

WebRTC を用いた低遅延 ライブストリーミング方式の検討

大阪電気通信大学 総合情報学研究科
木下透弥

2020/06/15 情報システム研究会

概要

- 研究背景
 - 生配信の増加と現状問題点
- 研究目的
 - 低遅延システムの提案、実装、評価
- **MPEG-DASH**
 - 仕様と遅延の問題点
- システム設計
 - サーバー視聴者間でWebRTCを利用
- 評価
 - 実ネットワークで評価を行い、提案方式の有効性を示す
- まとめ、今後の課題

研究の背景

- ウェブにて動画を見ることが一般的
 - YouTube などの動画共有サービス
 - Netflix などのVideo on Demand 方式のメディア配信
- 現在、リアルタイムで映像を多数の視聴者へライブストリーミングすることが増加
- インターネットでのユーザーによる生配信の規模は大きい
 - YouTube Live や Twitch.tv サービスが存在
 - PC やスマートフォンのアプリから個人で生放送することが可能

ウェブでの動画配信の方法

- MPEG-DASH

- 現在最も利用されている動画配信手法
 - Netflix、YouTube
- HTTPを用いて動画をストリーミングする
- 動画ファイルをHTTPで読み込みながら再生する
プログレッシブダウンロード方式

- RTMP (Real Time Messaging Protocol)

- リアルタイム性のある手法
- Flash Playerを利用して、ブラウザで利用
 - 開発が終わった為、Flash Player を用いることは適切ではない
- ウェブブラウザが対応していない為、ブラウザへ直接配信はできない

ライブ配信での問題

- MPEG-DASHは生放送にも利用した場合、**遅延が非常に大きくなりやすい**
 - 10秒以上の遅延が発生
- 配信者と視聴者でのコミュニケーションをとる際、遅延が大きいとコミュニケーションが取りづらい
 - 自然なコミュニケーションではなくなる

研究の目的

- 低遅延ライブストリーミングを可能とするシステムの提案および実装
 - ウェブブラウザやスマートフォンアプリで標準的に実装されている **WebRTC** を用いる
- ネットワークの状況に応じた画質ストリームを提供
 - ネットワークの状況が悪いときは、低画質でのストリーム
- 実ネットワーク環境において、評価

