TTC技術レポート
Technical Report

TR-1101

WebRTC に関する技術報告書
データ転送編

Technical Report on WebReal-Time Communication (WebRTC) Data Transport

第 1 版

2022 年 12 月 09 日制定

一般社団法人
情報通信技術委員会

THE TELECOMMUNICATION TECHNOLOGY COMMITTEE

TTC Telecommunication Technology Committee

目次

＜参考＞

1. 国際勧告等の関連

本技術レポートは、RFC8835、RFC8828、RFC8831 および RFC8832 を調査したものである。

2. 上記国際勧告等に対する追加項目等

なし

3. 改版の履歴

版数	制定日	改版内容
第 1 版	2022 年 12 月 09 日	制定

4. 参考文献

[RFC8835] Alvestrand, H., "Transports for WebRTC", RFC 8835, DOI 10.17487/RFC8835, January 2021, <https://www.rfc-editor.org/info/rfc8835>.

[RFC8828] Uberti, J. and G. Shieh, "WebRTC IP Address Handling Requirements", RFC 8828, DOI 10.17487/RFC8828, January 2021, <https://www.rfc-editor.org/info/rfc8828>.

[RFC8831] Jesup, R., Loreto, S., and M. Tüxen, "WebRTC Data Channels", RFC 8831, DOI 10.17487/RFC8831, January 2021, <https://www.rfc-editor.org/info/rfc8831>.

[RFC8832] Jesup, R., Loreto, S., and M. Tüxen, "WebRTC Data Channel Establishment Protocol", RFC 8832, DOI 10.17487/RFC8832, January 2021, <https://www.rfc-editor.org/info/rfc8832>.

5. 工業所有権

本標準に関わる「工業所有権等の実施の権利に係る確認書」の提出状況は、TTC ホームページでご覧になれます。

6. 技術レポート作成部門

第 1 版 ： 企業ネットワーク専門委員会

I. 本技術レポートの概要

近年、テレワークの推進により、Web 会議システムが急速に普及してきた。Web 会議システムにおいてはパソコンやスマートフォン、タブレットなどデバイスを選ばず、Web ブラウザからアクセスすることにより、いつでもどこでも会議を行うことができるというメリットがある。Web 会議システムの通信プロトコルはシステムによって WebSocket であったり独自仕様であったりと様々なプロトコルが使用されている。その中でも近年特に注目されているのが WebRTC である。

WebRTC はブラウザ同士の双方向通信のために 2012 年に規格が策定され、様々な Web ブラウザで実装されてきた。その後テレワークの推進により、さらに注目を浴び、2021 年に IETF による標準化が行われた。

本報告書では TR-1095 に引き続き、IETF によって標準化された WebRTC に関する以下の RFC について日本語に翻訳する。

- RFC8835：WebRTC で使用されるデータ転送プロトコル
- RFC8828：WebRTC 実装における IP アドレスの処理方法(プライバシーとメディアパフォーマンスのトレードオフの処理方法)
- RFC8831：WebRTC コンテキストにおける Stream Control Transmission Protocol (SCTP) の使用方法
- RFC8832：WebRTC データチャネル確立プロトコル

なお、TR-1095 では以下の RFC について日本語に翻訳している。

- RFC8825：IETF によって標準化された WebRTC の仕様の概要
- RFC8834：WebRTC で使用される RTP の取り決め

II.　RFC8835 原文の著作権について

III. RFC8835 の和訳

RFC 8835　WebRTC のトランスポート

概要

この文書では、ファイアウォール、リレー、NAT ボックスなどの中間ボックスとの相互作用に使用されるプロトコルを含む、Web Real-Time Communication (WebRTC) で使用されるデータ転送プロトコルについて説明する。

1. はじめに

WebRTC は、ブラウザ間、およびブラウザとその他のエンティティ間のリアルタイムマルチメディア交換を目的としたプロトコルスイートである。

WebRTC は、WebRTC 概要文書 [RFC8825] に記述されている。この文書では、「WebRTC エンドポイント」および「WebRTC ブラウザ」など、この文書で使用される用語も定義されている。

RTP ソースの用語は [RFC7656] から引用した。

この文書は、ファイアウォール、リレー、NAT ボックスのような中間ボックスとの相互作用のために使用されるプロトコルなど、適合する実装によって使用されるデータ転送プロトコルに焦点を当てる。

このプロトコルスイートは、WebRTC セキュリティ文書 [RFC8826] および [RFC8827] に記述されているセキュリティの考慮事項を満たすことを意図している。

この文書では、すべての WebRTC エンドポイントに適用される要件について説明する。WebRTC ブラウザにのみ適用される要件がある場合は、明示的に呼び出される。

2　要件言語

この文書のキーワードである「MUST」、「MUST NOT」、「REQUIRED」、「SHALL」、「SHALL NOT」、「SHOULD」、「SHOULD NOT」、「RECOMMENDED」、「NOT RECOMMENDED」、「MAY」、および「OPTIONAL」は、ここに示すように、すべて大文字で表示される場合にのみ、BCP 14 [RFC2119] [RFC8174] で記述されているように解釈される。

3　トランスポートとミドルボックスの仕様

3.1　システム提供のインタフェース

ここで使用されるプロトコル仕様は、WebRTC プロトコルの実装で次のプロトコルが使用可能であることを前提としている。

UDP [RFC0768]： これは、説明されているほとんどのプロトコル要素で想定されているプロトコルである。

TCP [RFC0793]： これは、HTTP/WebSocket、TURN/TLS、ICE-TCP に使用される。

どちらのプロトコルでも、IPv4 と IPv6 のサポートが想定されている。

UDP について、この仕様は、複数のメディアタイプが多重化されるときに、[RFC8837] (この文書の 4.2 節を参照) で記述された優先度付けを達成するために、パケットごとに開かれるソケットの Differentiated

Services Code Point (DSCP) を設定する能力を想定している。これはローカル設定の問題であるため、DSCP が優先されるとは想定せず、DSCP がゼロまたは変更される可能性があると想定する。

これらのインタフェースへのアクセスを許可しないプラットフォームでは、準拠 WebRTC エンドポイントをサポートできない。

この仕様は、実装が ICMP または raw IP にアクセスできることを想定していない。

次のプロトコルを使用できるが、WebRTC エンドポイントで実装できるため、「システム提供のインタフェース」として定義されていない。

TURN： NAT 周辺のリレーを使用したトラバーサル [RFC8656]
STUN： NAT のためのセッショントラバーサルユーティリティ [RFC5389]
ICE： インタラクティブコネクティビティの確立 [RFC8445]
TLS： トランスポート層セキュリティ [RFC8446]
DTLS： データグラムトランスポート層セキュリティ [RFC6347]

3.2 IPv4 および IPv6 を使用する機能

WebRTC ブラウザで実行されている Web アプリケーションは、利用可能な場合は IPv4 と IPv6 の両方を利用できなければならない[MUST]。つまり、2 つのピアが相互に IPv4 接続のみを持つ場合、または相互に IPv6 接続のみを持つ場合、WebRTC ブラウザで実行されているアプリケーションは通信できなければならない[MUST]。

TURN が使用され、TURN サーバがピアまたはピアの TURN サーバへの IPv4 または IPv6 接続を持つ場合、適切なタイプの候補がサポートされなければならない[MUST]。ICE [RFC8421] に対する「Happy Eyeballs」仕様は、サポートされるべきである[SHOULD]。

3.3 一時 IPv6 アドレスの使用

IPv6 デフォルトアドレス選択仕様 [RFC6724] は、一時アドレス [RFC4941] が永久アドレスより優先されることを指定する。これは [RFC3484] で規定された規則からの変更である。単一のアドレスを選択するアプリケーションでは、これは通常、[RFC5014] で指定された IPV6_PREFER_SRC_TMP 優先度フラグによって行われる。ただし、プライバシー強化アドレスが静的アドレスよりも優先して使用されるようにすることを目的としたこの規則は、すべてのアドレスが収集されてアプリケーションに公開される ICE では適切な効果を持たない。したがって、代わりに次のルールが適用される。

WebRTC エンドポイントがそのホスト上のすべての IPv6 アドレスを収集し、非推奨の一時アドレスと同じスコープの永続アドレスの両方が存在する場合、WebRTC エンドポイントは、アドレスをアプリケーションに公開する前、または ICE で使用する前に、永続アドレスを破棄する必要がある[SHOULD]。これは、[RFC6724] に記述されているデフォルトポリシーと一致する。

すべてではなく一部の一時 IPv6 アドレスが非推奨とマークされている場合、WebRTC エンドポイントは、進行中の接続で使用されていない限り、非推奨アドレスを破棄する必要がある[SHOULD]。ICE の再起動では、現在使用中の非推奨のアドレスが保持することができる[MAY]。

3.4 ミドルボックス関連の関数

ミドルボックスを処理するための主なメカニズムは ICE である。これは、内部からのトラフィックを受け入れる NAT ボックスとファイアウォールを処理するための適切な方法であるが、内部トラフィック (単純な

ステートフルファイアウォール) に応答している場合は外部からのトラフィックのみを処理する。

ICE [RFC8445] をサポートしなければならない[MUST]。実装は ICE-Lite ではなく、完全な ICE 実装でなければならない[MUST]。完全な ICE 実装では、ICE と ICE-Lite の両方の実装が適切に配置されている場合に、これらの実装とのインターワーキングが可能になる。

両当事者がエンドポイント依存マッピング ([RFC5128]、2.4 節で定義されているように) を実行するタイプの NAT の背後にいる状況に対処するために、TURN [RFC8656] がサポートされなければならない[MUST]。

WebRTC ブラウザは、ブラウザ構成とアプリケーションの両方から、STUN および TURN サーバの構成をサポートしなければならない[MUST]。

STUN および TURN サーバの検出および管理に関しては、サーバ検出のための [RFC8155] および [RETURN] を含む他の作業が存在することに注意されたい。

すべての UDP トラフィックをブロックするファイアウォールを処理するために、WebRTC エンドポイントと TURN サーバの間で TCP を使用する TURN のモードをサポートする必要がある[MUST]。また、WebRTC エンドポイントと TURN サーバの間で TLS over TCP を使用する TURN のモードをサポートする必要がある[MUST]。詳細については、[RFC8656] の 3.1 節を参照する。

一方が IPv4 ネットワーク上にあり、他方が IPv6 ネットワーク上にある状況を扱うために、IPv6 のための TURN 拡張がサポートされなければならない[MUST]。

TURN TCP 候補。WebRTC エンドポイントの TURN サーバからピアへの接続が TCP 接続である場合、[RFC6062] をサポートすることができる[MAY]。

しかしながら、以下の理由から、そのような候補者は大きな利益をもたらすとはみなされていない。

まず、TURN TCP 候補の使用は、両方のピアが接続を確立するために TCP を使用する必要がある場合にのみ関連する。

第 2 に、TCP 上で TURN を使用して UDP リレー候補を確立し、それぞれのリレーサーバに接続することで、このユースケースは異なる方法でサポートされる。

第 3 に、WebRTC エンドポイントの TURN サーバとピアとの間で TCP を使用すると、UDP を使用する場合よりもパフォーマンス上の問題が発生する可能性がある。

ICE-TCP 候補 [RFC6544] はサポートされなければならない[MUST]。これにより、アプリケーションは、TURN サーバを使用せずに、UDP ブロッキングファイアウォールを介してパブリック IP アドレスを持つピアと通信できる。

TCP 接続が使用される場合、[RFC4571] による RTP フレーミングがすべてのパケットに使用されなければならない[MUST]。これには、RTP パケット、データチャネルの伝送に使用される DTLS パケット、および STUN 接続性チェックパケットが含まれる。

[RFC5389] の 11 章 (300 Try Alternate) で指定された ALTERNATE-SERVER メカニズムをサポートしなければならない[MUST]。

WebRTC エンドポイントは、HTTP プロキシを介したインターネットへのアクセスをサポートすることができる[MAY]。その場合、[RFC7639] で規定されている「ALPN」ヘッダを含めなければならず、[RFC7231] および [RFC7235] の 4.3.6 項で規定されているプロキシ認証もサポートしなければならない[MUST]。

3.5 実装されているトランスポートプロトコル

メディアの転送には、セキュア RTP が使用される。使用される RTP プロファイルの詳細は、Media Transport and Use of RTP in WebRTC [RFC8834] に記述されており、サーキットブレーカ [RFC8083] の使用と輻輳制御を義務付けている (さらなるガイダンスについては、[RFC8836] を参照のこと)。

鍵交換は、[RFC8827] に記述されているように、DTLS-SRTP を使用して行われなければならない[MUST]。

WebRTC データチャネル [RFC8831] を介したデータ転送の場合、WebRTC エンドポイントは ICE を介した SCTP over DTLS をサポートしなければならない[MUST]。このカプセル化は、[RFC8261] で規定されている。Session Description Protocol (SDP) におけるこのトランスポートのネゴシエーションは、[RFC8841] で定義されている。I-DATA の SCTP 拡張 [RFC8260] がサポートされなければならない[MUST]。

[RFC8832] に記述されている WebRTC データチャネルの設定プロトコルをサポートしなければならない[MUST]。

> 注意：[RFC5764] で定義される DTLS-SRTP と [RFC8445] で定義される ICE との間の相互作用は、[RFC8842] の 6 章に記述される。この仕様の効果は、1 つのコンポーネントに関連付けられているすべての ICE 候補ペアが、同じ DTLS アソシエーションの一部になることである。したがって、複数の有効な候補ペアがある場合でも、DTLS ハンドシェイクは 1 つだけになる。

