

# 輻輳制御機能を備えた 低遅延ライブストリーミングシステムの 提案および評価

大阪電気通信大学 総合情報学研究科  
木下透弥

2020/11/15 関西支部連合大会

# 概要

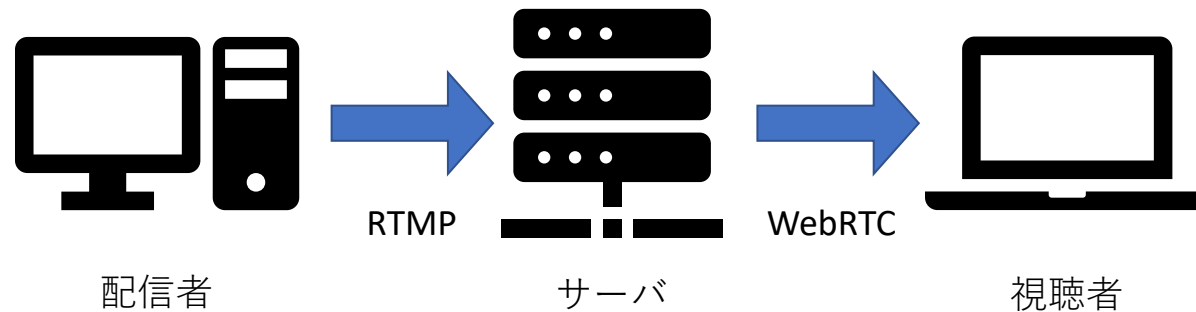
- 研究背景
  - ライブストリーミングにおける遅延の問題
- 研究目的
  - 輻輳制御の導入: ネットワークの状況に応じた画質のストリームを提供
- **MPEG-DASH**
  - 仕様と遅延の問題点
- 輻輳制御による映像品質の変更
- 評価
  - 実ネットワークで評価を行い、提案方式の有効性を示す
- まとめと今後の課題

# 研究の背景

- ウェブにてライブストリーミングを見ることが一般的
  - 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
  - MPEG-DASHをライブストリーミングに利用した場合、  
遅延が非常に大きくなりやすい
  - 遅延が大きいと、配信者と視聴者間のコミュニケーションにおいて  
問題となる

# 過去の我々の研究

- 低遅延ストリーミングシステムを提案、実装



- RTMP と WebRTC を用いて転送
- 複数の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**
  - **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**
- **システムには輻輳制御機構が存在していない**
  - 輻輳時に輻輳が悪化
  - 高品質ストリームは、そもそも視聴できない