MPEG-DASH

- DASH方
呼ぶ動
- それに

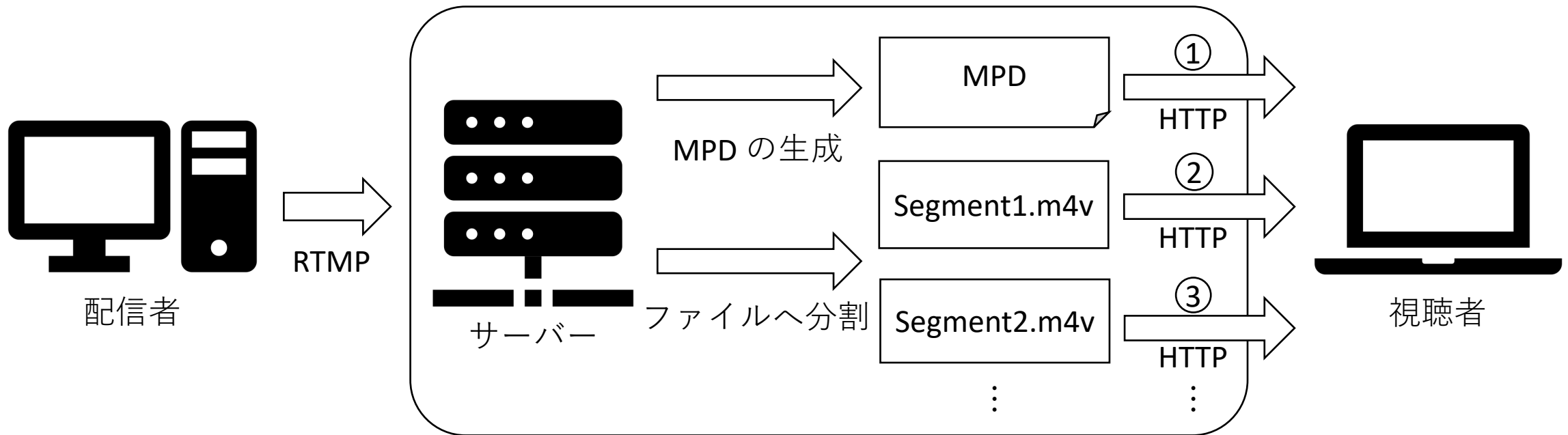
解像度
フレームレート

読み込

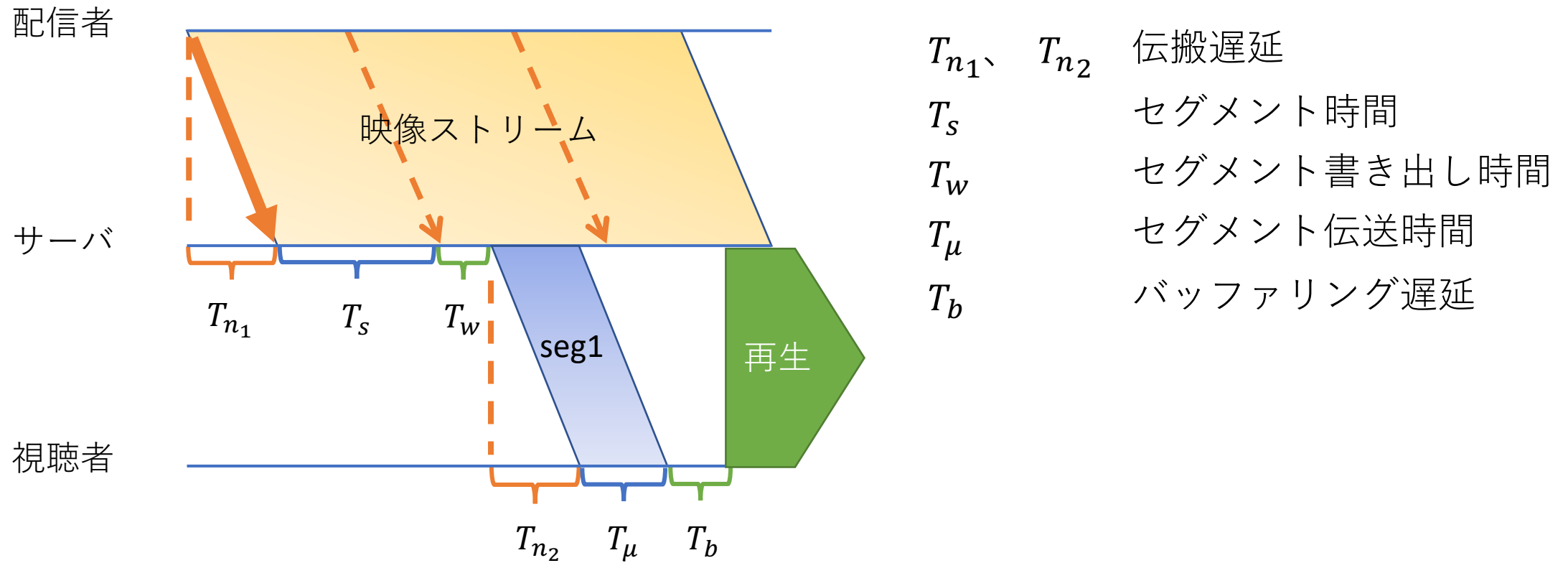
```
<?xml version="1.0"?>
<MPD
  type="dynamic"
  xmlns="urn:mpeg:dash:schema:mpd:2011"
  availabilityStartTime="2020-06-08T06:37:18Z"
  publishTime="2020-06-08T06:38:00Z"
  minimumUpdatePeriod="PT5S"
  timeShiftBufferDepth="PT21S"
  maxSegmentDuration="PT15S"
  profiles="urn:mpeg:dash:profile:isoff-live:2012,urn:mpeg:dash:profile:isoff-live:2011"
  xsi:schemaLocation="urn:mpeg:DASH:schema:MPD:2011 DASH-MPD.xsd">
  <Period start="PT0S" id="dash">
    <AdaptationSet
      id="1"
      segmentAlignment="true"
      maxWidth="1920"
      maxHeight="1080"
      maxFrameRate="30">
```