WebRTC エンドポイントは、DTLS-SRTP 仕様 [RFC5764] 5.1.2 項の説明に従って、同じポートペアを介した DTLS と RTP の多重化をサポートしなければならない[MUST]。この DTLS 接続上のすべてのアプリケーション層プロトコルペイロードは、SCTP パケットである。

プロトコル識別は、[RFC8833] で規定されているように、DTLS ハンドシェイクの一部として提供されなければならない[MUST]。

4　メディアの優先度付け

WebRTC 優先度付けモデルでは、アプリケーションは、API から制御されるメディアとデータの優先度について WebRTC エンドポイントに通知する。

このコンテキストでは、WebRTC API を介して特定の優先度が与えられるユニットに対して「フロー」が使用される。

メディアの場合、「オーディオフロー」または「ビデオフロー」とすることができる「メディアフロー」は、[RFC7656] が「メディアソース」と呼ぶものであり、「ソース RTP ストリーム」および 1 つ以上の「冗長 RTP ストリーム」となる。この仕様では、単一のメディアソースからの RTP ストリーム間の優先度付けについては説明しない。

WebRTC のすべてのメディアフローは、[RFC4594] で定義されているように、インタラクティブであると見なされる。メディアがインタラクティブか非インタラクティブかを示すブラウザ API はサポートされていない。

「データフロー」は、単一の WebRTC データチャネル上の送信データである。

メディアフローまたはデータフローに関連付けられた優先度は、「超低」、「低」、「中」、または「高」に分類される。API には 4 つの優先度レベルしかない。

優先度設定は、パケット送信シーケンスの決定とパケットマーキングという 2 つの動作に影響する。以下では、それぞれについて説明する。

4.1　ローカルの優先度付け

ローカル優先度付けは、パケットが送信される前にローカルノードで適用される。つまり、優先度付けは個々のパケットに関するデータに完全にアクセスでき、パケットが属するストリームに基づいて異なる処理を選択できる。

WebRTC エンドポイントが、同じ輻輳制御レジームで輻輳制御されている複数のストリームで送信するパケットを持っている場合、WebRTC エンドポイントは、各優先度レベルの各ストリームに、その下のレベルの約 2 倍の送信容量 (ペイロードバイトで測定) が与えられるような方法で、データを送信させるべきであ

る[SHOULD]。

したがって、輻輳が発生すると、両方に送信するデータがある場合、高優先度のフローは、非常に低い優先度のフローの 8 倍のデータを送信できる。この優先度付けは、メディアタイプに依存しない。最初に送信するパケットの詳細は、実装によって定義される。

たとえば、100 バイトのパケットを送信する優先度の高いオーディオフローと 1000 バイトのパケットを送信する優先度の低いビデオフローがあり、ペイロードバイトが 5000 を超える送信容量が存在する場合、送信決定の単一パスの結果として、4000 バイト (40 パケット) のオーディオと 1000 バイト (1 パケット) のビデオを送信することが適切である。

逆に、オーディオフローが低優先度とマークされ、ビデオフローが高優先度とマークされている場合、スケジューラは、ペイロードバイトが 2500 を超える送信容量が存在するときに、2 つのビデオパケット (2000 バイト) と 5 つのオーディオパケット (500 バイト) を送信することを決定できる。

優先度の高いオーディオフローが 2 つある場合、優先度の低いビデオフローが 1000 バイトを送信できるのと同じ期間に、それぞれが 4000 バイトを送信できるようになる。

2 つの実装戦略の例を次に示す。

- 使用可能な帯域幅が輻輳制御アルゴリズムからわかっている場合は、使用可能な帯域幅の共有に適したターゲット送信レートを使用して、各コーデックおよび各データチャネルを設定する。
- 輻輳制御によって、指定された数のパケットを送信できることが示された場合は、加重ラウンドロビン方式を使用して、送信可能なパケットを接続間で送信する。

これらの組み合わせ、または同じ効果を持つ他のスキームは、伝送容量の分布がほぼ正確である限り有効である。

メディアの場合、通常、送信にディープキューを使用することは適切ではない。たとえば、低いビットレートを実現するために、依存関係のない中間フレームをスキップする方が便利である。信頼性の高いデータには、キューが役立つ。

この仕様は、異なるストリームがいつ「同じ輻輳制御レジームの下で輻輳制御」されるかを規定していないことに注意されたい。輻輳制御装置の結合の問題は、[RFC8699] でさらに探求されている。

4.2 Quality of Service の使用：DSCP と多重化

パケットが送信されると、ネットワークは、通信の品質に影響を与える可能性のあるパケットのキューイングや廃棄について決定する。送信側は、パケットの DSCP フィールドを設定して、これらの決定に影響を与えることができる。

実装は、[RFC8837] のガイドラインに従って、送信されたパケットに QoS を設定するよう試みるべきである[SHOULD]。QoS マーキングが実装されていないプラットフォームで実行する場合は、この推奨事項から逸脱することが適切である。

実装は、特定の DSCP マーキングを持つパケットの優先度の反転やブロッキングなどの予期しない動作の兆候を検出した場合に、DSCP マーキングの使用をオフにする。このような動作のいくつかの例は、[ANRW16] に記載されている。これらの条件の検出は実装に依存することができる[MAY]。

特に難しい問題は、1 つのメディアトランスポートが複数の DSCP を使用する場合で、1 つはブロックされ、もう 1 つは許可される。これは、[RFC8837] のビデオの単一メディアフロー内でも許可される。実装はこのシナリオを診断する必要がある。1 つの可能な実装は、DSCP 0 で初期 ICE プローブを送信し、候補ペアが選択された後に使用されるすべての DSCP で ICE プローブを送信することである。1 つ以上の DSCP マーキングされたプローブが失敗した場合、送信側は DSCP 0 を使用するようにメディアタイプを切り替える。これは、初期メディアトラフィックと同時に実行できる。失敗した場合は、初期データを再送信する必要が

ある。このスイッチは、当然ながら、その時点までに収集された輻輳情報を無効にする。

コールの存続期間中に障害が発生することもある。このケースはまれであると予想され、ICE の再起動が含まれる可能性があるトランスポートエラーの通常のメカニズムによって処理できる。

DSCP が原因で配信不能になった場合は、メディアフロー全体を DSCP 0 に切り替える必要がある。これは、輻輳制御を行うために、1 つのメディアフローのすべてのトラフィックを同じキューに入れる必要があるためである。異なる DSCP を使用する同じトランスポート上の他のフローは、変更する必要はない。

データチャネルをサポートする SCTP アソシエーションからのデータを運ぶすべてのパケットは、単一の DSCP を使用しなければならない[MUST]。使用されるコードポイントは、[RFC8837] で推奨されている最も優先度の高いデータチャネルのコードポイントであるべきである[SHOULD]。これは、相対的な優先度に関係なく、すべてのデータパケットがネットワークによって同じように扱われることを意味する。

1 つの TCP 接続上のすべてのパケットは、それが何を運ぶかにかかわらず、単一の DSCP を使わなければならない[MUST]。

DSCP と RTP の使用、および DSCP と輻輳制御との関係についてのさらなるアドバイスは、[RFC7657] に記載されている。

DSCP に依存しないサービス品質を達成するための多くのスキームが存在している。これらの方式のいくつかは、5 タプル (送信元アドレス、送信元ポート、プロトコル、宛先アドレス、宛先ポート) または 6 タプル (5 タプル+DSCP) に基づいてトラフィックをフローに分類することに依存している。

したがって、異なる状況下では、送信側アプリケーションが次のいずれかの設定を選択することが適切な場合がある。

- 各メディアストリームは独自の 5 タプルを持っている
- メディアタイプごとに 5 タプルよってグループ化されたメディアストリーム (1 つの 5 タプルですべてのオーディオを伝送するなど)
- すべてのメディアが 1 つの 5 タプルで送信され、DSCP に基づいて 6 タプルに区別されるかどうか

上記の各構成において、データチャネルは、それ自体の 5 タプルで搬送されてもよく、またはメディアフローの 1 つと一緒に多重化されてもよい。

1 つの 5 タプル上に高優先度ビデオストリームを送信し、他の 5 タプル上に多重化された他のすべてのビデオストリームを送信するような、より複雑な構成も想定することができる。5 タプルへのメディアフローのマッピングの詳細については、[RFC8834] を参照する。

送信側の実装は、以下の設定をサポートできなければならない[MUST]。

- すべてのメディアとデータを 1 つの 5 タプルで多重化 (完全バンドル)
- 各メディアストリームを独自の 5 タプルで送信し、データを独自の 5 タプルで送信する (完全にバンドルされていない)

送信側の実装は、各メディアタイプ (オーディオ、ビデオ、またはデータ) を独自の 5 タプルにバンドルする (メディアタイプによるバンドル) など、他の構成をサポートすることを選択することができる[MAY]。

複数の 5 タプルを介したデータチャネルデータの送信はサポートされていない。

受信側の実装は、これらすべての設定でメディアとデータを受信できなければならない[MUST]。

5. IANA に関する考慮事項

この文書には IANA アクションはない。

6. セキュリティに関する考慮事項

WebRTC セキュリティの考慮事項は、[RFC8826] に列挙されている。

DSCP の使用に関するセキュリティ上の考慮事項は、[RFC8837] に列挙されている。

7. 参考資料

7.1 標準参照

[RFC0768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, DOI 10.17487/RFC0768, August 1980, <https://www.rfc-editor.org/info/rfc768>.

[RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, DOI 10.17487/RFC0793, September 1981, <https://www.rfc-editor.org/info/rfc793>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC4571] Lazzaro, J., "Framing Real-time Transport Protocol (RTP) and RTP Control Protocol (RTCP) Packets over Connection-Oriented Transport", RFC 4571, DOI 10.17487/RFC4571, July 2006, <https://www.rfc-editor.org/info/rfc4571>.

[RFC4594] Babiarz, J., Chan, K., and F. Baker, "Configuration Guidelines for DiffServ Service Classes", RFC 4594, DOI 10.17487/RFC4594, August 2006, <https://www.rfc-editor.org/info/rfc4594>.

[RFC4941] Narten, T., Draves, R., and S. Krishnan, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC 4941, DOI 10.17487/RFC4941, September 2007, <https://www.rfc-editor.org/info/rfc4941>.

[RFC5389] Rosenberg, J., Mahy, R., Matthews, P., and D. Wing, "Session Traversal Utilities for NAT (STUN)", RFC 5389, DOI 10.17487/RFC5389, October 2008, <https://www.rfc-editor.org/info/rfc5389>.

[RFC5764] McGrew, D. and E. Rescorla, "Datagram Transport Layer Security (DTLS) Extension to Establish Keys for the Secure Real-time Transport Protocol (SRTP)", RFC 5764, DOI 10.17487/RFC5764, May 2010, <https://www.rfc-editor.org/info/rfc5764>.

[RFC6062] Perreault, S., Ed. and J. Rosenberg, "Traversal Using Relays around NAT (TURN) Extensions for TCP Allocations", RFC 6062, DOI 10.17487/RFC6062, November 2010, <https://www.rfc-editor.org/info/rfc6062>.

[RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, DOI 10.17487/RFC6347, January 2012, <https://www.rfc-editor.org/info/rfc6347>.

[RFC6544] Rosenberg, J., Keranen, A., Lowekamp, B. B., and A. B. Roach, "TCP Candidates with Interactive Connectivity Establishment (ICE)", RFC 6544, DOI 10.17487/RFC6544, March 2012, <https://www.rfc-editor.org/info/rfc6544>.

[RFC6724] Thaler, D., Ed., Draves, R., Matsumoto, A., and T. Chown, "Default Address Selection for Internet Protocol Version 6 (IPv6)", RFC 6724, DOI 10.17487/RFC6724, September 2012, <https://www.rfc-editor.org/info/rfc6724>.

[RFC7231] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content", RFC 7231, DOI 10.17487/RFC7231, June 2014, <https://www.rfc-editor.org/info/rfc7231>.

[RFC7235] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Authentication", RFC 7235, DOI 10.17487/RFC7235, June 2014, <https://www.rfc-editor.org/info/rfc7235>.

[RFC7639] Hutton, A., Uberti, J., and M. Thomson, "The ALPN HTTP Header Field", RFC 7639, DOI 10.17487/RFC7639, August 2015, <https://www.rfc-editor.org/info/rfc7639>.

[RFC7656] Lennox, J., Gross, K., Nandakumar, S., Salgueiro, G., and B. Burman, Ed., "A Taxonomy of Semantics

and Mechanisms for Real-Time Transport Protocol (RTP) Sources", RFC 7656, DOI 10.17487/RFC7656, November 2015, <https://www.rfc-editor.org/info/rfc7656>.

[RFC7983] Petit-Huguenin, M. and G. Salgueiro, "Multiplexing Scheme Updates for Secure Real-time Transport Protocol (SRTP) Extension for Datagram Transport Layer Security (DTLS)", RFC 7983, DOI 10.17487/RFC7983, September 2016, <https://www.rfc-editor.org/info/rfc7983>.

[RFC8083] Perkins, C. and V. Singh, "Multimedia Congestion Control: Circuit Breakers for Unicast RTP Sessions", RFC 8083, DOI 10.17487/RFC8083, March 2017, <https://www.rfc-editor.org/info/rfc8083>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8260] Stewart, R., Tuexen, M., Loreto, S., and R. Seggelmann, "Stream Schedulers and User Message Interleaving for the Stream Control Transmission Protocol", RFC 8260, DOI 10.17487/RFC8260, November 2017, <https://www.rfc-editor.org/info/rfc8260>.

