

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 17, 2020

J. Uberti
Google
Jul 16, 2019

WebRTC Forward Error Correction Requirements
draft-ietf-rtcweb-fec-10

Abstract

This document provides information and requirements for how Forward Error Correction (FEC) should be used by WebRTC implementations.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 17, 2020.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Terminology	2
3. Types of FEC	2
3.1. Separate FEC Stream	3
3.2. Redundant Encoding	3
3.3. Codec-Specific In-band FEC	3
4. FEC for Audio Content	4
4.1. Recommended Mechanism	4
4.2. Negotiating Support	5
5. FEC for Video Content	5
5.1. Recommended Mechanism	5
5.2. Negotiating Support	6
6. FEC for Application Content	6
7. Implementation Requirements	7
8. Adaptive Use of FEC	7
9. Security Considerations	8
10. IANA Considerations	8
11. Acknowledgements	8
12. References	8
12.1. Normative References	8
12.2. Informative References	9
Appendix A. Change log	11
Author's Address	13

1. Introduction

In situations where packet loss is high, or perfect media quality is essential, Forward Error Correction (FEC) can be used to proactively recover from packet losses. This specification provides guidance on which FEC mechanisms to use, and how to use them, for WebRTC implementations.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Types of FEC

FEC describes the sending of redundant information in an outgoing packet stream so that information can still be recovered even in the face of packet loss. There are multiple ways this can be

accomplished for RTP media streams [RFC3550]; this section enumerates the various mechanisms available and describes their tradeoffs.

3.1. Separate FEC Stream

This approach, as described in [RFC5956], Section 4.3, sends FEC packets as an independent RTP stream with its own synchronization source (SSRC, [RFC3550]) and payload type, multiplexed with the primary encoding. While this approach can protect multiple packets of the primary encoding with a single FEC packet, each FEC packet will have its own IP+UDP+RTP+FEC header, and this overhead can be excessive in some cases, e.g., when protecting each primary packet with a FEC packet.

This approach allows for recovery of entire RTP packets, including the full RTP header.

3.2. Redundant Encoding

This approach, as described in [RFC2198], allows for redundant data to be piggybacked on an existing primary encoding, all in a single packet. This redundant data may be an exact copy of a previous payload, or for codecs that support variable-bitrate encodings, possibly a smaller, lower-quality representation. In certain cases, the redundant data could include encodings of multiple prior audio frames.

Since there is only a single set of packet headers, this approach allows for a very efficient representation of primary + redundant data. However, this savings is only realized when the data all fits into a single packet (i.e. the size is less than a MTU). As a result, this approach is generally not useful for video content.

As described in [RFC2198], Section 4, this approach cannot recover certain parts of the RTP header, including the marker bit, CSRC information, and header extensions.

3.3. Codec-Specific In-band FEC

Some audio codecs, notably Opus [RFC6716] and AMR [RFC4867], support their own in-band FEC mechanism, where redundant data is included in the codec payload. This is similar to the redundant encoding mechanism described above, but as it adds no additional framing, it can be slightly more efficient.

For Opus, audio frames deemed important are re-encoded at a lower bitrate and appended to the next payload, allowing partial recovery of a lost packet. This scheme is fairly efficient; experiments

performed indicate that when Opus FEC is used, the overhead imposed is only about 20-30%, depending on the amount of protection needed. Note that this mechanism can only carry redundancy information for the immediately preceding audio frame; as such the decoder cannot fully recover multiple consecutive lost packets, which can be a problem on wireless networks. See [RFC6716], Section 2.1.7, and this Opus mailing list post [OpusFEC] for more details.

For AMR/AMR-WB, packets can contain copies or lower-quality encodings of multiple prior audio frames. See [RFC4867], Section 3.7.1 for details on this mechanism.

In-band FEC mechanisms cannot recover any of the RTP header.

4. FEC for Audio Content

The following section provides guidance on how to best use FEC for transmitting audio data. As indicated in Section 8 below, FEC should only be activated if network conditions warrant it, or upon explicit application request.

4.1. Recommended Mechanism

When using variable-bitrate codecs without an internal FEC, redundant encoding (as described in Section 3.2) with lower-fidelity version(s) of the previous packet(s) is RECOMMENDED. This provides reasonable protection of the payload with only moderate bitrate increase, as the redundant encodings can be significantly smaller than the primary encoding.

When using the Opus codec, use of the built-in Opus FEC mechanism is RECOMMENDED. This provides reasonable protection of the audio stream against individual losses, with minimal overhead. Note that, as indicated above, the built-in Opus FEC only provides single-frame redundancy; if multi-packet protection is needed, the aforementioned redundant encoding with reduced-bitrate Opus encodings SHOULD be used instead.

When using the AMR/AMR-WB codecs, use of their built-in FEC mechanism is RECOMMENDED. This provides slightly more efficient protection of the audio stream than redundant encoding.

When using constant-bitrate codecs, e.g., PCMU [RFC5391], redundant encoding MAY be used, but this will result in a potentially significant bitrate increase, and suddenly increasing bitrate to deal with losses from congestion may actually make things worse.

Because of the lower packet rate of audio encodings, usually a single packet per frame, use of a separate FEC stream comes with a higher overhead than other mechanisms, and therefore is NOT RECOMMENDED.

As mentioned above, the recommended mechanisms do not allow recovery of parts of the RTP header that may be important in certain audio applications, e.g., CSRCs and RTP header extensions like those specified in [RFC6464] and [RFC6465]. Implementations SHOULD account for this and attempt to approximate this information, using an approach similar to those described in [RFC2198], Section 4, and [RFC6464], Section 5.

4.2. Negotiating Support

Support for redundant encoding of a given RTP stream SHOULD be indicated by including audio/red [RFC2198] as an additional supported media type for the associated m= section in the SDP offer [RFC3264]. Answerers can reject the use of redundant encoding by not including the audio/red media type in the corresponding m= section in the SDP answer.

Support for codec-specific FEC mechanisms are typically indicated via "a=fmtp" parameters.

For Opus, a receiver MUST indicate that it is prepared to use incoming FEC data with the "useinbandfec=1" parameter, as specified in [RFC7587]. This parameter is declarative and can be negotiated separately for either media direction.

For AMR/AMR-WB, support for redundant encoding, and the maximum supported depth, are controlled by the 'max-red' parameter, as specified in [RFC4867], Section 8.1. Receivers MUST include this parameter, and set it to an appropriate value, as specified in [TS.26114], Table 6.3.

5. FEC for Video Content

The following section provides guidance on how to best use FEC for transmitting video data. As indicated in Section 8 below, FEC should only be activated if network conditions warrant it, or upon explicit application request.

5.1. Recommended Mechanism

Video frames, due to their size, often require multiple RTP packets. As discussed above, a separate FEC stream can protect multiple packets with a single FEC packet. In addition, the "flexfec" FEC mechanism described in [I-D.ietf-payload-flexible-fec-scheme] is also

capable of protecting multiple RTP streams via a single FEC stream, including all the streams that are part of a BUNDLE [I-D.ietf-mmusic-sdp-bundle-negotiation] group. As a result, for video content, use of a separate FEC stream with the flexfec RTP payload format is RECOMMENDED.

To process the incoming FEC stream, the receiver can demultiplex it by SSRC, and then correlate it with the appropriate primary stream(s) via the CSRC(s) present in the RTP header of flexfec repair packets, or the SSRC field present in the FEC header of flexfec retransmission packets.

5.2. Negotiating Support

Support for a SSRC-multiplexed flexfec stream to protect a given RTP stream SHOULD be indicated by including one of the formats described in [I-D.ietf-payload-flexible-fec-scheme], Section 5.1.2, as an additional supported media type for the associated m= section in the SDP offer [RFC3264]. As mentioned above, when BUNDLE is used, only a single flexfec repair stream will be created for each BUNDLE group, even if flexfec is negotiated for each primary stream.

Answerers can reject the use of SSRC-multiplexed FEC, by not including the offered FEC formats in the corresponding m= section in the SDP answer.

Use of FEC-only m-lines, and grouping using the SDP group mechanism as described in [RFC5956], Section 4.1 is not currently defined for WebRTC, and SHOULD NOT be offered.

Answerers SHOULD reject any FEC-only m-lines, unless they specifically know how to handle such a thing in a WebRTC context (perhaps defined by a future version of the WebRTC specifications).

6. FEC for Application Content

WebRTC also supports the ability to send generic application data, and provides transport-level retransmission mechanisms to support full and partial (e.g. timed) reliability. See [I-D.ietf-rtcweb-data-channel] for details.

Because the application can control exactly what data to send, it has the ability to monitor packet statistics and perform its own application-level FEC, if necessary.

As a result, this document makes no recommendations regarding FEC for the underlying data transport.

7. Implementation Requirements

To support the functionality recommended above, implementations MUST be able to receive and make use of the relevant FEC formats for their supported audio codecs, and MUST indicate this support, as described in Section 4. Use of these formats when sending, as mentioned above, is RECOMMENDED.

The general FEC mechanism described in [I-D.ietf-payload-flexible-fec-scheme] SHOULD also be supported, as mentioned in Section 5.

Implementations MAY support additional FEC mechanisms if desired, e.g., [RFC5109].

8. Adaptive Use of FEC

Because use of FEC always causes redundant data to be transmitted, and the total amount of data must remain within any bandwidth limits indicated by congestion control and the receiver, this will lead to less bandwidth available for the primary encoding, even when the redundant data is not being used. This is in contrast to methods like RTX [RFC4588] or flexfec's retransmission mode ([I-D.ietf-payload-flexible-fec-scheme], Section 1.1.7), which only transmit redundant data when necessary, at the cost of an extra roundtrip and thereby increased media latency.

Given this, WebRTC implementations SHOULD prefer using RTX or flexfec retransmissions instead of FEC when the connection RTT is within the application's latency budget, and otherwise SHOULD only transmit the amount of FEC needed to protect against the observed packet loss (which can be determined, e.g., by monitoring transmit packet loss data from RTCP Receiver Reports [RFC3550]), unless the application indicates it is willing to pay a quality penalty to proactively avoid losses.

Note that when probing bandwidth, i.e., speculatively sending extra data to determine if additional link capacity exists, FEC data SHOULD be used as the additional data. Given that extra data is going to be sent regardless, it makes sense to have that data protect the primary payload; in addition, FEC can typically be applied in a way that increases bandwidth only modestly, which is necessary when probing.

When using FEC with layered codecs, e.g., [RFC6386], where only base layer frames are critical to the decoding of future frames, implementations SHOULD only apply FEC to these base layer frames.

Finally, it should be noted that although applying redundancy is often useful in protecting a stream against packet loss, if the loss is caused by network congestion, the additional bandwidth used by the redundant data may actually make the situation worse, and can lead to significant degradation of the network.

9. Security Considerations

In the WebRTC context, FEC is specifically concerned with recovering data from lost packets; any corrupted packets will be discarded by the SRTP [RFC3711] decryption process. Therefore, as described in [RFC3711], Section 10, the default processing when using FEC with SRTP is to perform FEC followed by SRTP at the sender, and SRTP followed by FEC at the receiver. This ordering is used for all the SRTP Protection Profiles used in DTLS-SRTP [RFC5763], which are enumerated in [RFC5764], Section 4.1.2.

Additional security considerations for each individual FEC mechanism are enumerated in their respective documents.

10. IANA Considerations

This document requires no actions from IANA.

11. Acknowledgements

Several people provided significant input into this document, including Bernard Aboba, Jonathan Lennox, Giri Mandyam, Varun Singh, Tim Terriberry, Magnus Westerlund, and Mo Zanaty.

12. References

12.1. Normative References

- [I-D.ietf-payload-flexible-fec-scheme]
Zanaty, M., Singh, V., Begen, A., and G. Mandyam, "RTP Payload Format for Flexible Forward Error Correction (FEC)", draft-ietf-payload-flexible-fec-scheme-20 (work in progress), May 2019.
- [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>>.

- [RFC2198] Perkins, C., Kouvelas, I., Hodson, O., Hardman, V., Handley, M., Bolot, J., Vega-Garcia, A., and S. Fosse-Parisis, "RTP Payload for Redundant Audio Data", RFC 2198, DOI 10.17487/RFC2198, September 1997, <<https://www.rfc-editor.org/info/rfc2198>>.
- [RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, DOI 10.17487/RFC3264, June 2002, <<https://www.rfc-editor.org/info/rfc3264>>.
- [RFC4867] Sjoberg, J., Westerlund, M., Lakaniemi, A., and Q. Xie, "RTP Payload Format and File Storage Format for the Adaptive Multi-Rate (AMR) and Adaptive Multi-Rate Wideband (AMR-WB) Audio Codecs", RFC 4867, DOI 10.17487/RFC4867, April 2007, <<https://www.rfc-editor.org/info/rfc4867>>.
- [RFC5956] Begen, A., "Forward Error Correction Grouping Semantics in the Session Description Protocol", RFC 5956, DOI 10.17487/RFC5956, September 2010, <<https://www.rfc-editor.org/info/rfc5956>>.
- [RFC7587] Spittka, J., Vos, K., and JM. Valin, "RTP Payload Format for the Opus Speech and Audio Codec", RFC 7587, DOI 10.17487/RFC7587, June 2015, <<https://www.rfc-editor.org/info/rfc7587>>.
- [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>>.
- [TS.26114] 3GPP, "IP Multimedia Subsystem (IMS); Multimedia telephony; Media handling and interaction", 3GPP TS 26.114 15.0.0, September 2017.

12.2. Informative References

- [I-D.ietf-mmusic-sdp-bundle-negotiation] Holmberg, C., Alvestrand, H., and C. Jennings, "Negotiating Media Multiplexing Using the Session Description Protocol (SDP)", draft-ietf-mmusic-sdp-bundle-negotiation-54 (work in progress), December 2018.
- [I-D.ietf-rtcweb-data-channel] Jesup, R., Loreto, S., and M. Tuexen, "WebRTC Data Channels", draft-ietf-rtcweb-data-channel-13 (work in progress), January 2015.

- [OpusFEC] Terriberry, T., "Opus FEC", January 2013.
- [RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, DOI 10.17487/RFC3550, July 2003, <<https://www.rfc-editor.org/info/rfc3550>>.
- [RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, DOI 10.17487/RFC3711, March 2004, <<https://www.rfc-editor.org/info/rfc3711>>.
- [RFC4588] Rey, J., Leon, D., Miyazaki, A., Varsha, V., and R. Hakenberg, "RTP Retransmission Payload Format", RFC 4588, DOI 10.17487/RFC4588, July 2006, <<https://www.rfc-editor.org/info/rfc4588>>.
- [RFC5109] Li, A., Ed., "RTP Payload Format for Generic Forward Error Correction", RFC 5109, DOI 10.17487/RFC5109, December 2007, <<https://www.rfc-editor.org/info/rfc5109>>.
- [RFC5391] Sollaud, A., "RTP Payload Format for ITU-T Recommendation G.711.1", RFC 5391, DOI 10.17487/RFC5391, November 2008, <<https://www.rfc-editor.org/info/rfc5391>>.
- [RFC5763] Fischl, J., Tschofenig, H., and E. Rescorla, "Framework for Establishing a Secure Real-time Transport Protocol (SRTP) Security Context Using Datagram Transport Layer Security (DTLS)", RFC 5763, DOI 10.17487/RFC5763, May 2010, <<https://www.rfc-editor.org/info/rfc5763>>.
- [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>>.
- [RFC6386] Bankoski, J., Koleszar, J., Quillio, L., Salonen, J., Wilkins, P., and Y. Xu, "VP8 Data Format and Decoding Guide", RFC 6386, DOI 10.17487/RFC6386, November 2011, <<https://www.rfc-editor.org/info/rfc6386>>.
- [RFC6464] Lennox, J., Ed., Ivov, E., and E. Marocco, "A Real-time Transport Protocol (RTP) Header Extension for Client-to-Mixer Audio Level Indication", RFC 6464, DOI 10.17487/RFC6464, December 2011, <<https://www.rfc-editor.org/info/rfc6464>>.

[RFC6465] Ivov, E., Ed., Marocco, E., Ed., and J. Lennox, "A Real-time Transport Protocol (RTP) Header Extension for Mixer-to-Client Audio Level Indication", RFC 6465, DOI 10.17487/RFC6465, December 2011, <<https://www.rfc-editor.org/info/rfc6465>>.

[RFC6716] Valin, JM., Vos, K., and T. Terriberry, "Definition of the Opus Audio Codec", RFC 6716, DOI 10.17487/RFC6716, September 2012, <<https://www.rfc-editor.org/info/rfc6716>>.

Appendix A. Change log

Changes in draft -10:

- o Additional editorial changes from IETF LC.

Changes in draft -09:

- o Editorial changes from IETF LC.
- o Added new reference for Opus FEC.

Changes in draft -08:

- o Switch to RFC 8174 boilerplate.

Changes in draft -07:

- o Clarify how bandwidth management interacts with FEC.
- o Make 3GPP reference normative.

Changes in draft -06:

- o Discuss how multiple streams can be protected by a single FlexFEC stream.
- o Discuss FEC for bandwidth probing.
- o Add note about recovery of RTP headers and header extensions.
- o Add note about FEC/SRTP ordering.
- o Clarify flexfec demux text, and mention retransmits.
- o Clarify text regarding offers/answers.
- o Make RFC2198 support SHOULD strength.

- o Clean up references.

Changes in draft -05:

- o No changes.

Changes in draft -04:

- o Discussion of layered codecs.
- o Discussion of RTX.
- o Clarified implementation requirements.
- o FlexFEC MUST -> SHOULD.
- o Clarified AMR max-red handling.
- o Updated references.

Changes in draft -03:

- o Added overhead stats for Opus.
- o Expanded discussion of multi-packet FEC for Opus.
- o Added discussion of AMR/AMR-WB.
- o Removed discussion of ssrc-group.
- o Referenced the data channel doc.
- o Referenced the RTP/RTCP RFC.
- o Several small edits based on feedback from Magnus.

Changes in draft -02:

- o Expanded discussion of FEC-only m-lines, and how they should be handled in offers and answers.

Changes in draft -01:

- o Tweaked abstract/intro text that was ambiguously normative.
- o Removed text on FEC for Opus in CELT mode.

- o Changed RFC 2198 recommendation for PCMU to be MAY instead of NOT RECOMMENDED, based on list feedback.
- o Explicitly called out application data as something not addressed in this document.
- o Updated flexible-fec reference.

Changes in draft -00:

- o Initial version, from sidebar conversation at IETF 90.

Author's Address

Justin Uberti
Google
747 6th St S
Kirkland, WA 98033
USA

Email: justin@uberti.name

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 3, 2020

J. Uberti
Google
July 2, 2019

WebRTC IP Address Handling Requirements
draft-ietf-rtcweb-ip-handling-12

Abstract

This document provides information and requirements for how IP addresses should be handled by WebRTC implementations.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 3, 2020.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Terminology	2
3. Problem Statement	2
4. Goals	4
5. Detailed Design	5
5.1. Principles	5
5.2. Modes and Recommendations	5
6. Implementation Guidance	7
6.1. Ensuring Normal Routing	7
6.2. Determining Associated Local Addresses	7
7. Application Guidance	8
8. Security Considerations	8
9. IANA Considerations	8
10. Acknowledgements	8
11. References	8
11.1. Normative References	8
11.2. Informative References	9
Appendix A. Change log	10
Author's Address	12

1. Introduction

One of WebRTC's key features is its support of peer-to-peer connections. However, when establishing such a connection, which involves connection attempts from various IP addresses, WebRTC may allow a web application to learn additional information about the user compared to an application that only uses the Hypertext Transfer Protocol (HTTP) [RFC7230]. This may be problematic in certain cases. This document summarizes the concerns, and makes recommendations on how WebRTC implementations should best handle the tradeoff between privacy and media performance.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119][RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Problem Statement

In order to establish a peer-to-peer connection, WebRTC implementations use Interactive Connectivity Establishment (ICE) [RFC8445], which attempts to discover multiple IP addresses using techniques such as Session Traversal Utilities for NAT (STUN)

[RFC5389] and Traversal Using Relays around NAT (TURN) [RFC5766], and then checks the connectivity of each local-address-remote-address pair in order to select the best one. The addresses that are collected usually consist of an endpoint's private physical or virtual addresses and its public Internet addresses.

