

モバイルネットワークにおけるビットレートマップを用いた可変ビットレート動画のための動画アダプティブストリーミング方式

有友 大輔[†] 木村 成伴[‡]

筑波大学 情報学群 情報メディア創成学類[†] 筑波大学 システム情報系 情報工学科[‡]

1 はじめに

近年、スマートフォンやタブレット端末をはじめとする、小型で軽量なモバイルデバイスの普及に伴い、これらのデバイスに向けた様々なネットワークサービスが提供されている。これらのネットワークサービスを、ユーザは様々なネットワーク環境下で、移動しながら利用することが多いことから、通信の切断やハンドオーバーなどが発生しやすく、ネットワークが不安定な状況下においても安定した通信を提供することが重要な課題となっている。

これを達成するため、ネットワークサービスにおける最も一般的な動画配信では、再生する動画の品質を、ネットワークの状況に応じて、ユーザが事前に選択していたが、一般のユーザにとってこれらを適切に選択するのは難しく、また、ネットワーク状況の変化に追従するのも困難であった。

そこで、再生中に、ネットワークで利用可能な帯域などに応じて、動画の品質を動的に切り替える HTTP アダプティブストリーミング方式が注目されている。

この方式では、動画配信サーバにおいて、あらかじめ、複数の品質の動画ストリームを作成しておき、それらを数秒ごとのセグメントに分割しておく。また、それぞれの品質の平均ビットレートは再生の開始前にクライアントに通知される。

クライアントは、HTTP を用いて、先頭セグメントから順次ダウンロードしていき、これらを再生する。このとき、ネットワーク帯域などを勘案して、どの品質のセグメントをダウンロードするかを選択するが、その選択アルゴリズムによって再生される動画の品質が変動し、ユーザの視聴体験に大きな影響を与える。

このため、この選択アルゴリズムが HTTP アダプティブストリーミングの根幹をなし、様々なアルゴリ

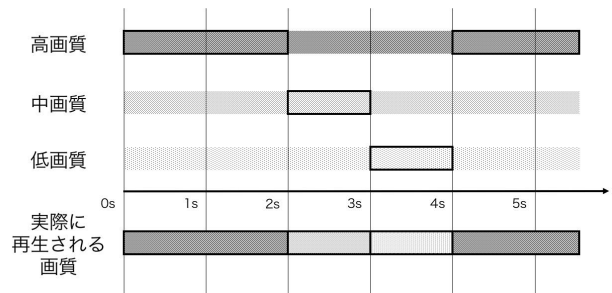


図 1: 高→中→低→高画質と遷移するストリーミング

ズムが提案されている。しかし、これらのアルゴリズムでは、モバイルネットワークにおいて、圧縮効率を高めるために用いられた可変ビットレート (VBR: Variable Bit Rate) 動画を再生する場合は問題があった。

2 動画の品質の選択アルゴリズム

HTTP アダプティブストリーミング方式のために提案されている、セグメントの品質 (ビットレート) の主な選択アルゴリズム [1] として一般的なものとして、クライアントが使用可能な帯域を推定し、それを越えない最大限のビットレートのストリームを選択する Throughput-Based アルゴリズムがある [2]。このアルゴリズムは、動画のビットレートが常に一定のビットレートである (CBR: Constant Bit Rate) ことを前提としているため、このアルゴリズムで可変ビットレート (VBR: Variable Bit Rate) で圧縮されている動画を扱うと、その平均ビットレートを基準としてストリームを選択してしまう。このため、平均値より高いビットレートのセグメントがあると、そのダウンロード時間が推定以上にかかり、これが頻繁に続くと、バッファが枯渇し、再生が停止するという問題があった。

3 提案方式

前章で述べたアルゴリズムの問題点を解決するため、VBR 動画にも対応できるように Throughput-Based アルゴリズムを改良した方式を提案する。

An Adaptive Streaming System Method Using Bitrate Maps for VBR Videos over Mobile Networks

[†]Daisuke Aritomo, College of Media Arts, Science and Technology of Informatics, University of Tsukuba

[‡]Shigetomo Kimura, Faculty of Engineering, Information and Systems, University of Tsukuba