[RFC8261] Tuexen, M., Stewart, R., Jesup, R., and S. Loreto, "Datagram Transport Layer Security (DTLS) Encapsulation of SCTP Packets", RFC 8261, DOI 10.17487/RFC8261, November 2017, <https://www.rfc-editor.org/info/rfc8261>.

[RFC8421] Martinsen, P., Reddy, T., and P. Patil, "Guidelines for Multihomed and IPv4/IPv6 Dual-Stack Interactive Connectivity Establishment (ICE)", BCP 217, RFC 8421, DOI 10.17487/RFC8421, July 2018, <https://www.rfc-editor.org/info/rfc8421>.

[RFC8445] Keranen, A., Holmberg, C., and J. Rosenberg, "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal", RFC 8445, DOI 10.17487/RFC8445, July 2018, <https://www.rfc-editor.org/info/rfc8445>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

[RFC8656] Reddy, T., Ed., Johnston, A., Ed., Matthews, P., and J. Rosenberg, "Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN)", RFC 8656, DOI 10.17487/RFC8656, February 2020, <https://www.rfc-editor.org/info/rfc8656>.

[RFC8825] Alvestrand, H., "Overview: Real-Time Protocols for Browser-Based Applications", RFC 8825, DOI 10.17487/RFC8825, January 2021, <https://www.rfc-editor.org/info/rfc8825>.

[RFC8826] Rescorla, E., "Security Considerations for WebRTC", RFC 8826, DOI 10.17487/RFC8826, January 2021, <https://www.rfc-editor.org/info/rfc8826>.

[RFC8827] Rescorla, E., "WebRTC Security Architecture", RFC 8827, DOI 10.17487/RFC8827, January 2021, <https://www.rfc-editor.org/info/rfc8827>.

[RFC8831] Jesup, R., Loreto, S., and M. Tüxen, "WebRTC Data Channels", RFC 8831, DOI 10.17487/RFC8831, January 2021, <https://www.rfc-editor.org/info/rfc8831>.

[RFC8832] Jesup, R., Loreto, S., and M. Tüxen, "WebRTC Data Channel Establishment Protocol", RFC 8832, DOI 10.17487/RFC8832, January 2021, <https://www.rfc-editor.org/info/rfc8832>.

[RFC8833] Thomson, M., "Application-Layer Protocol Negotiation (ALPN) for WebRTC", RFC 8833, DOI 10.17487/RFC8833, January 2021, <https://www.rfc-editor.org/info/rfc8833>.

[RFC8834] Perkins, C., Westerlund, M., and J. Ott, "Media Transport and Use of RTP in WebRTC", RFC 8834, DOI 10.17487/RFC8834, January 2021, <https://www.rfc-editor.org/info/rfc8834>.

[RFC8836] Jesup, R. and Z. Sarker, Ed., "Congestion Control Requirements for Interactive Real-Time Media", RFC 8836, DOI 10.17487/RFC8836, January 2021, <https://www.rfc-editor.org/info/rfc8836>.

[RFC8837] Jones, P., Dhesikan, S., Jennings, C., and D. Druta, "Differentiated Services Code Point (DSCP) Packet

Markings for WebRTC QoS", RFC 8837, DOI 10.17487/RFC8837, January 2021, <https://www.rfc-editor.org/info/rfc8837>.

[RFC8841] Holmberg, C., Shpount, R., Loreto, S., and G. Camarillo, "Session Description Protocol (SDP) Offer/Answer Procedures for Stream Control Transmission Protocol (SCTP) over Datagram Transport Layer Security (DTLS) Transport", RFC 8841, DOI 10.17487/RFC8841, January 2021, <https://www.rfc-editor.org/info/rfc8841>.

[RFC8842] Holmberg, C. and R. Shpount, "Session Description Protocol (SDP) Offer/Answer Considerations for Datagram Transport Layer Security (DTLS) and Transport Layer Security (TLS)", RFC 8842, DOI 10.17487/RFC8842, January 2021, <https://www.rfc-editor.org/info/rfc8842>.

7.2 参考文献

[ANRW16] Barik, R., Welzl, M., and A. Elmokashfi, "How to say that you're special: Can we use bits in the IPv4 header?", ANRW '16: Proceedings of the 2016 Applied Networking Research Workshop, pages 68-70, DOI 10.1145/2959424.2959442, July 2016, <https://irtf.org/anrw/2016/anrw16-final17.pdf>.

[RETURN] Schwartz, B. and J. Uberti, "Recursively Encapsulated TURN (RETURN) for Connectivity and Privacy in WebRTC", Work in Progress, Internet-Draft, draft-ietf-rtcweb-return-02, 27 March 2017, <https://tools.ietf.org/html/draft-ietf-rtcweb-return-02>.

[RFC3484] Draves, R., "Default Address Selection for Internet Protocol version 6 (IPv6)", RFC 3484, DOI 10.17487/RFC3484, February 2003, <https://www.rfc-editor.org/info/rfc3484>.

[RFC5014] Nordmark, E., Chakrabarti, S., and J. Laganier, "IPv6 Socket API for Source Address Selection", RFC 5014, DOI 10.17487/RFC5014, September 2007, <https://www.rfc-editor.org/info/rfc5014>.

[RFC5128] Srisuresh, P., Ford, B., and D. Kegel, "State of Peer-to-Peer (P2P) Communication across Network Address Translators (NATs)", RFC 5128, DOI 10.17487/RFC5128, March 2008, <https://www.rfc-editor.org/info/rfc5128>.

[RFC7657] Black, D., Ed. and P. Jones, "Differentiated Services (Diffserv) and Real-Time Communication", RFC 7657, DOI 10.17487/RFC7657, November 2015, <https://www.rfc-editor.org/info/rfc7657>.

[RFC8155] Patil, P., Reddy, T., and D. Wing, "Traversal Using Relays around NAT (TURN) Server Auto Discovery", RFC 8155, DOI 10.17487/RFC8155, April 2017, <https://www.rfc-editor.org/info/rfc8155>.

[RFC8699] Islam, S., Welzl, M., and S. Gjessing, "Coupled Congestion Control for RTP Media", RFC 8699, DOI 10.17487/RFC8699, January 2020, <https://www.rfc-editor.org/info/rfc8699>.

IV. RFC8828 原文の著作権について

V.　RFC8828 の和訳

RFC 8828　WebRTC の IP アドレス処理要件

概要

この文書では、Web Real-Time Communication (WebRTC) 実装による IP アドレスの処理方法に関する情報と要件について説明する。

1.　はじめに

WebRTC の主要な機能の 1 つは、ピアツーピア接続のサポートである。ただし、さまざまな IP アドレスからの接続試行を含むこのような接続を確立するときに、WebRTC は、ハイパーテキスト転送プロトコル (HTTP) [RFC7230] のみを使用するアプリケーションと比較して、Web アプリケーションがユーザに関する追加情報を学習できるようにする場合があり、これが問題となる場合もある。この文書では、問題を要約し、WebRTC 実装がプライバシーとメディアパフォーマンスのトレードオフをどのように最適に処理するかについて推奨する。

2.　用語

この文書のキーワードである「MUST」、「MUST NOT」、「REQUIRED」、「SHALL」、「SHALL NOT」、「SHOULD」、「SHOULD NOT」、「RECOMMENDED」、「NOT RECOMMENDED」、「MAY」、および「OPTIONAL」は、ここに示すように、すべて大文字で表示される場合にのみ、BCP 14 [RFC2119] [RFC8174] で記述されているように解釈される。

3.　問題ステートメント

ピアツーピア接続を確立するために、WebRTC 実装では Interactive Connectivity Establishment (ICE) [RFC8445] を使用する。ICE は、Session Traversal Utilities for NAT (STUN) [RFC5389] や Traversal Using Relays around NAT (TURN) [RFC5766] などの技術を使用して複数の IP アドレスの検出を試み、次に最適なアドレスを選択するために、各ローカルアドレスとリモートアドレスのペアの接続を確認する。通常、収集されるアドレスは、エンドポイントのプライベート物理アドレスまたは仮想アドレスと、そのパブリックインターネットアドレスで構成される。

これらのアドレスは、チェックのためにリモートエンドポイントに通信できるように、Web アプリケーションに提供される。これにより、アプリケーションは、Web サーバが 1 つのパブリックインターネットアドレス、つまり HTTP 要求の送信元アドレスのみを参照する一般的な HTTP シナリオよりも、ローカルネットワーク構成についてより多くの情報を取得できる。

表示される追加情報は、次の 3 つのカテゴリに分類される。

1. クライアントがマルチホームの場合、クライアントの追加のパブリック IP アドレスを学習できる。特に、クライアントが仮想プライベートネットワーク (VPN) を介して物理的な場所を隠そうとし、VPN およびローカル OS が複数のインタフェースでのルーティング (「スプリットトンネル」VPN) をサポートしている場合、WebRTC は VPN のパブリックアドレスだけでなく、VPN が実行されている ISP のパブリックアドレスも検出できる。

2. クライアントがネットワークアドレス変換 (NAT) の背後にある場合、クライアントのプライベート IP アドレス、多くの場合 [RFC1918] アドレスを学習できる。
3. クライアントがプロキシの背後にあるが ([RFC1919] 3 章で定義されているように、クライアントで構成された従来のアプリケーションプロキシ)、インターネットへの直接アクセスが許可されている場合、WebRTC の STUN チェックはプロキシをバイパスし、クライアントのパブリック IP アドレスを明らかにする。この懸念は、上記のように直接インターネットアクセスが許可されている場合、[RFC7478] 2.3.5.1 項で説明されているエンタープライズ TURN サーバシナリオにも適用される。しかし、この文書上で「プロキシ」という用語を使うときは、常に [RFC1919] プロキシサーバを指す。

これら 3 つの懸念事項のうち、最初のものが最も重要となる。一部のユーザにとって、VPN を使用する目的は匿名性のためである。ただし、VPN ユーザごとにニーズは異なり、一部の VPN ユーザ (企業 VPN ユーザなど) は、メディアトラフィックを VPN 経由ではなく直接送信するために、実際には WebRTC を好む場合がある。

第二の懸念は重要ではないが、それでも有効である。中心的な問題は、Web アプリケーションがインターネットに公開されていないアドレスを学習できることである。通常、これらのアドレスは IPv4 だが、NAT64 [RFC6146] の場合のように IPv6 にすることもできる。[RFC8835] によって推奨されている [RFC4941] IPv6 アドレスの開示は、意図的に短い寿命のためにかなり無害であるが、IPv4 アドレスはいくつかの課題を提示する。プライベート IPv4 アドレスは多くの場合最小エントロピー (例えば、かなり一般的なアドレスである 192.168.0.2) を含んでいるが、最悪の場合、無期限の 24 ビットのエントロピーを含むことができる。このように、それらはかなり重要な指紋面 (fingerprinting surface) となり得る。さらに、イントラネットの Web サイトは、その IPv4 アドレス範囲が外部に知られている場合、より簡単に攻撃される可能性がある。

プライベート IP アドレスは、分離された閲覧コンテキスト (例えば、通常の閲覧とプライベートな閲覧) で実行されている Web アプリケーションが同じデバイスで実行されていることを学習できるようにする識別子としても機能する。これにより、アプリケーションセッションを関連付けることができ、分離によって提供されるプライバシー保護の一部が無効になる可能性がある。プライベートアドレスは、この相関関係の潜在的なメカニズムの 1 つにすぎず、これは今後の研究領域であることに注意する。

3 番目の問題は最も一般的ではない。プロキシ管理者は既に組織のファイアウォールポリシーを介してこの動作を制御できる。また、一般に、プロキシサーバを介して WebRTC トラフィックを強制すると、プロキシとメディアの両方の品質に悪影響を及ぼす。

これらの懸念は WebRTC 以前からあることにも注意する。Adobe Flash Player は、2008 年にリアルタイムメディアフロープロトコル (RTMFP) サポート [RFC7016] を導入して以来、同様の機能を提供してきた。

4. 目標

WebRTC のセキュアなピアツーピア接続のサポートにより、分散システムの展開が容易になり、プライバシー上のメリットが得られる。その結果、WebRTC を無効にしたり、使用を大幅に困難にしたりする単純なソリューションは望ましくない。この文書では、次の目標を持つ、より微妙なアプローチを採用する。

- 問題を理解するためのフレームワークを提供して、WebRTC のパフォーマンスとプライバシーの問題に関してさまざまなトレードオフを行うためのコントロールを提供する。
- このフレームワークを使用して、パフォーマンスとプライバシーのバランスが異なるピアツーピア通信を有効にする設定を定義する。
- 最後に、ユーザの期待に反する方法でアドレス指定情報を公開することなく、適切なパフォーマンスを提供する既定の設定に関する推奨事項を提供する。

