

# 強化学習を用いたMPEG-DASHにおける 映像品質制御手法の実験と評価

生出 真人<sup>1,a)</sup> 阿部 亨<sup>1,2,b)</sup> 菅沼 拓夫<sup>1,2,c)</sup>

**概要:** 現在、映像をリアルタイムに配信するライブストリーミングサービスは、様々な特性の映像を対象に、様々な状況のネットワーク、様々な性能の受信端末（高性能なPCから性能が劣る携帯端末まで）で利用されている。既存のライブストリーミングサービスは、ネットワークの状況（利用可能なネットワーク資源）に基づき、受信端末へ配信する映像の品質を制御しているため、配信映像の品質が受信端末の性能（利用可能な計算機資源）に適したものであるとは限らない。そこで本研究では、映像の特性、利用可能なネットワーク資源、計算機資源と、受信端末でサービス利用者へ配信される映像品質の関係を強化学習することで、ネットワークの状況だけでなく受信端末の性能にも適応した品質の映像を配信するための映像品質制御手法を提案する。本稿では、MPEG-DASHを対象とした映像品質制御手法の設計と実装を行い、実機を用いた実験から、提案手法を用いることで資源の無駄な消費を抑えつつ高品質な映像を利用者へ配信可能であることを確認した。

## 1. はじめに

情報端末の高性能化と無線ネットワーク網の発展に伴い、インターネットを用いた様々なサービスが種々の環境で利用されるようになった。近年、サービスの高機能化により、個々のサービスで生じるネットワークトラフィックが増加したことに加え、サービスを手軽に利用可能な端末の台数が急増したため、インターネット上のトラフィックが爆発的に増加している [1, 2]。この結果、ネットワーク回線の混雑により、利用者に提供されるサービスの品質低下が懸念されており、各サービス利用者に対し、用途に応じた必要十分なレベルでネットワーク資源を割り当てる手法が求められている [3]。

現在、ネットワークトラフィックの大多数は映像配信によるもので占められている。特に、映像をリアルタイムに配信するライブストリーミングサービスでは、地上波の放送映像と4K映像を切り換えての視聴が可能な配信 [4] や、スポーツや音楽のイベントをVR映像として視聴可能な配信 [5] 等、トラフィックの増加を伴うサービスの高機能化が進められている。スマートフォンやタブレット端末等の

携帯端末の急速な普及と相俟って、ライブストリーミングサービスによるトラフィックは2016年から2021年の5年間で5倍に増加すると予測されている [2]。そこで本研究では、インターネット上のトラフィック量に大きな影響を与えるライブストリーミングサービスを対象としたネットワーク資源の効果的な割り当て手法に着目する。

映像をリアルタイムに配信するライブストリーミングサービスの実現手法として、適応型ライブストリーミングが多く採用されている [6–8]。本研究では、適応型ライブストリーミングサービスの実現手法として、Moving Picture Experts Group (MPEG) が標準化した Dynamic Adaptive Streaming over HTTP (MPEG-DASH) [9] を対象とする。MPEG-DASHは、送信端末が送信する映像をエンコーダが圧縮符号化してセグメントファイルに分割（セグメント化）して受信端末に配信する。配信サーバは、受信端末がセグメントの受信に要した時間から、受信端末で利用可能なネットワーク資源を推定する。これにより、ネットワーク資源の変動に対応した映像品質のセグメントをサービス利用者へ提供可能となる。

このように、MPEG-DASHはネットワークの状況（利用可能なネットワーク資源）を考慮した品質制御手法である [9, 10]。そのため、各々の端末性能に応じた適切な映像品質を提供し、サービス利用者が満足するサービス品質を提供することは困難である。そこで既存手法では、映像の特性や再生バッファの占有率を考慮することで、サービス

<sup>1</sup> 東北大学大学院情報科学研究科  
Graduate School of Information Sciences, Tohoku University

<sup>2</sup> 東北大学サイバーサイエンスセンター  
Cyberscience Center, Tohoku University

a) medio@ci.cc.tohoku.ac.jp

b) beto@tohoku.ac.jp

c) suganuma@tohoku.ac.jp

利用者の体感品質を向上させた [11-14]。しかし、これらの手法では配信された高品質な映像の復号・再生に起因する計算機資源への影響が考慮されておらず、受信端末の性能（利用可能な計算機資源）が乏しい端末では高品質でのサービス提供は困難である。

そこで本研究では、ネットワークの状況及び受信端末の性能に適応したライブストリーミングサービスの実現を目的とする。しかし、ライブストリーミングサービスは種々の環境で利用され、使用する端末や配信映像の特性も様々であるため、事例ベースのような品質制御手法では、受信端末の性能・状況に適応した適切な映像品質を決定できるとは限らない。