MPEG-DASHを用いたライブ配信

- 配信者がRTMPを用いてサーバーへ配信
- 視聴者へはサーバーがMPEG-DASHに変更
 - HTTP上で動画をストリーミングする



MPEG-DASH用いたライブ配信の遅延(1/2)



MPEG-DASH ライブ配信の遅延(2/2)

遅延に大きな
影響がある

$$T = T_{n_1} + T_s + T_w + T_{n_2} + T_\mu + T_b$$

- T_{n_1} T_{n_2} はネットワークの環境により小さくできる
 - T_μ もネットワークによる
 - CDNの利用で小さくできる
- T_w は小さい
- ライブ配信向けの場合 T_b は小さくできる
- 例えばセグメントの長さ T_s を5秒とするだけでも、少なくとも5秒の遅延がつくこととなる

MPEG-DASHのセグメントサイズの検討

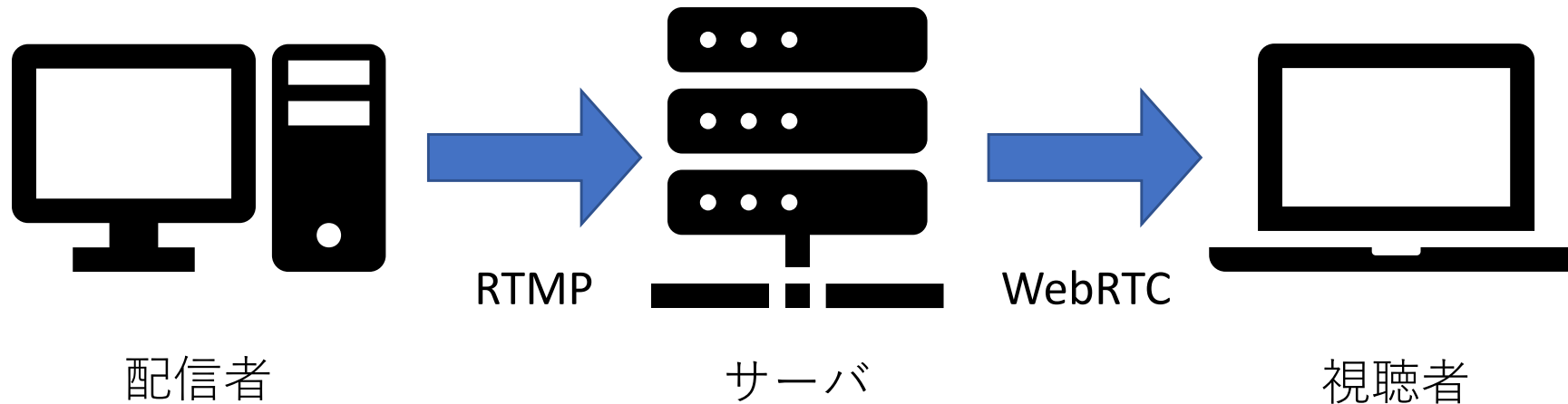
- セグメント長を短くすることによりセグメント待ちの遅延を短くした場合
 - セグメント数が増え、IOの増加
 - セグメント数分のHTTPのネットワークのリクエスト増加
 - バッファリングが止まる
 - 帯域が無駄に利用される
- セグメントサイズを1秒にしても結果的に2秒程の遅延ができる

オーバーヘッドが増え、遅延が増加

[11] T. Lohmar, T. Einarsson, P. Fröjdh, F. Gabin, and M. Kampmann, "Dynamic adaptive HTTP streaming of live content," in Proceedings of 2011 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, pp. 1–8, 2011.

