

マルチパス通信技術を活用した WebRTC におけるメディア ストリームトラフィックの柔軟な制御の実現

金子 直矢¹ 伊東 孝紘² 渡辺 敏暢² 阿部 博¹ 大西 亮吉^{1,2}

概要: 本研究では、WebRTC を用いたアプリケーションにおいてマルチメディア伝送の可用性と柔軟性を向上するためにマルチパス技術を導入し、メディアストリームごとのポリシーに基づき通信パスの利用形態を制御する手法を提案する。WebRTC を用いたアプリケーションとして車両やロボットの遠隔操作システムといった事例が存在する。遠隔操作システムは、操作に必要な環境情報を伝えるために複数のカメラやマイクを併用しており、操作者のニーズに応じて必要なメディアストリームを安定的に伝送する通信基盤が重要である。そこで WebRTC を用いる遠隔車両運転システムをターゲットとして、以下の 3 手法を導入し組み合わせることを提案する。(1) IP ToS 値を用いたメディアストリームの識別による Web ブラウザとネットワーク制御の連携 (2) マルチパス通信技術の導入による通信の選択肢多様化 (3) 優先度・重要度に基づくメディアストリームごとのパス利用形態制御。これらの手法を、遠隔車両運転システムが用いるモバイル通信サービスを利用して検証し、実現性と通信制御の柔軟性の向上に寄与することを示した。

1. 背景と目的

Web ブラウザの多機能化およびアプリケーションプラットフォームとしての広まりを受けて、映像や音声メディアを多用するソフトウェアを Web アプリケーションとして実現する例がある。WebRTC[1] は、こうしたマルチメディア情報を活用しユーザ間のコミュニケーションを可能にする Web アプリケーション向けの枠組みとして利用されている。本稿ではマルチメディア情報のトラフィックをメディアストリームと呼称する。WebRTC は、アプリケーション利用者のブラウザの間で P2P による通信路を確立し、リアルタイム性の高い映像・音声メディアストリーム伝送を可能としている [2]。

こうしたメディア伝送に特化した WebRTC をコミュニケーションのみならず、遠隔地でのインタラクティブな操作へ応用する事例が存在する。文献 [3] では WebRTC を用いて、遠隔からドローンの制御を行っている。

本研究では、WebRTC を遠隔車両運転システムに対して適用した事例をターゲットとする。遠隔車両では、操作者に対して運転に必要な外部環境を提示するためにカメラやマイクが複数個取り付けられている。遠隔車両運転システムが WebRTC を用いることで、遠隔地にいる操作者に

対し映像メディア、音声メディアとしての外部環境情報を低遅延に伝送することが可能となる。

遠隔運転のための映像および音声メディアは、遠隔車両から操作者に対して継続的に伝送される必要がある。一方で、遠隔車両は一般的に公道を含む環境で稼働することが想定されており、通信路として 3G/4G/5G といったモバイル通信サービスの品質に依存している。モバイル通信サービスでは、遠隔車両の位置および時間によってサービス品質が変化するため、単一回線ではメディア伝送に必要な品質を満たせない可能性がある。

単一の通信手段では十分な通信品質が得られない環境を対象として、マルチパス通信技術を導入する手法が提案されている [4][5]。複数の通信回線を併用することで、通信の可用性を高めることが可能である。

遠隔車両運転システムで用いられる映像や音声メディアは、運転状況に応じてそれぞれの重要度が変化する。文献 [6] では、360 度 VR 環境を対象に視野を中心として映像品質を段階的に制御し、必要な帯域幅を低減する手法を提案している。遠隔車両運転システムでは、同様に操作者の視野や遠隔車両の運転方向に応じてメディアの品質および伝送品質の制御を行うことで回線負荷を低減することが考えられる。例として視野や運転と一致する方向を指向するメディアは重要度高、逆方向は重要度低とし、それぞれメディア品質および伝送品質を制御することが考えられる。メディアの重要度・必要性に応じて品質を制御することで、

¹ トヨタ自動車株式会社

Toyota Motor Corporation

² ウーブン・プラネット・ホールディングス株式会社

Woven Planet Holdings, Inc.

通信手段および回線帯域が限られた環境で、重要なメディアの伝送を優先しつつ通信容量を最適化することが可能となる。

そこで本研究では、WebRTC が実現する低遅延メディアストリーム伝送へマルチパス通信技術を導入し、ストリームごとに通信パスを制御することができる手法を提案する。本手法により、特性の異なる複数の通信回線を束ねメディアの重要度に応じて回線の利用形態を制御することができ、可用性の向上や帯域利用の制御最適化を実現することが可能となる。

本論文の構成は以下の通りである。第 2 章では関連研究を紹介する。第 3 章では、WebRTC における通信技術を踏まえ、マルチパス通信技術の導入にあたっての課題を抽出する。第 4 章では、本研究で提案するマルチパス通信技術および、WebRTC におけるメディアストリーム識別とマルチパス制御の導入手法について述べる。第 5 章では、第 4 章で述べた手法の実装について解説する。第 6 章では、実験により本手法の実現性を評価する。第 7 章では、実験結果の考察を行う。最後にまとめと今後の課題について整理する。

2. 関連研究

複数回線を併用するマルチパス通信技術を実現するため様々な研究が進められている。代表的な例として Multipath TCP (MPTCP)[7], MP-QUIC[8] [9] や MP-DCCP[10] といったプロトコルが提案および標準化されている。MP-TCP を対象としてトラフィックごとのパス制御を行う手法として、文献 [11], 文献 [12] が提案されている。