本研究ではこの課題を解決するため、強化学習を用いた映像品質制御手法を提案する。具体的には、映像の特性、利用可能なネットワーク資源、計算機資源と、利用者へ配信される映像品質の関係を強化学習することで、ネットワークの状況だけでなく受信端末の性能にも適応した映像品質を決定する資源適応型映像品質制御機構を提案する。本稿では、資源適応型映像品質制御機構の提案と設計について述べ、実装したプロトタイプを用いた実験から、提案手法を用いることで資源の無駄な消費を抑えつつ高品質な映像を利用者へ配信可能であることを確認する。

## 2. 関連研究と課題

### 2.1 関連研究

MPEG-DASH における映像品質制御では、セグメントのデータ量と転送時間から受信端末で利用可能な帯域を配信サーバが推定し、その推定値に適応した品質（ビットレート）のセグメントを提供することが一般的である [10]。

帯域以外のネットワーク資源を考慮した手法として、再生バッファの占有率に基づいた手法がある [11,12]。これらの手法では、再生バッファに下限と上限の閾値を設定し、下限の閾値を下回った場合は映像の品質を劣化させ、上限の閾値を越えた場合は映像の品質を向上させる。これにより、再生バッファが枯渇することで生じる配信映像の再生中断回数を削減している。

映像の特性を考慮した手法として、文献 [13,14] では、映像が可変ビットレートで圧縮符号化されていることに着目している。利用可能な帯域と各セグメントのビットレートに基づいたネットワークの使用帯域を予測することで、帯域のみを考慮した場合と比較して高精度な予測が可能となり、より高品質なセグメントを受信端末に提供している。

### 2.2 課題

以上のように既存手法では、ネットワーク資源（帯域、バッファ占有率）や映像の特性（ビットレート）を考慮することで、サービス利用者の体感品質を向上させた。しかし、計算機資源が考慮されていないため、ネットワーク資

源に余裕がある環境においても映像の再生が中断してしまう可能性がある。近年、スポーツイベント等で高品質な4K/8K映像の配信が行われるが、これらの大容量な映像配信のためには、映像の符号化効率を高めるための映像圧縮アルゴリズムが用いられている。このような配信映像を受信端末が復号するためには、高い処理性能を備えた計算機資源を要するため、ライブストリーミングサービスにおいてネットワーク資源に加え計算機資源を考慮した映像品質制御手法が必要となる。

受信端末となるスマートフォンやタブレット端末等の携帯端末が備えることが可能な計算機資源及びネットワーク資源（以降、資源）には制限があり、各端末で利用可能な資源量は様々である。さらに、配信映像はその用途により映像の特性や求められる品質が異なるため、これらすべてのあらゆる状況に適応した映像品質制御の実現が不可欠となる。事例ベースによる映像品質制御では、あらかじめ利用可能な資源に対応した映像品質を定義する必要があるため、映像の特性とあらゆる資源の変動に適応した品質を決定できるとは限らない。したがって、受信端末の性能・状況に適応した適切な映像品質を決定するのは困難となる課題がある。

## 3. 強化学習を用いた資源適応型映像品質制御機構の提案

### 3.1 提案の概要

前章で述べた課題を解決するため本研究では、サービス利用時のネットワークの状況（利用可能なネットワーク資源）及び受信端末の性能（利用可能な計算機資源）に適応した映像品質を決定・制御するための映像品質制御手法を提案する。具体的には、強化学習を用いた資源適応型映像品質制御機構を提案し、課題の解決を図る。

資源適応型映像品質制御機構（以降、品質制御機構）は、映像の特性、利用可能なネットワーク資源、計算機資源と、利用者へ配信される映像品質の関係を強化学習することで、ネットワークの状況だけでなく受信端末の性能にも適応した映像品質を決定する。教師データが不要な強化学習を用いることで、新たな映像の特性や、端末性能を持つ受信端末が追加された場合においても逐次的に学習し、あらゆる状況に適応した映像品質が決定可能となる。なお、本研究では受信端末の利用者に対して高品質なサービス提供を行うため、送信端末及び配信サーバの負荷は考慮しないものとする。

### 3.2 品質制御機構の構成

図 1 に品質制御機構の構成を示す。品質制御機構は、映像の特性を解析するデータ解析機構、映像の特性からセグメントの長さを決定するセグメント長決定機構、受信端末で利用可能な資源に適応した映像品質を選択する資源学習

