...
Channel 4	S(4,1)	S(4,1)	S(4,1)	S(4,1)	S(4,1)	S(4,1)	S(4,1)	S(4,1)	S(4,1)	S(4,1)	...
Channel 5	S(5,1)	S(5,1)	S(5,1)	S(5,1)	S(5,1)	S(5,1)	S(5,1)	S(5,1)	S(5,1)	S(5,1)	...
Channel 6	S(1,2)	S(2,2)	S(1,2)	S(2,2)	S(1,2)	S(2,2)	S(1,2)	S(2,2)	S(1,2)	S(2,2)	...
Channel 7	S(3,2)	S(4,2)	S(3,2)	S(4,2)	S(3,2)	S(4,2)	S(3,2)	S(4,2)	S(3,2)	S(4,2)	...
Channel 8	S(5,2)	<i>idle</i>	S(5,2)	<i>idle</i>	S(5,2)	<i>idle</i>	S(5,2)	<i>idle</i>	S(5,2)	<i>idle</i>	...
Channel 9	S(1,3)	S(2,3)	S(3,3)	S(1,3)	S(2,3)	S(3,3)	S(1,3)	S(2,3)	S(3,3)	S(1,3)	...
Channel 10	S(4,3)	S(5,3)	<i>idle</i>	S(4,3)	S(5,3)	<i>idle</i>	S(4,3)	S(5,3)	<i>idle</i>	S(4,3)	...
Channel 11	S(1,4)	S(2,4)	S(3,4)	S(4,4)	S(1,4)	S(2,4)	S(3,4)	S(4,4)	S(1,4)	S(2,4)	...
Channel 12	S(5,4)	<i>idle</i>	<i>idle</i>	<i>idle</i>	S(5,4)	<i>idle</i>	<i>idle</i>	<i>idle</i>	S(5,4)	<i>idle</i>	...
Channel 13	S(1,5)	S(2,5)	S(3,5)	S(4,5)	S(5,5)	S(1,5)	S(2,5)	S(3,5)	S(4,5)	S(5,5)	...
Channel 14	S(1,6)	S(2,6)	S(3,6)	S(4,6)	S(5,6)	<i>idle</i>	S(1,6)	S(2,6)	S(3,6)	S(4,6)	...
Channel 15	S(1,7)	S(2,7)	S(3,7)	S(4,7)	S(5,7)	<i>idle</i>	<i>idle</i>	S(1,7)	S(2,7)	S(3,7)	...

図 3 分割放送型配信における複数動画の配信スケジュール例 (MV-B 法)

Fig. 3 Example of broadcasting schedule in MV-B method.

Time slot	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_{10}	...
Channel 1	S(1,1)	S(1,1)	S(1,1)	S(1,1)	S(1,1)	S(1,1)	S(1,1)	S(1,1)	S(1,1)	S(1,1)	...
Channel 2	S(2,1)	S(2,1)	S(2,1)	S(2,1)	S(2,1)	S(2,1)	S(2,1)	S(2,1)	S(2,1)	S(2,1)	...
Channel 3	S(3,1)	S(3,1)	S(3,1)	S(3,1)	S(3,1)	S(3,1)	S(3,1)	S(3,1)	S(3,1)	S(3,1)	...
Channel 4	S(4,1)	S(4,1)	S(4,1)	S(4,1)	S(4,1)	S(4,1)	S(4,1)	S(4,1)	S(4,1)	S(4,1)	...
Channel 5	S(5,1)	S(5,1)	S(5,1)	S(5,1)	S(5,1)	S(5,1)	S(5,1)	S(5,1)	S(5,1)	S(5,1)	...
Channel 6	S(1,2)	S(2,2)	S(1,2)	S(2,2)	S(1,2)	S(2,2)	S(1,2)	S(2,2)	S(1,2)	S(2,2)	...
Channel 7	S(3,2)	S(4,2)	S(3,2)	S(4,2)	S(3,2)	S(4,2)	S(3,2)	S(4,2)	S(3,2)	S(4,2)	...
Channel 8	S(5,2)	S(5,4)	S(5,2)	S(5,8)	S(5,2)	S(5,4)	S(5,2)	S(4,8)	S(5,2)	S(5,4)	...
Channel 9	S(1,3)	S(2,3)	S(3,3)	S(1,3)	S(2,3)	S(3,3)	S(1,3)	S(2,3)	S(3,3)	S(1,3)	...
Channel 10	S(4,3)	S(5,3)	S(5,6)	S(4,3)	S(5,3)	S(4,6)	S(4,3)	S(5,3)	S(5,6)	S(4,3)	...
Channel 11	S(1,4)	S(2,4)	S(3,4)	S(4,4)	S(1,4)	S(2,4)	S(3,4)	S(4,4)	S(1,4)	S(2,4)	...
Channel 12	S(1,5)	S(2,5)	S(3,5)	S(4,5)	S(5,5)	S(1,5)	S(2,5)	S(3,5)	S(4,5)	S(5,5)	...
Channel 13	S(1,6)	S(2,6)	S(3,6)	S(1,10)	S(2,10)	S(3,10)	S(1,6)	S(2,6)	S(3,6)	S(1,10)	...
Channel 14	S(1,7)	S(2,7)	S(3,7)	S(4,7)	S(4,10)	S(5,10)	S(1,7)	S(2,7)	S(3,7)	S(4,7)	...
Channel 15	S(1,8)	S(2,8)	S(3,8)	S(1,9)	S(2,9)	S(3,9)	S(4,9)	S(5,9)	S(1,8)	S(2,8)	...