システムの設計

- 配信者からサーバへの転送: RTMP
 - ffmpegやXsplit、OBSなど主要なソフトウェアで一般的に配信に使用されてる
- サーバから視聴者への転送: WebRTC

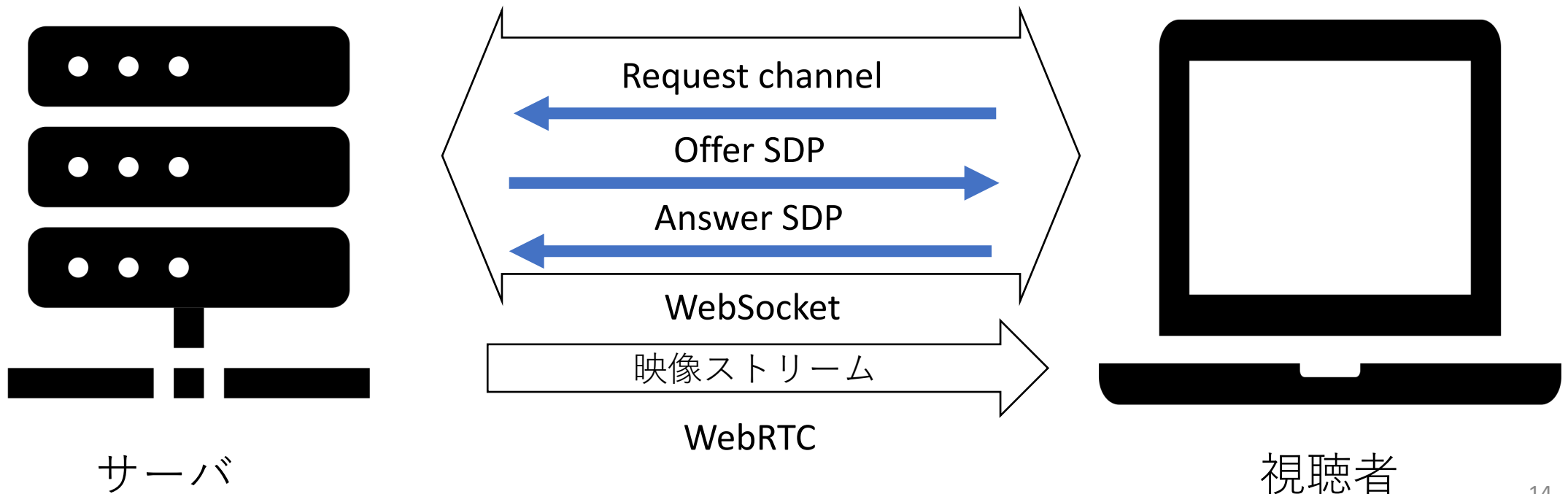


WebRTC

- **WebRTC**:ブラウザやスマートフォン上のアプリケーションでリアルタイムコミュニケーションを行う方法
 - UDPが採用されており、メディアチャンネルではRTPプロトコルで映像のやり取りができる
 - ブラウザ間などで通話をしたり、P2P方式でファイル共有を行うのに使われる
- サーバ→視聴者の転送において、WebRTCを用いることで、
 - 映像を直接ストリーミングすることが可能
 - MPEG-DASHのようなセグメント分割が不要で、セグメント分のデータの到着を待つ必要がなくなる