本研究では、WebRTC のトランスポートプロトコルである UDP を対象としてマルチパス通信技術の導入を提案する。輻輳制御や再送制御は WebRTC が内部で実装している Google Congestion Control [13] に一任することとし、マルチパス制御のみを行う点が異なる。既存研究およびプロトコルでは、トランスポート層で輻輳制御を導入している。

WebRTC の内部で用いられている、RTP(Realtime Transport Protocol) をマルチパス化する試みとして、文献 [14] では MPRTCP を提案している。本研究では UDP を対象としてマルチパス制御技術を導入している。Web ブラウザからはメディアストリーム識別のための情報を提示し、マルチパス制御は Web ブラウザ外で行う構成を取る点が異なる。

UDP を用いたマルチパス制御のオープンソースソフトウェアとして、MLVPN[15], Glorytun[16] が公開されている。前者をドローンの通信に対して適用した研究 [17] がある。本研究では VPN 方式ではなくプロキシ方式を提案している。本方式により特定のアプリケーションのトラフィックのみを対象とした選択的なマルチパス制御が可能

となる点が異なる。

マルチパスを活用した映像伝送技術として MP-DASH [18] が提案されている。文献 [18] では、MP-TCP を用いた通信パス選択によるコスト最適化の実現について言及されている。本研究では UDP ベースのマルチパス制御技術を WebRTC に対して適用している点が異なる。既存のフレームワークへと適用したことにより、アプリケーションでは引き続き WebRTC の利便性を活かすことができる。

車載通信環境を対象とし、通信コストやアプリケーションの需要に応じてトラフィックを制御する研究として文献 [19] の手法が存在する。文献 [19] では、トラフィックをフローとして抽象化し制御する UDP ベースのプロトコルを提案し、プロキシ方式での制御を行っている。本研究では、WebRTC をアプリケーションとして想定しメディアストリームを単位とした制御を行う点が異なる。またマルチパス通信技術を導入し複数回線を用いることで通信の柔軟性を実現している点が異なる。

WebRTC がネットワーク側に対してメディアストリームの QoS を設定する API として WebRTC Priority Control API[20] が定義されている。この API では”very-low” ”low” ”medium” ”high” の 4 つの区分を指定可能であり、区分に基づいてパケットの IP ToS/DSCP フィールドの値が設定される。本研究では IP ToS フィールド全体を利用することとし、最大で 127 個のメディアストリームをネットワーク側から識別可能とした点が異なる。文献 [20] の API では 6 つ以上のメディアストリームを識別した制御ができない。例として、遠隔車両のケースでは前後左右のカメラおよびマイクの計 8 メディアストリームが考えられる。

3. WebRTC における通信制御の課題

本章では、メディアストリームごとの制御を行うにあたり対象となる WebRTC の通信コンポーネントおよび課題についてまとめる。WebRTC の通信に関わるコンポーネントは図 1 にあるように概観できる。

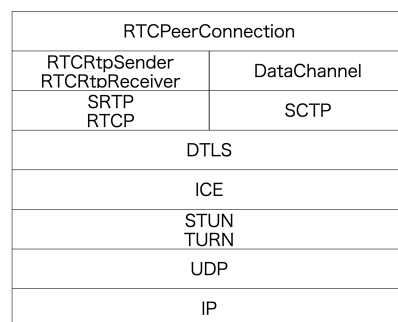


図 1 WebRTC の通信プロトコル、コンポーネントスタック

WebRTC は、RTCPeerConnection を介してメディアストリームとデータストリームを送受信するスキームを

提供する。内部ではメディアストリームの送受信を扱う RTCTpSender/RTCTpReceiver と、テキストデータを中心としたデータストリームを扱う DataChannel の 2 つに分かれる。それぞれのメディアストリームは SRTP (Secure RTP), データストリームは SCTP (Stream Control Transmission Protocol) にカプセル化され, DTLS (Datagram Transport Layer Security) により暗号化されたパケットとして送受信される。RTCP (RTP Control Protocol) は, SRTP を用いたメディア伝送の, 品質情報の交換のために用いられる。WebRTC は SRTP を対象として輻輳制御を導入しており, 例として Google Chrome では輻輳制御アルゴリズムとして Google Congestion Control を採用し実装している。

ブラウザ間の通信路は ICE (Interactive Connectivity Establishment) が確立する。ICE は Web ブラウザが動作する端末のネットワーク状況調査およびブラウザ間の疎通確認を行い, 収集した通信経路候補 (ICE Candidate) 情報を相手方のブラウザと交換し, 通信経路を決定する。疎通がある場合は直接通信が行われるが, ファイアウォールや NAT 装置が存在する場合は, STUN (Session Traversal Utilities for NATs) による UDP ホールパンチングまたは TURN (Traversal Using Relays around NAT) によるリレーを利用してブラウザ間の通信経路を確立する。複数の通信経路候補がある場合は, 優先度に基づいて ICE Candidate が選択される。Google Chrome では ICE として Trickle ICE[21] が用いられる。

上述の特性を備えた WebRTC を対象として, メディアストリームごとの制御を行うために以下のような点が課題となる。

課題 (1) メディアストリームパケットを識別できない

それぞれのメディアストリームは SRTP を用いてカプセル化される。ブラウザ内部では RTP パケットの SSRC(Synchronization Source, 同期送信元識別子) フィールドを用いてメディアストリームの識別が可能である。一方で, ブラウザ外では DTLS により暗号化されるためパケットのインスペクションによる識別ができない。ネットワーク側で制御を行うにあたってはそれぞれの DTLS/UDP パケットがどのメディアストリームに属しているかを識別する手法が必要となる。

