1U-03

WebRTC を利用したライブ動画配信システム

中村 日向子† 木村 友彦‡ 藤田 聡‡
広島大学 工学部† 広島大学 大学院工学研究院‡

1. はじめに

近年，インターネット上の様々なコンテンツが「情報メディア」としての役割を担う機会が多くなってきている．例えば YouTube のライブストリーミングを利用すれば，放送設備を持たない一般の人でも，Web カメラやマイクを揃えた PC を用意するだけで簡単に生放送を行うことができる．しかしながら，これらのシステムでは遅延が 30〜40 秒と長く，リアルタイム性には乏しいという欠点があった．今回我々は，ビデオ会議などで用いられるプロトコル群である WebRTC を用いることで，リアルタイム性に優れ，しかもスマートフォンなどの携帯端末のみでライブ動画を配信・視聴することができるシステムを開発したので報告する．

2. WebRTC

WebRTC は，ウェブブラウザ間でリアルタイムコミュニケーションを行うための API とそのプロトコル群であり，IETF によって関連プロトコルの標準化が， W3C によってブラウザ対応 API の標準化がそれぞれなされている．プラグインや専用ソフトウェアをインストールすることなく通常のブラウザのみで通信を行える点が注目を集めており，近年では iOS や Android などでも WebRTC を利用することができる．

WebRTC では，PeerJS と呼ばれるライブラリを用いることで，ピア間のコネクションを確立したり，音声や映像などのデータのリアルタイムな送受信を行なったりすることができる． 具体的には PeerJS では，各ピアに割り振られたユニークな ID を指定してコネクションの確立・切断を行うメソッドなどが提供されている． 後述の提案システムでは，PeerJS に NTT コミュニケーションが独自拡張を施したライブラリである SkyWay を用いている．SkyWay は，上記のような機能の他に，STUN サーバや TURN サーバ，シグナリングサーバなどの機能も備えている．

P2P live video streaming system using WebRTC
† Hinako NAKAMURA, Hiroshima University
‡ Tomohiko KIMURA, Satoshi FUJITA,Hiroshima University
本研究の一部は科学研究費補助金基盤研究(B)16H02807 の補助を受けています．

具体的には，SkyWay を用いることで，他のピアをコールする際にストリームを指定し．コールを受けたピアがストリームを指定して応答することで，コールをした相手へのストリーム配信が可能となる．また SkyWay にはルームと呼ばれる機能があり，この機能を用いることで，同じルーム内のピアたちとの間のコネクションを確立し，ルーム内のすべてのピアに向けたストリームの配信が実現される．

3. 提案システム

3.1 概要

我々が想定する提案システムの利用例は，大学祭などの大規模なイベント会場でのライブ動画の配信である．システムでは(例えば)イベント会場のブースごとにチャネルを用意し，各ユーザは，チャネルリストから興味のあるチャネルを選択してその視聴を行う．各チャネルは複数のアングル(カメラ)を持つことができ，アングルの切り替えとそのチャネルのストリーム配信は，チャネルを開設したユーザが責任を持って行う．以下では，チャネルの開設を行ったユーザを(そのチャネルの)配信者と呼び，それ以外のカメラの所有者を追加カメラと呼ぶ．ライブ動画の視聴者たちへの配信は，ルーム機能を用いてピア・ツー・ピアに行われる．以下では，他の視聴者への配信も担う視聴者のことを(そのチャネルの)受信マスターと呼び，それ以外の受信者と区別する．

ユーザは，自端末上でアプリを起動した時点で，1) 視聴チャネルの選択，2) チャネルの新規開設， 3) 既存チャネルへのカメラの追加のいずれかを選択する．後述するように，ユーザ端末間は，選択された役割に応じた P2P オーバーレイにより接続される．開設されたチャネルに関する情報は，各チャネルの視聴ユーザリストとともにサーバ上のデータベース(DB)に保存され，適宜参照・更新される．また本システムでは，視聴チャネル選択の補助として，アクセスランキング等の UI も提供している．

3.2 接続プロトコル

各ピアは，選択した役割に応じて以下の手順で

オーバーレイネットワークへの接続・離脱と他ピアとの通信を行う(図１参照).