These addresses are provided to the web application so that they can be communicated to the remote endpoint for its checks. This allows the application to learn more about the local network configuration than it would from a typical HTTP scenario, in which the web server would only see a single public Internet address, i.e., the address from which the HTTP request was sent.

The information revealed falls into three categories:

1. If the client is multihomed, additional public IP addresses for the client can be learned. In particular, if the client tries to hide its physical location through a Virtual Private Network (VPN), and the VPN and local OS support routing over multiple interfaces (a "split-tunnel" VPN), WebRTC can discover not only the public address for the VPN, but also the ISP public address over which the VPN is running.
2. If the client is behind a Network Address Translator (NAT), the client's private IP addresses, often [RFC1918] addresses, can be learned.
3. If the client is behind a proxy (a client-configured "classical application proxy", as defined in [RFC1919], Section 3), but direct access to the Internet is permitted, WebRTC's STUN checks will bypass the proxy and reveal the public IP address of the client. This concern also applies to the "enterprise TURN server" scenario described in [RFC7478], Section 2.3.5.1, if, as above, direct Internet access is permitted. However, when the term "proxy" is used in this document, it is always in reference to an [RFC1919] proxy server.

Of these three concerns, the first is the most significant, because for some users, the purpose of using a VPN is for anonymity. However, different VPN users will have different needs, and some VPN users (e.g., corporate VPN users) may in fact prefer WebRTC to send media traffic directly, i.e., not through the VPN.

The second concern is less significant but valid nonetheless. The core issue is that web applications can learn about addresses that are not exposed to the internet; typically these addresses are IPv4, but they can also be IPv6, as in the case of NAT64 [RFC6146]. While disclosure of the [RFC4941] IPv6 addresses recommended by

[I-D.ietf-rtcweb-transports] is fairly benign due to their intentionally short lifetimes, IPv4 addresses present some challenges. Although private IPv4 addresses often contain minimal entropy (e.g., 192.168.0.2, a fairly common address), in the worst case, they can contain 24 bits of entropy with an indefinite lifetime. As such, they can be a fairly significant fingerprinting surface. In addition, intranet web sites can be attacked more easily when their IPv4 address range is externally known.

Private IP addresses can also act as an identifier that allows web applications running in isolated browsing contexts (e.g., normal and private browsing) to learn that they are running on the same device. This could allow the application sessions to be correlated, defeating some of the privacy protections provided by isolation. It should be noted that private addresses are just one potential mechanism for this correlation and this is an area for further study.

The third concern is the least common, as proxy administrators can already control this behavior through organizational firewall policy, and generally, forcing WebRTC traffic through a proxy server will have negative effects on both the proxy and on media quality.

Note also that these concerns predate WebRTC; Adobe Flash Player has provided similar functionality since the introduction of Real-Time Media Flow Protocol (RTMFP) support [RFC7016] in 2008.

4. Goals

WebRTC's support of secure peer-to-peer connections facilitates deployment of decentralized systems, which can have privacy benefits. As a result, blunt solutions that disable WebRTC or make it significantly harder to use are undesirable. This document takes a more nuanced approach, with the following goals:

- o Provide a framework for understanding the problem so that controls might be provided to make different tradeoffs regarding performance and privacy concerns with WebRTC.
- o Using that framework, define settings that enable peer-to-peer communications, each with a different balance between performance and privacy.
- o Finally, provide recommendations for default settings that provide reasonable performance without also exposing addressing information in a way that might violate user expectations.

5. Detailed Design

5.1. Principles

The key principles for our framework are stated below:

1. By default, WebRTC traffic should follow typical IP routing, i.e., WebRTC should use the same interface used for HTTP traffic, and only the system's 'typical' public addresses (or those of an enterprise TURN server, if present) should be visible to the application. However, in the interest of optimal media quality, it should be possible to enable WebRTC to make use of all network interfaces to determine the ideal route.
2. By default, WebRTC should be able to negotiate direct peer-to-peer connections between endpoints (i.e., without traversing a NAT or relay server) when such connections are possible. This ensures that applications that need true peer-to-peer routing for bandwidth or latency reasons can operate successfully.
3. It should be possible to configure WebRTC to not disclose private local IP addresses, to avoid the issues associated with web applications learning such addresses. This document does not require this to be the default state, as there is no currently defined mechanism that can satisfy this requirement as well as the aforementioned requirement to allow direct peer-to-peer connections.
4. By default, WebRTC traffic should not be sent through proxy servers, due to the media quality problems associated with sending WebRTC traffic over TCP, which is almost always used when communicating with such proxies, as well as proxy performance issues that may result from proxying WebRTC's long-lived, high-bandwidth connections. However, it should be possible to force WebRTC to send its traffic through a configured proxy if desired.

5.2. Modes and Recommendations

Based on these ideas, we define four specific modes of WebRTC behavior, reflecting different media quality/privacy tradeoffs:

Mode 1: Enumerate all addresses: WebRTC MUST use all network interfaces to attempt communication with STUN servers, TURN servers, or peers. This will converge on the best media path, and is ideal when media performance is the highest priority, but it discloses the most information.

- Mode 2: Default route + associated local addresses: WebRTC MUST follow the kernel routing table rules, which will typically cause media packets to take the same route as the application's HTTP traffic. If an enterprise TURN server is present, the preferred route MUST be through this TURN server. Once an interface has been chosen, the private IPv4 and IPv6 addresses associated with this interface MUST be discovered and provided to the application as host candidates. This ensures that direct connections can still be established in this mode.
- Mode 3: Default route only: This is the the same as Mode 2, except that the associated private addresses MUST NOT be provided; the only IP addresses gathered are those discovered via mechanisms like STUN and TURN (on the default route). This may cause traffic to hairpin through a NAT, fall back to an application TURN server, or fail altogether, with resulting quality implications.
- Mode 4: Force proxy: This is the same as Mode 3, but when the application's HTTP traffic is sent through a proxy, WebRTC media traffic MUST also be proxied. If the proxy does not support UDP (as is the case for all HTTP and most SOCKS [RFC1928] proxies), or the WebRTC implementation does not support UDP proxying, the use of UDP will be disabled, and TCP will be used to send and receive media through the proxy. Use of TCP will result in reduced media quality, in addition to any performance considerations associated with sending all WebRTC media through the proxy server.

Mode 1 MUST NOT be used unless user consent has been provided. The details of this consent are left to the implementation; one potential mechanism is to tie this consent to `getUserMedia` (device permissions) consent, described in [I-D.ietf-rtcweb-security-arch], Section 6.2. Alternatively, implementations can provide a specific mechanism to obtain user consent.

In cases where user consent has not been obtained, Mode 2 SHOULD be used.

These defaults provide a reasonable tradeoff that permits trusted WebRTC applications to achieve optimal network performance, but gives applications without consent (e.g., 1-way streaming or data channel applications) only the minimum information needed to achieve direct connections, as defined in Mode 2. However, implementations MAY choose stricter modes if desired, e.g., if a user indicates they want all WebRTC traffic to follow the default route.

Future documents may define additional modes and/or update the recommended default modes.

Note that the suggested defaults can still be used even for organizations that want all external WebRTC traffic to traverse a proxy or enterprise TURN server, simply by setting an organizational firewall policy that allows WebRTC traffic to only leave through the proxy or TURN server. This provides a way to ensure the proxy or TURN server is used for any external traffic, but still allows direct connections (and, in the proxy case, avoids the performance issues associated with forcing media through said proxy) for intra-organization traffic.

6. Implementation Guidance

This section provides guidance to WebRTC implementations on how to implement the policies described above.

6.1. Ensuring Normal Routing

When trying to follow typical IP routing, as required by Modes 2 and 3, the simplest approach is to bind() the sockets used for peer-to-peer connections to the wildcard addresses (0.0.0.0 for IPv4, :: for IPv6), which allows the OS to route WebRTC traffic the same way as it would HTTP traffic. STUN and TURN will work as usual, and host candidates can still be determined as mentioned below.

6.2. Determining Associated Local Addresses

When binding to a wildcard address, some extra work is needed to determine the associated local address required by Mode 2, which we define as the source address that would be used for any packets sent to the web application host (assuming that UDP and TCP get the same routing treatment). Use of the web application host as a destination ensures the right source address is selected, regardless of where the application resides (e.g., on an intranet).

First, the appropriate remote IPv4/IPv6 address is obtained by resolving the host component of the web application URI [RFC3986]. If the client is behind a proxy and cannot resolve these IPs via DNS, the address of the proxy can be used instead. Or, if the web application was loaded from a file:// URI [RFC8089], rather than over the network, the implementation can fall back to a well-known DNS name or IP address.

Once a suitable remote IP has been determined, the implementation can create a UDP socket, bind() it to the appropriate wildcard address, and then connect() to the remote IP. Generally, this results in the

socket being assigned a local address based on the kernel routing table, without sending any packets over the network.

Finally, the socket can be queried using `getsockname()` or the equivalent to determine the appropriate local address.

7. Application Guidance

The recommendations mentioned in this document may cause certain WebRTC applications to malfunction. In order to be robust in all scenarios, the following guidelines are provided for applications:

- o Applications SHOULD deploy a TURN server with support for both UDP and TCP connections to the server. This ensures that connectivity can still be established, even when Mode 3 or 4 are in use, assuming the TURN server can be reached.
- o Applications SHOULD detect when they don't have access to the full set of ICE candidates by checking for the presence of host candidates. If no host candidates are present, Mode 3 or 4 above is in use; this knowledge can be useful for diagnostic purposes.

8. Security Considerations

This document describes several potential privacy and security concerns associated with WebRTC peer-to-peer connections, and provides mechanisms and recommendations for WebRTC implementations to address these concerns.

9. IANA Considerations

This document requires no actions from IANA.

10. Acknowledgements

Several people provided input into this document, including Bernard Aboba, Harald Alvestrand, Youenn Fablet, Ted Hardie, Matthew Kaufmann, Eric Rescorla, Adam Roach, and Martin Thomson.

11. References

11.1. Normative References

- [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>>.

11.2. Informative References

- [I-D.ietf-rtcweb-security-arch]
Rescorla, E., "WebRTC Security Architecture", draft-ietf-rtcweb-security-arch-18 (work in progress), February 2019.
- [I-D.ietf-rtcweb-transport]
Alvestrand, H., "Transports for WebRTC", draft-ietf-rtcweb-transport-17 (work in progress), October 2016.
- [RFC1918] Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G., 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>>.

Appendix A. Change log

Changes in draft -12:

- o Editorial updates from IETF LC review.

Changes in draft -11:

- o Editorial updates from AD review.

Changes in draft -10:

- o Incorporate feedback from IETF 102 on the problem space.
- o Note that future versions of the document may define new modes.

Changes in draft -09:

- o Fixed confusing text regarding enterprise TURN servers.

Changes in draft -08:

- o Discuss how enterprise TURN servers should be handled.

Changes in draft -07:

- o Clarify consent guidance.

Changes in draft -06:

- o Clarify recommendations.
- o Split implementation guidance into two sections.

Changes in draft -05:

- o Separated framework definition from implementation techniques.
- o Removed RETURN references.
- o Use origin when determining local IPs, rather than a well-known IP.

Changes in draft -04:

- o Rewording and cleanup in abstract, intro, and problem statement.
- o Added 2119 boilerplate.
- o Fixed weird reference spacing.
- o Expanded acronyms on first use.
- o Removed 8.8.8.8 mention.
- o Removed mention of future browser considerations.

Changes in draft -03:

- o Clarified when to use which modes.
- o Added 2119 qualifiers to make normative statements.
- o Defined 'proxy'.
- o Mentioned split tunnels in problem statement.

Changes in draft -02:

- o Recommendations -> Requirements
- o Updated text regarding consent.

Changes in draft -01:

- o Incorporated feedback from Adam Roach; changes to discussion of cam/mic permission, as well as use of proxies, and various editorial changes.
- o Added several more references.

Changes in draft -00:

- o Published as WG draft.

Author's Address

Justin Uberti
Google
747 6th St S
Kirkland, WA 98033
USA

Email: justin@uberti.name

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 28, 2017

B. Schwartz
J. Uberti
Google
March 27, 2017

Recursively Encapsulated TURN (RETURN) for Connectivity and Privacy in
WebRTC
draft-ietf-rtcweb-return-02

Abstract

In the context of WebRTC, the concept of a local TURN proxy has been suggested, but not reviewed in detail. WebRTC applications are already using TURN to enhance connectivity and privacy. This document explains how local TURN proxies and WebRTC applications can work together.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 28, 2017.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Visual Overview of RETURN	3
3. Goals	8
3.1. Connectivity	8
3.2. Independent Path Control	9
4. Concepts	9
4.1. Proxy	9
4.2. Virtual interface	10
4.3. Proxy configuration leakiness	10
4.4. Sealed proxy rank	10
5. Requirements	11
5.1. ICE candidates produced in the presence of a proxy	11
5.2. Leaky proxy configuration	11
5.3. Sealed proxy configuration	11
5.4. Proxy rank	11
5.5. Multiple physical interfaces	12
5.6. IPv4 and IPv6	12
5.7. Unspecified leakiness	12
5.8. Interaction with SOCKS5-UDP	12
5.9. Encapsulation overhead, fragmentation, and Path MTU	13
5.10. Interaction with alternate TURN server fallback	13
5.11. Reusing the same TURN server	13
6. Examples	14
6.1. Firewalled enterprise network with a basic application	14
6.2. Conflicting proxies configured by Auto-Discovery and local policy	15
7. Security Considerations	16
8. IANA Considerations	16
9. Acknowledgements	16
10. References	17
10.1. Normative References	17
10.2. Informative References	17
Authors' Addresses	18

1. Introduction

TURN [RFC5766] is a protocol for communication between a client and a TURN server, in order to route UDP traffic to and from one or more peers. As noted in [RFC5766], the TURN relay server "typically sits in the public Internet". In a WebRTC context, if a TURN server is to be used, it is typically provided by the application, either to provide connectivity between users whose NATs would otherwise prevent

it, or to obscure the identity of the participants by concealing their IP addresses from one another.

In many enterprises, direct UDP transmissions are not permitted between clients on the internal networks and external IP addresses, so media must flow over TCP. To enable WebRTC services in such a situation, clients must use TURN-TCP, or TURN-TLS. These configurations are not ideal: they send all traffic over TCP, which leads to higher latency than would otherwise be necessary, and they force the application provider to operate a TURN server because WebRTC endpoints behind NAT cannot typically act as TCP servers. These configurations may result in especially bad behaviors when operating through TCP or HTTP proxies that were not designed to carry real-time media streams.

To avoid forcing WebRTC media streams through a TCP stage, enterprise network operators may operate a TURN server for their network, which can be discovered by clients using TURN Auto-Discovery [I-D.ietf-tram-turn-server-discovery], or through a proprietary mechanism. This TURN server may be placed inside the network, with a firewall configuration allowing it to communicate with the public internet, or it may be operated by a third party outside the network, with a firewall configuration that allows hosts inside the network to communicate with it. Use of the specified TURN server may be the only way for clients on the network to achieve a high quality WebRTC experience. This scenario is required to be supported by the WebRTC requirements document [RFC7478] Section 2.3.5.1.

When the application intends to use a TURN server for identity cloaking, and the enterprise network administrator intends to use a TURN server for connectivity, there is a conflict. In current WebRTC implementations, TURN can only be used on a single-hop basis in each candidate, but using only the enterprise's TURN server reveals information about the user (e.g. organizational affiliation), and using only the application's TURN server may be blocked by the network administrator, or may require using TURN-TCP or TURN-TLS, resulting in a significant sacrifice in latency.

To resolve this conflict, we introduce Recursively Encapsulated TURN, a procedure that allows a WebRTC endpoint to route traffic through multiple TURN servers, and get improved connectivity and privacy in return.

2. Visual Overview of RETURN

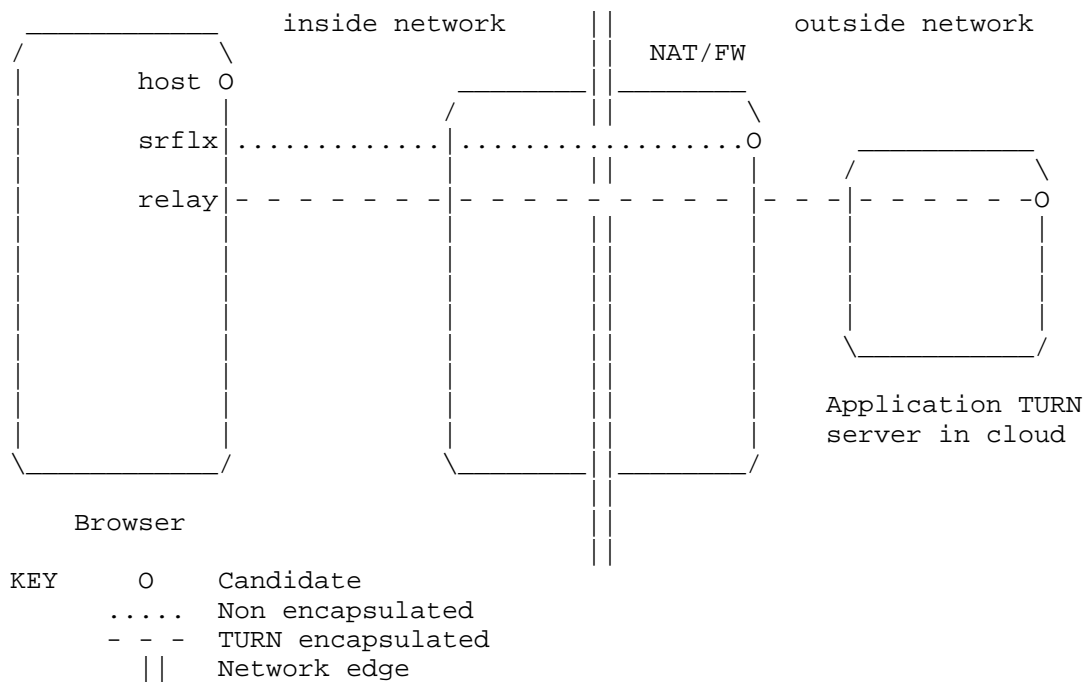


Figure 1: Basic WebRTC ICE Candidates with TURN Server

Figure 1 shows a browser located inside a home or enterprise network which connects to the Internet through a Network Address Translator and Firewall (NAT/FW). A TURN server in the Internet cloud is also shown, which is provided by the WebRTC application via the JavaScript RTCIceServers object.