図 4 分割放送型配信における複数動画の配信スケジュール例 (MV-R 法)

Fig. 4 Example of broadcasting schedule in MV-R method.

スロットがアイドル時間であり、チャンネル 8, 10, 12, 14, および 15 でアイドル時間が発生する。

3.5 MV-R 法

MV-R [10] 法は、3.4 節で述べた MV-B 法で発生するアイドル時間に他のセグメントを配信することで、MV-B 法より待ち時間を短縮する手法である。MV-R 法では、MV-B 法と同様にチャンネル 1 から順番にセグメントをスケジュールリングするが、あるタイムスロットでアイドル時間が発生した場合、他のセグメントを該当するタイムスロットにスケジュールリングする。

チャンネルのアイドル時間に他のセグメントを割り当てて補完する方法として、完全補完と強制補完の 2 つがある。完全補完は、アイドル時間に割り当てたセグメントがスケジュールリング条件を維持できる場合の補完方法である。逆に、スケジュールリング条件を維持できない場合は強制補完を行う。

MV-R 法の配信スケジュールを図 4 に示す。配信条件は 3.4 節と同じとする。MV-R 法では、MV-B 法でセグメントをスケジュールリングするとともに、MV-B 法で発生したアイドル時間を他のセグメントで補完することで、MV-B 法に比べて待ち時間を短縮できる。

3.6 シミュレーション環境

これまでのスケジューリング手法は、計算機間で発生する処理負荷を考慮しない理想的なシミュレーション環境下で評価を行った。ここで、本論文が想定するシミュレーション環境を以下に示す。

- (1) サーバにおいて、クライアントの受信要求開始からデータの配信開始までの間に発生する処理時間は 0 とする。
- (2) クライアントにおいて、データの受信開始から再生開始までの間に発生する処理時間は 0 とする。
- (3) クライアントの待ち時間は計算式で算出する。

シミュレーション環境では、サーバがセグメントを送信単位に分割してクライアントに送信するときの処理負荷、およびクライアントにおいてデータの受信完了からセグメントに復元して再生プレイヤーに送信開始するまでに発生する処理時間を考慮していない。また、計算機シミュレーションで行うため、サーバがクライアントにデータを配信する時間のみを待ち時間として、計算式で算出する。一方、本論文で想定する計算機ネットワーク環境では、これらの処理時間を含めた待ち時間および途切れ時間を評価する。

4. 分割放送型配信システム TeleCaS

4.1 概要

これまでに提案されてきたスケジューリング手法の多くは、計算機上でのシミュレーション環境を想定している。このため、システムの処理負荷やパケット損失による動画再生への影響を想定しておらず、計算機ネットワーク環境でスケジューリング手法を用いる場合に発生する影響を考慮していない。我々の研究グループでは、計算機ネットワーク環境でスケジューリング手法を評価可能な分割放送型配信システム *Telecommunication and BroadCasting System (TeleCaS)* [11] を提案してきた。しかし、*TeleCaS* は、複数動画の同期配信や同時再生に対応しておらず、MV-B 法や MV-R 法を適用できない。そこで、*TeleCaS* において、複数動画の同期配信および同時再生を実現する。課題として、セグメントの配信契機を考慮した同期配信方式の実現、および複数動画の同時再生方式の実現の 2 つがある。以降の節で、順番に説明する。

4.2 セグメントの配信契機を考慮した同期配信方式

分割放送型配信で提案されているスケジューリング手法は、再生中に途切れが発生しないようにセグメントをスケジュールリングしたうえで、各チャンネルで周期的にセグメントを配信する。シミュレーション環境では、すべてのチャンネルでセグメントの先頭部分の配信契機を同期できる。一方で、計算機ネットワーク環境でスケジューリング手法を用いて動画データを配信する場合、サーバはセグメントの受信に必要な情報（以下、配信開始部）をクライアントに

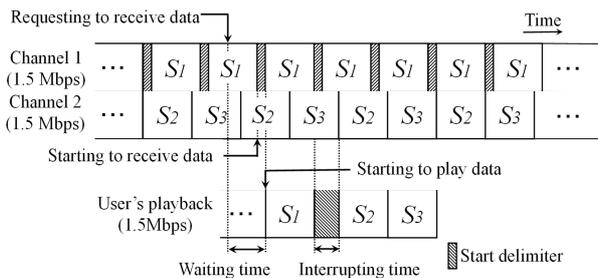


図 5 TeleCaSにおける非同期配信の例

Fig. 5 Example of asynchronous broadcasting schedule in TeleCaS.