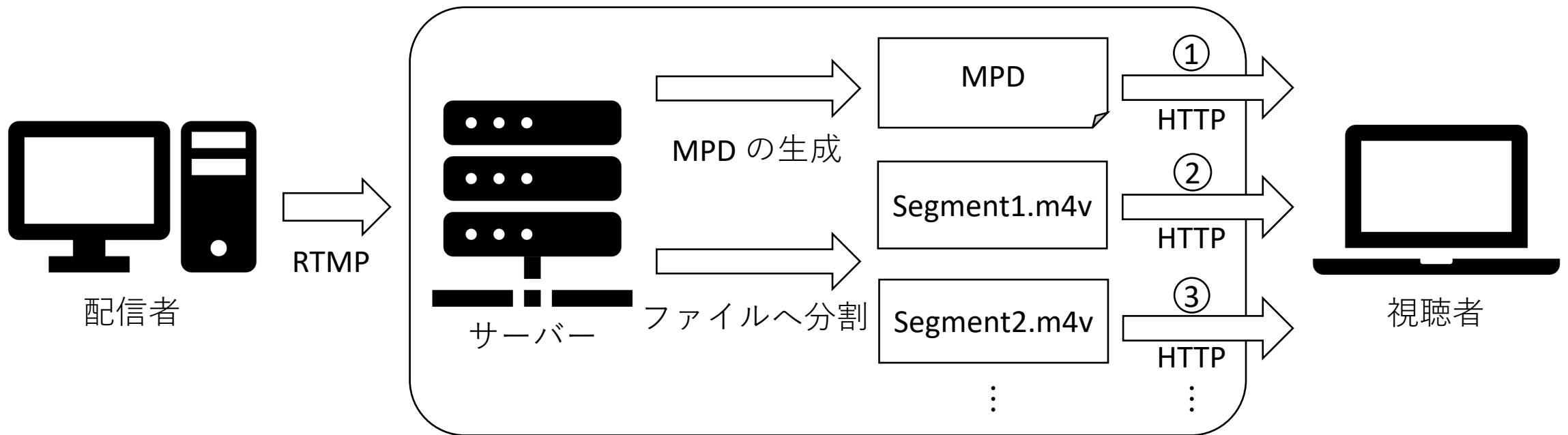
# 研究の目的

- 過去の研究の拡張
  - ネットワークの状況に応じた画質ストリームを提供
    - 輻輳の検知
    - ネットワークの状況が悪いとき->低画質でストリーム
    - ネットワークの状況が回復したとき->品質を上げてストリーム
- 実ネットワーク環境において評価
  - 輻輳が発生した場合の動作

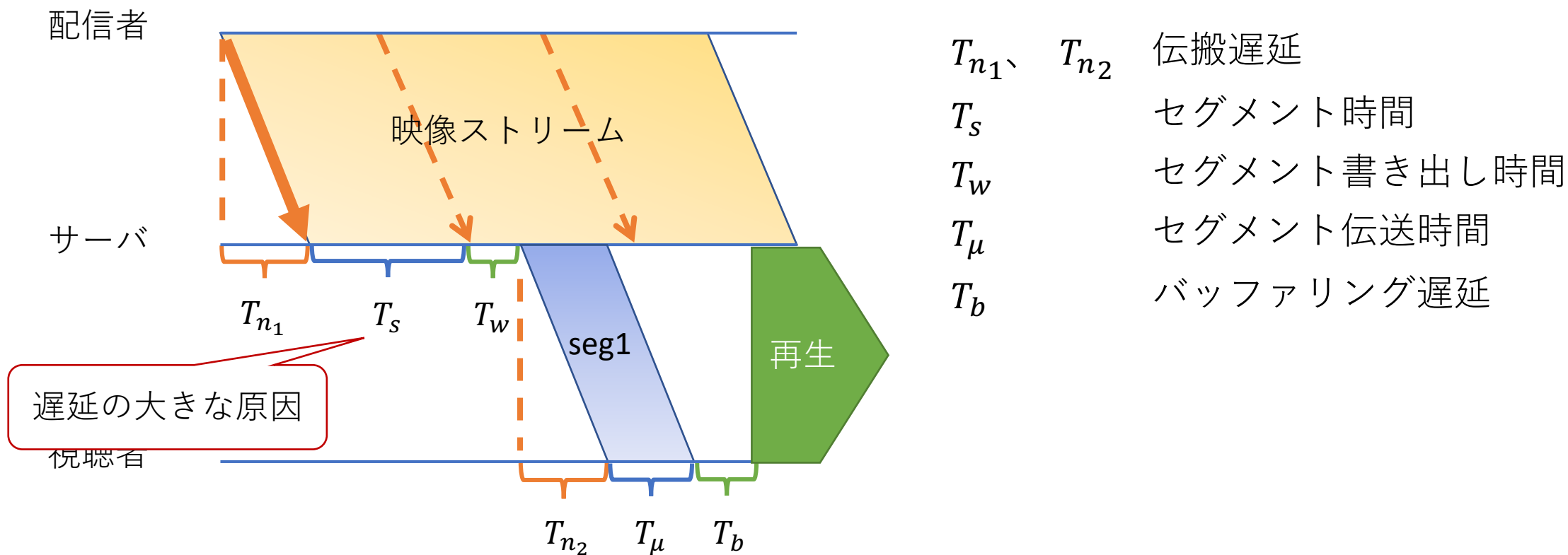
```
<?xml version="1.0"?>
<MPD
type="dynamic"
xmlns="urn:mpeg:dash:schema:mpd:2011"
availabilityStartTime="2020-06-08T06:37:18Z"
publishTime="2020-06-08T06:38:00Z"
minimumUpdatePeriod="PT5S"
minBufferTime="PT5S"
timeShiftBufferDepth="PT21S"
profiles="urn:hbbtv:dash:profile:isoff-live:2012,urn:mpeg:dash:profile:isoff-live:2011"
xmlns:xsi="http://www.w3.org/2011/XMLSchema-instance"
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">
```