A WebRTC application can use a TURN server to provide NAT traversal, but also to provide privacy, routing optimizations, logging, or possibly other functionality. The application can accomplish this by forcing all traffic to flow through the TURN server using the JavaScript RTCIceTransportPolicy object [I-D.ietf-rtcweb-jsep]. Since this TURN server is injected by the application, we will refer to it as an Application TURN server.

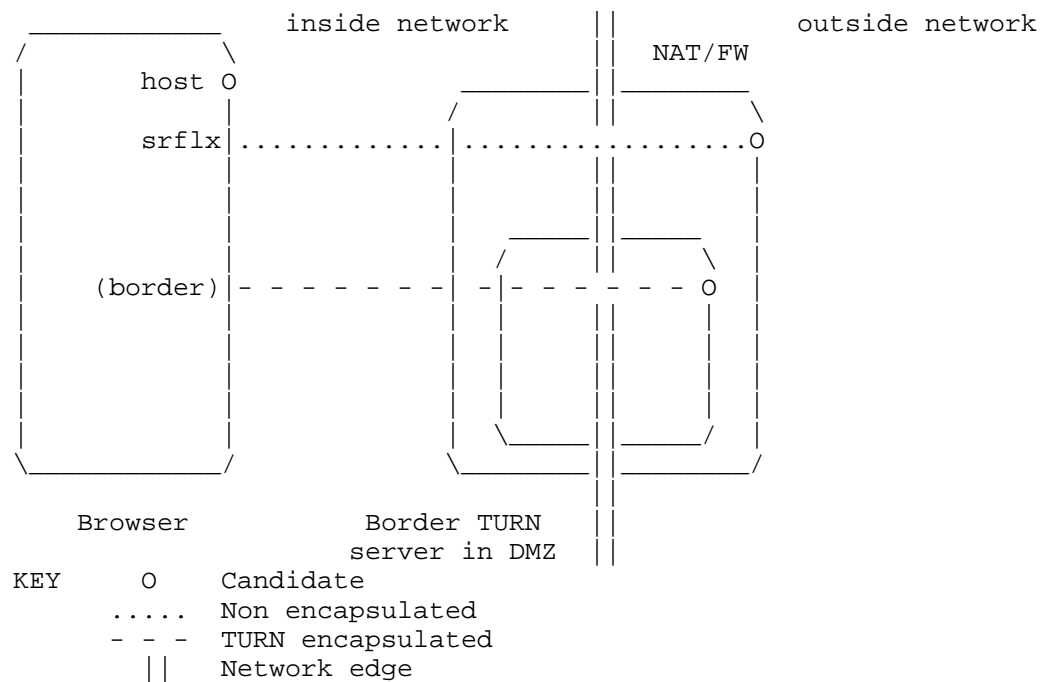


Figure 2: WebRTC ICE Candidates with DMZ TURN Server

Figure 2 shows a TURN server co-resident with the NAT/FW, i.e. in the DMZ of the FW. This TURN server might be used by an enterprise, ISP, or home network to enable WebRTC media flows that would otherwise be blocked by the firewall, or to improve quality of service on flows that pass through this TURN server. This TURN server is not part of a particular application, and is managed as part of the border control system, so we call it a Border TURN Server.

Figure 2 shows the port allocated on this TURN server as "(border)", not any particular candidate type, to distinguish it from the other ports, which have been represented as ICE candidates in accordance with the WebRTC specifications. This case is different, because unlike an Application TURN server, there is not yet any specification for how WebRTC should interact with a Border TURN server. Under what conditions should WebRTC allocate a port on a Border TURN server? How should WebRTC represent that port as an ICE candidate? This draft serves to answer these two questions.

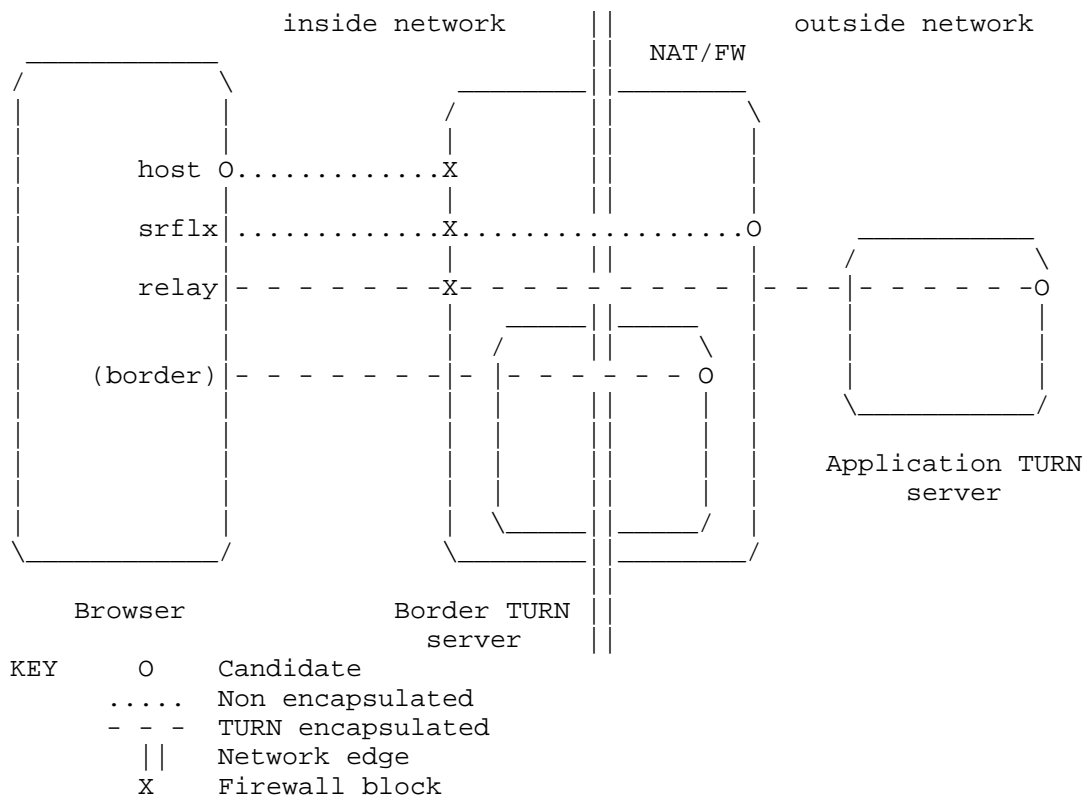


Figure 3: WebRTC ICE Candidates with Application and Border TURN Servers

In Figure 3, there is both an Application TURN server and a Border TURN server. The Firewall is blocking UDP traffic except for UDP traffic to/from the Border TURN server, so only the "(border)" port allocation will work. However, there is no specified way for WebRTC to use this port as a candidate. Moreover, this port on its own would not be sufficient to satisfy the user's needs. Both TURN servers provide important functionality, so we need a way for WebRTC to select a candidate that uses both TURN servers.

The solution proposed in this draft is for the browser to implement RETURN, which provides a candidate that traverses both TURN servers, as shown in Figure 4.

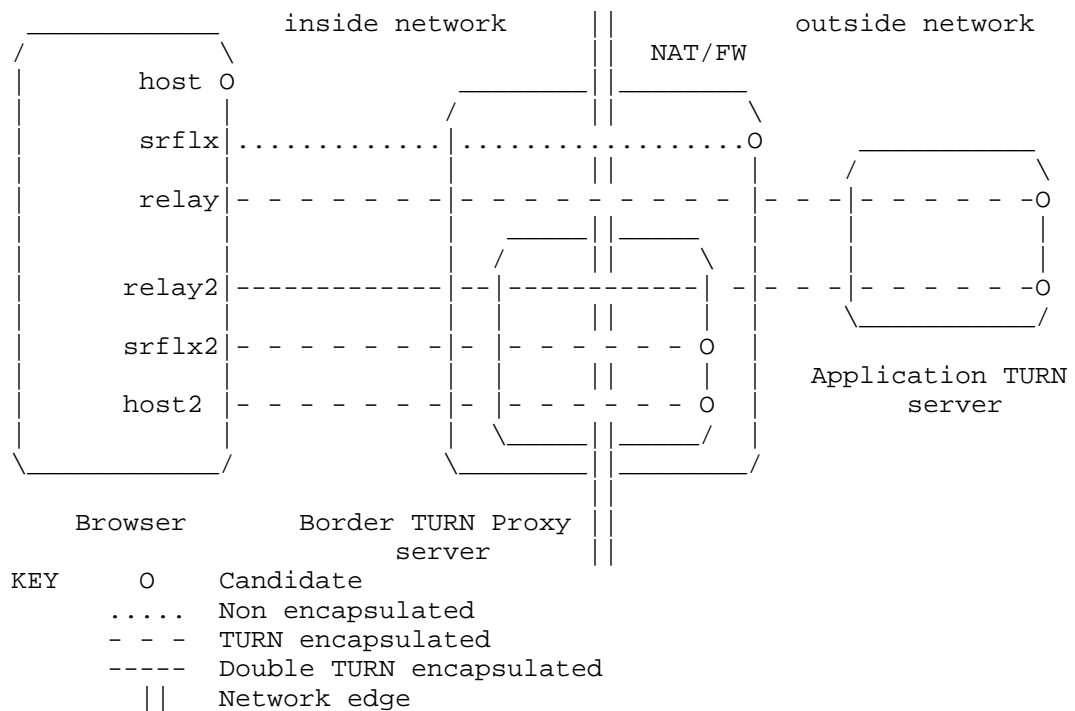


Figure 4: WebRTC ICE Candidates with Application TURN and Border TURN Proxy Servers

The Browser in Figure 4 implements RETURN, so it allocates a port on the Border TURN server, now referred to as a Border TURN Proxy by analogy to an HTTP CONNECT or SOCKS Proxy (see Figure 5), and then runs STUN and TURN over this allocation, resulting in three candidates: relay2, srflx2, and host2. The relay2 candidate causes traffic to flow through both TURN servers by encapsulating TURN within TURN - hence the name Recursively Encapsulated TURN (RETURN).

The host2 and srflx2 candidates are probably identical, so one will be dropped by ICE. If the NAT/FW blocks UDP and the application uses only relay candidates, then the relay2 candidate will be selected. Otherwise, the other candidates will be used, in accordance with the usual ICE procedure.

Only the browser needs to implement the RETURN behavior - both the Border TURN Proxy and Application TURN servers' TURN protocol usage is unchanged.

Note that this arrangement preserves the end-to-end security and privacy features of WebRTC media flows. The ability to steer the

media flows through multiple TURN servers while still allowing end-to-end encryption and authentication is a key benefit of RETURN.

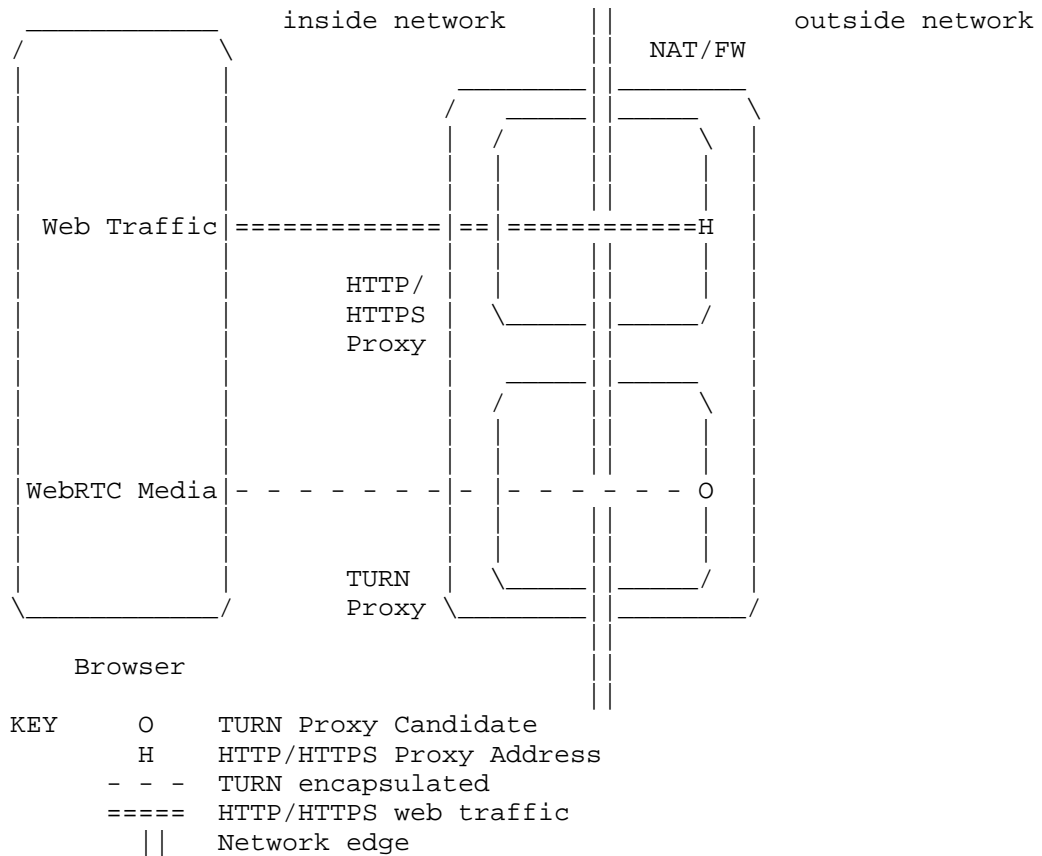


Figure 5: Similarity between HTTP/HTTPS Proxy and TURN Proxy

3. Goals

These goals are requirements on this document (not on implementations of the specification).

3.1. Connectivity

As noted in [RFC7478] Section 2.3.5.1 and requirement F20, a WebRTC browser endpoint MUST be able to direct UDP connections through a designated TURN server configured by enterprise policy (a "proxy").

It MUST be possible to configure a WebRTC endpoint that supports proxies to achieve connectivity no worse than if the endpoint were operating at the proxy's address.

For efficiency, network administrators SHOULD be able to prevent browsers from attempting to send traffic through routes that are already known to be blocked.

3.2. Independent Path Control

Both network administrators and application developers may wish to direct all their UDP flows through a particular TURN server. There are many goals that might motivate such a choice, including

- o improving quality of service by tunneling packets through a network that is faster than the public internet,
- o monitoring the usage of UDP services,
- o troubleshooting and debugging problematic services,
- o logging connection metadata for legal or auditing reasons,
- o recording the entire contents of all connections, or
- o providing partial IP address anonymization (as described in [I-D.ietf-rtcweb-security] Section 4.2.4 and [I-D.ietf-rtcweb-security-arch] Section 5.4).

4. Concepts

To achieve our goals, we introduce the following new concepts:

4.1. Proxy

In this document a "proxy" is any TURN server that was provided by any mechanism other than through the standard WebRTC-application ICE candidate provisioning API [I-D.ietf-rtcweb-jsep]. We call it a "proxy" by analogy with SOCKS proxies and similar network services, because it performs a similar function and can be configured in a similar fashion.

If a proxy is to be used, it will be the destination of traffic generated by the client. (There is no analogue to the transparent/intercepting HTTP proxy configuration, which modifies traffic at the network layer.) Mechanisms to configure a proxy include Auto-Discovery [I-D.ietf-tram-turn-server-discovery] and local policy ([I-D.ietf-rtcweb-jsep] Section 3.5.3, "ICE candidate policy").

In an application context, a proxy may be "active" (producing candidates) or "inactive" (not in use, having no effect on the context).

4.2. Virtual interface

A typical WebRTC browser endpoint may have multiple network interfaces available, such as wired ethernet, wireless ethernet, and WAN. In this document, a "virtual interface" is a procedure for generating ICE candidates that are not simply generated by a particular physical interface. A virtual interface can produce "host", "server-reflexive", and "relay" candidates, but may be restricted to only some type of candidate (e.g. UDP-only).

4.3. Proxy configuration leakiness

"Leakiness" is an attribute of a proxy configuration. This document defines two values for the "leakiness" of a proxy configuration: "leaky" and "sealed". Proxy configuration, including leakiness, may be set by local policy ([I-D.ietf-rtcweb-jsep], "ICE candidate policy") or other mechanisms.

A leaky configuration adds a proxy and also allows the browser to use routes that transit directly via the endpoint's physical interfaces (not through the proxy). In a leaky configuration, setting a proxy augments the available set of ICE candidates. Multiple leaky-configuration proxies may therefore be active simultaneously.

A sealed proxy configuration requires the browser to route all WebRTC traffic through the proxy, eliminating all ICE candidates that do not go through the proxy. Only one sealed proxy may be active at a time.

Leaky proxy configurations allow more efficient routes to be selected. For example, two peers on the same LAN can connect directly (peer to peer) if a leaky proxy is enabled, but must "hairpin" through the TURN proxy if the configuration is sealed. However, sealed proxy configurations can be faster to connect, especially if many of the peer-to-peer routes that ICE will try first are blocked by the network's firewall policies.

4.4. Sealed proxy rank

In some configurations, an endpoint may be subject to multiple sealed proxy settings at the same time. In that case, one of those settings will have highest rank, and it will be the active proxy. In a given application context (e.g. a webpage), there is at most one active sealed proxy. This document does not specify a representation for rank.

5. Requirements

5.1. ICE candidates produced in the presence of a proxy

When a proxy is configured, by Auto-Discovery or a proprietary means, the browser MUST NOT report a "relay" candidate representing the proxy. Instead, the browser MUST connect to the proxy and then, if the connection is successful, treat the TURN tunnel as a UDP-only virtual interface.

For a virtual interface representing a TURN proxy, this means that the browser MUST report the public-facing IP address and port acquired through TURN as a "host" candidate, the browser MUST perform STUN through the TURN proxy (if STUN is configured), and it MUST perform TURN by recursive encapsulation through the TURN proxy, resulting in TURN candidates whose "raddr" and "rport" attributes match the acquired public-facing IP address and port on the proxy.

Because the virtual interface has some additional overhead due to indirection, it SHOULD have lower priority than the physical interfaces if physical interfaces are also active. Specifically, even host candidates generated by a virtual interface SHOULD have priority 0 when physical interfaces are active (similar to [RFC5245] Section 4.1.2.2, "the local preference for host candidates from a VPN interface SHOULD have a priority of 0").

5.2. Leaky proxy configuration

If the active proxy for an application is leaky, the browser should undertake the standard ICE candidate discovery mechanism [RFC5245] on the available physical and virtual interfaces.

5.3. Sealed proxy configuration

If the active proxy for an application is sealed, the browser MUST NOT gather or produce any candidates on physical interfaces. The WebRTC implementation MUST direct its traffic from those interfaces only to the proxy, and perform ICE candidate discovery only on the single virtual interface representing the active proxy.

5.4. Proxy rank

Any browser mechanism for specifying a proxy SHOULD allow the caller to indicate a higher rank than the proxy provided by Auto-Discovery [I-D.ietf-tram-turn-server-discovery].

5.5. Multiple physical interfaces

Some operating systems allow the browser to use multiple interfaces to contact a single remote IP address. To avoid producing an excessive number of candidates, WebRTC endpoints **MUST NOT** use multiple physical interfaces to connect to a single proxy simultaneously. (If this were violated, it could produce a number of virtual interfaces equal to the product of the number of physical interfaces and the number of active proxies.)

Mechanisms for configuring a RETURN proxy **SHOULD** allow configuring a proxy that only applies to connections made from a single physical interface. This is useful to optimize efficiency in modes 2 and 3 of [I-D.ietf-rtcweb-ip-handling].