送る必要があるが、配信スケジュールには含まれていない。
 TeleCaSにおいて、図 2 に示す FB 法を計算機ネットワーク環境で導入する場合の配信スケジュールを図 5 に示す。配信開始部を考慮してセグメントをスケジューリングする場合、配信開始部を配信するチャンネルと配信しないチャンネルとの間で、配信開始時刻にずれが発生する。このとき、チャンネル間で各セグメントの先頭部分の配信契機を同期できず、シミュレーション環境で想定していない再生中の途切れが発生する。

4.3 複数動画の同時再生方式

TeleCaSで複数動画の同時再生に対応するためには、サーバが配信する複数動画の判別方法の提案、および複数動画の逐次再生方式の実現という 2 つの課題がある。以降で詳しく述べる。

4.3.1 複数動画の判別方法

複数動画の配信において、サーバは 1 つのチャンネルで複数の動画のセグメントを配信するようにスケジューリングすることで、各動画の配信契機を同期できる。クライアントは、複数動画を同時に視聴するため、複数のチャンネルから同時に受信したセグメントを各動画に復元する必要がある。これまでの TeleCaS のデータフォーマットは、配信開始部、およびセグメントを等分割した部分（以下、サブセグメント）をセグメントに復元する情報（以下、情報部）の 2 つで構成されていた。しかし、各セグメントがどの動画のどの部分かを示す情報はデータフォーマットに含まれていなかった。このため、クライアントは動画ごとに異なる再生プレイヤーに動画データを送信できず、複数動画を同時に再生できない。

4.3.2 複数動画を考慮した逐次再生方式の実現

これまでの TeleCaS における逐次再生方式を説明する。初めに、ブラウザ上の再生プレイヤーを読み込む。次に、情報部の情報をもとに、クライアントはサーバから受信したサブセグメントを配信前のセグメントに復元する。動画の先頭部分のサブセグメントを受信すると、クライアントは分割配信の通信プロトコルを逐次再生の通信プロトコルで

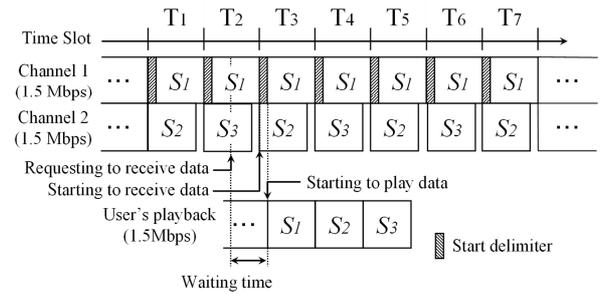


図 6 TeleCaSにおける同期配信の例

Fig. 6 Example of synchronous broadcasting schedule in TeleCaS.

ある Hypertext Transfer Protocol (HTTP) に変換して、サブセグメント単位で動画データをブラウザに送信することで、逐次再生を実現できる。

しかし、従来の TeleCaS では、複数の再生プレイヤーを同時に読み込むことができず、複数の動画を同時に逐次再生できないという問題がある。このため、受信した複数動画を同時に再生できず、複数動画の配信に対応したスケジューリング手法を TeleCaS に適用できない。

5. 実現方式

5.1 セグメントの配信契機を考慮した同期配信方式

4.2 節の課題に対処するため、我々の研究グループでこれまでに提案した配信契機の同期方法 [11] を用いて、複数動画の同期配信を実現する。この方法では、図 6 に示すように、各セグメントの配信契機が同期し、途切れ時間を短縮できる。また、複数動画の配信では、動画データを配信するチャンネル数が増加するため、単一の動画配信と比べて配信開始時刻のずれによる影響は大きくなる。そこで、複数の動画データを分割した各セグメントを単一動画のセグメントとして処理する。これにより、複数動画の配信におけるセグメントの処理は単一の動画配信と同様の処理となり、セグメントの配信契機の同期を実現できる。また、図 6 において、各チャンネルで行われる同期配信で発生する動画データを配信しない時間は、配信開始部のデータサイズは 1,472 Bytes であるため、 $1472 \times 8 / (1.5 \times 1000000) \times 1000 \approx 7.9$ ミリ秒となり、伝送効率の低下はほとんどない。

5.2 複数動画の同時再生方式

5.2.1 複数動画の判別方法

4.3.1 項の課題に対処するため、配信開始部を拡張して、複数動画の配信に対応する。実現方式のデータフォーマットを図 7 に示す。データフォーマットは、従来と同様に、配信開始部と情報部で構成する。従来の配信開始部は、識別子、タイムスロット番号、サブセグメントデータサイズ、セグメント番号、セグメントデータサイズ、バッファ時間、および再生時間で構成していた。また、情報部は、

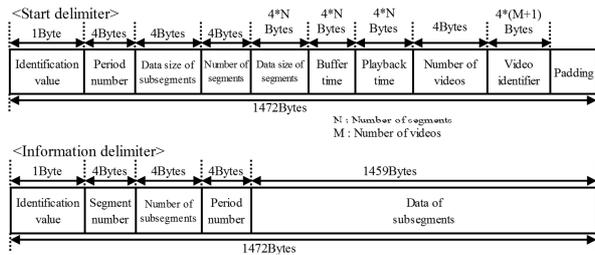


図 7 データフォーマット
Fig. 7 Data format.