課題 (2) 通信経路・プロトコルが動的に決定される

メディアストリーム伝送のために利用される通信経路は ICE により動的に決定される。この通信経路はブラウザの通信環境や WebRTC アプリケーションまたはシグナリングサーバの設定により異なるためマルチパス制御の対象とすべき通信の判別が困難である。例として Google Chrome では TURN over TCP の利用も可能であるため, 制御対

象とすべきトランスポートプロトコルの識別も行う必要がある。

4. 提案手法

第 3 章で述べた課題を解決し, WebRTC のマルチメディアストリームをマルチパス制御するために, 本研究では以下の手法を組み合わせる。

- (1) IP ToS 値を用いたパケットごとのメディアストリーム識別
- (2) TURN サーバを用いた通信経路の制限
- (3) UDP プロキシを介したマルチパス制御

図 2 に, 本研究で提案するシステムの概要を示す。本システムはカメラやマイクによる映像や音声メディアを送信するクライアント端末とメディアを受信し出力するリモート端末, および通信をリレーする中継サーバから構成される, 既存の WebRTC の枠組みを踏襲する。このうちクライアント端末において Web ブラウザがパケット単位でメディアストリームを識別可能な情報を付与し, クライアント端末と中継サーバ間において複数の通信パスを併用する点が従来の WebRTC からなるシステムと異なる。

次節以降, 各手法について説明を行う。

4.1 IP ToS 値を用いたパケットごとのメディアストリーム識別

ブラウザ内でのメディアストリームの区別をブラウザ外部のネットワーク制御で行うため, 本研究では IP ToS フィールドを用いる。本手法により第 3 章で述べた課題 (1) であるパケットの識別の問題を解決する。

提案するシステムは, Web ブラウザとネットワーク制御を行うコンポーネントのそれぞれにメディアストリームと IP ToS 値の対応関係を定義する。Web ブラウザはメディアストリーム (SSRC) と IP ToS の対応関係を保持し UDP パケットごとに IP ToS 値を変えて送信を行う。Web ブラウザ外でネットワーク制御を行うコンポーネントは, IP ToS 値をもとに各メディアストリームのパケットを区別し通信パスの制御を行う。本手法により, Web ブラウザ外のコンポーネントでも WebRTC の DTLS により暗号化されたパケットをメディアストリームごとに制御することが可能となる。

IP ToS を変更可能な API として WebRTC Priority Control API[20] が存在する。本研究では, この API を拡張し同様の枠組みで, IP ToS を値として設定可能な API を提供する。

4.2 TURN サーバでの利用による通信経路の制限

WebRTC のトラフィックを制御するため, 本研究では WebRTC アプリケーションに TURN サーバの利用を強制し, プロトコルとして UDP を用いる。直接通信または

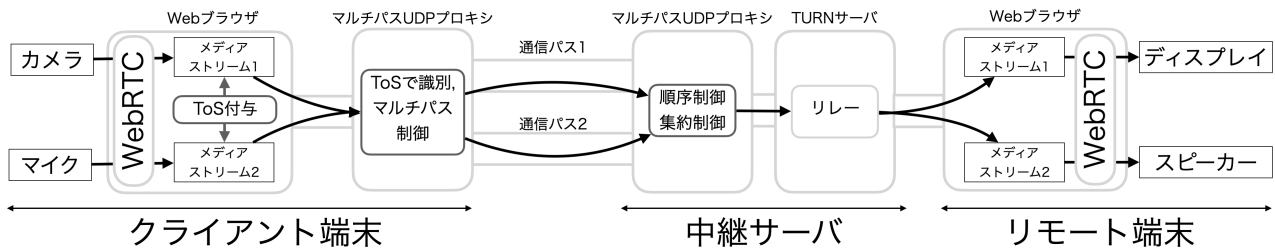


図 2 WebRTC メディアストリームトラフィックのマルチパス制御システムの概要

```
const pcConfig = {
  iceServers = [{
    urls: "turn:192.0.2.253:3478",
    username: "bob",
    credential: "hello"
  }],
  iceTransportPolicy: "relay"
};
pc = new RTCPeerConnection(pcConfig);
```

図 3 TURN サーバの利用を指定

STUN を利用した通信経路の利用を制限し、第 3 章で述べた課題 (2) である制御すべき通信経路および種別が定まらない問題を解決する。

WebRTC アプリケーションは、図 3 に示すように ICE サーバとして TURN を用い、後述する UDP プロキシをリレーサーバとして指定する。また TURN 以外の利用を制限するために、iceTransportPolicy を relay と定義する。

4.3 UDP プロキシを介したマルチパス制御

本研究では、通信の制御のためにプロキシ方式を用いる。プロキシはクライアント端末側とサーバ側で動作する 2 つのコンポーネントで構成される。それぞれのプロキシは UDP の通信を受けて、プロキシ間の複数の通信経路を束ねてパケットの送受信を行う。本稿ではこのプロキシをマルチパス UDP プロキシと呼称する。