5. 詳細設計

5.1 原則

我々の枠組みの主要な原則は以下の通りである。

1. 既定では、WebRTC トラフィックは一般的な IP ルーティング (つまり、WebRTC は HTTP トラフィックと同じインタフェースを使用する必要がある)に従う必要があり、システムの(またはエンタープライズ TURN サーバ (存在する場合)) パブリックアドレスのみがアプリケーションに表示される必要がある。ただし、最適なメディア品質のために、WebRTC がすべてのネットワークインタフェースを使用して理想的なルートを決定できるようにすることが可能である。
2. デフォルトでは、WebRTC は、このような接続が可能な場合に、エンドポイント (つまり、NAT またはリレーサーバを通過せずに) 間の直接ピアツーピア接続をネゴシエートできる必要がある。これにより、帯域幅や遅延の理由から真のピアツーピアルーティングを必要とするアプリケーションが正常に動作できるようになる。
3. このようなアドレスを学習する Web アプリケーションに関連する問題を回避するために、プライベートローカル IP アドレスを公開しないように WebRTC を設定できる必要がある。この文書では、これをデフォルト状態にする必要はない。これは、この要件を満たすメカニズムが現在定義されていないためである。また、前述のピアツーピアの直接接続を許可する要件もない。
4. デフォルトでは、WebRTC トラフィックはプロキシサーバを経由して送信されない。これは、WebRTC トラフィックを TCP 経由で送信することに関連するメディア品質の問題が原因で、このようなプロキシとの通信時にはほとんど常にこの問題が使用される。また、WebRTC の長時間の高帯域接続をプロキシすることによるプロキシパフォーマンスの問題もある。しかし、必要に応じて、WebRTC が構成されたプロキシを介してトラフィックを送信するように強制できる。

5.2 モードと推奨事項

これらのアイデアに基づいて、異なるメディア品質/プライバシートレードオフを反映した WebRTC 挙動の 4 つの特定モードを定義した。

モード 1 - 全アドレス列挙：

WebRTC は、STUN サーバ、TURN サーバ、またはピアツーピア接続の通信を試みるために、すべてのネットワークインタフェースを使用しなければならない[MUST]。これは、最適なメディアパスに収斂し、メディアパフォーマンスが最優先であるが、最も多くの情報を開示する場合に理想的である。

モード 2 - デフォルトルート+関連付けられたローカルアドレス：

WebRTC は、通常、メディアパケットがアプリケーションの HTTP トラフィックと同じルートを取るようにするカーネルルーティングテーブルルールに従わなければならない[MUST]。エンタープライズ TURN サーバが存在する場合、優先ルートはこの TURN サーバを経由しなければならない[MUST]。インタフェースが選択されると、このインタフェースに関連するプライベート IPv4 および IPv6 アドレスが発見され、ホスト候補としてアプリケーションに提供されなければならない[MUST]。これにより、このモードでも直接接続を確立できる。

モード 3 - デフォルトルートのみ：

これはモード 2 と同じであるが、関連するプライベートアドレスが提供されてはならない点が異なる[MUST NOT]。収集された IP アドレスは、STUN や TURN (デフォルトルート上) など

のメカニズムを介して検出されたものだけである。これにより、トラフィックが NAT を介してヘアピンされたり、アプリケーション TURN サーバにフォールバックされたり、完全に失敗したりすることがあり、結果として品質に影響を及ぼす可能性がある。

モード 4 - 強制プロキシ：

これはモード 3 と同じであるが、アプリケーションの HTTP トラフィックがプロキシを介して送信される場合、WebRTC メディアトラフィックもプロキシされなければならない[MUST]。このプロキシが UDP をサポートしていない場合 (すべての HTTP およびほとんどの SOCKS プロキシ [RFC1928] の場合と同様に)、または WebRTC 実装が UDP プロキシをサポートしていない場合は、UDP の使用が無効になり、プロキシ経由でメディアを送受信するために TCP が使用される。TCP を使用すると、プロキシサーバ経由ですべての WebRTC メディアを送信することに関連するパフォーマンスの考慮事項に加えて、メディアの品質が低下する。

モード 1 は、ユーザの同意がない限り使用してはならない[MUST NOT]。この同意の詳細は実装に委ねられる。考えられるメカニズムの一つは、[RFC8827] 6.2 節で説明されているように、この同意を getUserMedia (デバイス許可) 同意に結びつけることである。または、実装は、ユーザの同意を得るための特定のメカニズムを提供できる。

ユーザの同意が得られなかった場合、モード 2 を使用すべきである[SHOULD]。

これらのデフォルトは、信頼された WebRTC アプリケーションが最適なネットワークパフォーマンスを実現できるようにする一方で、同意のないアプリケーション (例：1 方向ストリーミングまたはデータチャネルアプリケーション) には、モード 2 で定義されているように、直接接続を実現するために必要な最小限の情報のみを与えるという、合理的なトレードオフを提供する。しかしながら、実装は、例えば、ユーザが全ての WebRTC トラフィックがデフォルトルートに従うことを望むならば、より厳密なモードを選択してもよい[MAY]。

将来の文書では、追加のモードを定義したり、推奨されるデフォルトモードを更新したりする場合がある。

すべての外部 WebRTC トラフィックがプロキシまたはエンタープライズ TURN サーバを通過する必要がある組織でも、WebRTC トラフィックがプロキシまたは TURN サーバを通過することだけを許可する組織ファイアウォールポリシーを設定するだけで、推奨されるデフォルトを使用できる。これにより、プロキシまたは TURN サーバが外部トラフィックに使用されるようになるが、組織内トラフィックには直接接続(また、プロキシの場合には、前記プロキシを介してメディアを強制することに関連するパフォーマンスの問題を回避する) が許可される。

6. 実装ガイダンス

この章では、上記のポリシーを実装する方法に関する WebRTC 実装のガイダンスを提供する。

6.1 通常ルーティングの確認

モード 2 および 3 で要求される一般的な IP ルーティングに従う場合、最も簡単な方法は、ピアツーピア接続に使用されるソケットをワイルドカードアドレス (IPv4 の場合は 0.0.0.0、IPv6 の場合は ::) にバインド (bind()) することである。これにより、OS は HTTP トラフィックと同じ方法で WebRTC トラフィックをルーティングできるようになる。STUN および TURN は通常通り動作し、以下のようにホスト候補を決定することができる。

6.2　関連付けられたローカルアドレスの確認

ワイルドカードアドレスにバインドする場合は、モード 2 で必要な関連するローカルアドレスを決定するために追加の作業が必要である。これは、Web アプリケーションホストに送信されるすべてのパケットに使用される送信元アドレスとして定義される (UDP と TCP が同じルーティング処理を受けることを前提とする)。Web アプリケーションホストを宛先として使用すると、アプリケーションの場所 (例えば、イントラネット上で) に関係なく、正しい送信元アドレスが選択される。

まず、Web アプリケーション URI [RFC3986] のホストコンポーネントを解決することによって、適切なリモート IPv4/IPv6 アドレスを取得する。クライアントがプロキシの背後にあり、DNS 経由でこれらの IP を解決できない場合は、代わりにプロキシのアドレスを使用できる。または、Web アプリケーションがネットワーク経由ではなく、file:// URI [RFC8089] から読み込まれた場合、実装は既知の DNS 名または IP アドレスにフォールバックすることができる。

適切なリモート IP が決定されると、実装は UDP ソケットを作成し、適切なワイルドカードアドレスにバインド (bind()) してからリモート IP に接続 (connect()) できる。一般的に、これにより、ソケットはネットワーク上にパケットを送信することなく、カーネルルーティングテーブルに基づいてローカルアドレスが割り当てられる。

最後に、getsockname() またはそれと同等のメソッドを使用して適切なローカルアドレスを取得し、ソケットに対して問い合わせを行う。

7.　アプリケーションガイダンス

この文書に記載されている推奨事項により、特定の WebRTC アプリケーションが誤動作する可能性がある。すべてのシナリオで堅牢であるために、アプリケーションには次のガイドラインが用意されている。

- アプリケーションは、サーバへの UDP および TCP 接続の両方をサポートする TURN サーバを配備すべきである[SHOULD]。これにより、モード 3 または 4 が使用されている場合でも、TURN サーバに到達できることを前提として、接続を確立できる。
- アプリケーションは、ホスト候補の存在をチェックすることによって、ICE 候補のフルセットにアクセスできない場合を検出すべきである[SHOULD]。ホスト候補が存在しない場合、モード 3 または 4 は使用中である。この知識は診断目的として有用である。

8.　セキュリティに関する考慮事項

この文書では、WebRTC ピアツーピア接続に関連するいくつかの潜在的なプライバシーとセキュリティの問題について説明し、これらの問題に対処するための WebRTC 実装のメカニズムと推奨事項を提供する。

9.　IANA に関する考慮事項

この文書には IANA アクションはない。

10. 参考資料

10.1　標準参照

[RFC2119]　Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005, <https://www.rfc-editor.org/info/rfc3986>.

[RFC5389] Rosenberg, J., Mahy, R., Matthews, P., and D. Wing, "Session Traversal Utilities for NAT (STUN)", RFC 5389, DOI 10.17487/RFC5389, October 2008, <https://www.rfc-editor.org/info/rfc5389>.

[RFC5766] Mahy, R., Matthews, P., and J. Rosenberg, "Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN)", RFC 5766, DOI 10.17487/RFC5766, April 2010, <https://www.rfc-editor.org/info/rfc5766>.

[RFC8089] Kerwin, M., "The "file" URI Scheme", RFC 8089, DOI 10.17487/RFC8089, February 2017, <https://www.rfc-editor.org/info/rfc8089>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8445] Keranen, A., Holmberg, C., and J. Rosenberg, "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal", RFC 8445, DOI 10.17487/RFC8445, July 2018, <https://www.rfc-editor.org/info/rfc8445>.

10.2 参考文献

[RFC1918] Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G. J., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, DOI 10.17487/RFC1918, February 1996, <https://www.rfc-editor.org/info/rfc1918>.

[RFC1919] Chatel, M., "Classical versus Transparent IP Proxies", RFC 1919, DOI 10.17487/RFC1919, March 1996, <https://www.rfc-editor.org/info/rfc1919>.

[RFC1928] Leech, M., Ganis, M., Lee, Y., Kuris, R., Koblas, D., and L. Jones, "SOCKS Protocol Version 5", RFC 1928, DOI 10.17487/RFC1928, March 1996, <https://www.rfc-editor.org/info/rfc1928>.

[RFC4941] Narten, T., Draves, R., and S. Krishnan, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC 4941, DOI 10.17487/RFC4941, September 2007, <https://www.rfc-editor.org/info/rfc4941>.

[RFC6146] Bagnulo, M., Matthews, P., and I. van Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", RFC 6146, DOI 10.17487/RFC6146, April 2011, <https://www.rfc-editor.org/info/rfc6146>.

[RFC7016] Thornburgh, M., "Adobe's Secure Real-Time Media Flow Protocol", RFC 7016, DOI 10.17487/RFC7016, November 2013, <https://www.rfc-editor.org/info/rfc7016>.

[RFC7230] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", RFC 7230, DOI 10.17487/RFC7230, June 2014, <https://www.rfc-editor.org/info/rfc7230>.

[RFC7478] Holmberg, C., Hakansson, S., and G. Eriksson, "Web Real-Time Communication Use Cases and Requirements", RFC 7478, DOI 10.17487/RFC7478, March 2015, <https://www.rfc-editor.org/info/rfc7478>.

[RFC8827] Rescorla, E., "WebRTC Security Architecture", RFC 8827, DOI 10.17487/RFC8827, January 2021, <https://www.rfc-editor.org/info/rfc8827>.

[RFC8835] Alvestrand, H., "Transports for WebRTC", RFC 8835, DOI 10.17487/RFC8835, January 2021, <https://www.rfc-editor.org/info/rfc8835>.

VI. RFC8831 原文の著作権について

VII. RFC8831 の和訳

RFC 8831　WebRTC データチャネル

概要

WebRTC フレームワークは、2 つのピアの Web ブラウザ間で音声、ビデオ、およびデータを使用した直接対話型のリッチ通信のプロトコルサポートを指定する。この文書では、WebRTC フレームワークの非メディアデータトランスポートの側面について説明する。ここでは、Web ブラウザがピア間で汎用データを交換できるようにする汎用トランスポートサービスとして、WebRTC コンテキストで Stream Control Transmission Protocol (SCTP) を使用する方法のアーキテクチャの概要について説明する。

1.　はじめに

WebRTC フレームワークでは、当事者間の通信はメディア (オーディオやビデオなど) と非メディアデータで構成される。メディアは Secure Real-time Transport Protocol (SRTP) を使用して送信され、ここでは指定しない。メディア以外のデータは、DTLS にカプセル化された Stream Control Transmission Protocol (SCTP) [RFC4960] を使用して処理される。DTLS 1.0 は [RFC4347] で定義されている。現在の最新バージョンである DTLS 1.2 は、[RFC6347] で定義されている。また、[TLS-DTLS13] では、次期バージョンである DTLS 1.3 が定義されている。

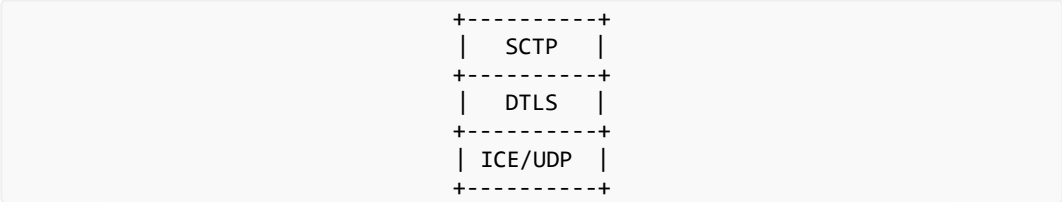


図 1：基本的なスタック図

ICE/UDP ([RFC8445] を参照) 上の SCTP over DTLS ([RFC8261] を参照) のカプセル化は、機密性、送信元認証、および整合性保護転送とともに NAT トラバーサルソリューションを提供する。このデータ転送サービスは SRTP メディア転送と並行して動作し、最終的にはすべてが 1 つの UDP ポート番号を共有できる。

