

無線環境における異なる TCP 上の MPEG-DASH/ HTTP Live Streaming の性能比較

武藤健史[†] 野崎寛也[†] 金井謙治[†] 大木哲史[†] 甲藤二郎[†]

モバイルビデオは 2017 年までにモバイルデータトラフィックの 66%以上を占める見込みであると Cisco から発表された。[1]このような現状においてストリーミング技術にたくさんの改善が求められる。近年では HTTP Streaming が主流となっており、ライブ放送を想定した HTTP Live Streaming の報告が行われ、さらに動的かつアダプティブにビットレートを変化させる MPEG Dynamic Adaptive Streaming over HTTP (MPEG-DASH) の国際標準化も完了している。本研究ではモバイル環境を想定し、異なる TCP 上 (TCP-Reno/ TCP-Vegas/ CUBIC-TCP) で MPEG-DASH/ HTTP Live Streaming の性能比較を行った。アダプティブにビットレートを変化させる新しいストリーミング技術の特性を確認することができた。

Performance Comparison of MPEG-DASH/ HTTP Live Streaming over different TCPs in wireless networks

Takeshi Muto[†] Tomoya Nozaki[†] Kenji Kanai[†] Tetsushi Ohki[†] Jiro Katto[†]

1. はじめに

近年、ストリーミング技術として HTTP Streaming が主流となっている。従来のストリーミング技術であった RTP や RTSP は通常品質を保証しない UDP (エラー処理やフロー制御を行わない) を想定し、Web サーバーとは異なる専用サーバーを必要とした。しかし、最近のネットワークの高速化と CDN (Content Delivery Network) の普及を背景に、HTTP Streaming の利点が顕著になりつつある。TCP を用いることでパケットロスの問題を解決し、HTTP のポート 80 番を用いることでファイヤーウォールのブロックの問題を解決し、かつ通常の Web サーバーを用いることができる。ライブ放送を想定した HTTP Live Streaming の報告も行われ、さらに、動的かつアダプティブにビットレートを変化させる MPEG Dynamic Adaptive Streaming over HTTP (MPEG-DASH) の国際標準化も完了している。

また現在のインターネット通信では TCP/IP が用いられているが、一番標準的な TCP である TCP-Reno は 1Gbps を超えるような広帯域を効率良く利用することができない。この問題を解決するべく、様々な輻輳制御方式が提案され、それらはパケットロスに応じて輻輳制御を行う Loss-Based 方式、RTT の計測値と推測値を用いてバッファ内で輻輳制御を行う Delay-Based 方式、また Loss-Based 方式と Delay-Based 方式を合わせた Hybrid 方式の 3 つに大きく分類することができる。

本研究では MPEG-DASH と HTTP Live Streaming を想定して評価実験を行った。いくつかある輻輳制御方式のうち、

Loss-based 手法で多くの OS に搭載されていた TCP-Reno、Delay-based 手法の TCP である TCP-Vegas、Loss-based 手法で、Linux が現在の標準の TCP としている CUBIC-TCP を評価対象としている。

2. HTTP Live Streaming

HTTP Live Streaming は Apple によって提案され、Web サーバーから HTTP を用いてビデオ・オーディオをライブ配信し、iPhone、iPad など主に iOS デバイスでの再生を想定している。また、ライブ放送、オンデマンド配信の両方に対応しているストリーミング技術である。手順としては、H.264 と AAC でエンコードを行い、MPEG-2 トランスポートストリームで多重化する。これはさらにセグメントにより小さなセグメントファイル(.ts)に分割され、メディアファイルのリストを含む index ファイル(.m3u8)を生成する。クライアントは始めにこの index ファイルを取得し、そこに記載されたセグメントを取得することによって動画再生が可能となる。ライブ配信時は index ファイルが随時更新され、クライアントは新たに追加されるセグメントの取得を継続する。図 2.1 に HTTP Live Streaming のストリーミング構成を示す。

[†]早稲田大学
Waseda University

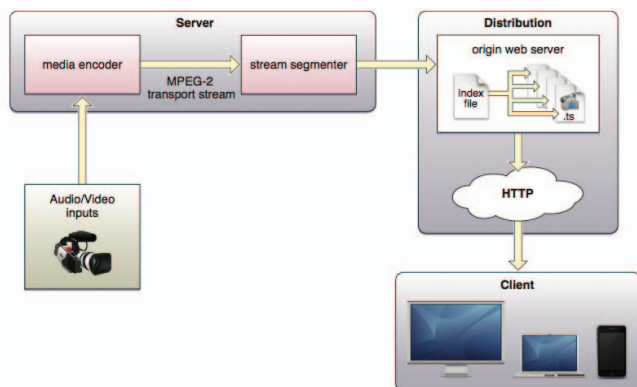


図 2.1 HTTP Live Streaming の設定[2]