配信者：自身の ID を DB に送信し，チャネルを開設する．開設したチャネルには，後述の追加カメラからのものを含めて複数のストリームが存在することになるが，そのうちのどれを実際に配信するかは，配信者が指定する(指定されなかったストリームは視聴者には届けられない)．指定されたストリームを後述の受信マスターに送信する責任は配信者が負い，開設チャネルは配信者がシステムから離脱した時点で自動的に閉鎖される．

追加カメラ：DB を参照して，カメラとして参加するチャネルの選択と配信者の ID を取得する．その後，自カメラのストリームの配信者への送信を行う．

受信マスター：受信マスターは，配信者からストリームを受け取り，それを他の配信者に転送する役割を担う(現在の実装ではピアの帯域幅などは考慮せず，最初にそのチャネルの視聴を始めたピアが自動的に受信マスターとなる)．配信者からの受信が始まると，受信マスターは自身の ID を DB に登録する．配信者によるチャネルの閉鎖を検知すると，受信マスターは配信者や他の受信者とのコネクションをすべて切断する．また受信マスターの離脱は配信者によって検知され，離脱を検知した配信者は，適切な方法で選択された受信者(次期マスター)にストリームの転送に関する責任を移譲し配信を継続する(ルーム機能を用いることでオーバーレイの再構築が迅速に行える)．

受信者：受信マスター以外の受信者は DB を参照して視聴するチャネル名を取得し，対応するルームに入室する．ルームに入室した時点で受信マスターからのストリーム配信が開始される．

なお追加カメラ以外の各ピアは，配信・受信を行っている間，DB に向けて５秒間隔で定期的に alive メッセージを送り，３０秒間更新が行われなかったピアやチャネルに関する情報は DB から削除される．これにより，各ピアは他ピアの離脱やチャネルの閉鎖を検知することができる．

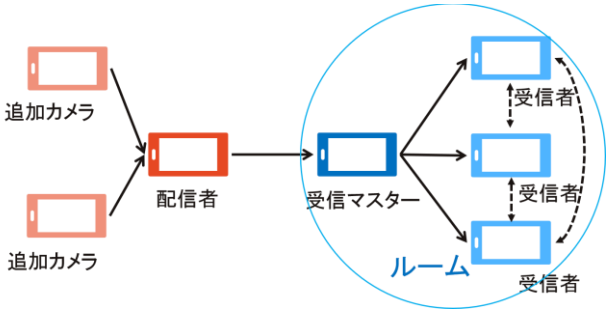

図 1. オーバーレイネットワーク

3. 3 受信者へのチャネル選択の補助

受信者へのチャネル選択の補助として，チャネルのリストにタグの表示を行う．タグは配信者がチャネル開設時に「大学祭」「屋台」などのチャネルに関係した文字列を登録する．タグごとのランキングの表示も可能であり，視聴者のチャネル情報取得時に視聴者数によるソートを行い，ソートされたリストを表示する．さらに，位置情報の提供を許可した配信者端末は alive メッセージ送信時に位置情報を DB に送信する．この位置情報により，チャネル選択時にマップへの表示を行い，位置によるチャネル選択も行うことができる．位置情報の取得は Google Places API for Android により行い，マップ表示は Google Maps Android API を利用する．

4. 実装

提案システムを Android 端末上のアプリとして実装した．使用した端末は ASUS 社製 ZenPad 10，OS: Android 7.0 であり，開発言語として Java を用いた．各端末は Wi-Fi を使ってネットワークに接続しており，学内のホスティングサービスで提供されているサーバ上に DB と Web サーバを準備した(Web サーバは学外にも公開されている)．DB 記述には SQLite を用い，ウェブサーバ上のプログラム記述は PHP で行なった．また Web サーバと端末間の通信は HTTP で行っている．

提案システムの性能評価のため，配信者について，チャネル開設から視聴可能，つまりデータベースへの登録が完了するまでの時間及び，受信マスター，受信者それぞれに対して，チャネルを選択してからストリームを受け取るまでの時間を計測した．

前者は約 270ms，後者は受信マスターが約 700ms，受信者が 650ms となった．

5. 参考文献

[1]木村友彦,藤田聡:WebRTC を用いたツインビュー型 P2P ビデオ会議システム,情報処理学会全国大会 2017

[２]WebRTC, https://webrtc.org/

[３]SkyWay, https://webrtc.ecl.ntt.com/

[４]Google Places API, https://developers.google.com/places/?hl=ja

[５]Google Maps API, https://developers.google.com/maps/?hl=ja