SCTP は、[RFC3758] で定義されている部分信頼性拡張 (PR-SCTP) と、[RFC7496] で定義されている追加ポリシーを持つ [RFC4960] で指定されているように、信頼性のある複数のストリームをネイティブに提供し、関連する部分信頼性のある、ユーザメッセージの配信モードを提供する。[RFC6525] で定義された再構成拡張を使用すると、SCTP アソシエーションのライフタイム中にストリームの数を増やすことができ、個々の SCTP ストリームをリセットすることができる。[RFC8260] を使用すると、大きなメッセージをインタリーブして独占を回避でき、SCTP ストリームの優先度付けのサポートが追加される。

この文書の残りの部分は、次のように構成されている。3 章と 4 章では、信頼性の低いピアツーピアデータチャネルと信頼性の高いピアツーピアデータチャネルの両方のユースケースと要件について説明する。5 章では SCTP over DTLS over UDP について説明している。6 章は、Web ブラウザ間でメディア以外のデータを転送するために WebRTC プロトコルフレームワークが SCTP を使用する方法を規定している。

2. 表記規則

この文書のキーワードである「MUST」、「MUST NOT」、「REQUIRED」、「SHALL」、「SHALL NOT」、「SHOULD」、「SHOULD NOT」、「RECOMMENDED」、「NOT RECOMMENDED」、「MAY」、および「OPTIONAL」は、ここに示すように、すべて大文字で表示される場合にのみ、BCP 14 [RFC2119] [RFC8174] で記述されているように解釈される。

3. ユースケース

この章では、データチャネル固有のユースケースを定義する。この章は情報提供のみである。

3.1 信頼性の低いデータチャネルの使用例

U-C 1： 位置およびオブジェクトの状態情報が 1 つまたは複数の信頼できないデータチャネルを介して送信されるリアルタイムゲーム。SRTP メディアチャネルが存在しない場合や、すべての SRTP メディアチャネルが非アクティブになっている場合がある。また、信頼できるデータチャネルが使用されている場合もある。

U-C 2： ビデオチャットまたは会議の状態更新の理由 (ミュート状態など) に関する重要でない情報をユーザに提供する。

3.2 信頼性の高いデータチャネルのユースケース

U-C 3： 制御情報などの重要な状態情報を転送する必要があるリアルタイムゲーム。このようなゲームには SRTP メディアチャネルがない場合もあれば、特定の時点で非アクティブになっている場合や、ゲーム内のアクションによってのみ追加される場合もある。

U-C 4： チャット中のユーザ間での非リアルタイムのファイル転送。これには、イメージのフォルダーやファイルのディレクトリを共有する場合など、シーケンシャルまたは並列に転送する多数のファイルが含まれる場合があることに注意すること。

U-C 5： 会議の個人または複数のユーザとの音声通話やビデオ通話中のリアルタイムテキストチャット。

U-C 6： PeerConnection の構成の再ネゴシエーション。

U-C 7： プロキシブラウジング。ブラウザが PeerConnection のデータチャネルを使用して、HTTP/HTTPS 要求とデータを送受信する。たとえば、ローカルインターネットフィルタリングまたは監視を回避する。

4. 要件

ここでは、2 つのブラウザ間のピアツーピア (P2P) データチャネルの要件を示す。この章は情報提供のみである。

要件 1： 複数の同時データチャネルがサポートされている必要がある。同じ PeerConnection 内のデータチャネルと並行して 0 個以上の SRTP メディアストリームが存在する場合があり、これらの SRTP メディアストリームの数と状態 (アクティブ/非アクティブ) はいつでも変更される可能性があることに注意すること。

要件 2： 信頼性のあるデータチャネルと信頼性のないデータチャネルの両方がサポートされている必要がある。

要件 3： PeerConnection のデータチャネルは、個別にクラスとして、または PeerConnection の SRTP メディアストリームと組み合わせて、輻輳を制御する必要がある。このため、データチャネルによってこれらの SRTP メディアストリームの輻輳問題が発生しなくなり、WebRTC PeerConnection が TCP 接続と並行して実行されている場合に過剰な問題が発生しなくなる。

要件 4： アプリケーションは、各データチャネルの相互および SRTP メディアストリームに対する相対的な優先度に関するガイダンスを提供できる必要がある。これは、輻輳制御アルゴリズムと相互作用する。

要件 5： データチャネルは、機密性、整合性、およびソース認証を可能にするセキュリティで保護する必要がある。詳細については、[RFC8826] および [RFC8827] を参照すること。

要件 6： JavaScript アプリケーションが送信するメッセージのサイズにかかわらず、IP 層の断片化を回避できるように、データチャネルはメッセージの断片化をサポートする必要がある。また、大規模なデータチャネル転送によって、他のデータチャネルのトラフィックが過度に遅延しないようにする必要がある。

要件 7： データチャネルトランスポートプロトコルは、そのプロトコルフィールド内にローカル IP アドレスをエンコードしてはならない。そうすることで、潜在的にプライベートな情報が明らかになり、アドレスに依存している場合は失敗につながる。

要件 8： データチャネルトランスポートプロトコルは、イメージファイル転送のようなもののためにアプリケーション層で無限長の「メッセージ」(すなわち、仮想ソケットストリーム) をサポートすべきである。実装によっては、妥当なメッセージサイズ制限が強制される場合がある。

要件 9： データチャネル転送プロトコルは、IP フラグメンテーションを回避する必要がある。Path MTU ディスカバリをサポートしている必要があり、特に Path MTU ディスカバリで ICMP または ICMPv6 が生成されたり、返されたりすることに依存してはならない。

要件 10： ユーザアプリケーション空間にプロトコルスタックを実装できる必要がある。

5. SCTP over DTLS over UDP に関する考慮事項

WebRTC コンテキストにおける SCTP の重要な機能は、次のとおりである。

- TCP フレンドリーな輻輳制御の使用
- SRTP メディアストリーム輻輳制御と統合するための変更可能な輻輳制御
- 順序付きメッセージ配信の独自の概念を提供する複数の単方向ストリームのサポート
- 順序付きおよび順序外メッセージ配信のサポート
- フラグメンテーションと再構成の提供により、任意サイズのユーザメッセージをサポート
- Path MTU ディスカバリのサポート
- 信頼性のあるメッセージトランスポートまたは部分的に信頼性のあるメッセージトランスポートのサポート

WebRTC データチャネルメカニズムは、SCTP マルチホーミングをサポートしない。SCTP レイヤは、DTLS レイヤ (コネクション型で信頼性の低いデータグラムサービス)が公開する抽象化であるため、シングルホームホストで実行されているかのように動作する。

[RFC8261] で定義されている SCTP over DTLS のカプセル化は、機密性、送信元認証、および整合性保護転送を提供する。Interactive Connectivity Establishment (ICE) [RFC8445] と組み合わせて DTLS over UDP を使用すると、IPv4 および IPv6 ベースのネットワークでミドルボックストラバーサルが可能になる。[RFC4960] で指定されている SCTP は、[RFC3758] で定義されている拡張と組み合わせて使用されなければならず

[MUST]、ブラウザ間で非メディアデータを転送するための以下の機能を提供する。

- 複数の単方向ストリームのサポート
- ユーザメッセージの順序付き配信と順序なし配信
- ユーザメッセージの信頼性の高い転送および部分的に信頼性の高い転送

各 SCTP ユーザメッセージには、送信側の上位層によって SCTP に渡され、受信側の上位層に提供されるペイロードプロトコル識別子 (PPID) が含まれる。PPID は、単一の SCTP アソシエーション上で複数の上位層を多重化/逆多重化するために使用できる。WebRTC コンテキストでは、PPID は、UTF-8 でエンコードされたユーザデータ、バイナリでエンコードされたユーザデータ、および [RFC8832] で定義されている Data Channel Establishment Protocol (DCEP) を区別するために使用される。PPID は JavaScript API 経由ではアクセスできないことに注意すること。

SCTP over DTLS のカプセル化は、上記の SCTP 機能とともに、4 章に記載されているすべての要件を満たす。

WebRTC のプロトコルの階層化を図 2 に示す。

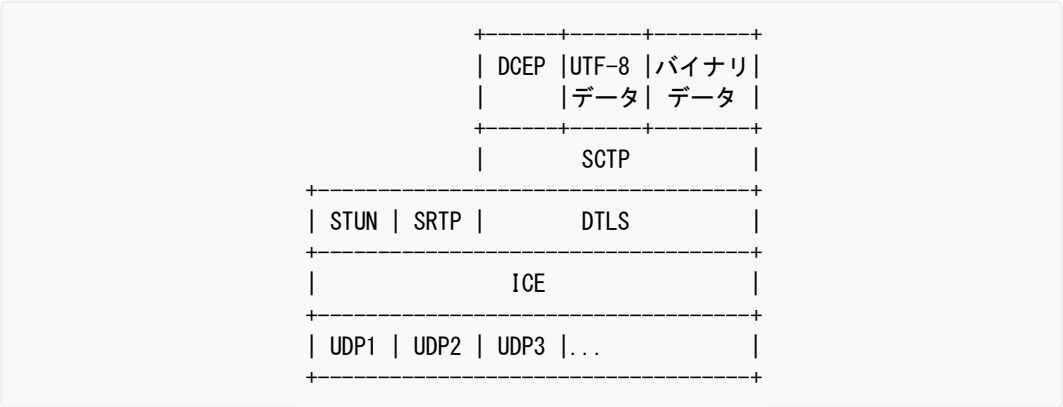


図 2：WebRTC プロトコルレイヤ

このスタック (特に DTLS over SCTP [RFC6083] および SCTP over UDP [RFC6951] とは対照的) は、次の理由で選択されている。

- 任意の大きさのユーザメッセージの送信をサポートする。
- PeerConnection の SRTP メディアチャネルと DTLS 接続を共有する。
- SCTP 制御情報のプライバシーを提供する。

図 2 に示されているプロトコルスタックを参照すること。

- UDP 上での DTLS 1.0 の使用は、[RFC4347] で規定されている。
- UDP 上での DTLS 1.2 の使用は、[RFC6347] で規定されている。
- UDP 上での DTLS 1.3 の使用は、今後の文書 [TLS-DTLS13] で指定される。
- DTLS 上での SCTP の使用は、[RFC8261] で規定されている。

NAT (STUN) [RFC5389] vs SRTP vs DTLS の逆多重化は、[RFC5764] の 5.1.2 項で説明されているように行われ、SCTP は DTLS の唯一のペイロードであることに注意すること。

DTLS は通常、ユーザアプリケーション空間に実装されるため、SCTP スタックもユーザアプリケーション空間スタックである必要がある。

ICE/UDP レイヤは、DTLS および SCTP レイヤとの対話を必要とせずに、セッション中の IP アドレスの変更を処理できる。しかし、アドレスが変更された時、SCTP は通知されるべきである[SHOULD]。この場合、

SCTP は Path MTU を再テストし、輻輳状態を初期状態にリセットする必要がある[SHOULD]。

[RFC4960] で規定されているようなウィンドウベースの輻輳制御の場合、これは輻輳ウィンドウとスロースタートしきい値を初期値に設定することを意味する。

着信 ICMP または ICMPv6 メッセージは、対応するアソシエーションを識別する方法がないため、SCTP レイヤで処理できない。したがって、SCTP は、[RFC4820] で指定されたプロービングメッセージを使用することによって、[RFC4821] で指定された ICMP または ICMPv6 に依存しない Path MTU ディスカバリの実行をサポートしなければならない[MUST]。 IP 層での初期 Path MTU は、IPv4 では 1200 バイト、IPv6 では 1280 バイトを超えてはいけない[SHOULD NOT]。

一般に、SCTP 実装の下位層インタフェースは、IPv4 と IPv6 (コネクションレス型) または DTLS (コネクション指向型) の違いに対処するように適合させる必要がある。

図 2 に示すプロトコルスタックを使用すると、DTLS は SCTP パケット全体を保護するため、SCTP パケット全体の機密性、整合性、および送信元認証を提供する。

SCTP は、アソシエーション単位で輻輳制御を行う。これは、1 つの SCTP アソシエーション内のすべての SCTP ストリームが同じ輻輳ウィンドウを共有することを意味する。SCTP 経由で送信されないトラフィックは、SCTP 輻輳制御の対象にならない。標準とは異なる輻輳制御を使用すると、並列 SRTP メディアストリームへの影響が改善される。

SCTP では、TCP および UDP と同じポート番号の概念が使用される。したがって、SCTP アソシエーションでは、各 SCTP エンドポイントに 1 つずつ、2 つのポート番号が使用される。

6. データチャネルでの SCTP の使用

6.1 SCTP プロトコルに関する考慮事項

[RFC8261] に記述されている SCTP パケットの DTLS カプセル化を使用しなければならない[MUST]。

この SCTP スタックとその上位層は、複数の SCTP ストリームの使用をサポートしなければならない[MUST]。ユーザメッセージは、順序付きまたは順序なしで部分的または完全な信頼性で送信できる。

次の SCTP プロトコル拡張が必要である。

- [RFC6525] で定義されたストリーム再構成拡張は、サポートされなければならない[MUST]。チャネルを閉じるために使用される。
- [RFC5061] で定義された動的アドレス再構成拡張は、[RFC6525] で定義されたストリームリセット拡張のサポートを通知するために使用されなければならない[MUST]。[RFC5061] の他の機能はオプションである[OPTIONAL]。
- [RFC3758] で定義された部分信頼性拡張は、サポートされなければならない[MUST]。[RFC3758] で定義されている時間指定信頼性 PR-SCTP ポリシーに加えて、[RFC7496] で定義されている限定再送ポリシーがサポートされなければならない[MUST]。再送信回数を 0 に制限すると、順序なし配信と組み合わせて、各ユーザメッセージが 1 回だけ送信され、受信した順序で配信される UDP のようなサービスが提供される。