# 既存の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を用いたライブ配信の遅延



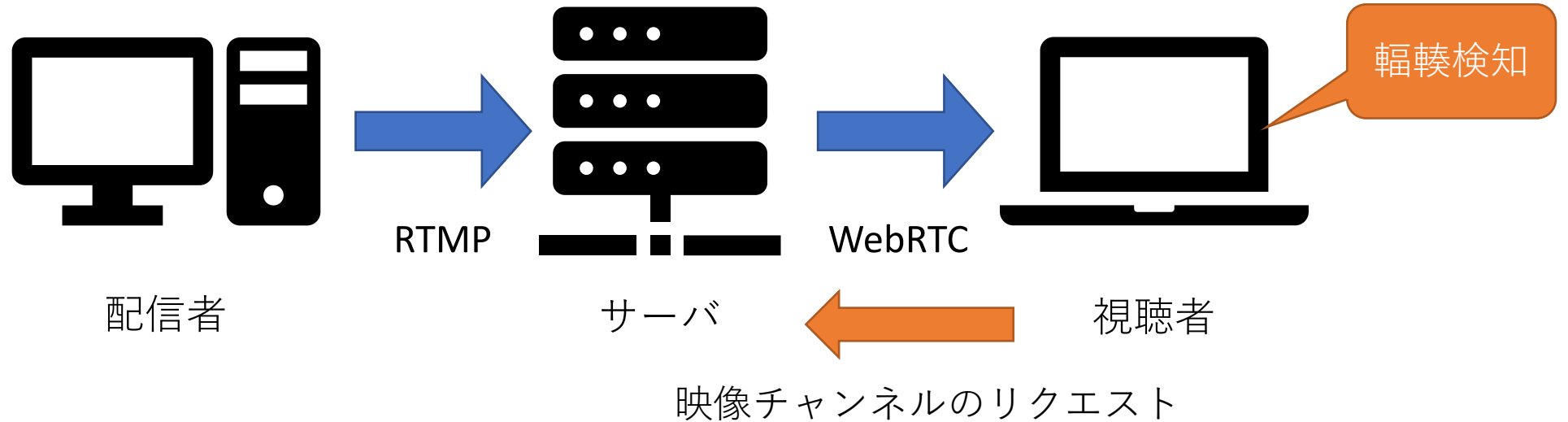
例えばセグメントの長さ $T_s$ を5秒とするだけでも、少なくとも5秒の遅延がつくこととなる



# 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

# 拡張したシステムの概要



- 視聴者側で輻輳を検知
- 視聴者がサーバに対して適切な品質のチャンネルを要求

# 輻輳の検知

- 再生した映像のフレームレートとTCPのスループットの推測値 [2]を利用して、適切な品質の映像を決定
  - フレームレート: 視聴者側の再生状況を示す
    - フレームレートが落ちている→パケット棄却の発生
  - TCPのスループットの推測値: ネットワークの空き状況を判断
    - TCPのスループット以上での転送は複層を悪化させうる

[2] Mark J. Handley and Jitendra Padhye and Sally Floyd and Joerg Widmer  
TCP Friendly Rate Control (TFRC): Protocol Specification

# 映像チャンネルの変更

- 複数の映像ストリームを提供する
  - 720p, 480p といった低い解像度、ビットレートのチャンネル
- 視聴者側で不安とされた場合に、**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
- 視聴者側で安定とされた場合に、**上位のストリームを要求**し、ストリームを切り替える

# 輻輳の判定

- 毎秒ごとにネットワークの輻輳状況の判定を行う
  - Quality of Experience (QoE) の観点から切り替えは素早く対応
    - すぐに品質を落とし、狭い帯域でも再生できるようにする
- 映像のデコードしたフレームレート < 規程フレームレート
  - 既定のフレームレート: 元の映像の80%
    - ネットワークが輻輳を起こしていると判定
    - 低いビットレートの映像へ変更