識別子, サブセグメント番号, タイムスロット番号, およびサブセグメントで構成していた. 実現方式では, 配信開始部を拡張して, 動画数および動画識別子を追加することで, 4.3.1 項の課題を解決した. クライアントは, 動画識別子を用いることで各セグメントがどの動画に対応するかを識別できる.

また, **TeleCaS**上で MV-B 法のアイドル時間を実現するため, サーバはアイドル時間中に, 配信する動画データと関連がない別の動画データをクライアントに配信する. このとき, クライアントが視聴する動画数と実際に配信する動画数は異なるため, クライアントは動画数のデータが必要となる. そこで, 追加した動画数のデータを用いることで, クライアントは動画数と同じ数の動画データをブラウザに送信し, スケジューリング手法で設定した数の動画データを再生できる. たとえば, 図 3 における MV-B 法の配信スケジュールの場合, サーバは 5 種類の動画データを同時に配信する. しかし, 実装した **TeleCaS**で MV-B 法を実現する場合, 配信スケジュールを維持するため, 5 種類の動画とは異なる動画をアイドル時間に配信する必要がある. そこで, クライアントは動画数の情報をもとに, クライアントが同時に再生する 5 種類の動画データをブラウザに送信することで, クライアントはスケジューリング手法で設定した数の動画を同時に再生できる.

5.2.2 複数動画を考慮した逐次再生方式の実現

4.3.2 項で述べた課題に対処するため, ブラウザと通信を行う処理部をスレッド化する. **TeleCaS**における複数動画の分割放送型配信の構成を図 8 に示す. サーバは, 分割放送型配信の通信プロトコルを使用して, サブセグメント単位でクライアントに動画データを送信する. クライアントは, ブラウザの読み込みから再生プレイヤーに動画データを転送するまでの部分をスレッド化したうえで, 並列に処理する. これにより, ブラウザに埋め込まれた複数の再生プレイヤーを同時に読み込むことができ, クライアントは複数の再生プレイヤーと同時に通信できる. また, 配信開始部の動画識別子をもとに, 各動画で先頭のサブセグメントを受信すると, クライアントは分割配信の通信プロトコルを逐次再生の通信プロトコルである HTTP に変換して, 対応する再生プレイヤーにサブセグメントを送信することで,

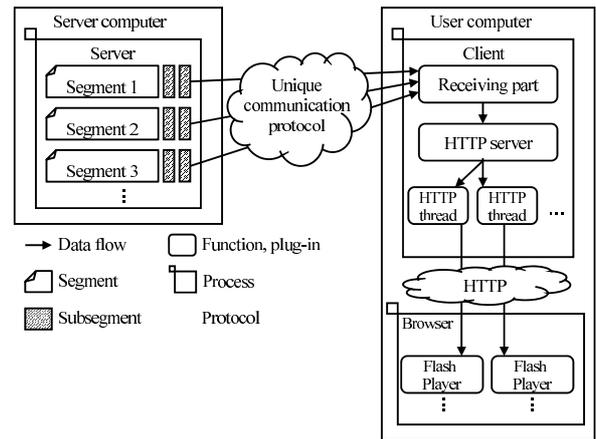


図 8 複数動画の分割放送型配信の構成
Fig. 8 Configuration of division-based broadcasting system for multiple videos.

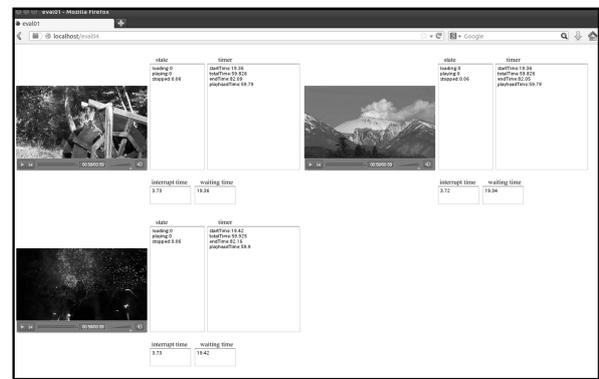


図 9 TeleCaSの再生画面
Fig. 9 Screen Shot in TeleCaS.

複数の動画を同時に逐次再生できる.

5.3 TeleCaSにおける実現方式の実装

5.3.1 想定環境

分割放送型配信システム **TeleCaS**において, 複数動画の同期配信および同時再生を実装するため, 想定する環境を以下に示す.