5.6. IPv4 and IPv6

A proxy **MAY** have both an IPv4 and an IPv6 address (e.g. if the proxy is specified by DNS and has both A and AAAA records). The client **MAY** try both of these addresses, but **MUST** select one, preferring IPv6, before allocating any remote addresses. This corresponds to the the Happy Eyeballs [RFC6555] procedure for dual-stack clients.

A proxy **MAY** provide both IPv4 and IPv6 remote addresses to clients [RFC6156]. A client **SHOULD** request both address families. If both requests are granted, the client **SHOULD** treat the two addresses as host candidates on a dual-stack virtual interface.

5.7. Unspecified leakiness

If a proxy configuration mechanism does not specify leakiness, browsers **SHOULD** treat the proxy as leaky. This is similar to current WebRTC implementations' behavior in the presence of SOCKS and HTTP proxies: the candidate allocation code continues to generate UDP candidates that do not transit through the proxy.

5.8. Interaction with SOCKS5-UDP

The SOCKS5 proxy standard [RFC1928] permits compliant SOCKS proxies to support UDP traffic. However, most implementations of SOCKS5 today do not support UDP. Accordingly, WebRTC browsers **MUST** by default (i.e. unless deliberately configured otherwise) treat SOCKS5 proxies as leaky and having lower rank than any configured TURN proxies.

5.9. Encapsulation overhead, fragmentation, and Path MTU

Encapsulating a link in TURN adds overhead on the path between the client and the TURN server, because each packet must be wrapped in a TURN message. This overhead is sometimes doubled in RETURN proxying. To avoid excessive overhead, client implementations SHOULD use ChannelBind and ChannelData messages to connect and send data through proxies and application TURN servers when possible. Clients MAY buffer messages to be sent until the ChannelBind command completes (requiring one round trip to the proxy), or they MAY use CreatePermission and Send messages for the first few packets to reduce startup latency at the cost of higher overhead.

Adding overhead to packets on a link decreases the effective Maximum Transmissible Unit on that link. Accordingly, clients that support proxying MUST NOT rely on the effective MTU complying with the Internet Protocol's minimum MTU requirement.

ChannelData messages have constant overhead, enabling consistent effective PMTU, but Send messages do not necessarily have constant overhead. TURN messages may be fragmented and reassembled if they are not marked with the Don't Fragment (DF) IP bit or the DONT-FRAGMENT TURN attribute. Client implementors should keep this in mind, especially if they choose to implement PMTU discovery through the proxy.

5.10. Interaction with alternate TURN server fallback

As per [RFC5766], a TURN server MAY respond to an Allocate request with an error code of 300 and an ALTERNATE-SERVER indication. When connecting to proxies or application TURN servers, clients SHOULD attempt to connect to the specified alternate server in accordance with [RFC5766]. The client MUST route a connection to the alternate server through the proxy if and only if the original connection attempt was routed through the proxy.

5.11. Reusing the same TURN server

It is possible that the same TURN server may appear more than once in the network path. For example, if both endpoints configure the same sealed proxy, then each peer will only provide candidates on this proxy. This is not a problem, and will work as expected.

It is also possible that the same TURN server could be used by both the enterprise and the application. It might appear attractive to connect to this server only once, rather than connecting to it through itself, in order to avoid imposing unnecessary server load. However,

a RETURN client MUST connect to the server twice, even when this appears redundant, to ensure correct session attribution.

For example, consider a TURN service operator that issues different authentication credentials to different customers, and then allows each customer to observe the source and destination IP addresses used with their credentials. Suppose the application and enterprise both have accounts on this service: the application uses it to prevent the enterprise from learning its peers' IP addresses, and the enterprise uses it to prevent the application from learning its employees' IP addresses. If the client only connects to the service once, then either the enterprise or the application will learn IP address information (via the TURN provider's metadata reporting) that was meant to be kept secret.

As a result of this requirement, it is possible for the same TURN server to appear up to four times in a RETURN network path: once as each peer's application's TURN server, and once as each peer's sealed proxy.

6. Examples

6.1. Firewalled enterprise network with a basic application

In this example, an enterprise network is configured with a firewall that blocks all UDP traffic, and a TURN server is advertised for Auto-Discovery in accordance with [I-D.ietf-tram-turn-server-discovery]. The proxy leakiness of the TURN server is unspecified, so the browser treats it as leaky.

The application specifies a STUN and TURN server on the public net. In accordance with the ICE candidate gathering algorithm RFC 5245 [RFC5245], it receives a set of candidates like:

1. A host candidate acquired from one interface.
 - * e.g. candidate:1610808681 1 udp 2122194687 [internal ip addr for interface 0] 63555 typ host generation 0
2. A host candidate acquired from a different interface.
 - * e.g. candidate:1610808681 1 udp 2122194687 [internal ip addr for interface 1] 54253 typ host generation 0
3. The proxy, as a host candidate.
 - * e.g. candidate:3458234523 1 udp 24584191 [public ip addr for the proxy] 54606 typ host generation 0

4. The virtual interface also generates a STUN candidate, but it is eliminated because it is redundant with the host candidate, as noted in [RFC5245] Sec 4.1.2..
5. The application-provided TURN server as seen through the virtual interface. (Traffic through this candidate is recursively encapsulated.)
 - * e.g. candidate:702786350 1 udp 24583935 [public ip addr of the application TURN server] 52631 typ relay raddr [public ip addr for the proxy] rport 54606 generation 0

There are no STUN or TURN candidates on the physical interfaces, because the application-specified STUN and TURN servers are not reachable through the firewall.

If the remote peer is within the same network, it may be possible to establish a direct connection using both peers' host candidates. If the network prevents this kind of direct connection, the path will instead take a "hairpin" route through the enterprise's proxy, using one peer's physical "host" candidate and the other's virtual "host" candidate, or (if that is also disallowed by the network configuration) a "double hairpin" using both endpoints' virtual "host" candidates.

6.2. Conflicting proxies configured by Auto-Discovery and local policy

Consider an enterprise network with TURN and HTTP proxies advertised for Auto-Discovery with unspecified leakiness (thus defaulting to leaky). The browser endpoint configures an additional TURN proxy by a proprietary local mechanism.

If the locally configured proxy is leaky, then the browser MUST produce candidates representing any physical interfaces (including SSLTCP routes through the HTTP proxy), plus candidates for both UDP-only virtual interfaces created by the two TURN servers.

There MUST NOT be any candidate that uses both proxies. Multiple configured proxies are not chained recursively.

If the locally configured proxy is "sealed", then the browser MUST produce only candidates from the virtual interface associated with that proxy.

If both proxies are configured for "sealed" use, then the browser MUST produce only candidates from the virtual interface associated with the proxy with higher rank.

7. Security Considerations

A RETURN proxy can capture, block, and otherwise interfere with all of its clients' WebRTC network activity. Therefore, browsers and other WebRTC endpoints MUST NOT use RETURN proxies that are provided by untrusted sources. For example, endpoints MUST NOT implement a configuration based on unauthenticated network multicast (e.g. mDNS) unless the endpoint will only be used on networks where all other users are fully trusted to intercept all WebRTC traffic. In contrast, endpoints MAY implement mechanisms to configure RETURN proxies by system-wide policy, which can only be modified by trusted system administrators.

This document describes web browser behaviors that, if implemented correctly, allow users to achieve greater identity-confidentiality during WebRTC calls under certain configurations.

If a site administrator offers the site's users a TURN proxy, websites running in the users' browsers will be able to initiate a UDP-based WebRTC connection to any UDP transport address via the proxy. Websites' connections will quickly terminate if the remote endpoint does not reply with a positive indication of ICE consent, but no such restriction applies to other applications that access the TURN server. Administrators should take care to provide TURN access credentials only to the users who are authorized to have global UDP network access.

TURN proxies and application TURN servers can provide some privacy protection by obscuring the identity of one peer from the other. However, unencrypted TURN provides no additional privacy from an observer who can monitor the link between the TURN client and server, and even encrypted TURN ([RFC7350] Section 4.6) does not provide significant privacy from an observer who sniff traffic on both legs of the TURN connection, due to packet timing correlations.

8. IANA Considerations

This document requires no actions from IANA.

9. Acknowledgements

Thanks to Harald Alvestrand, Philipp Hancke, Tirumaleswar Reddy, Alan Johnston, John Yoakum, and Cullen Jennings for suggestions to improve the content and presentation. Special thanks to Alan Johnston for contributing the visual overview in Section 2.

10. References

10.1. Normative References

- [I-D.ietf-rtcweb-jsep]
Uberti, J., Jennings, C., and E. Rescorla, "Javascript Session Establishment Protocol", draft-ietf-rtcweb-jsep-19 (work in progress), March 2017.
- [I-D.ietf-tram-turn-server-discovery]
Patil, P., Reddy, T., and D. Wing, "TURN Server Auto Discovery", draft-ietf-tram-turn-server-discovery-12 (work in progress), January 2017.
- [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, <<http://www.rfc-editor.org/info/rfc1928>>.
- [RFC5245] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", RFC 5245, DOI 10.17487/RFC5245, April 2010, <<http://www.rfc-editor.org/info/rfc5245>>.
- [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, <<http://www.rfc-editor.org/info/rfc5766>>.
- [RFC6156] Camarillo, G., Novo, O., and S. Perreault, Ed., "Traversal Using Relays around NAT (TURN) Extension for IPv6", RFC 6156, DOI 10.17487/RFC6156, April 2011, <<http://www.rfc-editor.org/info/rfc6156>>.
- [RFC6555] Wing, D. and A. Yourtchenko, "Happy Eyeballs: Success with Dual-Stack Hosts", RFC 6555, DOI 10.17487/RFC6555, April 2012, <<http://www.rfc-editor.org/info/rfc6555>>.

10.2. Informative References

- [I-D.ietf-rtcweb-ip-handling]
Uberti, J. and G. Shieh, "WebRTC IP Address Handling Requirements", draft-ietf-rtcweb-ip-handling-03 (work in progress), January 2017.

- [I-D.ietf-rtcweb-security]
Rescorla, E., "Security Considerations for WebRTC", draft-ietf-rtcweb-security-08 (work in progress), February 2015.
- [I-D.ietf-rtcweb-security-arch]
Rescorla, E., "WebRTC Security Architecture", draft-ietf-rtcweb-security-arch-12 (work in progress), June 2016.
- [RFC7350] Petit-Huguenin, M. and G. Salgueiro, "Datagram Transport Layer Security (DTLS) as Transport for Session Traversal Utilities for NAT (STUN)", RFC 7350, DOI 10.17487/RFC7350, August 2014, <<http://www.rfc-editor.org/info/rfc7350>>.
- [RFC7478] Holmberg, C., Hakansson, S., and G. Eriksson, "Web Real-Time Communication Use Cases and Requirements", RFC 7478, DOI 10.17487/RFC7478, March 2015, <<http://www.rfc-editor.org/info/rfc7478>>.

Authors' Addresses

Benjamin M. Schwartz
Google, Inc.
111 8th Ave
New York, NY 10011
USA

Email: bemasc@webrtc.org

Justin Uberti
Google, Inc.
747 6th Street South
Kirkland, WA 98033
USA

Email: justin@uberti.name

Network Working Group
Internet-Draft
Intended status: Informational
Expires: 20 June 2021

S. Nandakumar
C. Jennings
Cisco
17 December 2020

Annotated Example SDP for WebRTC
draft-ietf-rtcweb-sdp-14

Abstract

The Web Real Time Communications (WebRTC) family of protocols defines mechanism for direct interactive rich communication using audio, video, and data between two peers' web browsers. With in the WebRTC framework, the Session Description Protocol (SDP) is used for negotiating session capabilities between the peers. Such a negotiation happens based on the SDP offer/answer exchange mechanism

This document provides an informational reference in describing the role of SDP and the offer/answer exchange mechanism for the most common WebRTC use cases.

This document makes no changes to the SDP offer/answer exchange mechanism.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 20 June 2021.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. Terminology	3
3. SDP and WebRTC	3
4. Offer/Answer and the WebRTC	6
5. WebRTC Session Description Examples	6
5.1. Some Conventions	7
5.2. Basic Examples	9
5.2.1. Audio Only Session	9
5.2.2. Audio/Video Session	13
5.2.2.1. IPv4 Audio/Video Session	14
5.2.2.2. Dual Stack Audio/Video Session	19
5.2.3. Data-only Session	24
5.2.4. Audio Call On Hold	26
5.2.5. Audio with DTMF Session	30
5.2.6. One-way Audio/Video Session - Document Camera	34
5.2.7. Audio, Video Session with BUNDLE Support Unknown	39
5.2.8. Audio, Video, and Data Session	45
5.2.9. Audio and Video Session with BUNDLE Unsupported	50
5.2.10. Audio, Video BUNDLED, but Data Not BUNDLED	56
5.2.11. Audio Only, Add Video to BUNDLE	62
5.3. MultiResolution, RTX, FEC Examples	70
5.3.1. Send-only Simulcast Session with 2 Cameras and 2 Encodings per Camera	70
5.3.2. Successful SVC Video Session	78
5.3.3. Successful Simulcast Video Session with Retransmission	82
5.3.4. Successful One-way Simulcast Session with 2 resolutions and RTX - One resolution rejected	88
5.3.5. Simulcast Video Session with Forward Error Correction	93
5.4. Others	99
5.4.1. Audio Session - Voice Activity Detection	99
5.4.2. Audio Conference - Voice Activity Detection	103
5.4.3. Successful Legacy Interop Fallback with bundle-only	107
5.4.4. Legacy Interop with RTP/AVP profile	112
6. IANA Considerations	116

7. Security Considerations	117
8. Acknowledgments	117
9. Change Log	117
10. Informative References	121
Appendix A. Appendix	126
A.1. JSEP SDP Attributes Checklist	126
A.1.1. Common Checklist	126
A.1.2. RTP Media Description Checklist	127
A.1.3. Data Channel Media Description checklist	129
Authors' Addresses	129

1. Introduction

JavaScript Session Establishment Protocol (JSEP)

[I-D.ietf-rtcweb-jsep] specifies a generic protocol needed to generate [RFC3264] SDP offers and answers negotiated between the [WebRTC] peers for setting up, updating, and tearing down a WebRTC session. For this purpose, SDP is used for describing (media and non-media) streams as appropriate for the recipients of the session description to participate in the session.

The remainder of this document is organized as follows: Sections 3 and 4 provide an overview of SDP and the offer/answer exchange mechanism. Section 5 provides sample SDP generated for the most common WebRTC use cases.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

Readers should be familiar with the terminology defined in [RFC3264] and in [RFC7656].

3. SDP and WebRTC

The purpose of this section is to provide a general overview of SDP and its components. For a more in-depth understanding, the readers are advised to refer to [RFC4566].

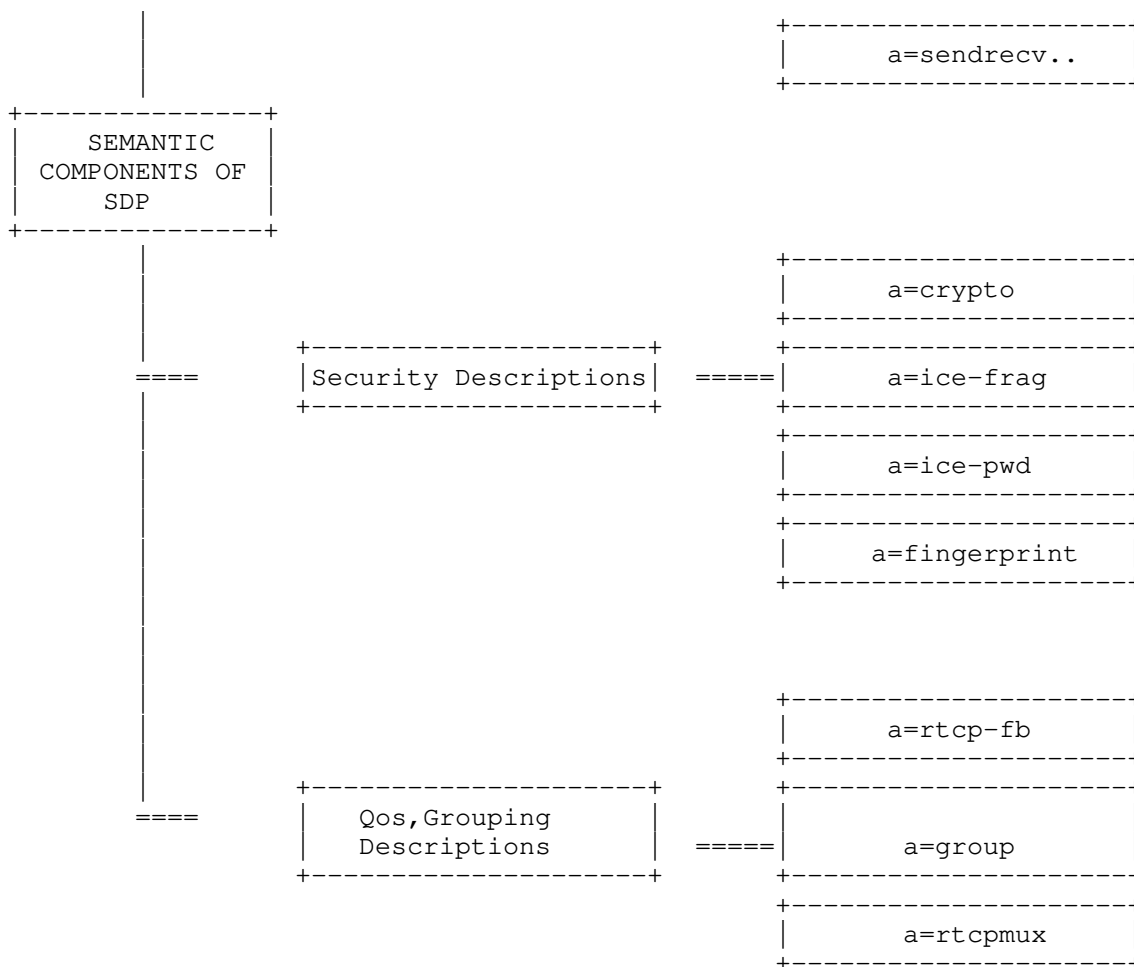


Figure 1: Semantic Components of SDP

[WebRTC] is architected in such a way that the design of the control plane is specified and implemented by the JavaScript application, as described in the JSEP specification [I-D.ietf-rtcweb-jsep]. JSEP provides mechanisms to create session characterization and media definition information to conduct the session based on SDP exchanges.

In this context, SDP serves two purposes:

1. Provide grammatical structure syntactically.
 2. Semantically convey a participant's intention and capabilities required to successfully negotiate a session.
4. Offer/Answer and the WebRTC

This section introduces the SDP offer/answer Exchange mechanism mandated by WebRTC for negotiating session capabilities while setting up, updating, and tearing down a WebRTC session. This section is intentionally brief in nature, and interested readers are recommended to refer to [RFC3264] for specific details on the protocol operations.

The offer/answer [RFC3264] model specifies rules for the bilateral exchange of Session Description Protocol (SDP) messages for creation of multimedia streams. It defines a protocol with the involved participants exchanging desired session characteristics with each other constructed as SDP messages to negotiate the session between them.

In the most basic form, the protocol operation begins by one of the participants sending an initial SDP offer describing its intent to start a multimedia communication session. The participant receiving the offer may generate an SDP answer accepting the offer or it may reject the offer. If the session is accepted the offer/answer model provides a common view of the multimedia session between the participants.