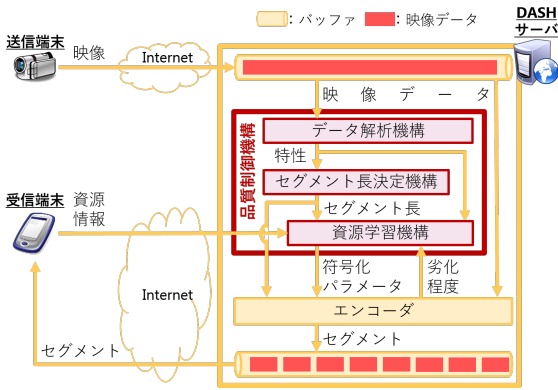


図 1: 品質制御機構の構成

機構から構成される。各機構の概要を以下に示す。

#### データ解析機構

送信端末が送信する映像を入力として、その特性を解析する。具体的には、映像を構成する複数フレーム単位でビットレート等の特性を解析する。これにより、映像の特性を考慮した映像品質を決定する。

#### セグメント長決定機構

データ解析機構で解析した映像の特性を入力として、セグメントの長さ（セグメント長）を決定する。セグメント長は映像再生時の受信端末の資源に影響を与えるため [15,16]、セグメント長を制御することにより、受信端末で利用可能な資源を考慮した映像品質を決定する。

#### 資源学習機構

資源学習機構は、データ解析機構で解析した映像の特性とセグメント長決定機構で決定したセグメント長、受信端末が送信する資源情報、エンコーダから獲得する圧縮符号化後の映像品質の劣化程度を入力として強化学習し、エンコーダが映像を圧縮符号化する際に設定する符号化パラメータを選択する。これにより、映像の特性、ネットワークの状況、受信端末の性能、圧縮符号化後の映像品質の劣化程度を総合的に考慮し、資源の無駄な消費を抑えつつ高品質な映像を利用者へ配信するための映像品質を決定する。

本稿では、資源学習機構に焦点を当てるため、次章では資源学習機構の詳細設計について述べる。

## 4. 設計

### 4.1 資源学習機構の設計

資源学習機構は、データ解析機構で判定した映像の特性  $char$  とセグメント長決定機構で決定したセグメント長  $len_{seg}$ 、受信端末の資源情報  $R$ 、圧縮符号化後の映像の劣化程度  $D$  に基づき、強化学習 [17] を用いて符号化パラメータ  $a$  を導出する。本稿では、 $a$  は映像解像度、フレームレート、画質で構成されるパラメータの集合とする。強化学習の枠組みに基づく資源学習機構の動作設計を以下に示す。

なお、学習を行う主体をエージェント (Ag) と呼ぶ。

(step 1) 受信端末が送信する計算機資源集合  $R_{CMP}$ 、ネットワーク資源集合  $R_{NW}$ 、データ解析機構とセグメント長決定機構で解析、決定した映像の性質集合  $C = \{char, len_{seg}\}$  から、Ag は受信端末の状態  $s$  を観測する。  $s$  の全体集合  $S$  を以下に定義する。

$$S = \{s_i \mid s_i = \langle R_{CMP_i}, R_{NW_i}, C_i \rangle; i = 1, 2, \dots, N_i\} \quad (1)$$

ここで、 $N_i$  は受信端末の端末の総数である。また、計算機資源集合  $R_{CMP}$  は以下の通りである。

$$R_{CMP} = \{cmp \mid cmp = \langle cpu, mem, bat \rangle\} \quad (2)$$

ここで、 $cpu(\in CPU)$  は CPU 使用率、 $mem(\in MEM)$  はメモリ使用率、 $bat(\in BAT)$  は充電残量である。符号化された映像の複号に必要な資源として CPU 使用率とメモリ使用率を考慮する。さらに、充電残量を考慮することで、充電残量が低い場合に低品質な映像を提供し、ライブストリーミングサービスを長時間継続可能とする。

ネットワーク資源集合  $R_{NW}$  は以下とする。

$$R_{NW} = \{nw \mid nw = \langle buf, rss, bwt \rangle\} \quad (3)$$

ここで、 $buf(\in BUF)$  は再生バッファの占有率、 $rss(\in RSS)$  は受信信号強度、 $bwt$  は受信端末の使用帯域である。再生バッファの占有率と使用帯域からネットワーク資源を、受信信号強度により通信の不安定さを推定する。これにより、利用可能なネットワーク資源の急激な変動に関して、映像の品質制御により対処すべきかどうかを判断する。