- (1) サーバは, 複数のチャンネルを用いてクライアントにセグメントを繰り返し配信する.
- (2) クライアントは, ブラウザ上で動画プレイヤーを用いて動画を視聴する.
- (3) クライアントは, 動画の再生を開始すると最後まで途切れずに再生できる.
- (4) クライアントは, すべてのセグメントを蓄積するために十分なバッファを持つ.

5.3.2 実装内容

5.1, 5.2 節で述べた実現方式をもとに, **TeleCaS**で複数動画の同期配信および同時再生に対応するように実装した. **TeleCaS**は, チャンネル数, 帯域幅, およびセグメントの配信順序が異なるスケジューリング手法を導入できる.



図 10 評価環境の構成

Fig. 10 Hardware configuration.

表 1 計算機の性能

Table 1 Computer spec for evaluations.

Server (1 PC)	CPU	Intel® Core™2 Duo CPU E7500 (2.93 GHz)
	Memory	2.0 GBytes
	OS	Ubuntu 12.10
	NIC	RTL8101E/RTL8102E
Dummynet (1 PC)	CPU	Intel® Core™2 Duo CPU E7500 (2.93 GHz)
	Memory	2.0 GBytes
	OS	FreeBSD 8.2-RELEASE
	NIC1	RTL8169SC
	NIC2	RTL8169SC
Client (3 PCs)	CPU	Intel® Core™2 Duo CPU E7500 (2.93 GHz)
	Memory	2.0 GBytes
	OS	Ubuntu 12.10
	NIC	RTL8101E/RTL8102E

TeleCaSにおける再生画面のスクリーンショットを図 9 に示す。図 9 は、3 種類の動画データを同時に再生している様子を示す。動画再生には、Adobe Flash Player を用いた。

6. 評価

6.1 評価環境

TeleCaSの評価環境を図 10 に示す。計算機ネットワーク環境を想定して、TeleCaSを導入した計算機を複数台用いてネットワークを構成した。サーバ計算機とクライアント計算機は Gigabit Ethernet で接続した。サーバ計算機 1 台とクライアント計算機 3 台の間は、帯域制御機能を持つ Dummynet を利用できる計算機を経由することで、ネットワークの帯域幅を制御する。評価に用いた各計算機の性能を表 1 に示す。本研究では、Dummynet により帯域幅を制御したうえで、単位時間あたりに受信するデータ量を帯域幅として算出する。また、マルチキャスト通信を用いることで、放送型による動画配信を実現した。

6.2 TeleCaSを用いた評価

計算機ネットワーク環境でスケジューリング手法の有用性を確認するため、TeleCaS上で待ち時間および途切れ時間の評価を行う。今回の評価では、複数動画を考慮したスケジューリング手法である MV-B 法と MV-R 法、および単一動画のスケジューリング手法であり複数動画の配信スケジュールに拡張可能な FB 法の 3 種類を用いる。これらのスケジューリング手法が提案された当時のネットワーク環境と比べて、現在のネットワーク品質は向上している。これにより、サーバが動画データの配信に使用できる帯域幅

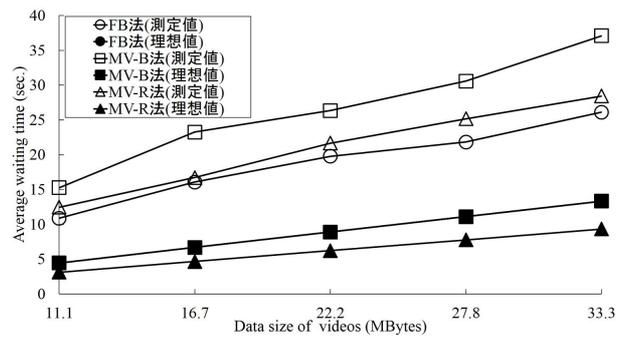


図 11 第 1 セグメントのデータサイズと平均待ち時間

Fig. 11 Data size of S_1 and average waiting time.

は増加しており、再生レートより大きな帯域幅を動画データの配信に使用することで、待ち時間および途切れ時間を短縮できる。一方で、現在のネットワーク環境では、サーバはより多くの動画データやサービスをクライアントに提供するため、動画データの配信に割り当てる帯域幅を削減するスケジューリング手法は必要である。

計算機シミュレーションで算出される待ち時間の値を理想値とし、TeleCaSによる配信で算出される待ち時間と途切れ時間の値を測定値とする。各スケジューリング手法における待ち時間の理想値は、3.1 節で説明したように、第 1 セグメントの受信時間の半分とした。理想値の算出で必要となる第 1 セグメントのデータサイズは、スケジューリング手法に基づき、評価に用いる各動画データを分割した第 1 セグメントのデータサイズの平均値とした。

