WebRTC を用いたツインビュー型 P2P ビデオ会議システム

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

WebRTC を用いたビデオ配信が近年高い注目を集めている．WebRTC とはブラウザを介してリアルタイムコミュニケーションを行うためのプロトコル群であり，WebRTC を使うことでプラグインなしにデータや動画などのストリームデータのやりとりを行うことができる．本稿では，WebRTC を用いたビデオ会議システムを P2P ネットワーク上に実現する方法を提案する．提案手法では，システムのスケーラビリティを確保するため，各ユーザが一度に視聴できるカメラ映像の個数を二つに制限する．その上で，発話によって他の話者に割り込みをかけたユーザの映像をリアルタイムに切り替えることで，話題の変化を容易にフォローできるようにしている．

２．WebRTC

WebRTC は W3C が提唱する Web ブラウザやモバイルアプリでリアルタイムコミュニケーションを実現するための API であり，対応ブラウザが搭載されていれば端末の種類や OS を問わずに利用することができる．現在対応しているブラウザは Google Chrome，Mozilla Firefox，Opera であり，低レイヤーでは Microsoft Edge も対応している(Safari や Internet Explorer に関してはプラグインが公開されており，プラグインを適用することによって対応可能)．WebRTC を利用した実アプリケーションの例として，チャットサービス，ゲーム，P2P ファイル共有サービスなどがあげられる．

３．提案システムの構成

提案システムは，専用ソフトをインストールすることなしに Web ブラウザのみで利用できることを意識し，Web 標準技術を利用して実装している．具体的には HTML5 や JavaScript を利用し，Web アプリケーションとして実装した．本節では実装の概要を述べる．

P2P video conference system of twin-view type using WebRTC
† Tomohiko KIMURA, Hiroshima University
‡ 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．

図 1．通信の様子

WebRTC を簡単に利用できるようにするために用意された既存ライブラリとして PeerJS がある．PeerJS は，各ピアに振られたユニークな ID を指定して通信の接続・切断が行えるメソッドや，通信中の様々な出来事に対するイベントなどを提供している．SkyWay は PeerJS に NTT コミュニケーションズが独自拡張を施したライブラリであり，2017 年 1 月現在無料で利用できる．また SkyWay では STUN サーバや TURN サーバ，シグナリングサーバとしての機能も提供されている．提案システムにおいて SkyWay は，ピア間の接続の仲立ちをしてくれるシグナリングサーバとしての役割を果たしている．また各ピアにユニークな ID を割り当て，ピアを接続・切断するための方法を提供することで，ID によるピアの管理を容易にしている．WebRTC では SDP(Session Description Protocol)でデータの種類や IP アドレス，暗号化の鍵などをやり取りすることによって P2P 通信を可能にしているが，SkyWay はこの部分も自動的に行ってくれる．さらに NAT traversal をするために STUN サーバを用いたポートマッピングをしたり，TURN サーバによって通信をリレーしたりすることもある．

上述の SkyWay によってピア間を接続することで WebRTC による P2P 通信路が確立する．提案システムでは，この通信路上でビデオデータや音声データ，テキストデータやバイナリデータなどを送ることでビデオ会議システムを実現している． SkyWay と WebRTC による各ピアの通信の様子を図 1 に示す．

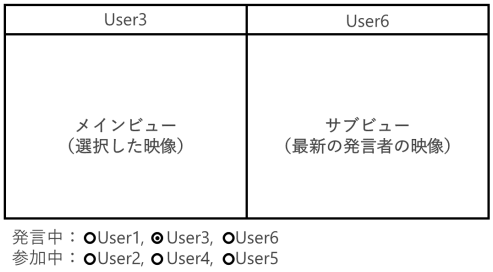

図 2. 提案システムの画面構成

提案システムのユーザ画面は，図 2 のようにメインビューとサブビューのツインビューからなる．メインビューにはユーザが選択したストリーム映像が表示される．映像の切り替えは，画面下の参加ユーザ一覧から見たいユーザにチェックすることで実現される．サブビューには，一番最近に発言を開始した発言者のストリーム映像が表示される．サブビューの映像をユーザ自身が選択することはできない．このような二つのビューを視聴することで，各ユーザは自分が注目する話者の映像を確認しつつ，現在行われている発言もチェックすることができ，話題の変化についていくことができる．