Within the context of WebRTC, the offer/answer model defines the state-machinery for WebRTC peers to negotiate session descriptions between themselves during the initial setup stages as well as for eventual session updates. The JSEP specification [I-D.ietf-rtcweb-jsep] for WebRTC provides the mechanism for generating [RFC3264] SDP offers and answers in order for both sides of the session to agree upon the details such as the list of media formats to be sent/received, bandwidth information, crypto parameters, and transport parameters, for example.

5. WebRTC Session Description Examples

A typical web-based real-time multimedia communication session can be characterized as follows:

- * It has zero or more audio-only, video-only or audio/video RTP sessions,
- * may contain zero or more non-media data sessions,

- * All the sessions are secured with DTLS-SRTP,
- * Supports NAT traversal using the ICE mechanism,
- * Provides RTCP-based feedback mechanisms,
- * Sessions can be over IPv4-only or IPv6-only or dual-stack based clients,
- * Supports BUNDLE-based grouping of media streams over a single 5-tuple transport.

5.1. Some Conventions

The examples given in this document follow the conventions listed below:

- * In all the examples, Alice and Bob are assumed to be the WebRTC peers.
- * It is assumed that for most of the examples, the support for [I-D.ietf-mmusic-sdp-bundle-negotiation] is established a priori either out-of-band or as a consequence of a successful offer/answer negotiation between Alice and Bob, unless explicitly stated otherwise.
- * Call-flow diagrams that accompany the use cases capture only the prominent aspects of the system behavior and intentionally are not detailed, to improve readability.
- * Even though the call-flow diagrams show SDP being exchanged between the parties, it doesn't represent the only way an WebRTC setup is expected to work. Other approaches may involve WebRTC applications to exchange the media setup information via non-SDP mechanisms as long as they conform to the [I-D.ietf-rtcweb-jsep] API specification.
- * The SDP examples deviate from actual on-the-wire SDP notation in several ways. This is done to facilitate readability and to conform to the restrictions imposed by the RFC formatting rules.
 - Visual markers/Empty lines in any SDP example are inserted to make functional divisions in the SDP clearer, and are not actually part of the SDP syntax.
 - Excepting the above two conventions, line endings are to be interpreted as <CR><LF> pairs (that is, a US-ASCII 13 followed by a US-ASCII 10).

- * Against each SDP line, pointers to the appropriate RFCs are provided for further informational reference. Also an attempt has been made to provide explanatory notes to enable better understanding of the SDP usage, wherever appropriate.
- * The following SDP details are common across all the use cases defined in this document unless mentioned otherwise.
 - DTLS fingerprint for SRTP (a=fingerprint)
 - RTP/RTCP Multiplexing (a=rtcp-mux)
 - RTCP Feedback support (a=rtcp-fb)
 - Host and server-reflexive candidate lines (a=candidate)
 - DTLS-SRTP Setup framework parameters (a=setup)
 - RTCP attribute (a=rtcp)
 - RTP header extension indicating audio-levels from client to the mixer

For specific details, readers must refer to the [I-D.ietf-rtcweb-jsep] specification.

- * The term "session" is used rather loosely in this document to refer to either a "communication session" or an "RTP session" or a "RTP stream" depending on the context.
- * Payload type 109 is usually used for OPUS, 0 for PCMU, 8 for PCMA, 99 for H.264, and 120 for VP8 in most of the examples to maintain uniformity.
- * The IP Address:Port combinations '192.0.2.4:61665' (host) and '203.0.113.141:54609' (Server Reflexive) are typically used for Alice.
- * The IP Address:Port combinations '198.51.100.7:51556' (host) and '203.0.113.77:49203' (Server Reflexive) are typically used for Bob.
- * The IPv6 addresses 2001:db8:8101:3a55:4858:a2a9:22ff:99b9 and 2001:db8:30c:1266:5916:3779:22f6:77f7 are used to represent Alice and Bob's host addresses respectively.

- * In actual use the values that represent SSRCS, ICE candidate foundations, WebRTC MediaStream, and MediaStreamTrack IDs values shall be much larger than and/or random in comparison to the ones shown in the examples.
- * tls-id attribute values 89J2LRATQ3ULA24G9AHWVR31VJWSLB68 and UKA29UQLTF69OJW4WNPNUO2Y0GF1FJOZ are used for Alice. The values CJ6FF9ZZMJW7MDRJIR7XVIQM48GE1G31 and 9AIFS8AQ009IXF5D6QQUJ7P8BXPEZJ8G are used for Bob.
- * identity attribute values are split across multiple lines to enhance readability, thus any line breaks and indentations in the value must be ignored.
- * SDP attributes in the examples closely follow the checklist defined in Appendix A.1.

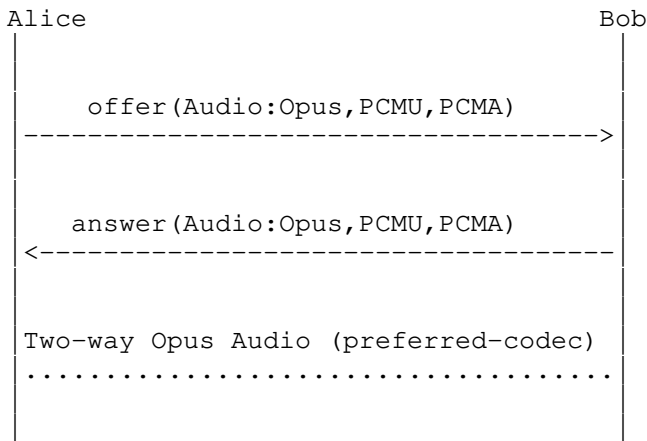
5.2. Basic Examples

5.2.1. Audio Only Session

This common scenario shows the SDP for a secure two-way audio session with Alice offering Opus, PCMU, PCMA and Bob accepting all the offered audio codecs.

This example also shows the endpoints being [RFC8445] compliant by including "ice2" ice-options attribute.

Two-way Audio Only Session



```

=====
| Offer SDP Contents
| RFC#/Notes
|
=====
| v=0
| [RFC4566]
|
-----
| o=- 20518 0 IN IP4 0.0.0.0
| [RFC4566] - Session Origin Information
|
-----
| s=-
| [RFC4566]
|
-----
| t=0 0
| [RFC4566]
|
-----
| a=group:BUNDLE audio
| [I-D.ietf-mmusic-sdp-bundle-negotiation]
|
-----
| a=ice-options:trickle
| [I-D.ietf-mmusic-trickle-ice]
|
-----
| a=ice-options:ice2
| [RFC8445]
|
-----
| a=identity:eyJpZHAiOnsiZG9tYWluIjoibmlpZi5odSIsInByb3RvY29sIjoiaWRwLmh0bWwifS
wiYXNzZXJ0a
| Section 7.4 of
| W9uIjoizX1KaGJHY2lPaUpTVXpJMU5pSXNJblI1Y0NjNk1rcFhVeUo5LmV5SmpiMjUwWlclMGN5ST
ZleUptYVclb
| [I-D.ietf-rtcweb-security-arch]
| lpYSndjbWx1ZENJNlczc2lZV3huYjNKcGRHaHRJam9pYzJoaExUSTFOaU1zSW1ScFoyVnPkQ0k2SW
prek9rTXdPa
|
| k16T2pKR09rRX1PakF3T2pBd09qQkVPalV4T2tGRE9rUX1PalUwT2pZMU9rWTBPak5DT2pkRU9qa3
1PalJET2pnN
|
| E9qTXpPalV4T2pJek9qUXdPamN5T2preE9qZ3pPalZDT2pBeE9qSkdPalV3T2pjNE9qTkdJbjFkZ1
N3aWFXUmxiB
|
| lJwZEhraU9pSnRhWE5wUuc1cGFXXVhSFVpZ1EuSTVQdGhKNFFDT05TOFVXd250OUh3MEdaTD13d0
RBVGRrTWtFW
|
| llmdlNVTTJ6Umd5R09WSGgzRmpnc2FPZk1kRnFsNUx6azBFbndVOTNQOU1CQ0xZOWtia3V1c0V1S2
5YRGVNLTNIN
|
| WfmdTJvZ19CTlZjUnB3MmdBdlNBbVR6S1ltcEppqMFETdmV0TmtVT1huZE9HLUIzT3ZGb3QwZVNNEN1
ZSNUdhb2wyc
|
| GduS3FStktOd3dacEZ1eUZZbFRodHJIdGniT19WV3o4QnZpTThKS25OdExWd1JxNUhMX2ZLTlRCNz
FDYkoyWmh5W
|
| XU1UEdwWDhXcJMWc1ybm5YSFY3RnhoTTh5OHdrLWd5cnRZazVnbF1ZeUFrcTVqZk1SXzRzWER5d1
9Qc1BWTW1aZ
|
| XltenVGV3BQTzVFWlJYR0ZpRjFET0o4Q0Q3Z3Zta2dUdlBXSwpkemtBIn0=
|
=====

```

```

|***** Audio m=line *****
|*****
+-----+
|m=audio 54609 UDP/TLS/RTP/SAVPF 109 0 8
| [RFC4566]
+-----+
|c=IN IP4 203.0.113.141
| [RFC4566]
+-----+
|a=mid:audio
| [RFC5888]
+-----+
|a=msid:ma ta
| Identifies MediaStream ID (ma) and
| MediaStreamTrack ID (ta)
+-----+
|a=sendrecv
| [RFC3264] - Alice can send and recv
| audio
+-----+
|a=rtpmap:109 opus/48000/2
| [RFC7587] - Opus Codec 48 kHz, 2
| channels
+-----+
|a=rtpmap:0 PCMU/8000
| [RFC3551] PCMU Audio Codec
+-----+

```

```
|a=rtpmap:8 PCMA/8000
|                                     | [RFC3551] PCMA Audio Codec          |
+-----+-----+
|a=maxptime:120
|                                     | [RFC4566]                            |
+-----+-----+
|a=ice-ufrag:074c6550
|                                     | [I-D.ietf-mmusic-ice-sip-sdp] - ICE user |
|                                     | fragment                               |
+-----+-----+
|a=ice-pwd:a28a397a4c3f31747dlee3474af08a068
|                                     | [I-D.ietf-mmusic-ice-sip-sdp] - ICE   |
|                                     | password                               |
+-----+-----+
|a=fingerprint:sha-256
|                                     | [RFC8122] - DTLS Fingerprint for SRTP |
|19:E2:1C:3B:4B:9F:81:E6:B8:5C:F4:A5:A8:D8:73:04:BB:05:2F:70:9F:04:A9:0E:05:E9
:26:33:E8:70:88:A2|
+-----+-----+
|a=setup:actpass
|                                     | [RFC5763] - Alice can act as DTLS client |
|                                     | or server                              |
+-----+-----+
|a=tls-id:89J2LRATQ3ULA24G9AHWVR31VJWSLB68
|                                     | [I-D.ietf-mmusic-dtls-sdp]          |
+-----+-----+
|a=rtcp-mux
|                                     | [RFC5761] - Alice can perform RTP/RTCP |
|                                     | Muxing                                |
+-----+-----+
|a=rtcp:60065 IN IP4 203.0.113.141
|                                     | [RFC3605]                            |
+-----+-----+
|a=rtcp-rsize
|                                     | [RFC5506] - Alice intends to use reduced |
|                                     | size RTCP for this session            |
+-----+-----+
|a=extmap:1 urn:ietf:params:rtp-hdext:ssrc-audio-level
|                                     | [RFC6464] Alice supports RTP header   |
|                                     | extension to indicate audio levels    |
+-----+-----+
```

```

|a=extmap:2 urn:iETF:params:rtp-hdext:sdes:mid
| [I-D.iETF-mmusic-sdp-bundle-negotiation] |
+-----+
|a=candidate:0 1 UDP 2122194687 192.0.2.4 61665 typ host
| [I-D.iETF-mmusic-ice-sip-sdp] - RTP Host |
|
| Candidate |
+-----+
|a=candidate:1 1 UDP 1685987071 203.0.113.141 54609 typ srflx raddr 192.0.2.4
rport 61665 | [I-D.iETF-mmusic-ice-sip-sdp] - RTP |
|
| Server Reflexive ICE Candidate |
+-----+
|a=candidate:0 2 UDP 2122194687 192.0.2.4 61667 typ host
| [I-D.iETF-mmusic-ice-sip-sdp] - RTCP |
|
| Host Candidate |
+-----+
|a=candidate:1 2 UDP 1685987071 203.0.113.141 60065 typ srflx raddr 192.0.2.4
rport 61667 | [I-D.iETF-mmusic-ice-sip-sdp] - RTCP |
|
| Server Reflexive ICE Candidate |
+-----+
|a=end-of-candidates
| [I-D.iETF-mmusic-trickle-ice] |
+-----+

```

Table 1: 5.2.1 SDP Offer


```

=====
| Answer SDP Contents |
| RFC#/Notes |
=====
| v=0 |
| [RFC4566] |
-----
| o=- 16833 0 IN IP4 0.0.0.0 |
| [RFC4566] - Session Origin Information |
-----
| s=- |
| [RFC4566] |
-----
| t=0 0 |
| [RFC4566] |
-----
| a=group:BUNDLE audio |
| [I-D.ietf-mmusic-sdp-bundle-negotiation] |
-----
| a=ice-options:trickle |
| [I-D.ietf-mmusic-trickle-ice] |
-----
| a=ice-options:ice2 |
| [RFC8445] |
-----
| a=identity:ew0KICAiaWRwIjpb7DQogICAgImRvbWFpbI6ICJjaXNjb3NwYXJrLmNvbSIsDQogIC
Ag | Section 7.4 of
| InByb3RvY29sIjogImRlZmFlbHQiDQogIH0sDQogICJhc3NlcnRpb24iOiAibEp3WkVocmFVOXBtb
1Jo | [I-D.ietf-rtcweb-security-arch]
| V0U1d1VWYzZjR0ZlV1hWaFNGVnBabEV1U1RWUWRHaEtORkZEVDAlVE9GV1hkMjVPT1VoM01FZGFUR
Gwz |
| ZDBSQ1ZHUnJUV3RGVw0KICAgICAgICAgICAgICBsbG1kbE5WVFRKN1VtZDVSMD1XU0dne1JtcG5jM
kZQ |
| Wmts a1JuRnNOVXg2YXpCRmJuZFZPVE5RT1VsQ1EweFpPV3RpbYTNWMMwVjFjVjZUkdWTkxUTk1OD
Qog |
| ICAgICAgICAgICAgIFdGbWRUSnZabD1DVGxaa1VuQjNNbWRCZGx0QmJWUjZTbGx0Y0VwcU1GRXRkb
VYw |
| VG10VlQxaHVaRT1ITFVJelQzWkdiM1F3WlZORU5sW1NOVWRoYjJ3eWMNCiAgICAgICAgICAgICAgR
2R1 |
| UzNGU1RrdE9kM2RhY0VaMwVWVlpiRlJvZlVhKSWRHTm1UMTlXVjNvNFpWnBUVGHUzI1T2RFefdkM
Up4 |
| TlVoTVgyWkxUbFJDnpgRFlrb3lXbWg1VyINCn0= |
-----
| ***** Audio m=line ***** |
| ***** |
-----

```

```

|m=audio 49203 UDP/TLS/RTP/SAVPF 109 0 8
| [RFC4566]
+-----+
|c=IN IP4 203.0.113.77
| [RFC4566]
+-----+
|a=mid:audio
| [RFC5888]
+-----+
|a=msid:ma ta
| [I-D.ietf-mmusic-msid] Identifies
|
| MediaStream ID (ma) and MediaStreamTrack
|
| ID (ta)
+-----+
|a=sendrecv
| [RFC3264] - Bob can send and recv audio
+-----+
|a=rtpmap:109 opus/48000/2
| [RFC7587] Opus Codec
+-----+
|a=rtpmap:0 PCMU/8000
| [RFC3551] PCMU Audio Codec
+-----+
|a=rtpmap:8 PCMA/8000
| [RFC3551] PCMA Audio Codec
+-----+
|a=maxptime:120
| [RFC4566]

```

```

+-----+
+-----+
|a=ice-ufrag:05067423
|           |[I-D.ietf-mmusic-ice-sip-sdp] - ICE user|
|           |fragment                               |
+-----+
+-----+
|a=ice-pwd:1747d1ee3474a28a397a4c3f3af08a068
|           |[I-D.ietf-mmusic-ice-sip-sdp] - ICE   |
|           |password parameter                   |
+-----+
+-----+
|a=fingerprint:sha-256
|           |[RFC8122] - DTLS Fingerprint for SRTP |
|6B:8B:F0:65:5F:78:E2:51:3B:AC:6F:F3:3F:46:1B:35:DC:B8:5F:64:1A:24:C2:43:F0:A1
:58:D0:A1:2C:19:08|
+-----+
+-----+
|a=setup:active
|           |[RFC5763] - Bob is the DTLS client   |
+-----+
+-----+
|a=tls-id:CJ6FF9ZZMJW7MDRJIR7XVIQM48GE1G31
|           |[I-D.ietf-mmusic-dtls-sdp]           |
+-----+
+-----+
|a=rtcp-mux
|           |[RFC5761] - Bob can perform RTP/RTCP |
|           |Muxing on port 49203                 |
+-----+
+-----+
|a=rtcp-rsize
|           |[RFC5506] - Bob intends to use reduced |
|           |size RTCP for this session           |
+-----+
+-----+
|a=extmap:1 urn:ietf:params:rtp-hdext:ssrc-audio-level
|           |[RFC6464] Bob supports audio level RTP |
|           |header extension as well             |
+-----+
+-----+
|a=extmap:2 urn:ietf:params:rtp-hdext:sdes:mid
|           |[I-D.ietf-mmusic-sdp-bundle-negotiation]|
+-----+
+-----+
|a=candidate:0 1 UDP 2122194687 198.51.100.7 51556 typ host
|           |[I-D.ietf-mmusic-ice-sip-sdp] - RTP/RTCP|
|           |Host ICE Candidate                   |
+-----+
+-----+
|a=candidate:1 1 UDP 1685987071 203.0.113.77 49203 typ srflx raddr 198.51.100.
7 rport 51556 |           |[I-D.ietf-mmusic-ice-sip-sdp] - RTP/RTCP|

```

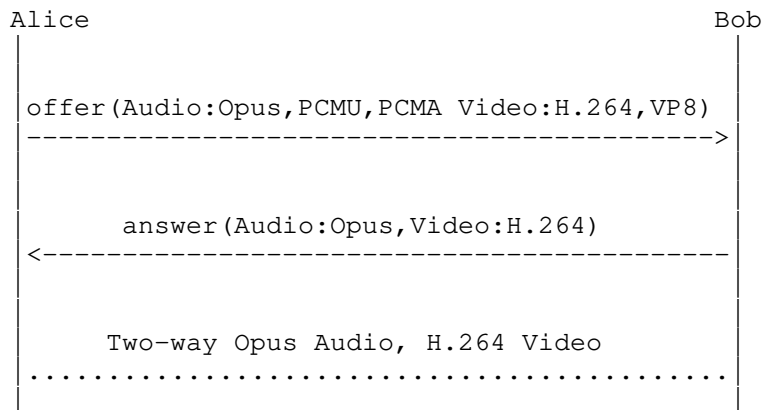
	Server Reflexive ICE Candidate	
+-----	+-----	+-----
a=end-of-candidates	[I-D.ietf-mmusic-trickle-ice]	
+-----	+-----	+-----
+-----	+-----	+-----

Table 2: 5.2.1 SDP Answer

5.2.2. Audio/Video Session

Alice and Bob establish a two-way audio and video session with Opus as the audio codec and H.264 as the video codec.