上記の図に示してあるように、HTTP Live Streaming はサーバーコンポーネント、ディストリビューションコンポーネント、クライアントソフトウェアの3つのパートから構成される。サーバーではメディアをエンコードし、エンコードされたメディアを分割、またそれらをファイルとして保存する。ディストリビューションシステムはメディアファイルとインデックスファイルを HTTP でクライアントに配信するウェブサーバーかウェブキャッシュシステムである。クライアントはサーバーに対してインデックスファイルを要求し、そのファイルに掲載されているメディアファイルを順番にダウンロードする。

3. MPEG-DASH

HTTP Streaming において、再生が途切れない(再生中にアンダーフロー状態にならない)ようにネットワークの帯域幅を観測しながら動的にビットレートを変化させるストリーミング技術であり、MPEG において標準化が行われた。実装例として DASHEncoder[3]が知られており、H.264 と AAC でエンコードを行い、MP4 ファイルを生成、その MP4 ファイルを.m4s セグメントファイルに分割し、インデックスファイル(MPD)を生成する。図 3.1 に示すようにクライアントは始めにこの index ファイルを取得し、m4s セグメントを逐次取得することで、動画再生が可能となる。また、複数ビットレートでエンコードを行い、ネットワークの状況に応じてクライアントにビットレート選択を行わせることで、適応制御を実現する。

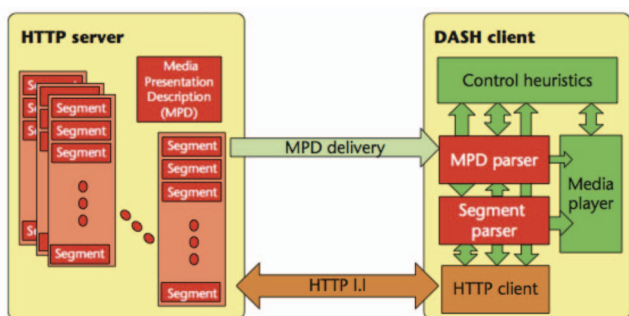


図 3.1 MPEG-DASH の構成[4]

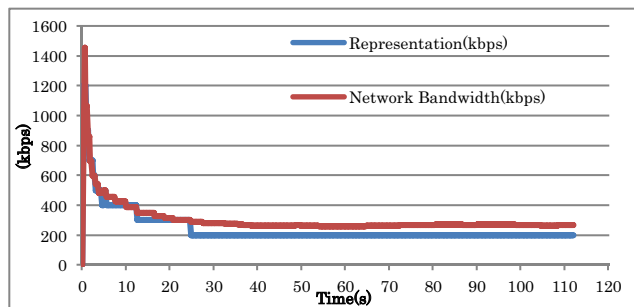


図 3.2 ネットワーク帯域と Representation の関係

図 3.2 は MPEG-DASH 配信におけるネットワーク帯域と Representation(指定したビットレート)の関係を表す。MPEG-DASH はネットワーク帯域を観測し、それに応じて適切なビットレートを選択する適応制御を実現している。

4. MPEG-DASH/ HTTP Live Streaming を用いた評価実験と性能比較

Server

MPEG-DASH のデータ(DASHEncoder を用いてエンコード、分割された.m4s ファイルと MPD ファイル)と HTTP Live Streaming のデータ(ffmpeg を用いてエンコードし、Segmenter によって分割された.ts ファイルと m3u8 ファイル)を格納した。

Client

VLC Media Player を用いた。VLC Media Player はマルチメディアプレイヤーであり、HTTP Live Streaming や MPEG-DASH など新しいアダプティブストリーミングプロトコルに対応している。

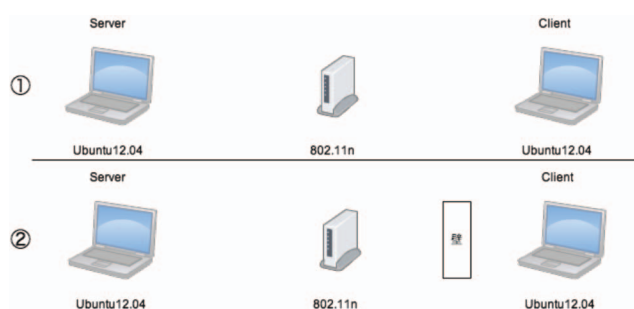


図 4.1 ネットワークトポロジー

サーバーとクライアントは図 4.1 のように 802.11n 規格で接続をし、クライアントの位置を変更し、以下の2通りにおいて実験を行った。

- ・ 無線通信状態が良い場合
(クライアントの位置が AP から 5 メートルの位置)
- ・ 無線通信状態が悪化した場合(伝送レートが低い時)
(クライアントの位置が AP から 5 メートルの位置で壁