[RFC8260] で定義されているメッセージインタリーブのサポートを使用すべきである[SHOULD]。

6.2 SCTP アソシエーション管理

WebRTC のコンテキストでは、SCTP アソシエーションは、WebRTC PeerConnection の 2 つのエンドポイントが、通常は Session Description Protocol (SDP) の交換である JavaScript Session Establishment Protocol (JSEP) によってネゴシエートされるように、それを開くことに合意したときに設定される。ICE を介して選

択された DTLS 接続を使用する。通常、これは、SRTP メディアストリームにキーを設定するために使用される BUNDLE または同等の DTLS 接続を介して共有される。

SCTP アソシエーションのセットアップ中にネゴシエートされるストリームの数は、アソシエーションのセットアップ中にネゴシエートできるストリームの最大数である 65535 にする必要がある[SHOULD]。

SCTP は、SCTP アソシエーションを終了する 2 つの方法をサポートする。最初の方法は、アソシエーションのシャットダウン中にメッセージが失われないことを保証する手順が使用される、適切な方法である。2 番目の方法は、一方がアソシエーションを中止できる非グレースフル方法である。

各 SCTP エンドポイントは、ユーザメッセージおよびテストメッセージの再送信回数を監視することによって、ピアの到達可能性を継続的に監視する。過剰な再送信の場合、アソシエーションは非グレースフルな方法で終了される。

SCTP アソシエーションが適切な方法で閉じられると、そのすべてのデータチャネルが閉じられる。正常でないティアダウンの場合、すべてのデータチャネルも閉じられるが、可能であればエラー表示を提供すべきである[SHOULD]。

6.3 SCTP ストリーム

SCTP は、別の SCTP エンドポイントへの SCTP アソシエーション内に存在する単方向論理チャネルとしてストリームを定義する。ストリームは、シーケンス内配信の概念を提供し、多重化するために使用される。各ユーザメッセージは、順序付けられているかどうかにかかわらず、特定のストリームで送信される。

順序付けは、同じストリームで送信された順序付けされたメッセージに対してのみ保持される。

6.4 データチャネル定義

データチャネルは、それに付随するアプリケーションレベルの API が WebSockets の API を厳密にミラーできるように定義されている。これは、データの双方向ストリームと、データチャネルの意味を識別するために使用される「label」と呼ばれるテキストフィールドを意味する。

データチャネルの実現は、同じ SCTP ストリーム識別子を有する 1 つの入力ストリームと 1 つの出力 SCTP ストリームの対である。これらの SCTP ストリーム識別子の選択方法は、プロトコルおよび実装に依存する。これにより、双方向通信が可能になる。

また、各データチャネルには、各方向に次のプロパティがある。

- reliable または unreliable メッセージ送信：
  unreliable 送信の場合は、同じレベルの unreliability が使用される。SCTP では、これは SCTP ユーザメッセージのプロパティであり、SCTP ストリームのプロパティではないことに注意すること。
- 送信されたメッセージの順序内または順序外のメッセージ配信：
  SCTP では、これは SCTP ストリームではなく SCTP ユーザメッセージのプロパティであることに注意すること。
- 優先度は 2 バイトの符号なし整数である：
  これらの優先度は、[RFC8260] でのインタリーブをサポートする対応するストリームスケジューラの定義に従って、重み付けされた公平なキューイングスケジューリング優先度として解釈されなければならない[MUST]。WebRTC で使用する場合、使用される値は、128 (below normal)、256 (normal)、512 (high)、または 1024 (extra high) のいずれかである必要がある[SHOULD]。
- オプションのラベル。
- オプションのプロトコル。

[RFC8832] で指定されたプロトコルとネゴシエートされるデータチャネルでは、上記のすべての特性が両方向で同じであることに注意すること。

6.5 データチャネルを開く

データチャネルは、SCTP アソシエーション内のネゴシエーション (インバンドネゴシエーションと呼ばれる) またはアウトオブバンドネゴシエーションを使用して開くことができる。帯域外ネゴシエーションとは、チャネルのパラメータとその作成に関して合意をもたらす任意の方法として定義される。詳細はこの文書の範囲外である。データチャネルを使用するアプリケーションは、両方のエンドポイントでネゴシエーション方式を一貫して使用する必要がある。

帯域内ネゴシエーションの簡易プロトコルは、[RFC8832] で規定される。

一方の側がアウトオブバンドネゴシエーションを使用してチャネルを開く場合は、ストリームを選択する。特に定義またはネゴシエートされない限り、ストリームは DTLS ロール (クライアントは偶数のストリーム識別子を取得し、サーバは奇数のストリーム識別子を取得する) に基づいて選択される。ただし、アプリケーションは既存のストリームとの競合を回避する必要がある。既存のデータチャネルの一部であるストリームを再利用しようとする場合、追加は失敗しなければならない[MUST]。ストリームの選択に加えて、アプリケーションはメッセージの送信に使用するオプションも決定すべきである[SHOULD]。アプリケーションは、アプリケーション固有の方法で、ピアのアプリケーションが使用する選択されたストリームと、その側からデータを送信するためのオプションも認識できることを保証しなければならない[MUST]。

6.6 データチャネルでのユーザデータの転送

オプションが変更されていたり、メッセージごとのオプションが上位レベルで指定されていない限り、双方向のデータチャネルで送信されるすべてのデータは、データチャネルが開かれたときに定義された信頼性を使用して、基になるストリームで送信されなければならない[MUST]。

SCTP のメッセージ方向は、ユーザメッセージのメッセージ境界を維持するために使用される。

したがって、送信者は、SCTP ユーザメッセージに複数のアプリケーションメッセージを入れてはならない[MUST NOT]。非推奨の PPID ベースのフラグメンテーションおよび再構成が使用されない限り、送信者は各 SCTP ユーザメッセージに 1 つのアプリケーションメッセージだけを含めなければならない[MUST]。

SCTP ペイロードプロトコル識別子 (PPID) は、「ペイロードデータ」の解釈を通知するために使用される。以下の PPID を使用しなければならない[MUST] (8 章を参照)。

WebRTC String：	UTF-8 でエンコードされた空でない JavaScript 文字列を識別する。
WebRTC String Empty：	UTF-8 でエンコードされた空の JavaScript 文字列を識別する。
WebRTC Binary：	空でない JavaScript バイナリデータを識別する。 (ArrayBuffer、ArrayBufferView、Blob)
WebRTC Binary Empty：	空の JavaScript バイナリデータを識別する。 (ArrayBuffer、ArrayBufferView、Blob)

SCTP では、空のユーザメッセージの送信はサポートされていない。したがって、空のメッセージを送信する必要がある場合は、適切な PPID (WebRTC String Empty または WebRTC Binary Empty) が使用され、1 つの 0 バイトの SCTP ユーザメッセージが送信される。これらの PPID のいずれかを含む SCTP ユーザメッセージを受信する場合、受信者は SCTP ユーザメッセージを無視し、空のメッセージとして処理しなければならない[MUST]。

PPID「WebRTC String Partial」および「WebRTC Binary Partial」の使用は推奨されない。これらは、信頼性

が高く順序付けされたデータチャネルに属するユーザメッセージの PPID ベースの断片化と再構築に使用された。

サポートされていない PPID を持つメッセージが受信された場合、または受信したメッセージに関連するエラー状態が受信者によって検出された場合 (例えば、不正な注文)、受信者は対応するデータチャネルを閉じるべきである[SHOULD]。これは特に、追加の PPID を使用する拡張は、事前ネゴシエーションなしでは使用できないことを意味する。

[RFC4960] で規定されている SCTP ベースプロトコルは、ユーザメッセージのインタリーブをサポートしていない。したがって、大きなユーザメッセージを送信すると、SCTP アソシエーションが独占される可能性がある。この制限を克服するために、[RFC8260] は、メッセージインタリービングをサポートするための拡張を定義し、それを使用するべきである[SHOULD]。メッセージのインタリーブがサポートされていない限り、送信者は独占を避けるために最大メッセージサイズを 16KB に制限すべきである[SHOULD]。

アプリケーションでは任意の大きさの単一メッセージをサポートできないため、メッセージサイズは特定のサイズの範囲内に収めることを推奨する。この制限は、例えば [RFC8841] を使ってネゴシエートしなければならない。

送信者は、遅延を最小にするために、Nagle アルゴリズム ([RFC1122] を参照) を無効にすべきである[SHOULD]。

6.7 データチャネルを閉じる

データチャネルのクローズは、対応する送信ストリーム [RFC6525] をリセットすることによって通知されなければならない[MUST]。これは、一方がデータチャネルを閉じることを決定した場合、対応する送信ストリームをリセットすることを意味する。ピアは、着信ストリームがリセットされたことを確認すると、対応する発信ストリームもリセットする。これが完了すると、データチャネルは閉じられる。ストリームをリセットすると、ストリームの Stream Sequence Number (SSN) が「0」に戻され、リセットが実行されたことがアプリケーション層に通知される。ストリームは、リセットの実行後に再利用できる。

[RFC6525] は、ストリームがリセットされる前に、すべてのメッセージが配信 (または破棄) される ことも保証する。

7. セキュリティに関する考慮事項

この文書は、[RFC8826] と [RFC8827] に記載されている考慮事項に、追加の考慮事項は加えない。

受信者は、任意の大きさのメッセージを送信しようとする送信者に備える必要があることに注意すべきである。

8. IANA に関する考慮事項

この文書では、すでに登録されている 6 つの SCTP ペイロードプロトコル識別子 (PPID) を使用している。「DOMString Last」、「Binary Data Partial」、「Binary Data Last」、「DOMString Partial」、「WebRTC String Empty」、「WebRTC Binary Empty」である。[RFC4960] は、これらの識別子が割り当てられた「SCTP ペイロードプロトコル識別子」レジストリを作成する。IANA は、この文書を指すようにこれら 6 つの割り当ての参照を更新し、最初の 4 つの PPID の名前を変更した。対応する日付は変更されない。

6 つの割り当てが次のように更新された。

値	SCTP PPID	参考	日付
WebRTC String	51	RFC 8831	2013-09-20
WebRTC Binary Partial (非推奨)	52	RFC 8831	2013-09-20
WebRTC Binary	53	RFC 8831	2013-09-20
WebRTC String Partial (非推奨)	54	RFC 8831	2013-09-20
WebRTC String Empty	56	RFC 8831	2014-08-22
WebRTC Binary Empty	57	RFC 8831	2014-08-22

表 1

9. 参考資料

9.1 標準参照

[RFC2119] Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels”, BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC3758] Stewart, R., Ramalho, M., Xie, Q., Tuexen, M., and P. Conrad, “Stream Control Transmission Protocol (SCTP) Partial Reliability Extension”, RFC 3758, DOI 10.17487/RFC3758, May 2004, <https://www.rfc-editor.org/info/rfc3758>.

[RFC4820] Tuexen, M., Stewart, R., and P. Lei, “Padding Chunk and Parameter for the Stream Control Transmission Protocol (SCTP)”, RFC 4820, DOI 10.17487/RFC4820, March 2007, <https://www.rfc-editor.org/info/rfc4820>.

[RFC4821] Mathis, M. and J. Heffner, “Packetization Layer Path MTU Discovery”, RFC 4821, DOI 10.17487/RFC4821, March 2007, <https://www.rfc-editor.org/info/rfc4821>.

[RFC4960] Stewart, R., Ed., “Stream Control Transmission Protocol”, RFC 4960, DOI 10.17487/RFC4960, September 2007, <https://www.rfc-editor.org/info/rfc4960>.

[RFC5061] Stewart, R., Xie, Q., Tuexen, M., Maruyama, S., and M. Kozuka, “Stream Control Transmission Protocol (SCTP) Dynamic Address Reconfiguration”, RFC 5061, DOI 10.17487/RFC5061, September 2007, <https://www.rfc-editor.org/info/rfc5061>.

[RFC6525] Stewart, R., Tuexen, M., and P. Lei, “Stream Control Transmission Protocol (SCTP) Stream Reconfiguration”, RFC 6525, DOI 10.17487/RFC6525, February 2012, <https://www.rfc-editor.org/info/rfc6525>.

[RFC7496] Tuexen, M., Seggelmann, R., Stewart, R., and S. Loreto, “Additional Policies for the Partially Reliable Stream Control Transmission Protocol Extension”, RFC 7496, DOI 10.17487/RFC7496, April 2015, <https://www.rfc-editor.org/info/rfc7496>.

[RFC8174] Leiba, B., “Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words”, BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8260] Stewart, R., Tuexen, M., Loreto, S., and R. Seggelmann, “Stream Schedulers and User Message Interleaving for the Stream Control Transmission Protocol”, RFC 8260, DOI 10.17487/RFC8260, November 2017, <https://www.rfc-editor.org/info/rfc8260>.

[RFC8261] Tuexen, M., Stewart, R., Jesup, R., and S. Loreto, “Datagram Transport Layer Security (DTLS) Encapsulation of SCTP Packets”, RFC 8261, DOI 10.17487/RFC8261, November 2017,

<https://www.rfc-editor.org/info/rfc8261>.

[RFC8445] Keranen, A., Holmberg, C., and J. Rosenberg, “Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal”, RFC 8445, DOI 10.17487/RFC8445, July 2018, <https://www.rfc-editor.org/info/rfc8445>.

[RFC8826] Rescorla, E., “Security Considerations for WebRTC”, RFC 8826, DOI 10.17487/RFC8826, January 2021, <https://www.rfc-editor.org/info/rfc8826>.