メインビューとサブビューのツインビューに制限するのは，各ピアが完全結合で接続した場合にネットワーク全体のトラフィックが飽和してしまうことを仕組みの上で防ぐためである．

４．発話によるサブビューの切り替え

提案システムにおけるサブビューの切り替えは次の手順で行われる．マイク入力を通してユーザの発話を検知したピアは，会議に参加しているすべてのピアの ID を取得し，各ピアに向けて自身の映像ストリームを送信する．ストリームを渡されたピアは受け取った映像ストリームの再生をサブビューで開始し，それまでサブビューで再生されていた映像ストリームを配信していたピアとの接続を遮断する．

発話の検知は Web Audio API や Web Speech API などの Web 標準技術を用いて行う．ここで Web Audio API は音響処理のための API であり，音の生成・エフェクト処理・分析などの高度な音響処理を行うことができる．また Web Speech API は，発話の開始や終了の検知に加えて音声合成や音声認識なども行うことができる API であり，音声データとテキストデータの間の双方向変換をブラウザで実現することができる．これらの API はいずれも Web 標準技術であるが，Web Speech API の音声認識の実装が各ブラウザにおいていまだ部分的な実装にとどまっていることもあり，今回実装したプロトタイプでは Web Audio API を用いることにした．

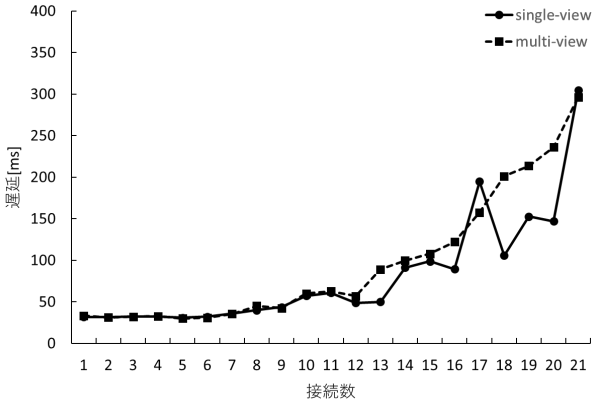

図 3. 接続要求が実際に到達するまでの時間

Web Audio API 上の発話検知は AnalyserNode によって行われる．具体的には，AnalyserNode によってマイクの音量を一定間隔で取得し，音量の平均が設定した閾値を超えた状態を発話中と判断する．閾値は事前に発話中のときの音量を測定し，その最小値から設定した．これにより発話の開始や終了のすばやい検知が可能になる．

５．予備実験

提案システムの性能を評価するため，あるピアが接続を要求してからその要求が実際に届くまでの時間を計測した．実験では，2 台の端末を用意し，各端末でブラウザを 1 タブのみ開き最新のひとつの映像だけシングルビューで再生した．同時に接続するピア数を増やしていき，ピア数の変化が接続所要時間にどう影響するかを調べた．実験は 5 回行い，外れ値を除外した上で平均をとった．また同様の実験を複数の映像を再生したマルチビュー環境でも行った．

結果を図 3 に示す．接続数が増えるにしたがって，接続にかかる遅延が加速度的に増加している．また遅延に関しては数十 ms 程度になっており，十分実用になる速度であることが確かめられた．

参考文献

[1] WebRTC, https://webrtc.org/
[2] MDN, https://developer.mozilla.org/ja/docs/Web
[3] SkyWay, http://nttcom.github.io/skyway/
[4] HTML5Experts.jp, https://html5experts.jp/