本プロキシは以下に挙げる制御を行う。

- 待ち受けポート、IP ToS に基づくパケットの弁別
- 送受信トラフィックをフローとして抽象化
- フローごとにマルチパスの利用形態を選択

第 4.2 節 で述べた TURN サーバの利用に従って、WebRTC はブラウザ間の通信をクライアントプロキシへと転送する。クライアントプロキシは、ポリシーに基づいて複数の通信パス経由でサーバ側プロキシへ転送する。サーバ側プロキシは、クライアント端末側プロキシから受信した通信の重複排除および順序制御を行い TURN サーバへ転送する。

プロキシ間では、それぞれのメディアストリームはクライアント側プロキシでの待ち受けポートおよび IP ToS に

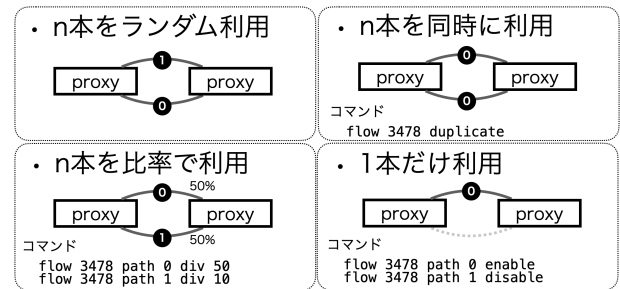


図 4 マルチパス UDP プロキシでの送信パス利用形態

応じてフローとして抽象化され、識別子をもとに制御される。図 4 に制御ポリシーの一覧を示す。本プロキシは利用可能な通信パスのうち 1 つ以上の任意のパスを、ランダムで併用する方式、比率をもとに併用する方式、同じパケットを送信し冗長利用する方式を選択することができる。またフローの送信に利用しないパスを選択することができる。

5. 実装

本研究では、Google Chromium 86.0.4240.198 を対象として、SSRC と ToS のテーブルをもとに SRTP パケット送信時にソケットの IP ToS 値の書き換えを行う機能を Web ブラウザへ組み込んだ。メディアストリームパケットの送信時のコールグラフおよび変更箇所の抜粋を図 5 に示す。

本システムでは図 5 に示されるように、webrtc::SrtpTransport::SendRtpPacket メソッドで RTP パケットの SSRC に応じてメタ情報 (rtc::PacketOptions.dscp) に ToS 値を付与する。ブラウザは、パケットをネットワークに送出する際にソケットオプションとしてこの値を用いる。IP ToS 値を設定していないメディアストリームおよび RTCP の制御パケットや DataChannel の SCTP パケットは IP ToS=0 となる。

図 6 に、それぞれのホストにおけるマルチパスプロキシの設定例を示す。プロキシの設定はサービス定義部とデータパス部からなる。クライアント側プロキシではサービス定義部に、待ち受けポートおよび IP ToS の値で示されるメディアストリームとフローとの対応関係を定義する。図 6 の例では UDP ポート 3478 で待ち受けたトラフィックをフロー識別子 (flow_id) 3478, 13478, 23478 と定義している。このうち先頭のフロー識別子をプライマリフローとし

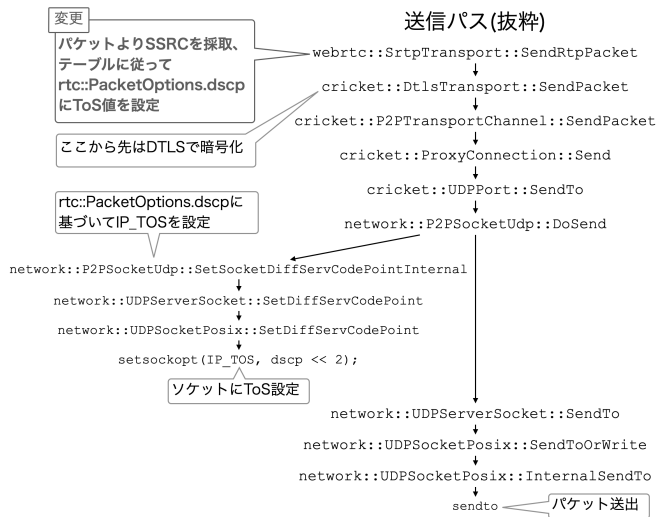


図 5 Chromium のメディアストリームパケットにおける SSRC と IP ToS の紐付け

て扱い、IP ToS の値に基づきトラフィックに応じてフローを弁別することができる。この例では IP ToS=4 は 13478, IP ToS=8 は 23478 とフロー識別子を分けて扱う。

サーバ側プロキシでは、フローと宛先ホストおよび宛先ポートの対応関係を定義する。図 6 では、flow_id 3478, 13478, 23478 のフローと、宛先ホスト 203.0.113.100, 宛先ポート 3478 のサービスとの対応関係が定義されている。サーバ側プロキシは、クライアント側プロキシから受信した該当フローのパケットを、指定された宛先のサービスへ転送する。サーバ側プロキシは、サービスより受信したパケットを、プライマリフローのトラフィックとして扱い、クライアント側プロキシへ送信する。

データパス定義部は、プロキシが用いる通信路を定義する。各通信路はパス識別子 (path_id) により区別される。この識別子はマルチパス利用ポリシー設定にあたり、フローごとに用いるパスを指定する際に利用される。