Two-way Audio/Video Session



5.2.2.1. IPv4 Audio/Video Session

This section shows the IPv4-only offer/answer exchange.

Offer SDP Contents	RFC#/Notes
v=0	[RFC4566]
o=- 20518 0 IN IP4 0.0.0.0	[RFC4566] - Session Origin Information
s=-	[RFC4566]
t=0 0	[RFC4566]
a=group:BUNDLE audio video	[I-D.ietf-mmusic-sdp-bundle-negotiation]
a=group:LS audio video	[RFC5888] - Alice wants to lip sync her audio and video streams
a=ice-options:trickle	[I-D.ietf-mmusic-trickle-ice]

```

|a=ice-options:ice2
| [RFC8445]
+-----+
+-----+
|***** Audio m=line *****
| *****
+-----+
+-----+
|m=audio 54609 UDP/TLS/RTP/SAVPF 109 0 8
| [RFC4566]
+-----+
+-----+
|c=IN IP4 203.0.113.141
| [RFC4566]
+-----+
+-----+

```

```

|a=mid:audio
| [RFC5888]
+-----+
|a=msid:ma ta
| [I-D.ietf-mmusic-msid] Identifies
|
| MediaStream ID (ma) and MediaStreamTrack
|
| ID (ta)
+-----+
|a=sendrecv
| [RFC3264] - Alice can send and recv
|
| audio
+-----+
|a=rtpmap:109 opus/48000/2
| [RFC7587] - Opus Codec 48 kHz, 2
|
| channels
+-----+
|a=rtpmap:0 PCMU/8000
| [RFC3551] PCMU Audio Codec
+-----+
|a=rtpmap:8 PCMA/8000
| [RFC3551] PCMA Audio Codec
+-----+
|a=maxptime:120
| [RFC4566]
+-----+
|a=ice-ufrag:074c6550
| [I-D.ietf-mmusic-ice-sip-sdp] - ICE user
|
| fragment
+-----+
|a=ice-pwd:a28a397a4c3f31747d1ee3474af08a068
| [I-D.ietf-mmusic-ice-sip-sdp] - ICE
|
| password parameter
+-----+
|a=fingerprint:sha-256
| [RFC8122] - DTLS Fingerprint for SRTP
| 19:E2:1C:3B:4B:9F:81:E6:B8:5C:F4:A5:A8:D8:73:04:BB:05:2F:70:9F:04:A9:0E:05:E9
:26:33:E8:70:88:A2
+-----+
|a=setup:actpass
| [RFC5763] - Alice can act as DTLS client
|
| or server

```

```

-----
|a=tls-id:89J2LRATQ3ULA24G9AHWVR31VJWSLB68
| [I-D.ietf-mmusic-dtls-sdp]
-----
|a=rtcp-mux
| [RFC5761] - Alice can perform RTP/RTCP
| Muxing
-----
|a=rtcp-mux-only
| [I-D.ietf-mmusic-mux-exclusive]
-----
|a=rtcp-rsize
| [RFC5506] - Alice intends to use reduced
| size RTCP for this session
-----
|a=extmap:1 urn:ietf:params:rtp-hdext:ssrc-audio-level
| [RFC6464]
-----
|a=extmap:2 urn:ietf:params:rtp-hdext:sdes:mid
| [I-D.ietf-mmusic-sdp-bundle-negotiation]
-----
|a=candidate:0 1 UDP 2122194687 192.0.2.4 61665 typ host
| [I-D.ietf-mmusic-ice-sip-sdp] - RTP/RTCP
| Host Candidate
-----
|a=candidate:1 1 UDP 1685987071 203.0.113.141 54609 typ srflx raddr 192.0.2.4
rport 61665 | [I-D.ietf-mmusic-ice-sip-sdp] - RTP/RTCP

```



```

|
|           |Server Reflexive ICE Candidate           |
+-----+-----+
|a=end-of-candidates
|           |[I-D.ietf-mmusic-trickle-ice]           |
+-----+-----+
|***** Video m=line *****
|           |*****                               |
+-----+-----+
|m=video 0 UDP/TLS/RTP/SAVPF 99 120
|           |[RFC4566]                               |
+-----+-----+
|c=IN IP4 203.0.113.141
|           |[RFC4566]                               |
+-----+-----+
|a=bundle-only
|           |[I-D.ietf-mmusic-sdp-bundle-negotiation]|
+-----+-----+
|a=mid:video
|           |[RFC5888]                               |
+-----+-----+
|a=msid:ma tb
|           |Identifies MediaStream ID (ma) and     |
|           |MediaStreamTrack ID (tb)               |
+-----+-----+
|a=sendrecv
|           |[RFC3264] - Alice can send and recv    |
|           |video                                   |
+-----+-----+
|a=rtpmap:99 H264/90000
|           |[RFC6184] - H.264 Video Codec          |
+-----+-----+
|a=fmtp:99 profile-level-id=4d0028;packetization-mode=1
|           |[RFC6184]                               |
+-----+-----+
|a=rtpmap:120 VP8/90000
|           |[RFC7741] - VP8 video codec            |
+-----+-----+
|a=rtcp-fb:99 nack
|           |[RFC4585] - Indicates NACK RTCP feedback|
|           |support                                 |
+-----+-----+

```

a=rtcp-fb:99	nack pli	[RFC4585] - Indicates support for	
		Picture loss Indication and NACK	
+-----+-----+			
a=rtcp-fb:99	ccm fir	[RFC5104] - Full Intra Frame Request-	
		Codec Control Message support	
+-----+-----+			
a=rtcp-fb:120	nack	[RFC4585] - Indicates NACK RTCP feedback	
		support	
+-----+-----+			
a=rtcp-fb:120	nack pli	[RFC4585] - Indicates support for	
		Picture loss Indication and NACK	
+-----+-----+			
a=rtcp-fb:120	ccm fir	[RFC5104] - Full Intra Request-Codec	
		Control Message support	
+-----+-----+			
a=extmap:2	urn:ietf:params:rtp-hdext:sdes:mid	[I-D.ietf-mmusic-sdp-bundle-negotiation]	
+-----+-----+			

Table 3: 5.2.2.1 SDP Offer

```

=====
| Answer SDP Contents
|                               | RFC#/Notes
|                               |
+=====
| v=0
|                               | [RFC4566]
+-----+
| o=- 16833 0 IN IP4 0.0.0.0
|                               | [RFC4566] - Session Origin Information
+-----+
| s=-
|                               | [RFC4566]
+-----+
| t=0 0
|                               | [RFC4566]
+-----+
| a=group:BUNDLE audio video
|                               | [I-D.ietf-mmusic-sdp-bundle-negotiation]
+-----+
| a=group:LS audio video
|                               | [RFC5888] - Bob agrees to do the same
+-----+
| a=ice-options:trickle
|                               | [I-D.ietf-mmusic-trickle-ice]
+-----+
| a=ice-options:ice2
|                               | [RFC8445]
+-----+
| ***** Audio m=line *****
|                               | *****
+-----+
| m=audio 49203 UDP/TLS/RTP/SAVPF 109
|                               | [RFC4566]
+-----+
| c=IN IP4 203.0.113.77
|                               | [RFC4566]
+-----+
| a=mid:audio
|                               | [RFC5888]
+-----+
| a=msid:ma ta
|                               | Identifies MediaStream ID (ma) and
|                               |
|                               | MediaStreamTrack ID (ta)
|                               |

```

```

-----+-----
|a=sendrecv                               |[RFC3264] - Bob can send and recv audio |
-----+-----
|a=rtpmap:109 opus/48000/2                 |[RFC7587] - Bob accepts only Opus Codec |
-----+-----
|a=maxptime:120                            |[RFC4566]                               |
-----+-----
|a=ice-ufrag:c300d85b                     |[I-D.ietf-mmusic-ice-sip-sdp] - ICE     |
|                                           |username frag                            |
-----+-----
|a=ice-pwd:de4e99bd291c325921d5d47efbabd9a2|[I-D.ietf-mmusic-ice-sip-sdp] - ICE     |
|                                           |password                                 |
-----+-----
|a=fingerprint:sha-256                    |[RFC8122] - DTLS Fingerprint for SRTP  |
|6B:8B:F0:65:5F:78:E2:51:3B:AC:6F:F3:3F:46:1B:35:DC:B8:5F:64:1A:24:C2:43:F0:A1|
:58:D0:A1:2C:19:08|
-----+-----
|a=setup:active                            |[RFC5763] - Bob is the DTLS client     |
-----+-----
|a=tls-id:CJ6FF9ZZMJW7MDRJIR7XVIQM48GE1G31|[I-D.ietf-mmusic-dtls-sdp]             |

```

```

+-----+
+-----+
|a=rtcp-mux          | [RFC5761] - Bob can perform RTP/RTCP |
|                   | Muxing                               |
+-----+
+-----+
|a=rtcp-mux-only    | [I-D.ietf-mmusic-mux-exclusive]     |
+-----+
+-----+
|a=rtcp-rsize       | [RFC5506] - Bob intends to use reduced |
|                   | size RTCP for this session          |
+-----+
+-----+
|a=extmap:1 urn:ietf:params:rtp-hdext:ssrc-audio-level |
|                   | [RFC6464]                           |
+-----+
+-----+
|a=extmap:2 urn:ietf:params:rtp-hdext:sdes:mid        |
|                   | [I-D.ietf-mmusic-sdp-bundle-negotiation] |
+-----+
+-----+
|a=candidate:0 1 UDP 3618095783 198.51.100.7 49203 typ host |
|                   | [I-D.ietf-mmusic-ice-sip-sdp] - RTP/RTCP |
|                   | Host ICE Candidate                     |
+-----+
+-----+
|a=candidate:1 1 UDP 565689203 203.0.113.77 49203 typ srflx raddr 198.51.100.7 |
| rport 51556      | [I-D.ietf-mmusic-ice-sip-sdp] - RTP/RTCP |
|                   | Server Reflexive ICE Candidate         |
+-----+
+-----+
|a=end-of-candidates | [I-D.ietf-mmusic-trickle-ice]       |
+-----+
+-----+
|***** Video m=line ***** |
|*****                       |
+-----+
+-----+
|m=video 0 UDP/TLS/RTP/SAVPF 99 |
| [RFC4566]                     |
+-----+
+-----+
|c=IN IP4 203.0.113.77        |
| [RFC4566]                   |
+-----+
+-----+
|a=bundle-only              | [I-D.ietf-mmusic-sdp-bundle-negotiation] |
+-----+
+-----+

```

a=mid:video	[RFC5888]	
+-----+-----+		
a=msid:ma tb	Identifies MediaStream ID (ma) and	
	MediaStreamTrack ID (tb)	
+-----+-----+		
a=sendrecv	[RFC3264] - Bob can send and recv video	
+-----+-----+		
a=rtpmap:99 H264/90000	[RFC6184] - Bob accepts H.264 Video	
	Codec.	
+-----+-----+		
a=fmtp:99 profile-level-id=4d0028;packetization-mode=1	[RFC6184]	
+-----+-----+		
a=rtcp-fb:99 nack	[RFC4585] - Indicates support for NACK-	
	based RTCP feedback	
+-----+-----+		
a=rtcp-fb:99 nack pli	[RFC4585] - Indicates support for	
	Picture loss Indication and NACK	
+-----+-----+		
a=rtcp-fb:99 ccm fir	[RFC5104] - Full Intra Request-Codec	

	Control Message support	
+-----+		+-----+
	a=extmap:2 urn:ietf:params:rtp-hdext:sdes:mid	
	[I-D.ietf-mmusic-sdp-bundle-negotiation]	
+-----+		+-----+

Table 4: 5.2.2.1 SDP Answer

5.2.2.2. Dual Stack Audio/Video Session

This section captures offer/answer exchange when Alice and Bob support both IPv4 and IPv6 host addresses.

+=====		+=====
=====		=====
	Offer SDP Contents	
	RFC#/Notes	
+=====		+=====
	v=0	
	[RFC4566]	
+-----+		+-----+
	o=- 20518 0 IN IP4 0.0.0.0	
	[RFC4566] - Session Origin Information	
+-----+		+-----+
	s=-	
	[RFC4566]	
+-----+		+-----+
	t=0 0	
	[RFC4566]	
+-----+		+-----+
	a=group:BUNDLE audio video	
	[I-D.ietf-mmusic-sdp-bundle-negotiation]	
+-----+		+-----+
	a=group:LS audio video	
	[RFC5888] - Alice wants to lip sync her	
	audio and video streams	
+-----+		+-----+
	a=ice-options:trickle	
	[I-D.ietf-mmusic-trickle-ice]	
+-----+		+-----+
	a=ice-options:ice2	
	[RFC8445]	
+-----+		+-----+
	***** Audio m=line *****	

```

+-----+
|*****|
+-----+
|m=audio 54609 UDP/TLS/RTP/SAVPF 109 0 8
| [RFC4566]
+-----+
|c=IN IP4 203.0.113.141
| [RFC4566]
+-----+
|a=mid:audio
| [RFC5888]
+-----+
|a=msid:ma ta
| Identifies MediaStream ID (ma) and
| MediaStreamTrack ID (ta)
+-----+
|a=sendrecv
| [RFC3264] - Alice can send and recv
| audio
+-----+
|a=rtpmap:109 opus/48000/2
| [RFC7587] - Opus Codec 48 kHz, 2

```



```

|
|           |channels|
+-----+
|a=rtpmap:0 PCMU/8000
|           |[RFC3551] PCMU Audio Codec|
+-----+
|a=rtpmap:8 PCMA/8000
|           |[RFC3551] PCMA Audio Codec|
+-----+
|a=maxptime:120
|           |[RFC4566]|
+-----+
|a=ice-ufrag:074c6550
|           |[I-D.ietf-mmusic-ice-sip-sdp] - ICE user|
|           |fragment|
+-----+
|a=ice-pwd:a28a397a4c3f31747dlee3474af08a068
|           |[I-D.ietf-mmusic-ice-sip-sdp] - ICE
|           |password parameter|
+-----+
|a=fingerprint:sha-256
|           |[RFC8122] - DTLS Fingerprint for SRTP|
|19:E2:1C:3B:4B:9F:81:E6:B8:5C:F4:A5:A8:D8:73:04:BB:05:2F:70:9F:04:A9:0E:05:E9
:26:33:E8:70:88:A2|
+-----+
|a=setup:actpass
|           |[RFC5763] - Alice can act as DTLS client|
|           |or server|
+-----+
|a=tls-id:89J2LRATQ3ULA24G9AHWVR31VJWSLB68
|           |[I-D.ietf-mmusic-dtls-sdp]|
+-----+
|a=rtcp-mux
|           |[RFC5761] - Alice can perform RTP/RTCP|
|           |Muxing|
+-----+
|a=rtcp-mux-only
|           |[I-D.ietf-mmusic-mux-exclusive]|
+-----+
|a=rtcp-rsize
|           |[RFC5506] - Alice intends to use reduced|
|           |size RTCP for this session|

```

```

-----+-----
|a=extmap:1 urn:iETF:params:rtp-hdext:ssrc-audio-level
| [RFC6464]
-----+-----
|a=extmap:2 urn:iETF:params:rtp-hdext:sdes:mid
| [I-D.iETF-mmusic-sdp-bundle-negotiation]
-----+-----
|a=candidate:0 1 UDP 2122194687 192.0.2.4 61665 typ host
| [I-D.iETF-mmusic-ice-sip-sdp] - RTP/RTCP
|
| Host Candidate
-----+-----
|a=candidate:0 1 UDP 2122194687 2001:db8:8101:3a55:4858:a2a9:22ff:99b9 61665 t
yp host
| [I-D.iETF-mmusic-ice-sip-sdp] - RTP/RTCP
|
| IPv6 Host Candidate
-----+-----
|a=end-of-candidates
| [I-D.iETF-mmusic-trickle-ice]
-----+-----
|***** Video m=line *****
| *****
-----+-----
|m=video 0 UDP/TLS/RTP/SAVPF 99 120
| [RFC4566]
-----+-----
|c=IN IP4 203.0.113.141
| [RFC4566]
-----+-----

```

a=bundle-only	[I-D.ietf-mmusic-sdp-bundle-negotiation]
a=mid:video	[RFC5888]
a=msid:ma tb	Identifies MediaStream ID (ma) and
	MediaStreamTrack ID (tb)
a=sendrecv	[RFC3264] - Alice can send and recv
	video
a=rtpmap:99 H264/90000	[RFC6184] - H.264 Video Codec
a=fmtp:99 profile-level-id=4d0028;packetization-mode=1	[RFC6184]
a=rtpmap:120 VP8/90000	[RFC7741] - VP8 video codec
a=rtcp-fb:99 nack	[RFC4585] - Indicates NACK RTCP feedback
	support
a=rtcp-fb:99 nack pli	[RFC4585] - Indicates support for
	Picture loss Indication and NACK
a=rtcp-fb:99 ccm fir	[RFC5104] - Full Intra Request-Codec
	Control Message support
a=rtcp-fb:120 nack	[RFC4585] - Indicates NACK RTCP feedback
	support
a=rtcp-fb:120 nack pli	[RFC4585] - Indicates support for

	Picture loss Indication and NACK	
a=rtcp-fb:120	ccm fir	
	[RFC5104] - Full Intra Request-Codec	
	Control Message support	
a=extmap:2	urn:ietf:params:rtp-hdext:sdes:mid	
	[I-D.ietf-mmusic-sdp-bundle-negotiation]	

Table 5: 5.2.2.2 SDP Offer

Answer SDP Contents	RFC#/Notes	
v=0	[RFC4566]	
o=- 16833 0 IN IP4 0.0.0.0	[RFC4566] - Session Origin Information	
s=-	[RFC4566]	

```
+-----+
+-----+
|t=0 0                                     |
|                                     | [RFC4566] |
+-----+
+-----+
|a=group:BUNDLE audio video              |
|                                     | [I-D.ietf-mmusic-sdp-bundle-negotiation] |
+-----+
+-----+
|a=group:LS audio video                  |
|                                     | [RFC5888] - Bob agrees to do the same |
+-----+
+-----+
|a=ice-options:trickle                   |
|                                     | [I-D.ietf-mmusic-trickle-ice] |
+-----+
+-----+
|a=ice-options:ice2                      |
|                                     | [RFC8445] |
+-----+
+-----+
|***** Audio m=line *****           |
|*****                               |
+-----+
+-----+
|m=audio 49203 UDP/TLS/RTP/SAVPF 109    |
|                                     | [RFC4566] |
+-----+
+-----+
|c=IN IP4 203.0.113.77                   |
|                                     | [RFC4566] |
+-----+
+-----+
|a=mid:audio                             |
|                                     | [RFC5888] |
+-----+
+-----+
|a=msid:ma ta                            |
|                                     | Identifies MediaStream ID (ma) and |
|                                     | MediaStreamTrack ID (ta) |
+-----+
+-----+
|a=sendrecv                              |
|                                     | [RFC3264] - Bob can send and recv audio |
+-----+
+-----+
|a=rtpmap:109 opus/48000/2               |
|                                     | [RFC7587] - Bob accepts only Opus Codec |
+-----+
+-----+
|a=maxptime:120                          |
|                                     | [RFC4566] |
+-----+
+-----+
|a=ice-ufrag:c300d85b                   |
|                                     | [I-D.ietf-mmusic-ice-sip-sdp] - ICE |
```

```

|
|                               |username frag                               |
+-----+-----+
|a=ice-pwd:de4e99bd291c325921d5d47efbabd9a2
|                               |[I-D.ietf-mmusic-ice-sip-sdp] - ICE                               |
|                               |password                               |
+-----+-----+
|a=fingerprint:sha-256
|                               |[RFC8122] - DTLS Fingerprint for SRTP                               |
|6B:8B:F0:65:5F:78:E2:51:3B:AC:6F:F3:3F:46:1B:35:DC:B8:5F:64:1A:24:C2:43:F0:A1
:58:D0:A1:2C:19:08|
+-----+-----+
|a=setup:active
|                               |[RFC5763] - Bob is the DTLS client                               |
+-----+-----+
|a=tls-id:CJ6FF9ZZMJW7MDRJIR7XVIQM48GE1G31
|                               |[I-D.ietf-mmusic-dtls-sdp]                               |
+-----+-----+
|a=rtcp-mux
|                               |[RFC5761] - Bob can perform RTP/RTCP                               |
|                               |Muxing                               |
+-----+-----+
|a=rtcp-mux-only
|                               |[I-D.ietf-mmusic-mux-exclusive]                               |
+-----+-----+
|a=rtcp-rsize
|                               |[RFC5506] - Bob intends to use reduced                               |
|                               |size RTCP for this session                               |