一枚が障害物としてある場合)

4.1 データセット

	HTTP Live Streaming	MPEG-DASH
Source (Length)	Animation: Elephants Dream (10:54)	
Resolution	720P	
Segment Length	2s	
Specified Bitrate	1.5Mb/s	Minimum: 100kb/s Maximum: 1.5Mb/s
Segment Size	約 3.2Mb	約 3.8Mb

なお、MPEG-DASH は 100kb/s, 200kb/s, 300kb/s, 400kb/s, 500kb/s, 600kb/s, 700kb/s, 800kb/s, 900kb/s, 1Mb/s, 1.2Mb/s, 1.5Mb/s で複数のビットレートエンコードを行った。

4.2 実験結果

本実験は 2 つの実験環境の下で HTTP Live Streaming と MPEG-DASH 各々の輻輳ウィンドウサイズ、ラウンドトリップタイム、スループットの 3 つのパラメータ値を取得した。

4.2.1 輻輳ウィンドウサイズ(Congestion Window)

図 4.2、4.3 に障害物がある場合における MPEG-DASH/ HTTP Live Streaming の輻輳ウィンドウサイズの結果を示す。

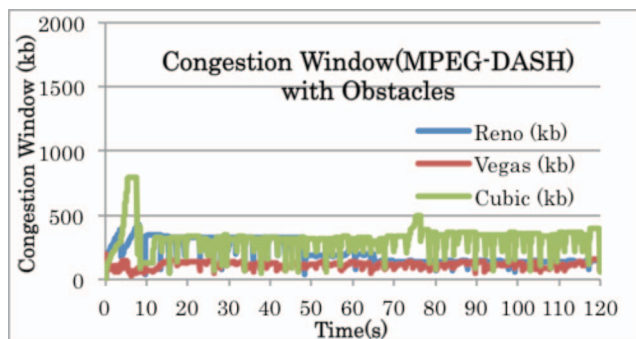


図 4.2 輻輳ウィンドウサイズ(MPEG-DASH)障害物あり

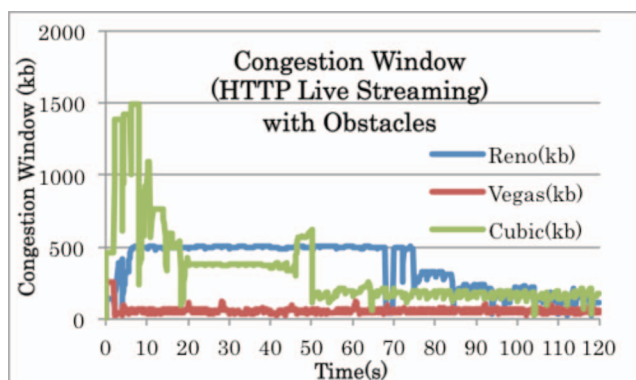


図 4.3 輻輳ウィンドウサイズ(HTTP Live Streaming) 障害物あり

図 4.2 より MPEG-DASH は 3 つの TCP 上で輻輳ウィンドウサイズの値が大幅に変動せず、パケットを多く送っていないことが分かる。TCP-Vegas は遅延ベース方式であり、輻輳ウィンドウサイズを調節し通信速度を抑えているため、一番低い値となり、CUBIC-TCP、TCP-Reno の結果に差はあまりなく、お互い約 300kb を平均として輻輳制御を行っていた。これは障害物がない場合においても同じような結果であり、MPEG-DASH は伝送レートが低下しても輻輳ウィンドウサイズの変動は TCP に依存せず、変化があまり見られない結果となった。

反対に HTTP Live Streaming の結果は MPEG-DASH と大幅に異なった。無線通信状態が良い場合では TCP-Vegas は安定し、反対に CUBIC-TCP は輻輳ウィンドウサイズの増加や変動が大きい結果となり、TCP-Reno は大きな変動はあまり見せなく、一定の値で安定した。CUBIC-TCP はパケットロス検出時の輻輳ウィンドウサイズまで素早く回復をさせ、バッファを占有するアルゴリズムであるためこのような結果となった。しかし、環境が悪化すると輻輳ウィンドウサイズは変化した。TCP-Vegas はバッファ内で制御しているため、あまり変化はないが、TCP-Reno と Cubic-TCP の輻輳ウィンドウサイズは減少した。障害物があるとボトルネックとなり、データが届かなくなってしまうため、パケットロスが頻繁におこり、輻輳ウィンドウサイズが減少する結果となった。