(step 2) Ag は、観測した受信端末の状態  $s$  から、受信端末の消費資源量を抑え、かつ、サービス利用者の体感品質が向上するような符号化パラメータ  $a$  を方策  $\pi$  に基づき選択する。  $a$  の全体集合  $A$  を以下に定義する。

$$A = \{a_j \mid a_j = \langle rsl, fps, crf \rangle; j = 1, 2, \dots, N_j\} \quad (4)$$

ここで、 $rsl(\in RSL)$  は映像解像度、 $fps(\in FPS)$  はフレームレート、 $crf(\in CRF)$  は映像を圧縮符号化する際に指定する画質の程度 Constant Rate Factor (CRF) である。また、 $N_j$  は符号化パラメータの総数である。

(step 3) step 2 で決定した  $a$  に基づき受信端末の状態に与えた影響として Ag に報酬  $rwd$  を与える。  $rwd$  は、計算機資源への負荷  $\Delta R_{CMP}$ 、ネットワーク資源への負荷  $\Delta R_{NW}$ 、圧縮符号化後の映像品質の劣化程度  $\Delta D$  を用いて以下の式で求める。

$$rwd = -\Delta R_{CMP} - \Delta R_{NW} - \Delta D \quad (5)$$

このとき、 $\Delta R_{CMP}$ 、 $\Delta R_{NW}$  が増加すると、それぞれ計算機資源、ネットワーク資源の消費量が大きくなるため、 $rwd$  は小さい値を設定する。同様に、 $\Delta D$  が増加した場合、圧縮符号化後の映像の品質劣化が大きくなり、サービスの利用者

**Algorithm 1** Q 学習に基づく資源学習機構

- 1: Initialize Q values  $Q(s, a)$  for all state-action pairs to 0.0
- 2: **repeat**
- 3: Get resources ( $R_{CMP_t}$  and  $R_{NW_t}$ ) from the Receiver at time  $t$
- 4: Observe a state  $s_t \in S$
- 5: Determine an encoding parameter  $a_t \in A$  according to the policy  $\pi$
- 6: Send  $a_t$  to the Encoder
- 7: Get resources ( $\Delta R_{CMP_{t+1}}$  and  $\Delta R_{NW_{t+1}}$ ) from the Receiver at time  $t+1$
- 8: Get a quality degradation  $\Delta D_{t+1}$  from the Encoder
- 9: Calculate a reward  $rw_{d_{t+1}}$  according to equation (5)
- 10: Observe the next state  $s_{t+1}$
- 11: Update the Q value  $Q(s_t, a_t)$  according to equation (6)
- 12: Transition to next time  $t \leftarrow t+1$
- 13: **until** the streaming service is finished.

に対して与えるサービス品質が劣化するため、 $rw_{d}$  は小さい値を設定する。

以上の動作により、 $Ag$  は  $rw_{d}$  が最大となるような  $a$  を選択していくことで、映像の特性、受信端末の利用可能な資源、圧縮符号化後の映像品質の劣化程度を総合的に考慮した  $a$  を選択可能となる。

#### 4.2 Q 学習を適用した資源学習機構の設計

本稿では、強化学習アルゴリズムとして Q 学習 [18] を使用する。Q 学習は、 $Ag$  が符号化パラメータ  $a$  を決定する際の報酬  $rw_{d}$  の期待値を Q 値として推定し、Q 値を最大化するような行動選択を行うアルゴリズムである。

Algorithm 1 に Q 学習に基づく資源学習機構の疑似コードを示す。Q 値はすべての端末の状態  $s$  と  $a$  の組み合わせ数だけ保存する。そして時刻  $t$  において観測した状態  $s_t$  から方策  $\pi$  に基づき符号化パラメータ  $a_t$  を選択する。このとき、 $\pi$  は  $\epsilon$ -greedy アルゴリズムに基づく決定方法とする。具体的には、確率  $\epsilon$  で受信端末が利用可能な帯域に適応した  $a_t$  を、確率  $(1-\epsilon)$  で Q 値が最大となる  $s_t$  に対応した  $a_t$  を選択する。次に、受信端末の資源情報である  $\Delta R_{CMP_{t+1}}, \Delta R_{NW_{t+1}}$  とエンコーダから獲得する圧縮符号化後の映像品質の劣化程度  $\Delta D_{t+1}$  から式 (5) に基づき報酬  $rw_{d_{t+1}}$  を求める。そして、状態  $s_{t+1}$  を再度観測し、以下の式で時刻  $t$  における Q 値  $Q_t (= Q(s_t, a_t))$  を更新する。

$$Q_t \leftarrow Q_t + \alpha \{rw_{d_{t+1}} + \gamma \max_{a' \in A} Q_{t+1} - Q_t\} \quad (6)$$