映像ストリームのコネクション確立

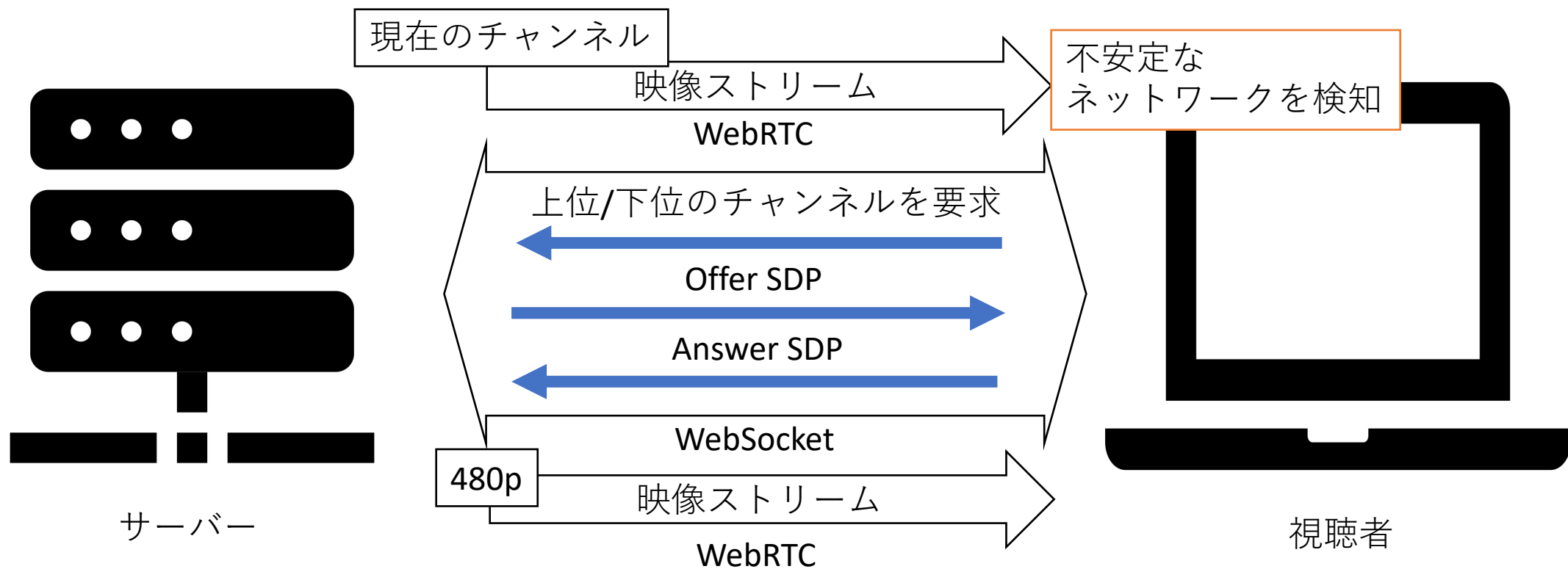
- WebRTCのリモートピアからクライアントのコネクションの確立にWebSocketを用いる
- WebRTCでメディアのやり取りに必要なSDPのやり取りを行う



映像チャンネルの変更 (1/2)

- 品質の異なる複数のチャンネルを提供
 - ライブストリーミングを視聴する人の帯域が狭い場合やネットワークの状態が不安定な場合、元の映像では不安定になることが考えれる
- サーバーで低い解像度とビットレートに映像をトランスコードし、ストリーム
- 視聴者側で不安定とされた場合に、視聴者が低位のストリームを要求し、ストリームを切り替える
 - SDPのやり取りを行ったWebSocketのコネクション上で要求や新しいSDPのやり取りをする

映像チャンネルの変更 (2/2)