[RFC8827] Rescorla, E., “WebRTC Security Architecture”, RFC 8827, DOI 10.17487/RFC8827, January 2021, <https://www.rfc-editor.org/info/rfc8827>.

[RFC8829] Uberti, J., Jennings, C., and E. Rescorla, Ed., “JavaScript Session Establishment Protocol (JSEP)”, RFC 8829, DOI 10.17487/RFC8829, January 2021, <https://www.rfc-editor.org/info/rfc8829>.

[RFC8832] Jesup, R., Loreto, S., and M. Tüxen, “WebRTC Data Channel Establishment Protocol”, RFC 8832, DOI 10.17487/RFC8832, January 2021, <https://www.rfc-editor.org/info/rfc8832>.

[RFC8841] Holmberg, C., Shpount, R., Loreto, S., and G. Camarillo, “Session Description Protocol (SDP) Offer/Answer Procedures for Stream Control Transmission Protocol (SCTP) over Datagram Transport Layer Security (DTLS) Transport”, RFC 8841, DOI 10.17487/RFC8841, January 2021, <https://www.rfc-editor.org/info/rfc8841>.

9.2 参考文献

[RFC1122] Braden, R., Ed., "Requirements for Internet Hosts - Communication Layers", STD 3, RFC 1122, DOI 10.17487/RFC1122, October 1989, <https://www.rfc-editor.org/info/rfc1122>.

[RFC4347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security", RFC 4347, DOI 10.17487/RFC4347, April 2006, <https://www.rfc-editor.org/info/rfc4347>.

[RFC5389] Rosenberg, J., Mahy, R., Matthews, P., and D. Wing, "Session Traversal Utilities for NAT (STUN)", RFC 5389, DOI 10.17487/RFC5389, October 2008, <https://www.rfc-editor.org/info/rfc5389>.

[RFC5764] McGrew, D. and E. Rescorla, "Datagram Transport Layer Security (DTLS) Extension to Establish Keys for the Secure Real-time Transport Protocol (SRTP)", RFC 5764, DOI 10.17487/RFC5764, May 2010, <https://www.rfc-editor.org/info/rfc5764>.

[RFC6083] Tuexen, M., Seggelmann, R., and E. Rescorla, "Datagram Transport Layer Security (DTLS) for Stream Control Transmission Protocol (SCTP)", RFC 6083, DOI 10.17487/RFC6083, January 2011, <https://www.rfc-editor.org/info/rfc6083>.

[RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, DOI 10.17487/RFC6347, January 2012, <https://www.rfc-editor.org/info/rfc6347>.

[RFC6951] Tuexen, M. and R. Stewart, "UDP Encapsulation of Stream Control Transmission Protocol (SCTP) Packets for End-Host to End-Host Communication", RFC 6951, DOI 10.17487/RFC6951, May 2013, <https://www.rfc-editor.org/info/rfc6951>.

[TLS-DTLS13] Rescorla, E., Tschofenig, H., and N. Modadugu, "The Datagram Transport Layer Security (DTLS) Protocol Version 1.3", Work in Progress, Internet-Draft, draft-ietf-tls-dtls13-39, 2 November 2020, <https://tools.ietf.org/html/draft-ietf-tls-dtls13-39>.

VIII. RFC8832 原文の著作権について

IX. RFC8832 の和訳

RFC 8832　WebRTC データチャネル確立プロトコル

概要

WebRTC フレームワークは、2 つのピアの Web ブラウザ間で音声、ビデオ、およびデータを使用する直接対話型リッチ通信のプロトコルサポートを指定する。この文書は、ピア間の対称データチャネルを確立するための簡単なプロトコルを規定する。これは双方向ハンドシェイクを使用し、ハンドシェイクの完了を待たずにユーザデータを送信できる。

1.　はじめに

Data Channel Establishment Protocol (DCEP) は、WebRTC データチャネルコンテキスト [RFC8831] において、対称データチャネルを開くための簡単なインバンド方式を提供するように設計されている。[RFC8831] で議論されているように、プロトコルは、データグラムトランスポート層セキュリティ (DTLS) にカプセル化されたストリーム制御伝送プロトコル (SCTP) [RFC4960] を使用する ([RFC8261] で記述されている)。

これにより、SCTP および DTLS のすでに標準化されたトランスポートおよびセキュリティ機能の恩恵を DCEP が受けることがでる。DTLS 1.0 は [RFC4347] で定義されている。現在の最新バージョンである DTLS 1.2 は、[RFC6347] で定義されている。また、[TLS-DTLS13] では、次期バージョンである DTLS 1.3 が定義されている。

2.　表記規則

この文書のキーワードである「MUST」、「MUST NOT」、「REQUIRED」、「SHALL」、「SHALL NOT」、「SHOULD」、「SHOULD NOT」、「RECOMMENDED」、「NOT RECOMMENDED」、「MAY」、および「OPTIONAL」は、ここに示すように、すべて大文字で表示される場合にのみ、BCP 14 [RFC2119] [RFC8174] で記述されているように解釈される。

3.　用語

このマニュアルでは、次の用語を使用する。

関連付け：　SCTP アソシエーション。

ストリーム：　SCTP アソシエーションの単方向ストリーム。SCTP ストリーム識別子 (0～65534) によって一意に識別される。

注意：SCTP ストリーム識別子 65535 は、SCTP INIT チャンクおよび INIT-ACK チャンクのために予約されており、最大 65535 ストリーム (0～65534) しかネゴシエートできない。

ストリーム識別子：　ストリームを一意に識別する SCTP ストリーム識別子。

データチャネル：　同じストリーム識別子を持つ 2 つのストリーム (各方向に 1 つずつ) は、一緒に管理される。

4.　プロトコルの概要

Data Channel Establishment Protocol は、SCTP アソシエーション上で一貫した一連のプロパティを使用して双方向データチャネルを確立する、簡単でオーバーヘッドの少ない方法である。

一貫性のあるプロパティのセットには、次のものが含まれる。

- 信頼できる、または信頼できないメッセージ送信。信頼性のない送信の場合は、同じレベルの信頼性のない送信が使用される。
- in-order または out-of-order メッセージ配信。
- データチャネルの優先度。
- データチャネルのオプションのラベル。
- データチャネルのオプションのプロトコル。
- ストリーム。

このプロトコルは、双方向ハンドシェイクを使用してデータチャネルを開く。ハンドシェイクは、同じストリーム識別子を持つ 1 つの受信ストリームと 1 つの送信ストリームをペアにして、単一の双方向データチャネルにする。データチャネルのオープンを開始するピアは、対応する受信ストリームと送信ストリームが使用されていないストリーム識別子を選択し、送信ストリームで DATA_CHANNEL_OPEN メッセージを送信する。ピアは次のように応答する。

対応する送信ストリームの DATA_CHANNEL_ACK メッセージ。次に、データチャネルが開く。DCEP メッセージは、データチャネルに属するユーザメッセージと同じストリームで送信される。DCEP は特定の PPID を使用するため、逆多重化は SCTP ペイロードプロトコル識別子 (PPID) に基づいている。

注意:開始側は、DATA_CHANNEL_ACK が受信される前にユーザメッセージを送信してもよい[MAY]。

両側が同じストリーム識別子を使用してデータチャネルを開こうとする競合を避けるために、各側は、DATA_CHANNEL_OPEN メッセージを送信するときに、偶数または奇数のストリーム識別子を持つストリームを使用する必要がある[MUST]。SCTP over DTLS [RFC8261] を使用する場合、奇数または偶数を使用する側を決定するために使用される方法は、基になる DTLS 接続ロールに基づいている。DTLS クライアントとして機能する側は、偶数のストリーム識別子を持つストリームを使用する必要がある[MUST]。 DTLS サーバとして機能する側は、奇数のストリーム識別子を持つストリームを使用する必要がある[MUST]。

注意：ラベルの一意性は保証されない。両側が同時に「x」というラベルの付いたデータチャネルを開くと、「x」というラベルの付いた 2 つのデータチャネルが存在する。1 つは偶数のストリームペア、もう 1 つは奇数のペアである。

プロトコルフィールドの目的は、[RFC6455] で定義されている WebSocket サブプロトコル名レジストリから IANA に登録された文字列によって渡されるユーザデータを識別することで、アプリケーション間の相互運用 (フェデレーション) を容易にすることである。このフィールドは、複数の種類のデータチャネルを作成する可能性のある同種のアプリケーションに役立つ。プロトコルフィールドの一意性は保証されないことに注意が必要である。

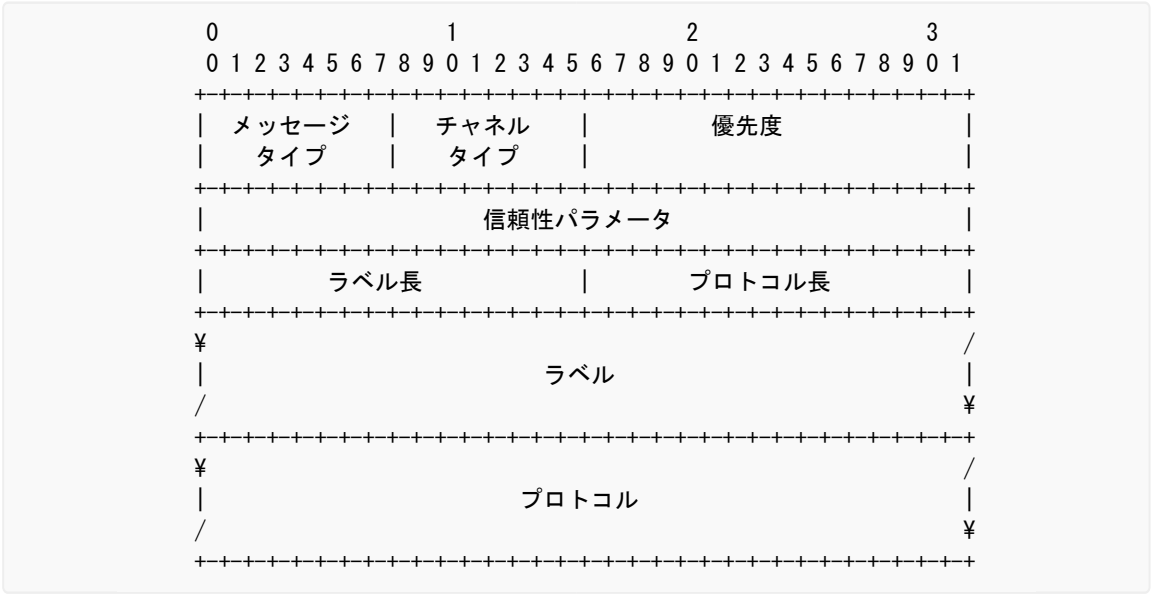
5. メッセージフォーマット

すべての DCEP メッセージは、メッセージのタイプを示す 「メッセージタイプ」 と呼ばれる 1 バイトのフィールドで始まる。対応する値は IANA によって管理される (8.2.1 項を参照) 。

5.1 DATA_CHANNEL_OPEN メッセージ

このメッセージは、最初に、ユーザメッセージに使用されるストリームのデータチャネルを使用して送信

される。



メッセージタイプ： 1 バイト (符号なし整数)

このフィールドは、DATA_CHANNEL_OPEN メッセージの IANA 定義のメッセージタイプを保持する。このフィールドの値は、8.2.1 項で指定されているように、0x03 である。

チャネルタイプ： 1 バイト (符号なし整数)

このフィールドでは、開くデータチャネルのタイプを指定する。値は IANA によって管理されている (8.2.2 項を参照) 。

DATA_CHANNEL_RELIABLE (0x00)： データチャネルは、信頼性の高い順方向双方向通信を提供する。

DATA_CHANNEL_RELIABLE_UNORDERED (0x80)： データチャネルは、信頼性の高い無順序双方向通信を提供する。

DATA_CHANNEL_PARTIAL_RELIABLE_REXMIT (0x01)： データチャネルは、部分的に信頼性のある順方向双方向通信を提供する。ユーザメッセージは、信頼性パラメータで指定された回数を超えて再送信されない。

DATA_CHANNEL_PARTIAL_RELIABLE_REXMIT_UNORDERED (0x81)： データチャネルは、部分的に信頼性の高い順序なし双方向通信を提供する。ユーザメッセージは、信頼性パラメータで指定された回数を超えて再送信されない。

DATA_CHANNEL_PARTIAL_RELIABLE_TIMED (0x02)： データチャネルは、部分的に信頼性のある順方向双方向通信を提供する。信頼性パラメータにミリ秒単位で指定されたライフタイムが経過すると、ユーザメッセージが送信または再送信されない場合がある。このライフタイムは、ユーザメッセージをプロトコルスタックに提供するときに開始される。

DATA_CHANNEL_PARTIAL_RELIABLE_TIMED_UNORDERED (0x82)： データチャネルは、部分的に信頼性の高い順序なし双方向通信を提供する。信頼性パラメータにミリ秒単位で指定されたライフタイムが経過すると、ユーザメッセージが送信または再送信されない場合がある。このライフタイムは、ユーザメッセージをプロトコルスタックに提供するときに開始される。

優先度： 2 バイト (符号なし整数)

[RFC8831] で説明されているデータチャネルの優先度。

信頼性パラメータ： 4 バイト (符号なし整数)

信頼できるデータチャネルの場合、このフィールドは送信側では 0 に設定しなければならず [MUST]、受信側では無視しなければならない[MUST]。再送信回数が制限されている部分的に信頼できるデータチャネルが使用されている場合、このフィールドは再送信回数を指定する。ライフタイムが制限された部分的に信頼できるデータチャネルが使用される場合、このフィールドは最大ライフタイムをミリ秒単位で指定する。次の表にこれを要約する。