```

```

+-----+
+-----+
|a=extmap:1 urn:ietf:params:rtp-hdext:ssrc-audio-level
| [RFC6464] |
+-----+
+-----+
|a=extmap:2 urn:ietf:params:rtp-hdext:sdes:mid
| [I-D.ietf-mmusic-sdp-bundle-negotiation] |
+-----+
+-----+
|a=candidate:0 1 UDP 3618095783 198.51.100.7 49203 typ host
| [I-D.ietf-mmusic-ice-sip-sdp] - RTP/RTCP |
|
| Host ICE Candidate |
+-----+
+-----+
|a=candidate:0 1 UDP 3618095783 2001:db8:30c:1266:5916:3779:22f6:77f7 49203 ty
p host | [I-D.ietf-mmusic-ice-sip-sdp] - RTP/RTCP |
|
| IPv6 Host ICE Candidate |
+-----+
+-----+
|a=end-of-candidates
| [I-D.ietf-mmusic-trickle-ice] |
+-----+
+-----+
|***** Video m=line *****
| ***** |
+-----+
+-----+
|m=video 0 UDP/TLS/RTP/SAVPF 99
| [RFC4566] |
+-----+
+-----+
|c=IN IP4 203.0.113.77
| [RFC4566] |
+-----+
+-----+
|a=bundle-only
| [I-D.ietf-mmusic-sdp-bundle-negotiation] |
+-----+
+-----+
|a=mid:video
| [RFC5888] |
+-----+
+-----+
|a=msid:ma tb
| Identifies MediaStream ID (ma) and |
|
| MediaStreamTrack ID (tb) |
+-----+
+-----+
|a=sendrecv
| [RFC3264] - Bob can send and recv video |
+-----+
+-----+
|a=rtpmap:99 H264/90000
| [RFC6184] - Bob accepts H.264 Video |

```

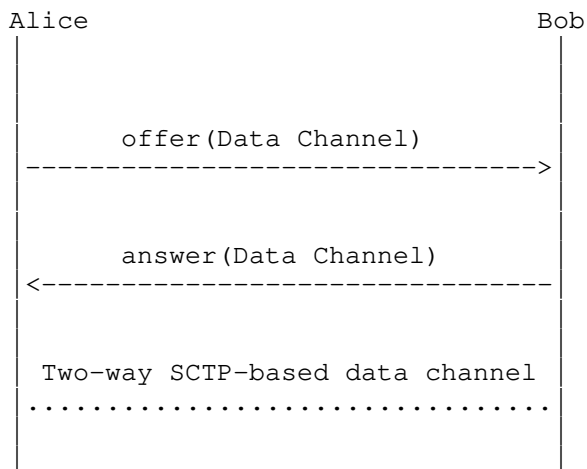
	Codec.	
+-----+		+-----+
	a=fmtp:99 profile-level-id=4d0028;packetization-mode=1	
+-----+	[RFC6184]	+-----+
	a=rtcp-fb:99 nack	
+-----+	[RFC4585] - Indicates support for NACK-	+-----+
	based RTCP feedback	
+-----+		+-----+
	a=rtcp-fb:99 nack pli	
+-----+	[RFC4585] - Indicates support for	+-----+
	Picture loss Indication and NACK	
+-----+		+-----+
	a=rtcp-fb:99 ccm fir	
+-----+	[RFC5104] - Full Intra Request- Codec	+-----+
	Control Message support	
+-----+		+-----+
	a=extmap:2 urn:ietf:params:rtp-hdext:sdes:mid	
+-----+	[I-D.ietf-mmusic-sdp-bundle-negotiation]	+-----+
+-----+		+-----+

Table 6: 5.2.2.2 SDP Answer

5.2.3. Data-only Session

This scenario illustrates the SDP negotiated to set up a data-only session based on the SCTP Data Channel, thus enabling use cases such as file transfer or real-time game control, for example.

Two-way Data channel Session



```

=====
+-----+-----+
| Offer SDP Contents          |
|           RFC#/Notes       |
+-----+-----+
| v=0                         | [RFC4566] |
+-----+-----+
| o=- 20518 0 IN IP4 0.0.0.0  | [RFC4566] - Session Origin Information |
+-----+-----+
| s=-                         | [RFC4566] |
+-----+-----+
| t=0 0                       | [RFC4566] |
+-----+-----+
| a=group:BUNDLE data        | [I-D.ietf-mmusic-sdp-bundle-negotiation] |
+-----+-----+
| a=ice-options:trickle      | [I-D.ietf-mmusic-trickle-ice] |
+-----+-----+
| a=ice-options:ice2         | [RFC8445] |
+-----+-----+

```

```
-----+-----+
|***** Application m=line *****|
|*****|*****|
+-----+-----+
|m=application 54609 UDP/DTLS/SCTP webrtc-datachannel|
| [I-D.ietf-rtcweb-data-channel]|
+-----+-----+
|c=IN IP4 203.0.113.141|
| [RFC4566]|
+-----+-----+
-----+-----+
```

```

|a=mid:data
| [RFC5888]
+-----+
|a=sendrecv
| [RFC3264] - Alice can send and recv non-
| media data
+-----+
|a=sctp-port:5000
| [I-D.ietf-mmusic-sctp-sdp]
+-----+
|a=max-message-size:100000
| [I-D.ietf-mmusic-sctp-sdp]
+-----+
|a=setup:actpass
| [RFC5763] - Alice can act as DTLS client
| or server
+-----+
|a=tls-id:89J2LRATQ3ULA24G9AHWVR31VJWSLB68
| [I-D.ietf-mmusic-dtls-sdp]
+-----+
|a=ice-ufrag:074c6550
| [I-D.ietf-mmusic-ice-sip-sdp] - Session
| Level ICE parameter
+-----+
|a=ice-pwd:a28a397a4c3f31747dlee3474af08a068
| [I-D.ietf-mmusic-ice-sip-sdp] - Session
| Level ICE parameter
+-----+
|a=fingerprint:sha-256
| [RFC8122] - Session DTLS Fingerprint for
| 19:E2:1C:3B:4B:9F:81:E6:B8:5C:F4:A5:A8:D8:73:04:BB:05:2F:70:9F:04:A9:0E:05:E9
:26:33:E8:70:88:A2 | SRTP
+-----+
|a=candidate:0 1 UDP 2113667327 192.0.2.4 61665 typ host
| [I-D.ietf-mmusic-ice-sip-sdp]
+-----+
|a=candidate:1 1 UDP 1694302207 203.0.113.141 54609 typ srflx raddr 192.0.2.4
rport 61665 | [I-D.ietf-mmusic-ice-sip-sdp]
+-----+
|a=end-of-candidates
| [I-D.ietf-mmusic-trickle-ice]
+-----+

```

Table 7: 5.2.3 SDP Offer

Answer SDP Contents	RFC#/Notes
v=0	[RFC4566]
o=- 16833 0 IN IP4 0.0.0.0	[RFC4566] - Session Origin Information
s=-	[RFC4566]
t=0 0	[RFC4566]
a=group:BUNDLE data	[I-D.ietf-mmusic-sdp-bundle-negotiation]
***** Application m=line *****	*****

```

|m=application 49203 UDP/DTLS/SCTP webrtc-datachannel
| [I-D.ietf-mmusic-sctp-sdp] |
+-----+
|c=IN IP4 203.0.113.77
| [RFC4566] |
+-----+
|a=mid:data
| [RFC5888] |
+-----+
|a=sendrecv
| [RFC3264] - Bob can send and recv non- |
| media data |
+-----+
|a=sctp-port:5000
| [I-D.ietf-mmusic-sctp-sdp] |
+-----+
|a=max-message-size:100000
| [I-D.ietf-mmusic-sctp-sdp] |
+-----+
|a=setup:active
| [RFC5763] - Bob is the DTLS client |
+-----+
|a=tls-id:CJ6FF9ZZMJW7MDRJIR7XVIQM48GE1G31
| [I-D.ietf-mmusic-dtls-sdp] |
+-----+
|a=ice-ufrag:c300d85b
| [I-D.ietf-mmusic-ice-sip-sdp] - Session |
| Level ICE username frag |
+-----+
|a=ice-pwd:de4e99bd291c325921d5d47efbabd9a2
| [I-D.ietf-mmusic-ice-sip-sdp] - Session |
| Level ICE password |
+-----+
|a=fingerprint:sha-256
| [RFC8122] - Session DTLS Fingerprint for|
| 6B:8B:F0:65:5F:78:E2:51:3B:AC:6F:F3:3F:46:1B:35:DC:B8:5F:64:1A:24:C2:43:F0:A1
:58:D0:A1:2C:19:08 | SRTP |
+-----+
|a=candidate:0 1 UDP 2113667327 198.51.100.7 51556 typ host
| [I-D.ietf-mmusic-ice-sip-sdp] |
+-----+
|a=candidate:1 1 UDP 1694302207 203.0.113.77 49203 typ srflx raddr 198.51.100.
7 rport 51556 | [I-D.ietf-mmusic-ice-sip-sdp] |

```

```

+-----+
|a=end-of-candidates
| [I-D.ietf-mmusic-trickle-ice]
+-----+

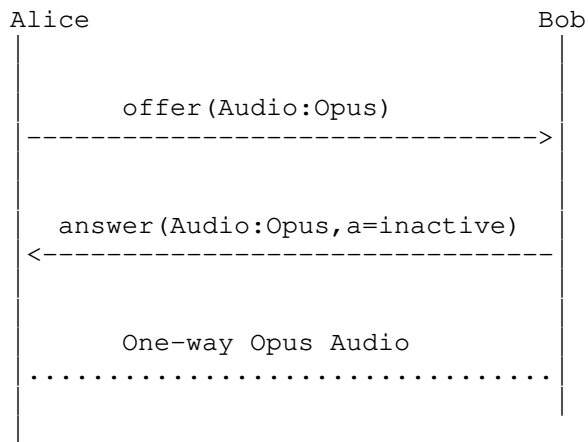
```

Table 8: 5.2.3 SDP Answer

5.2.4. Audio Call On Hold

Alice calls Bob, but when Bob answers he places Alice on hold by setting the SDP direction attribute to a=inactive in the answer.

Audio On Hold



```

=====
+-----+-----+
| Offer SDP Contents |
|                   | RFC#/Notes |
+-----+-----+
+-----+-----+
| v=0                | [RFC4566] |
+-----+-----+
+-----+-----+
| o=- 20518 0 IN IP4 0.0.0.0 |
|                   | [RFC4566] - Session Origin Information |
+-----+-----+
+-----+-----+
| s=-                | [RFC4566] |
+-----+-----+
+-----+-----+
| t=0 0              | [RFC4566] |
+-----+-----+
+-----+-----+
| a=group:BUNDLE audio |
|                   | [I-D.ietf-mmusic-sdp-bundle-negotiation] |
+-----+-----+
+-----+-----+
| a=ice-options:trickle |
|                   | [I-D.ietf-mmusic-trickle-ice] |
+-----+-----+
+-----+-----+
| a=ice-options:ice2   |
|                   | [RFC8445] |
+-----+-----+
+-----+-----+
| ***** Audio m=line ***** |
| ***** |
+-----+-----+
+-----+-----+
| m=audio 54609 UDP/TLS/RTP/SAVPF 109 |
|                   | [RFC4566] |
+-----+-----+

```

```

-----+-----
|c=IN IP4 203.0.113.141
| [RFC4566]
-----+-----
|a=mid:audio
| [RFC5888]
-----+-----
|a=msid:ma ta
| Identifies MediaStream ID (ma) and
|
| MediaStreamTrack ID (ta)
-----+-----
|a=sendrecv
| [RFC3264] - Alice can send and recv
|
| audio

```



```
+-----+
+-----+
|a=rtpmap:109 opus/48000/2
|           |[RFC7587] - Opus Codec 48 kHz, 2
|           |channels
+-----+
+-----+
|a=maxptime:120
|           |[RFC4566]
+-----+
+-----+
|a=ice-ufrag:074c6550
|           |[I-D.ietf-mmusic-ice-sip-sdp] - ICE user
|           |fragment
+-----+
+-----+
|a=ice-pwd:a28a397a4c3f31747dlee3474af08a068
|           |[I-D.ietf-mmusic-ice-sip-sdp] - ICE
|           |password
+-----+
+-----+
|a=fingerprint:sha-256
|           |[RFC8122] - DTLS Fingerprint for SRTP
|19:E2:1C:3B:4B:9F:81:E6:B8:5C:F4:A5:A8:D8:73:04:BB:05:2F:70:9F:04:A9:0E:05:E9
:26:33:E8:70:88:A2|
+-----+
+-----+
|a=setup:actpass
|           |[RFC5763] - Alice can act as DTLS client
|           |or server
+-----+
+-----+
|a=tls-id:89J2LRATQ3ULA24G9AHWVR31VJWSLB68
|           |[I-D.ietf-mmusic-dtls-sdp]
+-----+
+-----+
|a=rtcp-mux
|           |[RFC5761] - Alice can perform RTP/RTCP
|           |Muxing
+-----+
+-----+
|a=rtcp-mux-only
|           |[I-D.ietf-mmusic-mux-exclusive]
+-----+
+-----+
|a=rtcp-rsize
|           |[RFC5506]
+-----+
+-----+
|a=extmap:1 urn:ietf:params:rtp-hdext:ssrc-audio-level
|           |[RFC6464]
+-----+
+-----+
```

```

|a=extmap:2 urn:ietf:params:rtp-hdext:sdes:mid
| [I-D.ietf-mmusic-sdp-bundle-negotiation] |
+-----+
|a=candidate:0 1 UDP 2113667327 192.0.2.4 61665 typ host
| [I-D.ietf-mmusic-ice-sip-sdp] |
+-----+
|a=candidate:1 1 UDP 1685987071 203.0.113.141 54609 typ srflx raddr 192.0.2.4
rport 61665 | [I-D.ietf-mmusic-ice-sip-sdp] |
+-----+
|a=end-of-candidates
| [I-D.ietf-mmusic-trickle-ice] |
+-----+

```

Table 9: 5.2.4 SDP Offer

```

+=====+
|Answer SDP Contents
| RFC#/Notes |
+=====+
|v=0
| [RFC4566] |
+-----+
|o=- 16833 0 IN IP4 0.0.0.0
| [RFC4566] - Session Origin Information |
+-----+

```

```

|s=-
|                                     | [RFC4566]                                     |
+-----+-----+
|t=0 0
|                                     | [RFC4566]                                     |
+-----+-----+
|a=group:BUNDLE audio
|                                     | [I-D.ietf-mmusic-sdp-bundle-negotiation] |
+-----+-----+
|***** Audio m=line *****
|                                     | *****                                     |
+-----+-----+
|m=audio 49203 UDP/TLS/RTP/SAVPF 109
|                                     | [RFC4566]                                     |
+-----+-----+
|c=IN IP4 203.0.113.77
|                                     | [RFC4566]                                     |
+-----+-----+
|a=mid:audio
|                                     | [RFC5888]                                     |
+-----+-----+
|a=msid:ma ta
|                                     | Identifies MediaStream ID (ma) and       |
|                                     | MediaStreamTrack ID (ta)                 |
+-----+-----+
|a=inactive
|                                     | [RFC3264] - Bob puts call On Hold       |
+-----+-----+
|a=rtpmap:109 opus/48000/2
|                                     | [RFC7587] - Bob accepts Opus Codec      |
+-----+-----+
|a=maxptime:120
|                                     | [RFC4566]                                     |
+-----+-----+
|a=ice-ufrag:c300d85b
|                                     | [I-D.ietf-mmusic-ice-sip-sdp] - ICE     |
|                                     | username frag                             |
+-----+-----+
|a=ice-pwd:de4e99bd291c325921d5d47efbabd9a2
|                                     | [I-D.ietf-mmusic-ice-sip-sdp] - ICE     |
|                                     | password                                   |
+-----+-----+

```

```

|a=fingerprint:sha-256
| [RFC8122] - DTLS Fingerprint for SRTP |
|6B:8B:F0:65:5F:78:E2:51:3B:AC:6F:F3:3F:46:1B:35:DC:B8:5F:64:1A:24:C2:43:F0:A1
:58:D0:A1:2C:19:08|
+-----+
|a=setup:active
| [RFC5763] - Bob is the DTLS client |
+-----+
|a=tls-id:CJ6FF9ZZMJW7MDRJIR7XVIQM48GE1G31
| [I-D.ietf-mmusic-dtls-sdp] |
+-----+
|a=rtcp-mux
| [RFC5761] - Bob can perform RTP/RTCP |
| Muxing |
+-----+
|a=rtcp-mux-only
| [I-D.ietf-mmusic-mux-exclusive] |
+-----+
|a=rtcp-rsize
| [RFC5506] |
+-----+
|a=extmap:1 urn:ietf:params:rtp-hdext:ssrc-audio-level
| [RFC6464] |
+-----+
|a=extmap:2 urn:ietf:params:rtp-hdext:sdes:mid
| [I-D.ietf-mmusic-sdp-bundle-negotiation] |
+-----+
|a=candidate:0 1 UDP 2113667327 198.51.100.7 51556 typ host
| [I-D.ietf-mmusic-ice-sip-sdp] - Host |