6. 評価

6.1 評価環境

提案システムの評価にあたり図 7 に示される環境を構築した。異なる事業者のモバイル通信サービスを 2 回線用いて、実環境での評価を行う。

クライアント端末では、マルチパス UDP プロキシ (mp-proxy) のクライアント側プログラム、および IP ToS を可変にした Google Chromium 86.0.4240.198 を動作させた。クライアント端末は、通信回線として異なる 2 事業者の 3G/4G サービスを用いた。以降それぞれの通信回線を path 0, path 1 と呼称する。クライアント端末は、1 つの Web カメラおよび 1 つのマイクを備えており、WebRTC を経由して 2 つのメディアストリームをリモート端末に伝送する。

中継サーバ側では coturn 4.5.1.2 およびマルチパスプロ

キシのサーバ側プログラムを動作させた。リモート端末は、クライアント端末から伝送されたメディアストリームを受け映像および音声をディスプレイとスピーカに出力する。リモート端末では、通信回線として安定した固定回線を用いており、Web ブラウザとして Google Chrome を動作させた。

6.2 モバイル回線冗長利用の効果評価

本節では、提案するマルチパス制御を介してモバイル回線を冗長利用した場合と、従来手法を用いて 1 本のみを利用した場合を比較する。

それぞれの場合で、10 分間のメディア伝送を行い、モバイル回線のインタフェース統計情報および WebRTC の統計情報を採取することで比較を行う。インタフェース統計情報は、クライアント端末側の path0 および path1 のインタフェースから採取する。WebRTC の統計情報は、リモート端末側の Chrome の webrtc-internals から採取する。

メディア伝送を行った際の、インタフェース統計情報より採取した送信スループットの推移を図 8 に示す。冗長利用 (path 0) および冗長利用 (path 1) で示される黒色の実線および破線は、マルチパス UDP プロキシを用いて 2 つの通信回線を冗長利用した場合の、path 0 と path 1 のそれぞれのインタフェースで観測された送信スループットを示す。単独利用 (path 0) で示される灰色の実線は、マルチパス UDP プロキシを利用せず従来通りの手法を用い path 0 のみでメディア伝送を行った場合の送信スループットを示す。単独利用 (path 0) の実験時には、クライアント端末とリモート側端末とは ICE に従って選択された第三者の TURN サーバを介して Web ブラウザ間通信を行っていた。

メディア伝送を行った際の WebRTC 統計情報のうち、映像メディアにおけるパケットロス数の推移を複数回線利用と単独回線利用とで比較した結果を図 9 に示す。冗長利用 (path 0 + path 1) の実線は、マルチパス UDP プロキシを用いて 2 つの通信回線を冗長利用した場合のパケットロス数を示し、単独利用 (path 0) の実線は、マルチパス UDP プロキシを利用せず従来通りの手法を用い path 0 のみでメディア伝送を行った場合のパケットロス数を示す。

6.3 回線切替時の通信継続性評価

本節では、提案するシステムを用いて WebRTC の通信パスを切り替えた場合の挙動について評価を行う。あらかじめメディアストリームがある回線を利用している状況から、別の回線へと切り替えを行った場合に WebRTC およびアプリケーションに対して与える影響を評価する。

初期状態で path 0 のみを有効化し、path 1 は無効化した状態でメディアストリーム伝送を開始する。メディアストリーム伝送が行われている間に、path 0 から path 1 への切

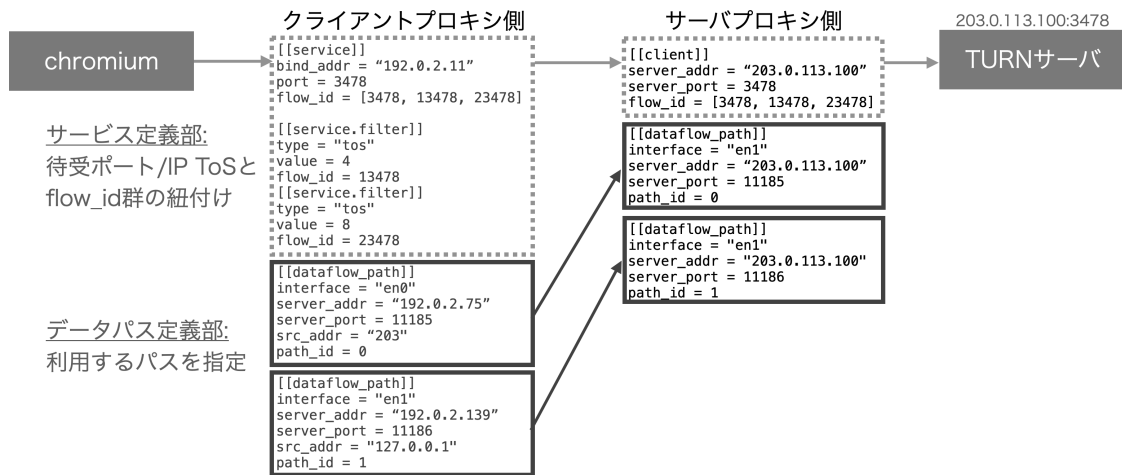


図 6 マルチパス UDP プロキシの設定例

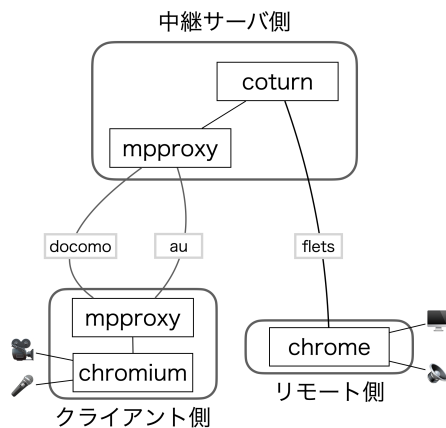


図 7 評価環境

り替えを行う。統計情報として、クライアント端末側のインタフェース統計情報およびリモート端末側の WebRTC 統計情報を採取する。回線切り替えにあたり用いたコマンドは図 10 の通りである。