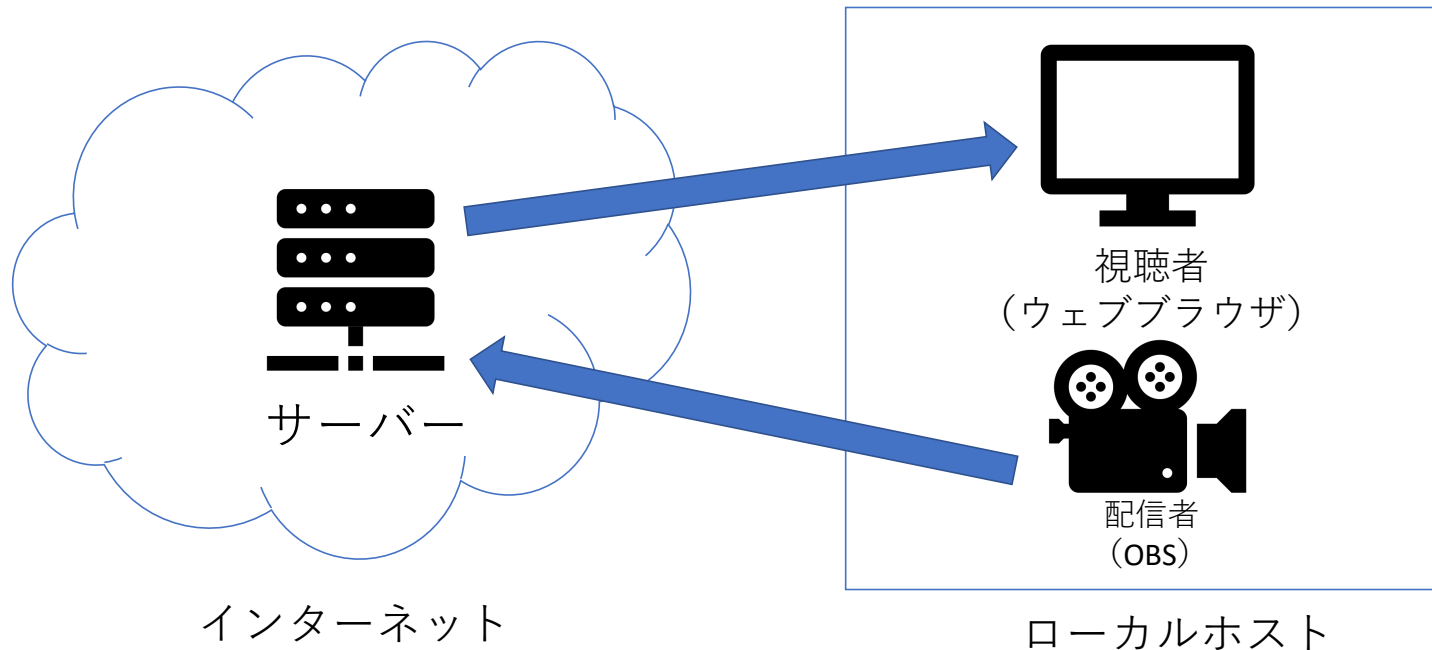


映像チャンネルの切り替え

- 視聴者のプレイヤーで一定時間、フレームがドロップした場合、**低画質のチャンネル要求**をサーバーに送信
- 一定時間デコードしたフレームレートが規定以上の割合だった場合、**高画質のチャンネル要求**をサーバーに送信
- **視聴者による手動**でのチャンネル要求が可能

実験環境 (1/2)

- 実ネットワークにおいて評価する
- 視聴者と配信者は同一のホスト上で行い、インターネット上のサーバーを介して配信、視聴を行う



ローカルホスト→サーバーの帯域
34.6Mbit/s

サーバー→ローカルホストの帯域
33.9Mbit/s

ローカルホスト・サーバー間の
RTT: 23.744ms

実験環境 (2/2)

- チャンネルとして、オリジナル、720p、480pを視聴者に提供
 - オリジナルは、配信者が配信したもの
 - 720p、480p は、サーバにてトランスコード
 - OBSのエンコーダはx264、2500kbps、30fps
- オリジナル、480p、MPEG-DASH の3種類の遅延を評価
 - 480p: トランスコードのオーバーヘッドの確認のため
 - MPEG-DASH: 提案手法との比較のため
 - セグメント時間を10秒とした
- それぞれ、60 秒間の配信を行い、遅延を測定