クライアントがひと続きの動画データを再生終了まで途切れなく再生するため、再生中に発生する途切れ時間の分だけ動画の再生開始を遅らせる必要がある。そこで、待ち時間の測定値（以下、実待ち時間）は、TeleCaSで測定する待ち時間と途切れ時間の合計値とした。また、TeleCaSで測定する待ち時間と途切れ時間はクライアント計算機 3 台の平均値とし、クライアントが動画データの受信をサーバに要求した時刻から測定を開始する。また、評価では、逐次再生方式を用いることで、第 1 セグメントのうち先頭部分となるサブセグメントを受信すると再生を開始できる。

6.3 待ち時間に関する評価

複数動画の配信において 3 種類のスケジューリング手法を導入し、計算機ネットワーク環境における待ち時間の短縮効果を確認する。動画のデータサイズを変化させた場合の実待ち時間の変化を図 11 に示す。横軸は動画のデータサイズ、縦軸はクライアント計算機で各動画の実待ち時間を同一条件で 10 回求めた平均値とした。また、再生時間が異なる動画データを使用して、動画のデータサイズを変化させた。MV-B 法および MV-R 法の配信スケジュールは、図 3 および図 4 とそれぞれ同じとする。動画のデータサイズ以外の配信条件として、チャンネル帯域幅は 1.5 Mbps,

動画の再生レートは 1.5 Mbps, ブラウザ内の再生プレイヤー数は 5 個, チャンネル数は 15, 動画数は 5 種類とする. 3 台すべてのクライアントが受信を開始するタイムスロットはランダムとする.

まず, 理想値について説明する. 図 11 より, FB 法と MV-B 法における平均待ち時間の理想値はどちらも等しい. 15 個のチャンネルで 5 種類の動画データを配信する場合, FB 法と MV-R 法の動画分割数は 7 で同じであり, 第 1 セグメントの大きさは等しいため, 第 1 セグメントを受信する時間の半分である平均待ち時間の理想値は一致する. 一方で, MV-R 法の動画分割数は 10 であり, 第 1 セグメントの大きさは他の 2 つの手法より小さくなるため, MV-R 法における平均待ち時間の理想値は他の手法より短くなる.

次に, 測定値の評価結果について説明する. 図 11 より, すべてのスケジューリング手法について, 測定値は理想値に比べて長大化する. たとえば, 動画データサイズが 22.2 MBytes において, FB 法の理想値は第 1 セグメント (約 3.18 MBytes) を受信する時間の半分であり, 約 8.9 秒となる. 一方, FB 法の測定値は, 待ち時間が約 8.8 秒, 途切れ時間が約 11.0 秒であり, 実待ち時間は合計で約 19.8 秒となる. この時間差は, シミュレーション環境では考慮していない動画再生中の途切れ時間の影響によるものである.

また, 動画データサイズが 22.2 MBytes の場合, MV-R 法の測定値は, 待ち時間が約 6.2 秒, 途切れ時間が約 15.5 秒であり, 実待ち時間は合計で約 21.7 秒となるため, FB 法より長い. 測定値では, 動画再生中に発生する途切れ時間の影響により, スケジューリングが FB 法に比べて複雑である MV-R 法の実待ち時間が大きく長大化する. 一方, 理想値では, MV-R 法の待ち時間は FB 法より短い.

6.4 同時再生する動画数の変化による途切れ時間の評価

6.3 節より, 動画再生中に発生する途切れ時間は実待ち時間に大きな影響を与える. そこで, 同時再生する動画数の変化による途切れ時間を評価し, 再生中に発生する途切れの要因について考察する.

同時再生する動画数を変化させた場合の途切れ時間の変化を図 12 に示す. 横軸は同時再生する動画数, 縦軸はクライアント計算機で各動画の途切れ時間を同一条件で 10 回求めた平均値とした. 動画データのデータサイズは 11.1 MBytes とし, これ以外の条件は 6.3 節と同じとした.

まず, 同時再生する動画数を変化させた場合の途切れ時間の変化について評価する. 図 12 より, FB 法では, 同時再生する動画数が 1 のときに途切れ時間は約 3.1 秒, 同時再生する動画数が 5 のときに途切れ時間は約 7.7 秒となる. 同時再生する動画数が増加すると, 途切れ時間は長大化する. これは, 同時再生する動画数の増加により, ブラ

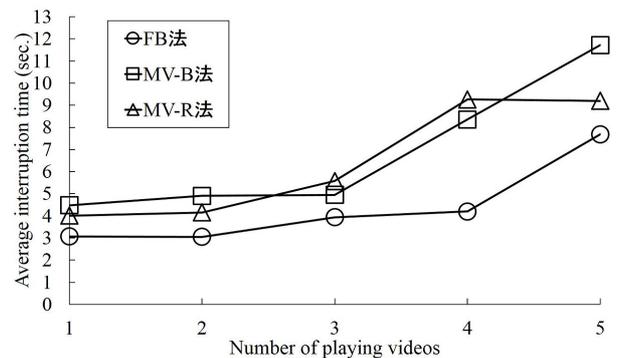


図 12 再生動画数と平均途切れ時間

Fig. 12 Number of videos and average interruption time.