図 11 に、回線切替時の path 0 および path 1 の送信スループットの推移を示す。path 0 で示される実線は path 0、path 1 で示される破線は path 1 の送信スループットを示す。00:05:00 から 00:06:00 の間に path 0 から path 1 への切り替えを行った。

図 12 に、WebRTC 統計情報より採取した、回線切替時のリモート端末側における映像メディア受信スループットを図示する。図 13 では、回線切り替えの前後における遅延の分布を比較する。path 0 利用時は 00:05:00 以前の path 0 のみを用いていた時間、path 1 利用時は 00:06:00 以降の path 1 のみを用いていた時間における、WebRTC 統計情報での映像メディアの遅延の分布を示す。

7. 考察

図 8 は複数回線を冗長して用いた場合の 2 つの回線の送信スループットを図示しており、同量程度のトラフィック

が 2 つの回線に流れていることがわかる。Web ブラウザからは従来通り ICE の設定に従って TURN サーバ宛の通信が 1 パスで行われるが、マルチパスプロキシの制御を受けて中継サーバとの間で複数回線が冗長利用されていると考えられる。

図 9 では、複数回線を冗長利用した場合と 1 つの回線を用いた場合の、映像メディアの packet loss 数の推移を比較した。単独回線よりも複数回線の冗長利用の方が packet loss の発生が低く抑えられている。複数回線の冗長利用により、packet loss の低減および可用性の向上に資すると思われる。

図 11 ではマルチパス制御の 1 オペレーション例としてアンダーレイとなる回線を切り替えた場合の挙動を評価した。切り替えタイミングを契機に path 0 の送信が停止し、path 1 の送信が開始しており、回線利用の制御が行われていることが確認できる。図 12 では、WebRTC の統計情報においても切り替え制御を行った時刻を跨いで連続してメディアが伝送されていることが確認できる。図 13 に見られる遅延特性の変化は、path 0 から回線の切り替えがなされ path 1 の特性に替わったためと考えられる。

WebRTC アプリケーションでは、回線切替を契機とした ICE の切断や再接続イベントは発生しなかった。WebRTC の統計情報より受信スループットが連続して記録されていることから、本システムが WebRTC に対してアンダーレイの通信パスの変化を隠蔽できていることがわかる。通信パスの変化が隠蔽されていることにより、WebRTC およびアプリケーションに影響を与えずに通信パスを制御することができていると考えられる。

実環境の WebRTC アプリケーションは、RTCPeerConnection を確立するために SDP (Session Description Protocol)[22] にしたがって記述されたメッセージを交換するシグナリング用の通信路を必要とする。シグナリングの通信路および用いられるトランスポートプロトコルは規定さ

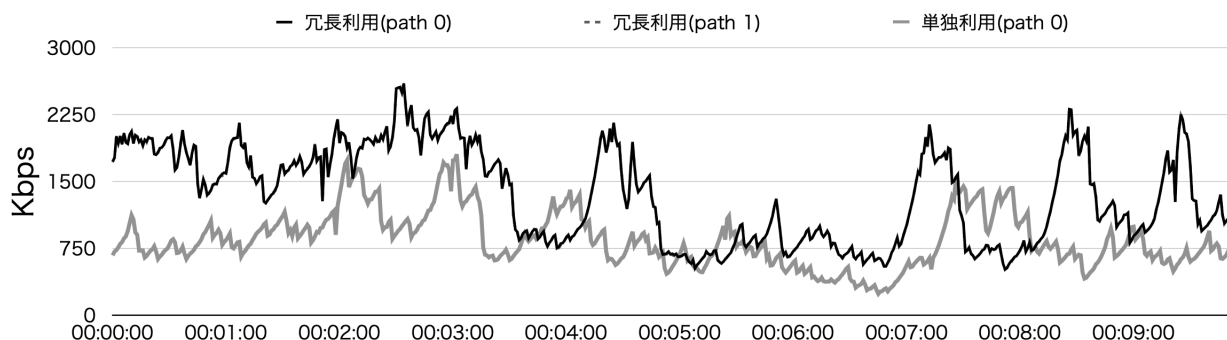


図 8 複数回線併用時と単独回線利用時の送信スループット

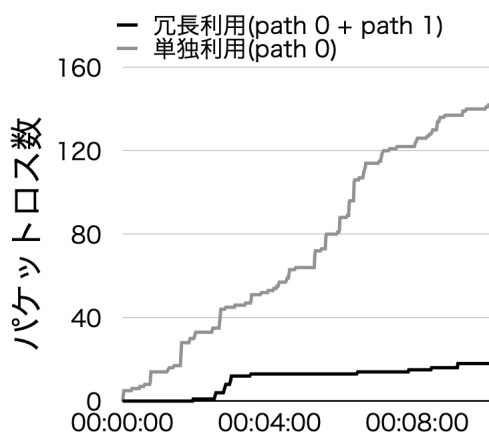


図 9 複数回線併用時と単独回線利用時のパケットロス数の比較

```
# 初期状態: 2 本目のパスを無効化
% flow 3478 path 1 disable

# 2 本目のパスを有効化
% flow 3478 path 1 enable

# 1 本目のパスを無効化
% flow 3478 path 0 disable
```