表 4-1 輻輳ウィンドウサイズの平均値

	TCP-Reno (kb)	TCP-Vegas (kb)	CUBIC-TCP (kb)
MPEG-DASH	369	118	342
MPEG-DASH with obstacles	216	113	327
HTTP Live Streaming	412	105	1020
HTTP Live Streaming with obstacles	215	112	327

表 4-1 はそれぞれの輻輳ウィンドウサイズの平均値を示す。これより平均として結果をみると、MPEG-DASH は環境が悪化すると TCP-Reno が一番減少を示すが、TCP-Vegas、CUBIC-TCP に関しては減少が少ないことが分かる。反対に HTTP Live Streaming は無線通信状態が悪化すると TCP-Reno、CUBIC-TCP 上で輻輳ウィンドウサイズが大幅に減少することが分かる。

4.2.2 RTT(Round Trip Time)

図 4.4、4.5 に障害物がある場合における MPEG-DASH/ HTTP Live Streaming の RTT の結果を示す。

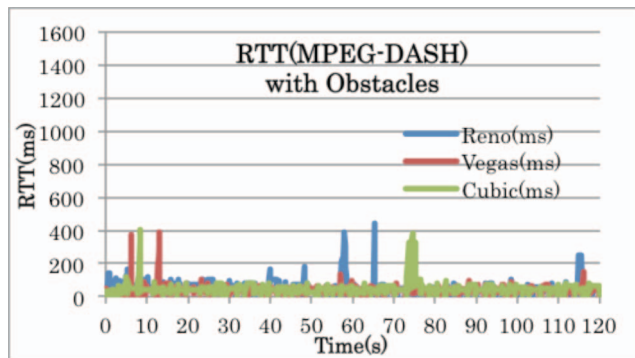


図 4.4 RTT(MPEG-DASH) 障害物あり

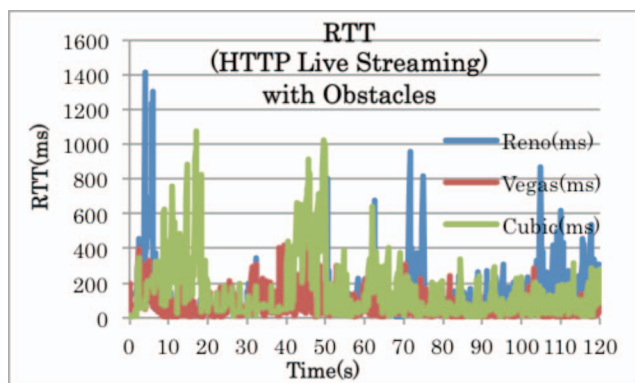


図 4.5 RTT(HTTP Live Streaming) 障害物あり

MPEG-DASH において、2 つの環境での結果を比較すると、全ての TCP において RTT の変化があまり見られないことが分かった。障害物がない場合においても図 4.4 の結果とほぼ変わらない。障害物によって伝送レートが低下しても、3 つの TCP 上での RTT の平均値は 1.2% しか増加しない結果となった。MPEG-DASH はネットワーク帯域に適応することができる。環境が悪化した場合にはそれに適したビットレートの動画を伝送するため、バッファに過度に蓄積されず、遅延の増加を防いでいる。

障害物がない場合は HTTP Live Streaming の結果は MPEG-DASH の結果とは変わらず、全ての TCP 上で安定していた。しかし、障害物があると RTT は全ての TCP 上で大幅に増加した。これは図 4.5 で見られる。HTTP Live Streaming は adaptive な動画配信技術ではないので、伝送レートが低くなった場合でも同じ大きさのデータを送信する。実際に実験時にはバッファリング状態となってしまう、データが届いていなかった。データがバッファに過度に蓄積されてしまうため、データが届かなくなり、RTT が増加していた。

表 4-2 RTT の平均値

	TCP-Reno (ms)	TCP-Vegas (ms)	CUBIC-TCP (ms)
MPEG-DASH	48	39	43
MPEG-DASH with obstacles	54	41	49
HTTP Live Streaming	24	12	52
HTTP Live Streaming with obstacles	125	64	143

表 4.2 はそれぞれの RTT の平均値を表す。平均値として結果を見ると、障害物がない場合、TCP-Vegas における HTTP Live Streaming が最小の遅延を示していることが分かる。環境が悪化すると MPEG-DASH の遅延は全ての TCP 上で増加が少ない。反対に HTTP Live Streaming に関しては全ての TCP において遅延が増大し、TCP-Vegas はバッファ内で制御しているため増加があまり多くないが、TCP-Reno は遅延が 100ms も増加していることが分かる。