ここで、 $\alpha$  は学習率、 $\gamma$  は割引率であり、ともに値の範囲は  $(0.0, 1.0]$  である。一般に、学習率は Q 値の更新の程度を制御し、0.0 に近い値が用いられる。また、割引率は次の状態における Q 値を割引くためのパラメータで、1.0 に近い値が用いられる。



図 2: 実験環境

表 1: 実験で使った映像

パラメータ名	パラメータ
名称	Big Buck Bunny
ファイル	bbb_sunflower_2160p_60fps_normal.mp4
符号化形式	H.264/MPEG-4 AVC
サイズ $size_{BBB}$	657.4 (MB)
再生時間 $len_{BBB}$	634 (s)
動画解像度	2160p
フレームレート	60 (fps)

## 5. 実験と評価

### 5.1 実験概要

前章の設計に基づき品質制御機構を実装し、実装したプロトタイプを用いて品質制御機構の動作を確認する実験を行った。実験では、品質制御機構が受信端末で利用可能な資源の無駄な消費を抑えつつ高品質な映像を利用者へ配信可能であることを確認するため、利用可能な帯域のみで符号化パラメータを選択する単純手法と、品質制御機構を用いて符号化パラメータを選択する提案手法を比較した。

図 2 に実験環境を示す。実験では送信端末を使用せず、DASH サーバ内に配置した映像を受信端末に配信した。配信映像は、MPEG-DASH 関連の研究で広く用いられている 3DCG アニメーション映像 Big Buck Bunny (BBB) [19] を使用した。表 1 に BBB の概要を示す。受信端末はスマートフォン (Nexus 5) を使用し、Wi-Fi を用いて DASH サーバへ接続した。

実験手順は以下の通りである。実験手順 (3) において、配信映像の再生時間に応じて利用可能な帯域と CPU 使用率を変動させることで、受信端末で利用可能な資源に適応した符号化パラメータを決定可能であることを確認した。

- (1) 配信サーバが配信映像をすべての品質 (=  $|A|$  個) で圧縮符号化、セグメント化することで、圧縮符号化後の映像品質の劣化程度を獲得
- (2) 受信端末がすべての品質で圧縮符号化された映像を再生することで、資源情報を獲得
- (3) 獲得した映像品質の劣化程度と資源情報を入力として資源学習機構が符号化パラメータを選択
  - (3-1) 50 秒後から 100 秒間、利用可能な帯域を制限
  - (3-2) 200 秒後から 100 秒間、受信端末の CPU 使用率を増加

表 2: 実験で使したパラメータ

集合	パラメータ名	パラメータ
$R_{CMP}$	$CPU$	{0, 25, 50, 75, 100} (%)
	$MEM$	{0, 50, 100} (%)
$R_{NW}$	$BUF$	{0, 100} (%)
$I_{DAT}$	$char$	$size_{BBB} \times 8 / len_{BBB}$ (Mbps)
	$len_{seg}$	2 (s)
A	$CRF$	{18, 23, 28}
	$FPS$	{60, 30, 24, 20} (fps)
	$RSL$	{2160p, 1080p, 720p, 360p, 180p}
N/A	$\alpha$	0.1
	$\gamma$	0.9
	$\epsilon$	0.1

(3-3)350 秒後から 100 秒間, 利用可能な帯域を制限し, CPU 使用率を増加

評価指標は, 画質評価指標 Structural SIMilarity (SSIM) [20] の平均  $SSIM_{avg}$ , 映像を再生しているプロセスの平均 CPU 使用率  $CPU_{avg}$ , 累計ドロップフレーム数  $DROP_{avg}$  とした. SSIM は, 値が 1.0 に近いほど元映像からの劣化が少ないとされる指標である.

表 2 に実験で使したパラメータを示す. 実験では, データ解析機構が解析する映像の特性  $char$  と, セグメント長決定機構が決定するセグメント長  $len_{seg}$  は手動で設定した値を使用した.

計算機資源への負荷  $\Delta R_{CMP}$  は以下の式で求めた.

$$\Delta R_{CMP} = \Delta cpu + \Delta mem \quad (7)$$

ここで,  $\Delta cpu, \Delta mem$  は映像を再生するアプリケーションの CPU 使用率とメモリ使用率である. また, 確率  $\epsilon$  で利用可能な帯域  $bwt$  を考慮した符号化パラメータ  $a$  を選択するため, ネットワーク資源への負荷  $\Delta R_{NW}$  はバッファの占有率  $\Delta buf$  のみ考慮した. 実験は, スマートフォンを定常状態で充電しながら行ったため, 充電残量  $bat$  と受信信号強度  $rss$  は考慮しなかった.