ウザへ同時に送信する動画のデータ量が増えてシステム全体の処理量が増加することで, 途切れ回数が増加したためである.

次に, 各スケジューリング手法における途切れ時間の変化について評価する. FB 法では, ほぼ線形的に途切れ時間が長大化した. *TeleCaS* で FB 法を適用する場合, 各動画の配信スケジュールはすべて同じになり, すべてのタイムスロットにおいて受信する各動画データのセグメント数は等しい. このため, 動画数の増加に比例して途切れ時間は長大化する. 一方で, MV-B 法, および MV-R 法では, 一定数以上の動画を同時再生すると, 途切れ時間は急激に長大化する. MV-B 法, および MV-R 法は, 複数の動画データを同期配信するようにセグメントをスケジューリングするため, 配信スケジュールは複雑になる. このとき, 各動画データがタイムスロットごとに受信するセグメント数は, タイムスロットによって偏りが生じる. このため, クライアントがデータの受信を開始する契機に応じて, 途切れ時間は大きく変化する.

また, MV-B 法において, 動画数が 2 から 3 に増加した場合, 平均途切れ時間は短縮する. MV-B 法は, 複数の動画をスケジューリングしているため, クライアントが受信を開始するタイムスロットに応じて, 3 番目の動画データが 2 番目の動画データより早く受信開始する場合が発生する. このような場合, 3 番目の動画データで逐次再生の対象となるセグメント数が 2 番目の動画データで逐次再生の対象となるセグメント数より少なくなり, 再生する 3 種類の動画データにおける平均途切れ時間は短縮する. MV-R 法において, 動画数が 4 から 5 に増加する場合も同様である.

6.5 受信を開始するタイムスロットの変化による途切れ時間の評価

6.4 節より, 動画再生中に発生する途切れ時間は, クライアントが受信を開始するタイムスロットに主に影響される. そこで, クライアントが受信を開始するタイムスロットを変化させた場合における途切れ時間の変化を図 13 に

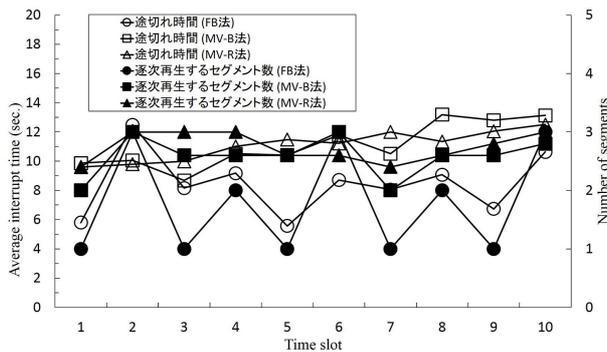


図 13 受信を開始するタイムスロットと平均途切れ時間
Fig. 13 Time slot and average interruption time.

示す。横軸は、クライアントが受信を開始するタイムスロット T ($1 \leq T \leq 10$) とする。縦軸は、あるタイムスロットでクライアントが受信を開始した場合に発生する途切れ時間を同一条件で 10 回求めた平均値、および各動画においてあるタイムスロットで受信を開始する場合に再生中に逐次再生を行うセグメント数の平均値である。3 台すべてのクライアントが受信を開始するタイムスロットは同じとする。動画のデータサイズは 11.1 MBytes とし、これ以外の条件は 6.3 節と同じとした。

図 13 より、FB 法は、逐次再生を行うセグメント数は他の 2 つの手法より少なく、動画再生中に発生する途切れ時間は短い。一方、MV-B 法、および MV-R 法は、逐次再生するセグメント数はほぼ同じであるため、動画再生中に発生する途切れ時間はほぼ等しい。図 13 において、タイムスロット 10 までの平均逐次再生セグメント数は、FB 法で約 1.8 個、MV-B 法で約 2.6 個、および MV-R 法で約 2.7 個であった。また、タイムスロット 10 までの平均途切れ時間は、FB 法で約 8.4 秒であり、MV-B 法および MV-R 法ではともに約 11.1 秒であった。以上より、逐次再生を行うセグメント数の変化の大きさに応じて、動画再生中に発生する途切れ時間は大きく変化することが分かる。

7. おわりに

本研究では、分割放送型配信システム *TeleCaS* において複数動画の同期配信を実現するため、セグメントの配信契機を考慮した同期配信方式、および複数動画の同時再生方式を提案した。また、課題と対処をもとに、これらの方式を *TeleCaS* に実装して評価を行い、実待ち時間の短縮効果の評価した。まず、3 種類のスケジューリング手法を用いて、実待ち時間を評価した。複数動画に対応したスケジューリング手法の *TeleCaS* 上での評価結果より、シミュレーションによる待ち時間の理想値と実待ち時間の差異を確認した。次に、同時再生する動画数の変化による途切れ時間について評価した。評価結果より、同時再生する動画数の増加による途切れ時間への影響を確認した。最後に、クライアントが受信を開始するタイムスロットの変化によ