図 10 回線切り替えのコマンド例

れておらず実装に依存する。HTTP(S) や WebSocket など TCP トランスポートを用いる場合、本研究で提案したマルチパスプロキシの制御対象とならない。シグナリング通信に対しても冗長性が必要な場合、MP-TCP の導入やマルチパスプロキシの TCP 対応が必要となる。

8. まとめと今後の課題

本研究では WebRTC のメディアストリームトラフィックを対象としてマルチパス通信技術を導入し制御を行う手法を提案した。提案手法により、メディアストリームごとに通信パスを動的に変更することが可能であること、および複数回線の冗長利用が可能であることを示した。

本稿では、WebRTC のトラフィック誘導のため、TURN サーバを強制的に利用することとした。ブラウザ間で直接通信あるいは STUN を用いた通信が可能であった環境でも新たなボトルネック要因としてリレーサーバが介在することとなる。TURN サーバの性能およびスケラビリティの評価の必要性が今後の課題となる。

マルチパス制御を介することによる輻輳制御への影響についても今後調査が必要である。本研究では回線の冗長利用や回線切り替えのケースを取り扱った。複数回線を冗長利用ではなく並行して利用した場合に Google Congestion Control の挙動に与える影響について検証が必要である。

謝辞 本研究はオープン・プラネット・ホールディングス株式会社との共同研究により実施されたものである。

参考文献

- [1] Alvestrand, H. T.: Overview: Real-Time Protocols for Browser-Based Applications, RFC 8825 (2021).
- [2] Jansen, B., Goodwin, T., Gupta, V., Kuipers, F. and Zussman, G.: Performance Evaluation of WebRTC-Based Video Conferencing, *SIGMETRICS Perform. Eval. Rev.*, Vol. 45, No. 3, p. 56–68 (online), DOI: 10.1145/3199524.3199534 (2018).
- [3] Chodorek, A., Chodorek, R. R. and Wajda, K.: Media and non-media WebRTC communication between a terrestrial station and a drone: the case of a flying IoT system to monitor parking, *2019 IEEE/ACM 23rd International Symposium on Distributed Simulation and Real Time Applications (DS-RT)*, pp. 1–4 (online), DOI: 10.1109/DS-RT47707.2019.8958706 (2019).
- [4] Ishac, J. and Sargent, M. T.: *Improving Scientific Payload Communication Over Multiple, Low-Capacity Links*, National Aeronautics and Space Administration, Glenn Research Center (2018).
- [5] Li, L., Xu, K., Li, T., Zheng, K., Peng, C., Wang, D., Wang, X., Shen, M. and Mijumbi, R.: A Measurement Study on Multi-Path TCP with Multiple Cellular Carriers on High Speed Rails, *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication, SIGCOMM '18*, New York, NY, USA, Association for Computing Machinery, p. 161–175 (online), DOI: 10.1145/3230543.3230556 (2018).
- [6] Lungaro, P., Sjöberg, R., Valero, A. J. F., Mittal, A. and Tollmar, K.: Gaze-Aware Streaming Solutions for the Next Generation of Mobile VR Experiences,