符号化後の映像品質の劣化程度  $\Delta D$  は, 選択した符号化パラメータ  $a_t$  中の  $fps_t (\in FPS)$  に対して受信端末が配信映像を 1 秒間で画面に表示したフレーム数の割合  $\Delta fps$ ,  $fps_t$  に対して受信端末が配信映像を 1 秒間で画面に表示せずドロップしたフレーム数の割合  $\Delta drop$ , 再生中の配信映像の SSIM である  $\Delta ssim$  を用いて以下の式で求めた.

$$\Delta D = \Delta fps + \Delta drop + \Delta ssim \quad (8)$$

## 5.2 実験結果

図 3, 図 4 に受信端末で利用可能な帯域と CPU 使用率の変動を示す. また, 表 3 に評価指標を用いて単純手法と提案手法を比較した結果を示す. 図 3, 図 4 の横軸は時

表 3: 実験結果 (評価指標を用いた比較)

	単純手法	提案手法	減少率 (%)
$SSIM_{avg}$	0.990	0.989	0.101
$CPU_{avg}$ (%)	72.192	39.520	45.257
$DROP_{total}$ (frames)	7028	2324	66.932

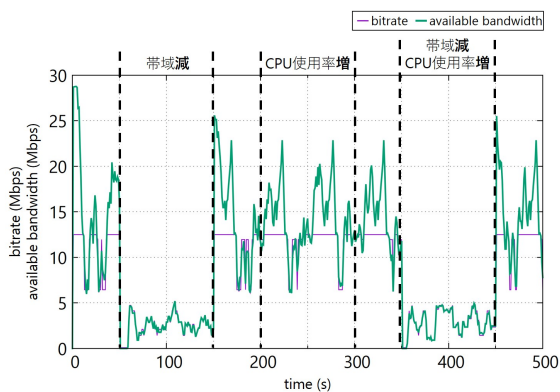
間 (time) である. 図 3 の縦軸は資源学習機構が選択した符号化パラメータ (映像品質) に対応する映像ビットレート (bitrate), 利用可能な帯域 (available bandwidth) である. また, 図 4 の縦の第一軸は資源学習機構が選択した符号化パラメータに対応する映像ビットレート (bitrate), 第二軸は CPU 使用率 (CPU usage) である.

図 3 より, 利用可能な帯域が減少した際, 単純手法は帯域に適応した符号化パラメータを決定し続けたのに対し, 提案手法は低ビットレートの符号化パラメータを決定するようになった. また図 4 より, 単純手法は, 受信端末の総 CPU 使用率が増加した際, セグメントの再生プロセスの CPU 使用率は高い状態が見られ, 高品質な符号化パラメータを決定し続けた. これに対し提案手法は, セグメントの再生プロセスの CPU 使用率が低くなるような低ビットレートの符号化パラメータを決定するようになった. さらに表 3 より, 提案手法は単純手法と比較して平均 SSIM は 0.101%, 平均 CPU 使用率は 45.257%, 累計フレームドロップ数は 66.932% 減少したことを確認した.

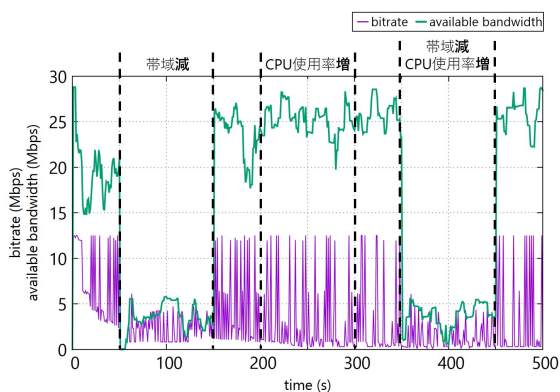
## 5.3 考察

受信端末で利用可能な帯域と CPU 使用率の変動を示した実験結果より, 提案手法は利用可能な帯域が減少した際には, 利用可能な帯域に余裕を持たせるために低ビットレートの符号化パラメータを徐々に決定するようになり, 総 CPU 使用率が増加した際には, 配信映像の再生負荷を軽減するために低ビットレートの符号化パラメータを徐々に決定するようになった. これは, 学習の初期段階で低ビットレートの符号化パラメータを選択することでエージェントが得られる報酬が大きくなる (消費資源量が少ない, 圧縮符号化後の映像品質の劣化が少ない) ことを学習したためである.