る途切れ時間を比較して評価した。評価結果より、再生中に逐次再生を行うセグメント数の増加による途切れ時間への影響を確認した。

今後の予定について、順番に説明する。現在の実装方式では、実際のネットワーク環境で想定される帯域変動の影響で、クライアントのデータ再生中に途切れが発生する場合は考慮していない。このため、Resource reSerVation Protocol (RSVP) [12] といった帯域保証プロトコルの実装を行う予定である。また、今回は単一種類のクライアント計算機による計算機ネットワーク環境で評価を行ったが、今後は帯域幅が異なる複数種類の計算機が混在する計算機ネットワーク環境で待ち時間を短縮できるスケジューリング手法 [13] を *TeleCaS* に実装して評価する。さらに、パケット損失を考慮したスケジューリング手法 [14], [15] の導入、および Planetlab [16] を用いた実際のインターネット環境における評価がある。

謝辞 本研究は、文部科学省科学研究費補助金・若手研究 (B) (26730059)、基盤研究 (B) (15H02702)、基盤研究 (C) (16K01065)、および (公財) 山陽放送学術文化財団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] Gnzo: fv ソリューション, Gnzo (オンライン), 入手先 (<http://gnzo.com/fabricvideoapi/>) (参照 2016-05-22).
- [2] 第一興商: DAM ★とも [club DAM.com], DAM ★とも (オンライン), 入手先 (<http://www.clubdam.com/app/damtop/top/Index.do>) (参照 2016-05-22).
- [3] NHK: NHK オンデマンド, NHK (オンライン), 入手先 (<https://www.nhk-ondemand.jp>) (参照 2016-05-22).
- [4] YouTube: YouTube, Youtube (オンライン), 入手先 (<https://www.youtube.com>) (参照 2016-05-22).
- [5] (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 (<http://www.eiseihoso.org/guide/howto.html>) (参照 2016-05-22).
- [6] 義久智樹, 塚本昌彦, 西尾章治郎: 再生単位を考慮したスケジューリング手法における使用チャンネル数について, 日本データベース学会 Letters, Vol.4, No.3, pp.5-8 (2005).
- [7] Juhn L. and Tseng L.: Harmonic broadcasting for video-on-demand service, *IEEE Trans. Broadcasting*, Vol.43, No.3, pp.268-271 (1997).
- [8] 義久智樹, 塚本昌彦, 西尾章治郎: 分割放送方式におけるチャンネルの帯域幅を考慮した連続メディアデータの分割手法, 情報処理学会論文誌 B, Vol.J91-B, No.3, pp.300-308 (2008).
- [9] Juhn, L. and Tseng, L.: Fast Data Broadcasting and Receiving Scheme for Popular Video Service, *IEEE Trans. Broadcasting*, Vol.44, No.1, pp.100-105 (1998).
- [10] Chen, Y. and Huang, K.: Multiple Videos Broadcasting Scheme for Near Video-on-Demand Services, *Proc. IEEE Int. Conf. on Signal Image Technology and Internet Based Systems 2008 (SITIS '08)*, pp.52-58 (2008).
- [11] 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).
- [12] RFC: Internet Group Management Protocol, Version 3, RFC (online), available from (<https://tools.ietf.org/html/rfc3376>) (accessed 2016-08-09).

- [13] 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, No.2, pp.98–114 (2013).
- [14] 青木輝勝, 安里 諒, 沼澤潤二: 分割放送スケジューリング方式における遅延揺らぎ特性の解析とその改善に関する一検討, 電子情報通信学会論文誌 B, Vol.J93-B, No.1, pp.112–122 (2010).
- [15] Gotoh, Y. and Taniguchi, H.: A Scheduling Scheme for Improving Error Resilience on Media Data Broadcasting, *Proc. 1st Workshop on Advances in Data Engineering and Mobile Computing (DEMoC-2012)*, pp.453–458 (2012).
- [16] Planetlab : Planetlab, Planetlab (オンライン), 入手先 (<https://www.planet-lab.org>) (参照 2016-08-09).



井上 祐輔 (学生会員)

2015年岡山大学工学部情報系学科卒業。現在、同大学大学院自然科学研究科博士前期課程在学中。インターネット放送システムの研究に興味を持つ。



木村 明寛

2011年岡山大学工学部情報工学科卒業。2013年同大学大学院自然科学研究科博士前期課程修了。放送型配信システムの構築に興味を持つ。



後藤 佑介 (正会員)

2005年岡山大学工学部情報工学科卒業。2007年京都大学大学院情報学研究科システム科学専攻修士課程修了。2009年同専攻博士後期課程修了。博士(情報学)。2009年岡山大学大学院自然科学研究科助教を経て、2014年同准教授。この間、豪ラトロープ大学客員研究員。放送コンピューティングおよび位置情報システムの研究に興味を持つ。電子情報通信学会, IEEE 各会員。本会シニア会員。