チャネルタイプ	信頼性パラメータ
DATA_CHANNEL_RELIABLE	無視
DATA_CHANNEL_RELIABLE_UNORDERED	無視
DATA_CHANNEL_PARTIAL_RELIABLE_REXMIT	RTX の数
DATA_CHANNEL_PARTIAL_RELIABLE_REXMIT_UNORDERED	RTX の数
DATA_CHANNEL_PARTIAL_RELIABLE_TIMED	ライフタイム (ミリ秒)
DATA_CHANNEL_PARTIAL_RELIABLE_TIMED_UNORDERED	ライフタイム (ミリ秒)

表 1

ラベル長： 2 バイト (符号なし整数)

ラベルフィールドの長さ (バイト単位) 。

プロトコル長： 2 バイト (符号なし整数)

プロトコルフィールドのバイト単位の長さ。

ラベル： 可変長 (一連の文字)

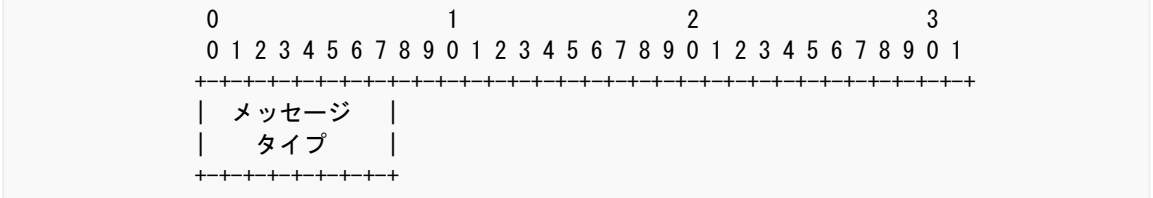
[RFC3629] で指定されている、UTF-8 でエンコードされた文字列としてのデータチャネルの名前。空の文字列を指定できる。

プロトコル： 可変長 (文字列)

これが空の文字列の場合、プロトコルは指定されない。空でない文字列の場合は、[RFC6455] で作成された WebSocket サブプロトコル名レジストリに登録されているプロトコルを指定する。この文字列は、[RFC3629] で指定されているように、UTF-8 でエンコードされる。

5.2 DATA_CHANNEL_ACK メッセージ

このメッセージは、DATA_CHANNEL_OPEN_RESPONSE メッセージへの応答として送信される。データチャネルを使用して、ユーザメッセージに使用されるストリームで送信される。このメッセージを受信すると、データチャネル設定ハンドシェイクが完了したことがオープナに通知される。



メッセージタイプ： 1 バイト (符号なし整数)

このフィールドは、DATA_CHANNEL_ACK メッセージの IANA 定義のメッセージタイプを保持する。このフィールドの値は、8.2.1 項で指定されているように、0x02 である。

6. 手順

すべての DCEP メッセージは、順序付けられた配送と信頼できる送信を使用して送信されなければならない[MUST]。それらは、対応するデータチャネルに属するユーザメッセージと同じ送信ストリームで送信されなければならない[MUST]。多重化および逆多重化は、SCTP PPID を使用して行われる。したがって、DCEP メッセージは、データチャネル確立プロトコル(8.1 節を参照)に割り当てられた PPID とともに送信されなければならない[MUST]。他のメッセージはこの PPID を使用して送信してはならない[MUST NOT]。

データチャネルのオープンを開始するピアは、対応する着信ストリームと発信ストリームが使用されていないストリーム識別子を選択する。その側が DTLS クライアントとして動作している場合、偶数ストリーム識別子を選択しなければならない[MUST]。サイドが DTLS サーバとして動作している場合は、奇数を選択する必要がある[MUST]。開始側ピアは、DATA_CHANNEL_OPEN メッセージのパラメータを入力し、選択したストリームで送信する。

未使用のストリームで DATA_CHANNEL_OPEN メッセージが受信され、ストリーム識別子がピアのロールに対応し、DATA_CHANNEL_OPEN メッセージ内のすべてのパラメータが有効である場合、対応する DATA_CHANNEL_ACK メッセージは、DATA_CHANNEL_OPEN メッセージが受信されたものと同じストリーム識別子を持つストリームに送信される。

DATA_CHANNEL_OPEN メッセージが上記の条件を満たさない場合、受信者は、[RFC8831] で説明されている手順を使用して、対応するデータチャネルを閉じなければならず[MUST]、受信したメッセージに応答して DATA_CHANNEL_ACK メッセージを送信してはならない[MUST NOT]。これは、たとえば、既に使用されているストリームで DATA_CHANNEL_OPEN メッセージを受信した場合、DATA_CHANNEL_OPEN メッセージ内のパラメータに問題がある場合、奇数/偶数の規則に違反している場合、または DATA_CHANNEL_OPEN メッセージ自体が適切な形式ではない場合に発生する可能性がある。したがって、DATA_CHANNEL_ACK メッセージを受信していないストリームに対する SCTP ストリームリセット要求を受信すると、対応する DATA_CHANNEL_OPEN メッセージの送信者に対して、データチャネル設定プロシージャの失敗が示される。対応する送信ストリームのリセットにも成功した後、ピアによって開始されたデータチャネルの閉鎖が完了し、ストリームで新しい DATA_CHANNEL_OPEN メッセージを送信できる。DATA_CHANNEL_OPEN メッセージが送信された後、そのメッセージの送信者は、対応する DATA_CHANNEL_ACK メッセージの受信を待たずに、ユーザデータを含むメッセージの送信を開始してもよい場合がある[MAY]。ただし、DATA_CHANNEL_ACK メッセージまたはその他のメッセージがデータチャネルで受信される前に、データチャネルが順序付けられているかどうかに関係なく、ユーザデータを含み、このデータチャネルに属する他のすべてのメッセージを順序付けて送信する必要がある[MUST]。DATA_CHANNEL_ACK またはその他のメッセージがデータチャネルで受信された後、ユーザデータを含むメッセージは、順序付けられたデータチャネルで送信する必要があり[MUST]、順序付けられていないデータチャネルで順序なしで送信する必要がある[MUST]。したがって、未使用のストリームでユーザデータを含むメッセージを受信すると、エラーが発生する。その場合、[RFC8831] に記述されているように、対応するデータチャネルは閉じられなければならない[MUST]。

7. セキュリティに関する考慮事項

DATA_CHANNEL_OPEN メッセージには、プロトコルとラベルの 2 つの可変長フィールドが含まれる。受信者は、これらのフィールドの最大長が 65535 バイトである DATA_CHANNEL_OPEN メッセージを受信するように準備する必要がある[MUST]。フィールド長の不整合、不明なパラメータ値の使用、奇数/偶数の規則違反などのエラーの場合も、対応するデータチャネルを閉じることで処理する必要がある[MUST]。また、ピアが最大数のデータチャネルを開くためのエンドポイントも準備する必要がある[MUST]。このプロ

トコルは、プライバシー、整合性、または認証を提供しない[MUST]。これらすべてを含むプロトコルスイートの一部として使用する必要がある[MUST]。このようなプロトコルスイートは、[RFC8261] で規定されている。

一般的な考慮事項については、[RFC8826] および [RFC8827] を参照。

8. IANA に関する考慮事項

IANA は、既存の SCTP PPID 割り当て (8.1 節) の参照を更新し、2 つの新しい登録テーブル (8.2.1 項と 8.2.2 項) を含む、独自の DCE 用 URL (8.2 節) を持つ新しいスタンドアロンレジストリを作成した。

8.1 SCTP ペイロードプロトコル識別子

この文書では、「WebRTC Control」として事前に登録されている SCTP ペイロードプロトコル識別子 (PPID) を使用する。[RFC4960] は、この識別子が割り当てられた「SCTP ペイロードプロトコル識別子」レジストリを作成した。IANA は PPID 名を「WebRTC Control」から「WebRTC DCEP」に更新し、この文書を指すように参照を更新した。対応する日付は保持されている。

したがって、この割り当ては次のように表示される。

値	SCTP PPID	参考	日付
WebRTC DCEP	50	RFC 8832	2013-09-20

表 2

8.2 DCEP 用の新しいスタンドアロンレジストリ

IANA は 「Data Channel Establishment Protocol (DCEP) Parameters」レジストリを作成した。ここには、8.2.1 項および 8.2.2 項に記載した 2 つの表が含まれる。

8.2.1 新しいメッセージタイプレジストリ

IANA は、DCEP メッセージ内の 1 バイトのメッセージタイプフィールドを管理するために、DCEP 用のメッセージタイプレジストリを作成した (5 章を参照) 。この登録表は、8.2 節に記載されているレジストリのサブレジストリである。

新しいメッセージタイプの割り当ては、[RFC8126] で定義されている RFC Required アクションを通じて行われる。新しいメッセージタイプの文書は、以下の情報を含まなければならない[MUST]。

1. 新しいメッセージ種類の名前である。
2. 各メッセージタイプが DCEP 内でどのように使用されるかについての詳細な手順説明。

初期登録は次のとおりである。

名前	タイプ	参考
予約済み	0x00	RFC 8832
予約済み	0x01	RFC 8832
DATA_CHANNEL_ACK	0x02	RFC 8832
DATA_CHANNEL_OPEN	0x03	RFC 8832
未割り当て	0x04-0xfe	
予約済み	0xff	RFC 8832

表 3

値 0x00 と 0x01 は、文書のドラフトバージョンで使用されているため、相互運用性の問題を回避するために予約されていることに注意すること。値 0xff は将来の拡張のために予約されている。指定できる値の範囲は 0x00～0xff である。

8.2.2 新しいチャネルタイプレジストリ

IANA は、DATA_CHANNEL_OPEN メッセージ (5.1 節を参照) の 1 バイトの 「チャネルタイプ」 フィールドを管理するために、DCEP の 「チャネルタイプ」 レジストリを作成した。この登録表は、8.2 節に記載されているレジストリ内のサブレジストリである。

新しいメッセージタイプの割り当ては、[RFC8126] で定義されている RFC Required アクションを通じて行われる。新しいチャネルタイプの文書は、次の情報を含まなければならない。

1. 新しいチャネルタイプの名前。
2. この新しいチャネルタイプを使用するデータチャネルのユーザメッセージ処理の詳細な手順説明。

新しいチャネルタイプが順序付きおよび順序なしメッセージ配送をサポートする場合、メッセージ配送が順序なしかどうかを示すために上位ビットを使用しなければならない[MUST]。

初期登録は次のとおりである。

名前	タイプ	参考
DATA_CHANNEL_RELIABLE	0x00	RFC 8832
DATA_CHANNEL_RELIABLE_UNORDERED	0x80	RFC 8832
DATA_CHANNEL_PARTIAL_RELIABLE_REXMIT	0x01	RFC 8832
DATA_CHANNEL_PARTIAL_RELIABLE_REXMIT_UNORDERED	0x81	RFC 8832
DATA_CHANNEL_PARTIAL_RELIABLE_TIMED	0x02	RFC 8832
DATA_CHANNEL_PARTIAL_RELIABLE_TIMED_UNORDERED	0x82	RFC 8832
予約済み	0x7f	RFC 8832
予約済み	0xff	RFC 8832
未割り当て	rest	

表 4

値 0x7f と 0xff は将来の拡張のために予約されている。指定できる値の範囲は 0x00～0xff である。

9. 参考資料

9.1 標準参照

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, DOI 10.17487/RFC3629, November 2003, <https://www.rfc-editor.org/info/rfc3629>.

[RFC4960] Stewart, R., Ed., "Stream Control Transmission Protocol", RFC 4960, DOI 10.17487/RFC4960, September 2007, <https://www.rfc-editor.org/info/rfc4960>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

[RFC8261] Tuexen, M., Stewart, R., Jesup, R., and S. Loreto, "Datagram Transport Layer Security (DTLS) Encapsulation of SCTP Packets", RFC 8261, DOI 10.17487/RFC8261, November 2017, <https://www.rfc-editor.org/info/rfc8261>.

[RFC8831] Jesup, R., Loreto, S., and M. Tüxen, "WebRTC Data Channels", RFC 8831, DOI 10.17487/RFC8831, January 2021, <https://www.rfc-editor.org/info/rfc8831>.

9.2 参考文献

[RFC4347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security", RFC 4347, DOI 10.17487/RFC4347, April 2006, <https://www.rfc-editor.org/info/rfc4347>.

[RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, DOI 10.17487/RFC6347, January 2012, <https://www.rfc-editor.org/info/rfc6347>.

[RFC6455] Fette, I. and A. Melnikov, "The WebSocket Protocol", RFC 6455, DOI 10.17487/RFC6455, December 2011, <https://www.rfc-editor.org/info/rfc6455>.

[RFC8826] Rescorla, E., "Security Considerations for WebRTC", RFC 8826, DOI 10.17487/RFC8826, January 2021, <https://www.rfc-editor.org/info/rfc8826>.

[RFC8827] Rescorla, E., "WebRTC Security Architecture", RFC 8827, DOI 10.17487/RFC8827, January 2021, <https://www.rfc-editor.org/info/rfc8827>.

[TLS-DTLS13] Rescorla, E., Tschofenig, H., and N. Modadugu, "The Datagram Transport Layer Security (DTLS) Protocol Version 1.3", Work in Progress, Internet-Draft, draft-ietf-tls-dtls13-39, 2 November 2020, <https://tools.ietf.org/html/draft-ietf-tls-dtls13-39>.