評価指標を用いて比較した実験結果より, 提案手法は単純手法よりも SSIM が減少し画質の劣化が見られたが, 平均 CPU 使用率と累計フレームドロップ数は減少し, 消費資源量を削減した. 単純手法は平均 CPU 使用率と累計フレームドロップ数が大きいことから, 配信映像の再生時に頻繁にバッファリングが発生し, スムーズな再生ができなかった一方で, 提案手法は, 平均 CPU 使用率と累計フレームドロップ数が小さいため, 配信映像の再生に問題は生じなかった. また, 低ビットレートの符号化パラメータを決定したことで SSIM が減少したが, その減少率は非常に小

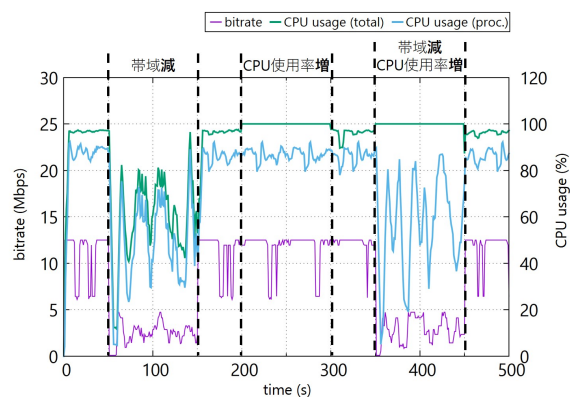


(a) 単純手法

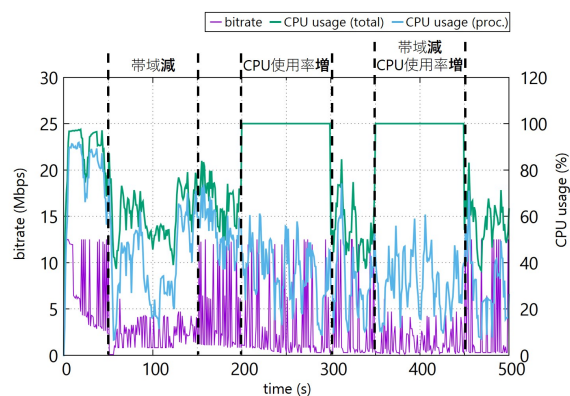


(b) 提案手法

図 3: 実験結果 (受信端末で利用可能な帯域)



(a) 単純手法



(b) 提案手法

図 4: 実験結果 (受信端末の CPU 使用率)

さく、利用者が携帯端末上でサービスを楽しむ上では問題ないと考えられる。

以上より、映像の特性、受信端末で利用可能な資源、符号化後の映像品質の劣化程度との関係を強化学習した提案手法を用いることで、受信端末の資源の無駄な消費を抑えつつ高品質な映像を利用者へ配信可能であることを確認した。

## 6. おわりに

本研究では、適応型ライブストリーミングサービスの実現手法の1つである、Moving Picture Experts Group (MPEG) が標準化した Dynamic Adaptive Streaming over HTTP (MPEG-DASH) を対象とし、ネットワークの状況及び受信端末の性能に適応したライブストリーミングサービスを実現するため、強化学習を用いた映像品質制御手法を提案した。具体的には、映像の特性、利用可能なネットワーク資源、計算機資源と、利用者へ配信される映像品質の関係を強化学習することで、映像をリアルタイムに配信するライブストリーミングサービスを利用する様々な環境に適応した映像品質を決定する資源適応型映像品質制御機構を提案した。実験結果から、提案手法を用いることで資源の無駄な消費を抑えつつ高品質な映像を利用者へ配信可能であることを確認した。

今後は、データ解析機構とセグメント長決定機構の詳細

設計を行い、これらの機構が配信映像と受信端末で利用可能な資源に応じて適切な映像の特性とセグメント長を決定可能であることを確認する。さらに、品質制御機構を実際のライブストリーミングサービスに適用し、実端末を用いた実験により、提案手法の有効性を評価する。

## 参考文献

- [1] 総務省：平成 29 年版情報通信白書，総務省（オンライン），入手先 <http://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h29/pdf/n2100000.pdf>（参照 2018-07-23）。
- [2] Cisco Systems, Inc.: Cisco Visual Networking Index: Forecast and Methodology, 2016–2021, Cisco Systems, Inc. (online), available from <https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.pdf> (accessed 2018-07-23)。
- [3] Kesidis, G., Urgaonkar, B., Nasiriani, N. and Wang, C.: Neutrality in Future Public Clouds: Implications and Challenges, *Proc. 8th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 2016)*, USENIX Association, pp. 1–6 (2016)。
- [4] 伊藤正史：現行地上波で HD/4K サイマル視聴サービスを実現するハイブリッドキャストコンテンツ（4K ランドスケープ），映像情報メディア学会誌，Vol. 71, No. 4, pp. J125–J130 (2017)。
- [5] KDDI 株式会社：『EAFF E-1 サッカー選手権 2017