表 1: 実験の結果

アルゴリズム	許容ビットレートを超えた時間
従来方式	40 秒
提案方式	13 秒

提案方式では、動画から、複数のビットレートのセグメントを作成する際に、すべてのセグメントのサイズなどを含むビットレートマップを作成しておく。そして、動画を再生するクライアントは、以下の手順でダウンロードするセグメントのビットレートを選択する。

1. 動画の再生開始前に、サーバからビットレートマップを要求し、選択可能な N 種類の動画品質（ビットレート $B_0 < B_1 < \dots < B_{N-1}$ ）を取得する。
2. 初期ビットレートを B_0 とし、再生開始までに蓄積が必要な数のセグメントをダウンロードして、これをバッファに格納する。このダウンロードにかかった時間を計測し、そのスループットから利用可能な帯域 $B_{\text{throughput}}$ を推定する。
3. ダウンロードするセグメントのビットレートをビットレートマップから取得し、 $B_{\text{throughput}}$ を超えない最大のビットレート B_i のセグメントをサーバにリクエストする。
4. 手順 3. にかかった時間を計測し、現在の利用可能帯域 $B_{\text{throughput}}$ を更新した上で、手順 3. に戻る。

提案方式では、ビットレートマップの導入によって、クライアントは次にダウンロードするセグメントの正確なサイズが得られるため、より正確なダウンロード時間を推定することができる。これにより、バッファの枯渇を防ぎつつ、現在のネットワークの状況で可能な、最も高いビットレートのセグメントを選択することが可能となる。

4 評価実験

本章では、提案したアルゴリズムの有効性を確認するための評価実験を行う。

HTTP アダプティブストリーミングを実装したクライアントをインストールした仮想マシンと、提案方式および文献 [2] の Throughput-Based アルゴリズム (以下、従来方式と呼ぶ) を実装したサーバをインストールした仮想マシンを、ネットワークシミュレータ ns-3

を用いて IEEE802.11g で接続した。端末は地下鉄での利用を想定し、無線 LAN の利用可能な帯域は、実験開始直後から 1 分後までは 54Mbps、それ以降は 1 分おきに 0Mbps と 54Mbps を交互させ、クライアントに動画をダウンロードさせた。ここで、セグメントあたりの秒数 4 秒、バッファ長 16 秒 (セグメント数 4) とし、初期バッファに蓄積するセグメント数は 4 とする。また、再生した動画は 10 分 (600 秒) のアニメーション映像であり、動画の解像度と平均ビットレートは 320x240 (200kbps と 400kbps), 480x360 (600kbps), 640x360 (800kbps と 1000kbps), 768x432 (1.5Mbps), 1024x576 (2.5Mbps), 1280x720 (4Mbps), 1920x1080 (8Mbps), 3840x2160 (12Mbps) の 10 段階を用意した。

そして、1 秒おきに、その時点でダウンロードを試みているビットレートと、利用可能なネットワーク帯域とを比較し、前者が後者を超えた時間を計測した結果を表 1 に示す。この結果、提案方式は、従来方式に比べて、ネットワークが許容できないビットレートのダウンロードを試みている時間は半分未満になり、それによるバッファの枯渇・リバッファリングに伴う再生の停止を防ぐ効果が高いことを確認できた。

5 まとめ

本論文では、モバイルネットワークを対象に、ビットレートマップを導入し、VBR 動画に対応できるセグメントのビットレート選択アルゴリズムを提案した。そして、評価実験によって従来方式よりもネットワークの変化に適切に追従できることを示した。今後は、ネットワーク環境や動画の種類を変えて実験により、提案方式を評価する予定である。

参考文献

- [1] Karagkioules, T., Concolato, C., Tsilimantou, D. and Valentin, S.: A Comparative Case Study of HTTP Adaptive Streaming Algorithms in Mobile Networks, *Proceedings of the 27th Workshop on Network and Operating Systems Support for Digital Audio and Video, NOSSDAV'17*, ACM, pp. 1–6 (2017).
- [2] Beben, A., Wiśniewski, P., Batalla, J. M. and Krawiec, P.: ABMA+: Lightweight and Efficient Algorithm for HTTP Adaptive Streaming, *Proceedings of the 7th International Conference on Multimedia Systems, MMSys '16*, ACM, pp. 2:1–2:11 (2016).