遅延の評価方法

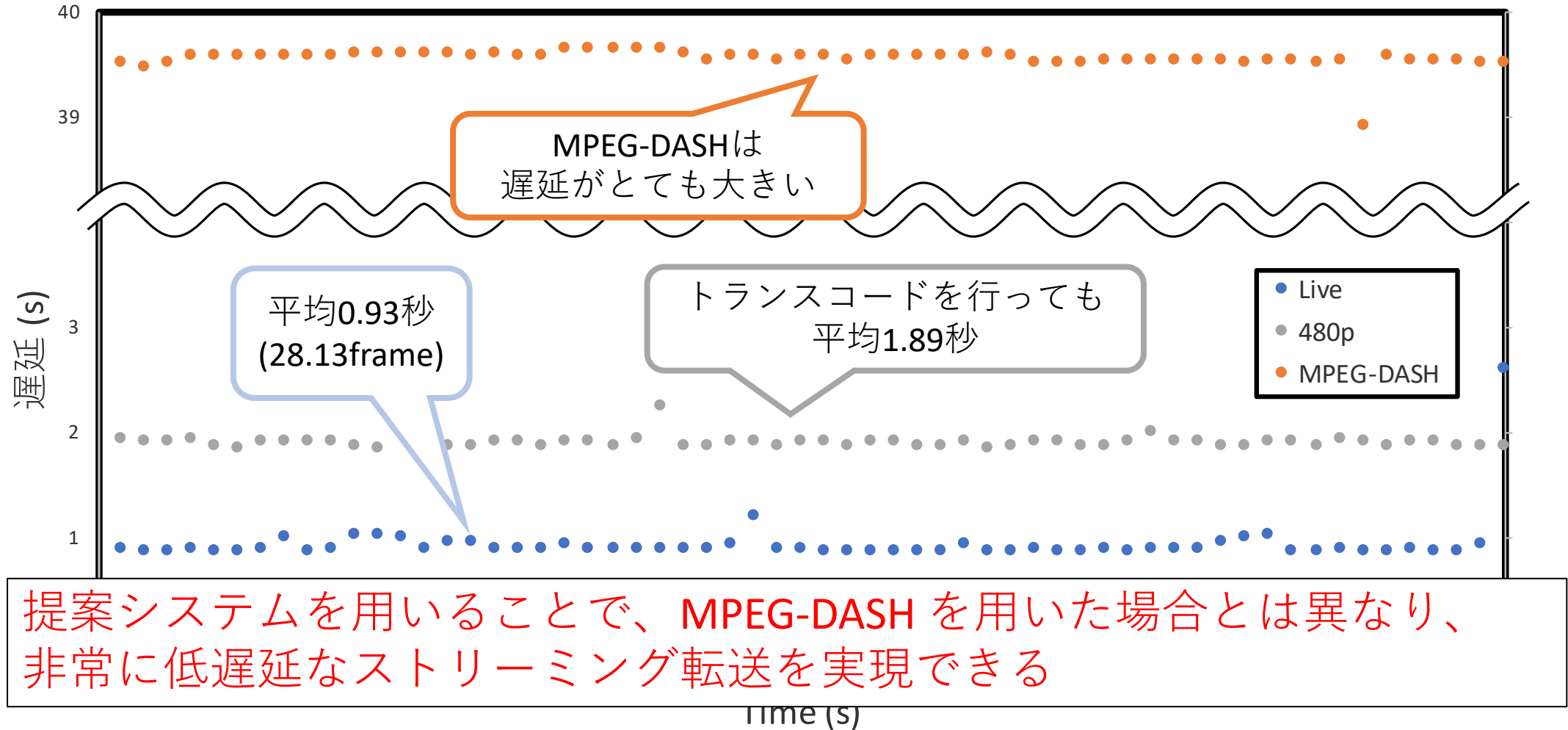
- 配信映像にフレーム番号を含める
- 配信者画面と視聴者画面のフレーム番号差で遅延を計測
 - glass-to-glass 遅延測定

The image shows a side-by-side comparison of two video feeds. On the left, the OBS Studio interface displays a video frame with the number **6052** and a timestamp of 'Thu Jun 11 2020 18:36:22 GMT+0900 (日本標準時)'. A blue callout bubble labeled '配信者' (streamer) points to this frame. On the right, a web browser window shows a video frame with the number **5839** and a timestamp of 'Thu Jun 11 2020 18:36:21 GMT+0900 (日本標準時)'. A blue callout bubble labeled '視聴者' (viewer) points to this frame. Below the browser window, technical details for the video stream are visible, including 'id: RTCInboundRTPVideoStream_279434614', 'timestamp: 1591868182004', 'type: track', 'trackIdentifier: 2ae57f10-e622-4699-bb42-440e0903df86', 'remoteSource:', 'ended:', and 'detached:'.

30fpsの映像の為
30fで1秒
1fで約0.033秒
の遅延

6052-5839
213フレーム
7秒の遅延

評価結果:配信者と視聴者間の遅延



まとめ

- 低遅延ストリーミングシステムを提案、実装
 - 配信者→サーバ: RTMP を用いて転送
 - サーバ→視聴者: WebRTC を用いて転送
 - ネットワークの状態に応じた品質のストリームを提供
- 実ネットワークにおいて、提案システムを評価
 - 提案システムの遅延: 1秒以下
 - トランスコードを行った場合2秒以下
 - 提案システムが低遅延ストリームを実現することを示した

今後の課題

- 今後の課題としてより詳細な評価が必要
 - MPEG-DASHの設定の変更
 - 映像切り替え
 - 配信中にネットワーク環境が変わった場合
 - 多数の視聴者に向けた配信
- 効率的な配信方法の検討
 - CDNを用いて複数のサーバが存在する場合の評価