- 決勝大会』で国内初となるスタジアムでの VR 観戦など新しいサッカー観戦体験！「au Future Stadium」, KDDI 株式会社 (オンライン), 入手先 (<https://www.au.com/information/topic/mobile/2017-082/>) (参照 2018-07-23).
- [6] YouTube: Ten years of YouTube video tech in ten videos, YouTube (online), available from (<https://youtube-eng.googleblog.com/2015/05/ten-years-of-youtube-video-tech-in-ten.html>) (accessed 2018-07-23).
- [7] Park, A. and Watson, M.: The Netflix Tech Blog: HTML5 Video at Netflix, Netflix (online), available from (<http://techblog.netflix.com/2013/04/html5-video-at-netflix.html>) (accessed 2018-07-23).
- [8] 馬場美由紀: AbemaTV がリニア型配信で「MPEG-DASH」をサポートした理由と、その使い方とは?, CodeIQ (オンライン), 入手先 (<http://getnews.jp/archives/1973058>) (参照 2018-07-23).
- [9] ISO/IEC: Information technology – Dynamic adaptive streaming over HTTP (DASH) – Part 1: Media presentation description and segment formats.
- [10] Li, Z., Zhu, X., Gahm, J., Pan, R., Hu, H., Begeen, A. C. and Oran, D.: Probe and Adapt: Rate Adaptation for HTTP Video Streaming At Scale, *IEEE Journal on Selected Areas in Communications*, Vol. 32, No. 4, pp. 719–733 (online), DOI: 10.1109/JSAC.2014.140405 (2014).
- [11] Huang, T., Johari, R., McKeown, N., Trunnell, M. and Watson, M.: A Buffer-Based Approach to Rate Adaptation: Evidence from a Large Video Streaming Service, *Proc. ACM conference on Special Interest Group on Data Communication (SIGCOMM 2014)*, ACM Press, pp. 187–198 (2014).
- [12] Juluri, P., Tamarapalli, V. and Medhi, D.: QoE Management in DASH Systems Using the Segment Aware Rate Adaptation Algorithm, *Proc. IEEE/IFIP Network Operations and Management Symposium (NOMS 2016)*, IEEE, pp. 129–136 (2016).
- [13] Lee, S., Lee, E. and Lee, H.: Quality adaptation scheme for improving QoE of MPEG DASH, *Proc. International Conference on Information and Communication Technology Convergence (ICTC 2016)*, IEEE, pp. 368–370 (2016).
- [14] Kim, M., Park, J. and Chung, K.: Content-aware rate adaptation scheme to improve stability in HTTP Adaptive Streaming, *Proc. International Conference on Information Networking (ICOIN 2017)*, IEEE, pp. 401–405 (2017).
- [15] Petrangeli, S., Bouten, N., Dejonghe, E., Famaey, J., Leroux, P. and De Turck, F.: Design and Evaluation of a DASH-compliant Second Screen Video Player for Live Events in Mobile Scenarios, *Proc. IFIP/IEEE International Symposium on Integrated Network Management (IM 2015)*, IEEE, pp. 894–897 (2015).
- [16] Hassan, Y. M., Helmy, A. and Rehan, M. M.: Effect of Varying Segment Size on DASH Streaming Quality for Mobile User, *Proc. International Conference on Engineering and Technology (ICET 2014)*, IEEE, pp. 1–4 (2014).
- [17] 牧野貴樹, 澁谷長史, 白川真一 ほか: これからの強化学習, 森北出版 (2016).
- [18] Watkins, C. J. C. H. and Dayan, P.: Q-Learning, *Machine Learning*, Vol. 8, No. 3, pp. 279–292 (online), DOI: 10.1007/BF00992698 (1992).
- [19] Blender Foundation: Big Buck Bunny, Blender Foundation (online), available from (<https://peach.blender.org/>) (accessed 2018-07-23).
- [20] Wang, Z., Bovik, A., Sheikh, H. and Simoncelli, E.: Image Quality Assessment: From Error Visibility to Structural Similarity, *IEEE Transactions on Image Processing*, Vol. 13, No. 4, pp. 600–612 (online), DOI: 10.1109/TIP.2003.819861 (2004).