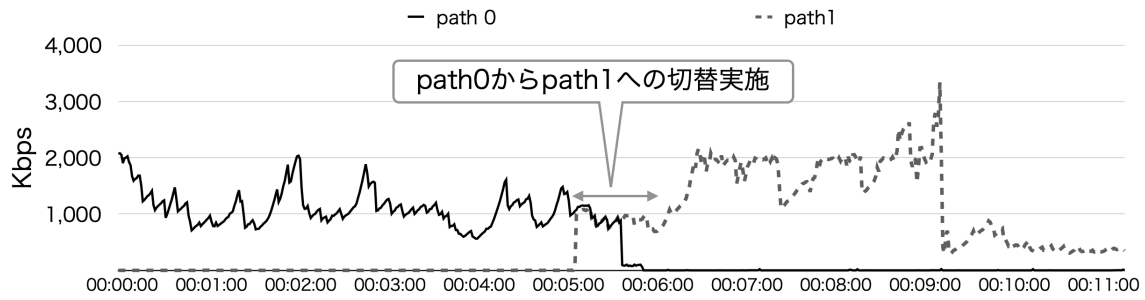


図 11 通信パス切り替え時の挙動: 各通信パスの送信スループットの比較

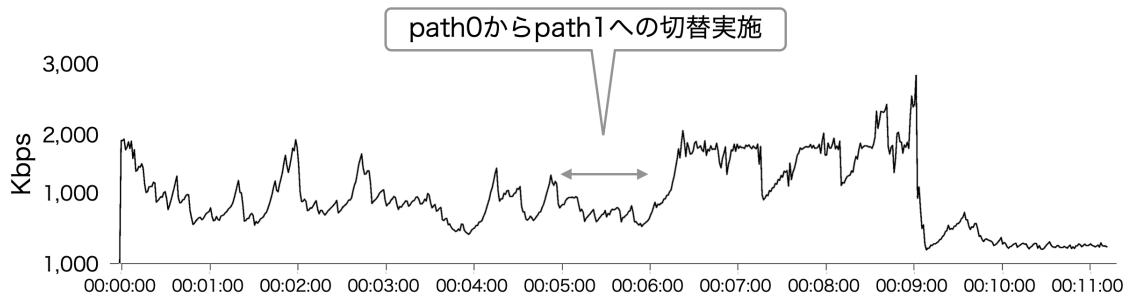


図 12 通信パス切り替え時の挙動: WebRTC の映像メディアの受信スループット

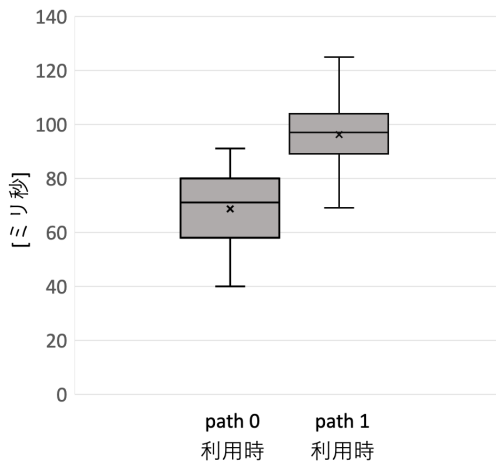


図 13 回線切替前と切り替え後の遅延比較

IEEE Transactions on Visualization and Computer Graphics, Vol. 24, No. 4, pp. 1535–1544 (online), DOI: 10.1109/TVCG.2018.2794119 (2018).

- [7] Ford, A., Raiciu, C., Handley, M. J. and Bonaventure, O.: TCP Extensions for Multipath Operation with Multiple Addresses, RFC 6824 (2013).
- [8] De Coninck, Q. and Bonaventure, O.: Multipath QUIC: Design and Evaluation, *Proceedings of the 13th International Conference on Emerging Networking Experiments and Technologies*, CoNEXT '17, New York, NY, USA, Association for Computing Machinery, p. 160–166 (online), DOI: 10.1145/3143361.3143370 (2017).
- [9] De Coninck, Q. and Olivier, B.: Multipath Extension for QUIC draft-deconinck-quic-multipath-06 (2017).
- [10] Amend, M., Bogenfeld, E., Cvjetkovic, M., Rakocevic, V., Pieska, M., Kessler, A. and Brunstrom, A.: A Framework for Multiaccess Support for Unreliable Internet Traffic using Multipath DCCP, *2019 IEEE 44th Confer-*

ence on Local Computer Networks (LCN), pp. 316–323 (online), DOI: 10.1109/LCN44214.2019.8990746 (2019).

- [11] 怜土堀込, 成良山井, 直哉北川, 智大坐島: MPTCP における外部プログラムによるプライマリサブフローの切替, 技術報告 40, 東京農工大学, 東京農工大学, 東京農工大学, 電気通信大学 (2018).
- [12] 将斗佐藤, 怜土堀込, 龍次浅倉, 成良山井, 直哉北川, 智大坐島: 任意のプロセスの MPTCP サブフロー情報取得および優先度設定を行うシステムコール群の実装, 技術報告 39, 東京農工大学, 東京農工大学, 東京農工大学, 東京農工大学, 東京農工大学, 電気通信大学 (2019).
- [13] Carlucci, G., De Cicco, L., Holmer, S. and Mascolo, S.: Analysis and Design of the Google Congestion Control for Web Real-Time Communication (WebRTC), *Proceedings of the 7th International Conference on Multimedia Systems*, MMSys '16, New York, NY, USA, Association for Computing Machinery, (online), DOI: 10.1145/2910017.2910605 (2016).
- [14] Singh, V., Ahsan, S. and Ott, J.: MP RTP: Multipath Considerations for Real-Time Media, *Proceedings of the 4th ACM Multimedia Systems Conference*, MMSys '13, New York, NY, USA, Association for Computing Machinery, p. 190–201 (online), DOI: 10.1145/2483977.2484002 (2013).
- [15] Coustet, L.: MLVPN - Multi-Link Virtual Public Network, <https://zeshome.github.io/MLVPN/>.
- [16] Gallouët, A.: Glorytun - Multipath UDP Tunnel, <https://github.com/angt/glorytun>.
- [17] Rametta, C., Beritelli, F., Avanzato, R. and Russo, M.: A Smart VPN Bonding Technique for Drone Communication Applications, *2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pp. 612–618 (online), DOI: 10.1109/DCOSS.2019.00112 (2019).
- [18] Han, B., Qian, F., Ji, L. and Gopalakrishnan, V.: MP-DASH: Adaptive Video Streaming Over Preference-Aware Multipath, *Proceedings of the 12th International on Conference on Emerging Networking Experiments*

and Technologies, CoNEXT '16, New York, NY, USA, Association for Computing Machinery, p. 129–143 (online), DOI: 10.1145/2999572.2999606 (2016).

- [19] Asai, H., Doi, Y. and Onishi, R.: Toward Just-in-Time Data Communications over Shared Networks and Computational Resources on Massive Client Environment, *Transportation Research Record*, Vol. 0, No. 0, p. 03611981211006731 (online), DOI: 10.1177/03611981211006731 (0).
- [20] Jones, P., Dhesikan, S., Jennings, C. and D.Druta: DSCP Packet Markings for WebRTC QoS draft-ietf-tsvwg-rtcweb-qos-18 (2016).
- [21] Ivov, E., Rescorla, E., Uberti, J. and Saint-Andre, P.: Trickle ICE: Incremental Provisioning of Candidates for the Interactive Connectivity Establishment (ICE) Protocol, RFC 8838 (2021).
- [22] Perkins, C., Handley, M. J. and Jacobson, V.: SDP: Session Description Protocol, RFC 4566 (2006).