4.2.3 スループット(Throughput)

図 4.5、4.6 に障害物がある場合における MPEG-DASH/ HTTP Live Streaming のスループットの結果を示す。

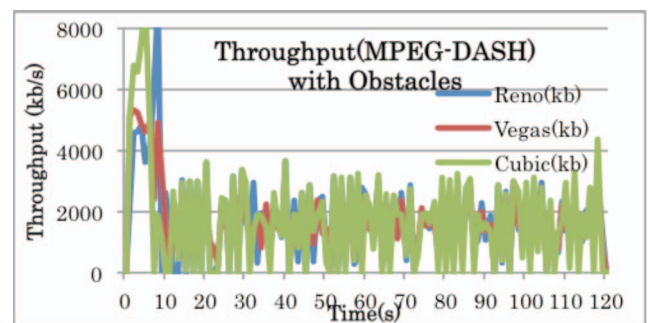


図 4.6 スループット(MPEG-DASH) 障害物あり

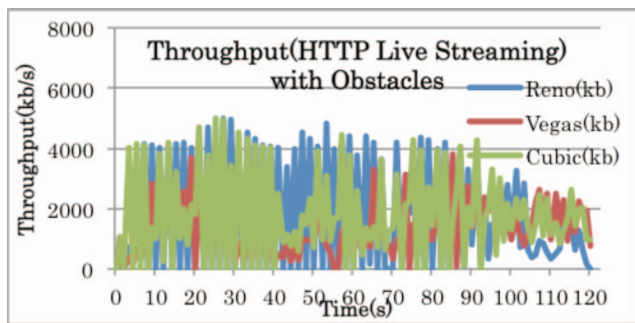


図 4.7 スループット(HTTP Live Streaming) 障害物あり

MPEG-DASH においては環境が変化してもスループットの値にあまり変化が見られない。TCP-Vegas は輻輳ウィンドウサイズの増加があまりなく、RTT は安定していたためスループットが一番低い。TCP-Reno、CUBIC-TCP に関しては輻輳ウィンドウサイズ、RTT の値がほぼ同じであったため、スループットも同じような値となった。障害物がある実験での MPEG-DASH は動的にビットレートを変化させているが、図 4.6 の結果より伝送レートが低下してもスループットがあまり下がっていないことから、障害物がある場合においてもビットレートを大幅に下げていないことが分かる。

反対に HTTP Live Streaming は環境が変わるとスループットが減少する結果となった。無線通信状態が良い場合は TCP に依存せずに同じ値を示していたが、環境が悪化するとすべての TCP 上でスループットは減少した。図 4.5 の RTT の結果を見ると、x 軸が 10~20 秒、40~50 秒、100~120 秒の時に急激に RTT が増加していることが分かる。図 4.3 の輻輳ウィンドウサイズもこの時間帯において値が減少していることが分かる。伝送レートが高くなりデータが届かない、あるいはパケットロスによる再送制御が頻繁に行われたため、このような結果となった。このため、図 4.7 ではそれらの時間帯にスループットの値が減少してしまっている。

5. まとめ

本論文では無線環境における各種の TCP の下で MPEG-DASH、HTTP Live Streaming の特性比較を行った。MPEG-DASH はネットワーク帯域に適応することができるため、環境が悪化し、伝送レートが低下した場合にはそれに適したビットレートの動画を伝送する。このため環境が悪化していても、輻輳ウィンドウサイズ、RTT とスループットは TCP に依存せずに変化があまり見られなかった。また、MPEG-DASH の特性からバッファに過度に蓄積されないため、遅延を防ぐことができる。反対に HTTP Live Streaming はアダプティブにビットレートを切り替えることができないので環境が悪化すると輻輳ウィンドウサイズ

が大幅に変化、遅延が増大し、スループットが減少してしまう。実際には障害物がある場合ではバッファリング状態となり、データが届かなくなってしまっていた。この結果として輻輳ウィンドウサイズ、スループットの減少や RTT の増大が結果として見られた。

6. 参考文献

- [1]Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2012-2017, http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-520862.html, February 6,2013
- [2] HTTP Live Streaming, Apple Inc., 2011
- [3] DASHEncoder at ITEC/Alpen-Adria-Universität Klagenfurt, http://www-itec.uni-klu.ac.at/dash/?page_id=282 (last access: Dec. 2011)
- [4] I. Sodagar, “The MPEG-DASH Standard for Multimedia Streaming Over the Internet”,