# 輻輳の解消の判定

- 5秒間連続してフレームレートが規定以上であった場合、**TCP**の転送速度の推測値  $\lambda$  を用い余裕があるかを判定
  - 5秒間連続: 品質を上げるときは
  - **TCP**の転送速度の推測値

$$\lambda = \frac{s}{R \left( \sqrt{\frac{2p}{3}} + 12 \sqrt{\frac{3p}{8}} p (1 + 32p^2) \right)}$$

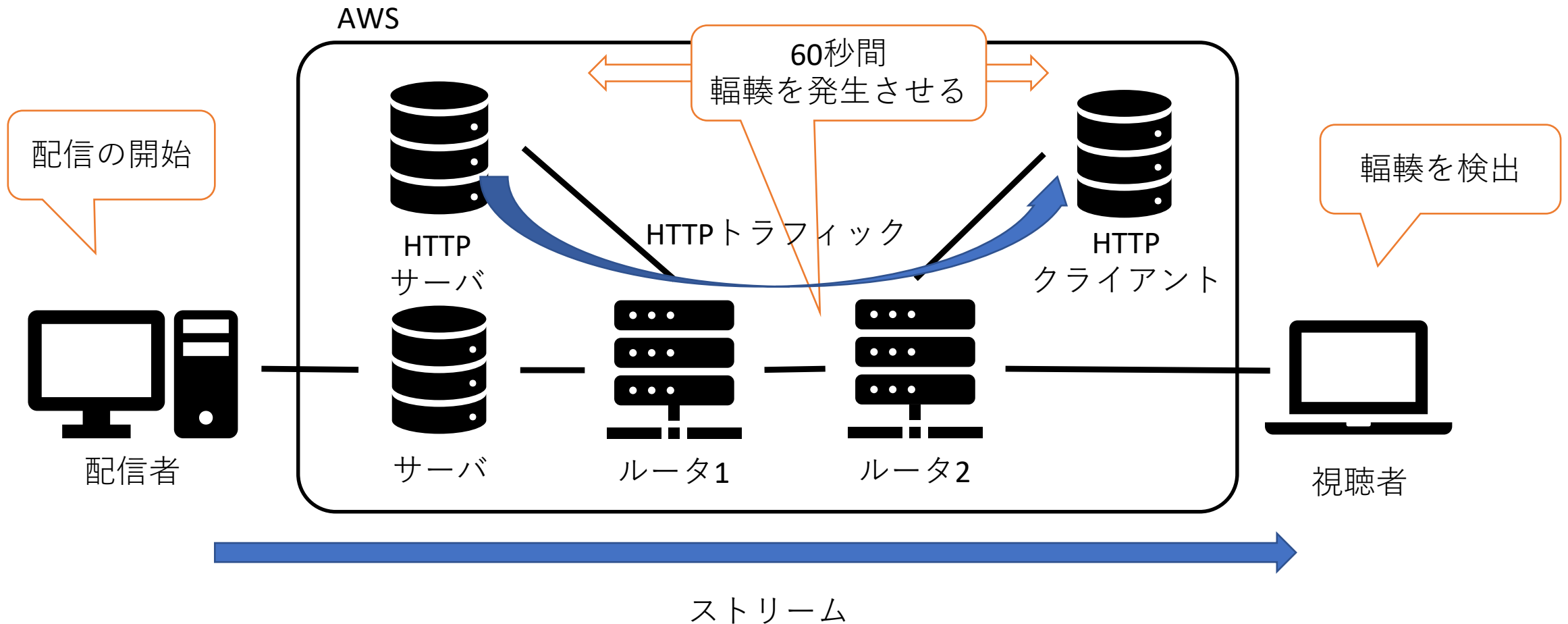
$R$ : RTT,  $s$ : セグメントサイズ,  $p$ : 棄却率

- $\lambda$ と映像のビットレートを比較し、帯域に余裕があるか判定

# 実験環境

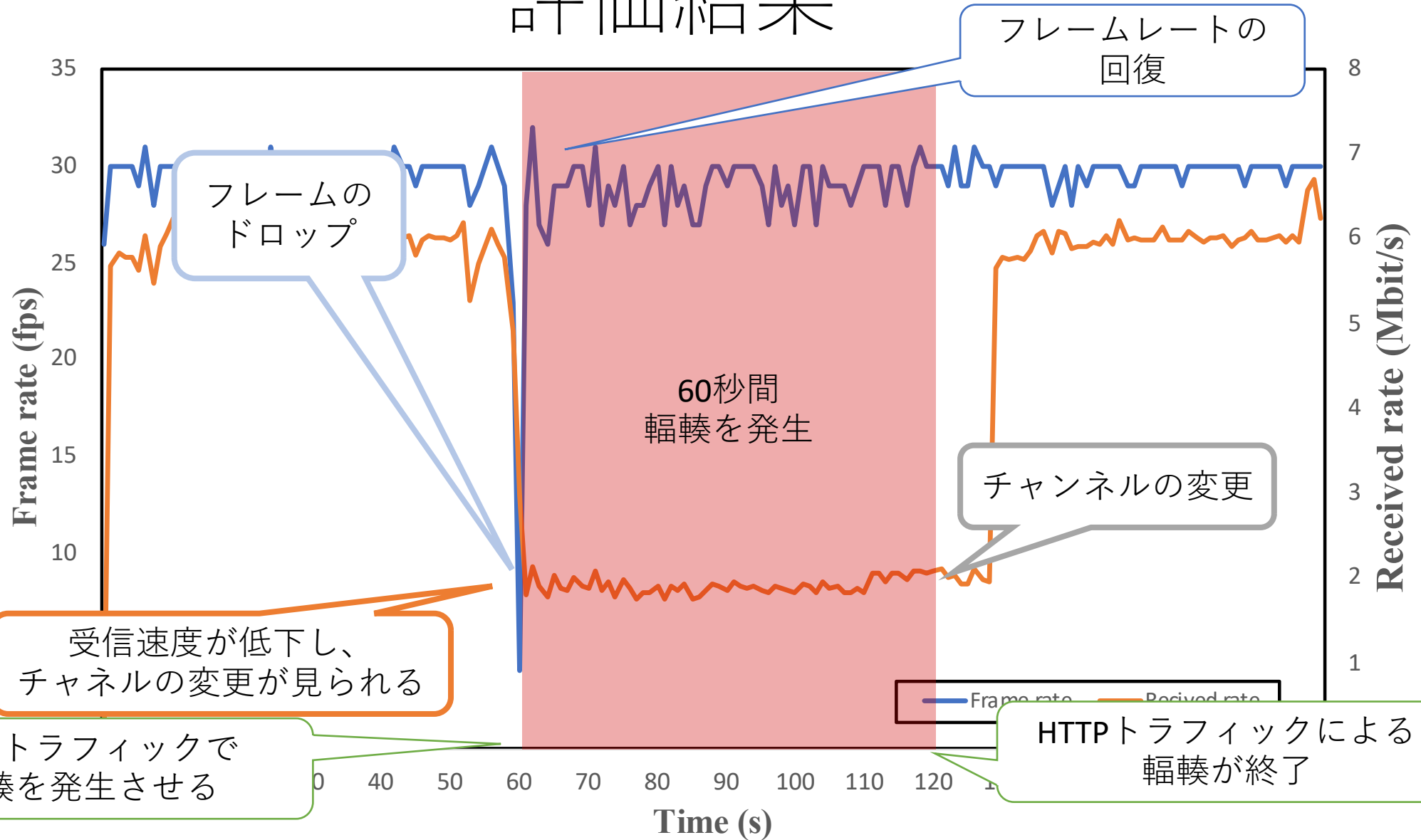
- 提案方式を実装し、実ネットワークにおいて評価
- チャンネル: オリジナル、720p、480pを視聴者に提供
  - オリジナルは、配信者が配信したもの
  - 720p、480p は、サーバにてトランスコード
- 180秒間の実験を行う
  - 実験開始60秒後から60秒間HTTPトラフィックを発生させ、負荷をかける

# 実験に用いるネットワーク





# 評価結果



# まとめと今後の課題

- まとめ
  - 既存のシステムを輻制御機能を有するシステムへ拡張
    - 再生している映像のフレームレートとTCPの転送速度の推測値を用いたネットワークの輻輳状況の判定
  - 実ネットワークにおいて、提案システムを評価
    - 適切に動作することを示した
- 今後の課題
  - より複雑なシナリオでの評価
  - 提案システムの **Quality of Experience (QoE)** の評価
  - モバイルネットワークにおける評価