```

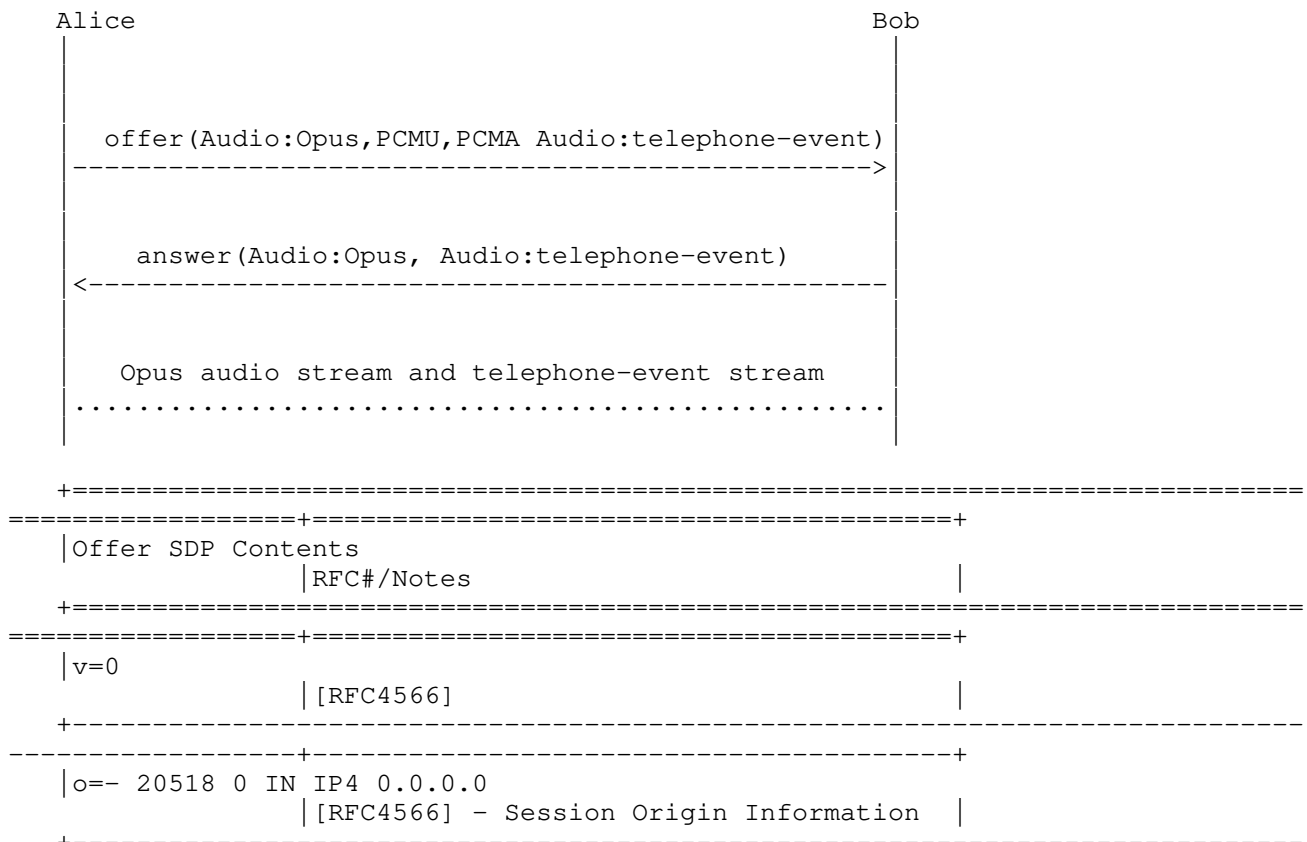
	candidate	
a=candidate:1 1 UDP 1685987071 203.0.113.141 49203 typ srflx raddr 198.51.100.7 rport 51556	[I-D.ietf-mmusic-ice-sip-sdp] - Server	
	Reflexive candidate	
a=end-of-candidates	[I-D.ietf-mmusic-trickle-ice]	

Table 10: 5.2.4 SDP Answer

5.2.5. Audio with DTMF Session

In this example, Alice wishes to establish two separate audio streams, one for normal audio and the other for telephone-events. Alice offers the first audio stream with three codecs and the other with [RFC4733] tones (for DTMF). Bob accepts both the audio streams by choosing Opus as the audio codec and telephone-event for the other stream.

Audio Session with DTMF



s=-	[RFC4566]	
t=0 0	[RFC4566]	

```

+-----+
+-----+
|a=group:BUNDLE audio dtmf
|                                     |[I-D.ietf-mmusic-sdp-bundle-negotiation]|
+-----+
+-----+
|a=ice-options:trickle
|                                     |[I-D.ietf-mmusic-trickle-ice]|
+-----+
+-----+
|a=ice-options:ice2
|                                     |[RFC8445]|
+-----+
+-----+
|***** Audio m=line *****
|                                     |*****
+-----+
+-----+
|m=audio 54609 UDP/TLS/RTP/SAVPF 109 0 8
|                                     |[RFC4566]|
+-----+
+-----+
|c=IN IP4 203.0.113.141
|                                     |[RFC4566]|
+-----+
+-----+
|a=mid:audio
|                                     |[RFC5888]|
+-----+
+-----+
|a=msid:ma ta
|                                     |Identifies MediaStream ID (ma) and
|                                     |MediaStreamTrack ID (ta)
+-----+
+-----+
|a=sendrecv
|                                     |[RFC3264] - Alice can send and recv
|                                     |audio
+-----+
+-----+
|a=rtpmap:109 opus/48000/2
|                                     |[RFC7587] - Opus Codec 48 kHz, 2
|                                     |channels
+-----+
+-----+
|a=rtpmap:0 PCMU/8000
|                                     |[RFC3551] PCMU audio Codec
+-----+
+-----+
|a=rtpmap:8 PCMA/8000
|                                     |[RFC3551] PCMA audio Codec
+-----+
+-----+
|a=maxptime:120
|                                     |[RFC4566]|

```

```

-----+-----
|a=ice-ufrag:074c6550
|           |[I-D.ietf-mmusic-ice-sip-sdp] - ICE user|
|           |fragment                               |
-----+-----
|a=ice-pwd:a28a397a4c3f31747dlee3474af08a068
|           |[I-D.ietf-mmusic-ice-sip-sdp] - ICE   |
|           |password parameter                   |
-----+-----
|a=fingerprint:sha-256
|           |[RFC8122] - DTLS Fingerprint for SRTP |
|19:E2:1C:3B:4B:9F:81:E6:B8:5C:F4:A5:A8:D8:73:04:BB:05:2F:70:9F:04:A9:0E:05:E9
:26:33:E8:70:88:A2|
-----+-----
|a=setup:actpass
|           |[RFC5763] - Alice can act as DTLS client|
|           |or server                               |
-----+-----
|a=tls-id:89J2LRATQ3ULA24G9AHWVR31VJWSLB68
|           |[I-D.ietf-mmusic-dtls-sdp]           |
-----+-----
|a=rtcp-mux
|           |[RFC5761] - Alice can perform RTP/RTCP |
|           |Muxing                               |
-----+-----
|a=rtcp-mux-only
|           |[I-D.ietf-mmusic-mux-exclusive]       |

```



```

+-----+
+-----+
|a=rtcp-rsize                               |
|                                             |
| [RFC5506]                                  |
+-----+
+-----+
|a=extmap:1 urn:iETF:params:rtp-hdext:ssrc-audio-level |
|                                             |
| [RFC6464]                                  |
+-----+
+-----+
|a=extmap:2 urn:iETF:params:rtp-hdext:sdes:mid        |
| [I-D.iETF-mmusic-sdp-bundle-negotiation] |
+-----+
+-----+
|a=candidate:0 1 UDP 2122194687 192.0.2.4 61665 typ host |
| [I-D.iETF-mmusic-ice-sip-sdp]                |
+-----+
+-----+
|a=candidate:1 1 UDP 1685987071 203.0.113.141 54609 typ srflx raddr 192.0.2.4 |
| rport 61665 [I-D.iETF-mmusic-ice-sip-sdp]      |
+-----+
+-----+
|a=end-of-candidates                          |
| [I-D.iETF-mmusic-trickle-ice]                |
+-----+
+-----+
|***** DTMF m=line *****                  |
|*****                                       |
+-----+
+-----+
|m=audio 0 UDP/TLS/RTP/SAVPF 126              |
| [RFC4566]                                    |
+-----+
+-----+
|c=IN IP4 203.0.113.141                       |
| [RFC4566]                                    |
+-----+
+-----+
|a=bundle-only                                |
| [I-D.iETF-mmusic-sdp-bundle-negotiation] |
+-----+
+-----+
|a=mid:dtmf                                    |
| [RFC5888]                                    |
+-----+
+-----+
|a=msid:ma tb                                  |
|                                             |
| Identifies MediaStream ID (ma) and         |
|                                             |
| MediaStreamTrack ID (tb)                   |
+-----+
+-----+
|a=sendonly                                    |
| [RFC3264] - Alice can send DTMF Events    |
+-----+
+-----+
|a=rtpmap:126 telephone-event/8000           |
| [RFC4733]                                    |

```

```

-----
|a=extmap:2 urn:iETF:params:rtp-hdext:sdes:mid
| [I-D.iETF-mmusic-sdp-bundle-negotiation] |
-----

```

Table 11: 5.2.5 SDP Offer

```

=====
|Answer SDP Contents
| RFC#/Notes |
=====
|v=0
| [RFC4566] |
-----
|o=- 16833 0 IN IP4 0.0.0.0
| [RFC4566] - Session Origin Information |
-----
|s=-
| [RFC4566] |
-----
|t=0 0
| [RFC4566] |
-----
|a=group:BUNDLE audio dtmf
| [I-D.iETF-mmusic-sdp-bundle-negotiation] |
-----

```

```

-----
| ***** Audio m=line *****
| *****
-----
|m=audio 49203 UDP/TLS/RTP/SAVPF 109
| [RFC4566]
-----
|c=IN IP4 203.0.113.77
| [RFC4566]
-----
|a=mid:audio
| [RFC5888]
-----
|a=msid:ma ta
| Identifies MediaStream ID (ma) and
| MediaStreamTrack ID (ta)
-----
|a=sendrecv
| [RFC3264] - Bob can send and receive
| Opus audio
-----
|a=rtpmap:109 opus/48000/2
| [RFC7587] - Bob accepts Opus Codec
-----
|a=maxptime:120
| [RFC4566]
-----
|a=ice-ufrag:c300d85b
| [I-D.ietf-mmusic-ice-sip-sdp] - ICE
| username frag
-----
|a=ice-pwd:de4e99bd291c325921d5d47efbabd9a2
| [I-D.ietf-mmusic-ice-sip-sdp] - ICE
| password
-----
|a=fingerprint:sha-256
| [RFC8122] - Fingerprint for SRTP
| 6B:8B:F0:65:5F:78:E2:51:3B:AC:6F:F3:3F:46:1B:35:DC:B8:5F:64:1A:24:C2:43:F0:A1
| :58:D0:A1:2C:19:08
-----
|a=setup:active
| [RFC5763] - Bob is the DTLS client

```

```

-----
|a=tls-id:CJ6FF9ZZMJW7MDRJIR7XVIQM48GE1G31
|                                     |[I-D.ietf-mmusic-dtls-sdp]
-----
|a=rtcp-mux
|                                     |[RFC5761] - Bob can perform RTP/RTCP
|                                     |Muxing on port 49203
-----
|a=rtcp-mux-only
|                                     |[I-D.ietf-mmusic-mux-exclusive]
-----
|a=rtcp-rsize
|                                     |[RFC5506] - Alice intends to use
|                                     |reduced-size RTCP for this session
-----
|a=extmap:1 urn:ietf:params:rtp-hdext:ssrc-audio-level
|                                     |[RFC6464]
-----
|a=extmap:2 urn:ietf:params:rtp-hdext:sdes:mid
|                                     |[I-D.ietf-mmusic-sdp-bundle-negotiation]
-----
|a=candidate:0 1 UDP 2122194687 198.51.100.7 51556 typ host
|                                     |[I-D.ietf-mmusic-ice-sip-sdp]
-----
|a=candidate:1 1 UDP 1685987071 203.0.113.77 49203 typ srflx raddr 198.51.100.
7 rport 51556 | [I-D.ietf-mmusic-ice-sip-sdp]
-----

```

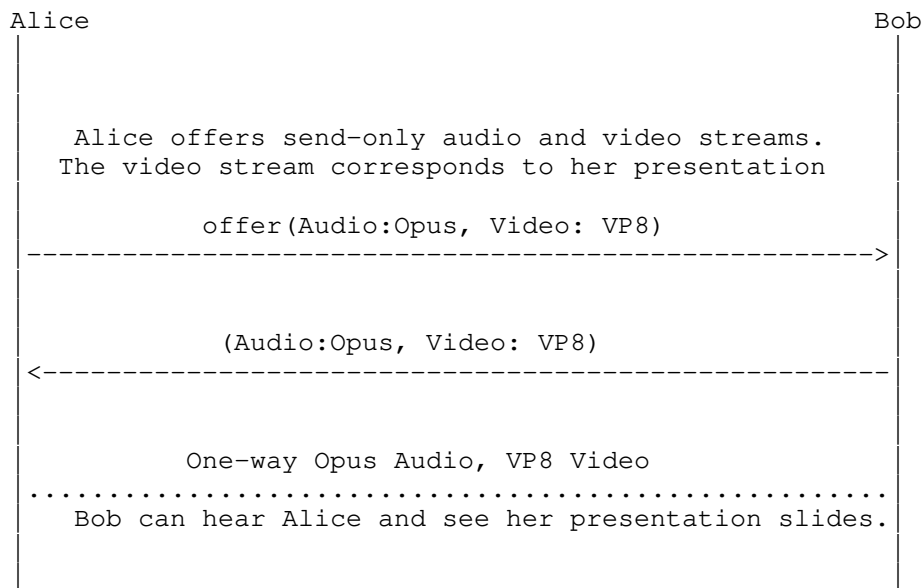
a=end-of-candidates	[I-D.ietf-mmusic-trickle-ice]	
+-----+-----+		
***** DTMF m=line *****	*****	
+-----+-----+		
m=audio 0 UDP/TLS/RTP/SAVPF 126	[RFC4566]	
+-----+-----+		
c=IN IP4 203.0.113.77	[RFC4566]	
+-----+-----+		
a=bundle-only	[I-D.ietf-mmusic-sdp-bundle-negotiation]	
+-----+-----+		
a=mid:dtmf	[RFC5888]	
+-----+-----+		
a=msid:ma tb	Identifies MediaStream ID (ma) and	
	MediaStreamTrack ID (tb)	
+-----+-----+		
a=recvonly	[RFC3264] - Alice can receive DTMF	
	events	
+-----+-----+		
a=rtpmap:126 telephone-event/8000	[RFC4733]	
+-----+-----+		
a=extmap:2 urn:ietf:params:rtp-hdext:sdes:mid	[I-D.ietf-mmusic-sdp-bundle-negotiation]	
+-----+-----+		

Table 12: 5.2.5 SDP Answer

5.2.6. One-way Audio/Video Session - Document Camera

In this scenario Alice and Bob engage in a One-way audio and video session with Bob receiving Alice’s audio and her presentation slides as video stream.

One-way Audio & Video Session - Document Camera



Offer SDP Contents	RFC#/Notes
v=0	[RFC4566]
o=- 20519 0 IN IP4 0.0.0.0	[RFC4566]
s=-	[RFC4566]
t=0 0	[RFC4566]
a=group:BUNDLE audio video	[I-D.ietf-mmusic-sdp-bundle-negotiation]
a=group:LS audio video	[RFC5888]
a=ice-options:trickle	[I-D.ietf-mmusic-trickle-ice]

```
|a=ice-options:ice2
| [RFC8445]
+-----+
+-----+
|***** Audio m=line *****
| *****
+-----+
+-----+
|m=audio 54609 UDP/TLS/RTP/SAVPF 109
| [RFC4566]
+-----+
+-----+
|c=IN IP4 203.0.113.141
| [RFC4566]
|
```



```

-----
|a=extmap:1 urn:iETF:params:rtp-hdext:ssrc-audio-level
| [RFC6464]
-----
|a=extmap:2 urn:iETF:params:rtp-hdext:sdes:mid
| [I-D.iETF-mmusic-sdp-bundle-negotiation]
-----
|a=candidate:0 1 UDP 2122194687 203.0.113.141 54609 typ host
| [I-D.iETF-mmusic-ice-sip-sdp]
-----
|a=end-of-candidates
| [I-D.iETF-mmusic-trickle-ice]
-----
|***** Video m=line *****
| *****
-----
|m=video 0 UDP/TLS/RTP/SAVPF 120
| [RFC4566]
-----
|c=IN IP4 203.0.113.141
| [RFC4566]
-----
|a=bundle-only
| [I-D.iETF-mmusic-sdp-bundle-negotiation]
-----
|a=mid:video
| [RFC5888]
-----

```

a=msid:ma tb	Identifies MediaStream ID (ma) and	
	MediaStreamTrack ID (tb)	
+-----+		
a=sendonly	[RFC3264] - Send-only video stream	
+-----+		
a=rtpmap:120 VP8/90000	[RFC7741]	
+-----+		
a=content:slides	[RFC4796] - Alice's presentation video	
	stream	
+-----+		
a=rtcp-fb:120 nack	[RFC4585]	
+-----+		
a=rtcp-fb:120 nack pli	[RFC4585]	
+-----+		
a=rtcp-fb:120 ccm fir	[RFC5104]	
+-----+		
a=extmap:2 urn:iETF:params:rtp-hdext:sdes:mid	[I-D.iETF-mmusic-sdp-bundle-negotiation]	
+-----+		

Table 13: 5.2.6 SDP Offer

+=====+		
Answer SDP Contents	RFC#/Notes	
+=====+		
v=0	[RFC4566]	
+-----+		
o=- 16833 0 IN IP4 0.0.0.0	[RFC4566]	
+-----+		
s=-	[RFC4566]	
+-----+		

```

|t=0 0
| [RFC4566]
+-----+
|a=group:BUNDLE audio video
| [I-D.ietf-mmusic-sdp-bundle-negotiation]
+-----+
|a=group:LS audio video
| [RFC5888]
+-----+
|a=ice-options:trickle
| [I-D.ietf-mmusic-trickle-ice]
+-----+
|a=ice-options:ice2
| [RFC8445]
+-----+
|***** Audio m=line *****
|*****
+-----+
|m=audio 49203 UDP/TLS/RTP/SAVPF 109
| [RFC4566]
+-----+
|c=IN IP4 203.0.113.77
| [RFC4566]
+-----+
|a=mid:audio
| [RFC5888]

```

```

+-----+
+-----+
|a=msid:ma ta          | Identifies MediaStream ID (ma) and |
|                      | MediaStreamTrack ID (ta)          |
+-----+
+-----+
|a=recvonly           | [RFC3264] - Receive-only audio stream |
+-----+
+-----+
|a=rtpmap:109 opus/48000/2 | [RFC7587]                          |
+-----+
+-----+
|a=maxptime:120       | [RFC4566]                          |
+-----+
+-----+
|a=ice-ufrag:c300d85b | [I-D.ietf-mmusic-ice-sip-sdp]      |
+-----+
+-----+
|a=ice-pwd:de4e99bd291c325921d5d47efbabd9a2 | [I-D.ietf-mmusic-ice-sip-sdp]      |
+-----+
+-----+
|a=fingerprint:sha-256 | [RFC8122]                          |
|6B:8B:F0:65:5F:78:E2:51:3B:AC:6F:F3:3F:46:1B:35:DC:B8:5F:64:1A:24:C2:43:F0:A1 |
|:58:D0:A1:2C:19:08    |                                         |
+-----+
+-----+
|a=setup:active       | [RFC5763] - Bob is the DTLS client  |
+-----+
+-----+
|a=tls-id:CJ6FF9ZZMJW7MDRJIR7XVIQM48GE1G31 | [I-D.ietf-mmusic-dtls-sdp]        |
+-----+
+-----+
|a=rtcp-mux           | [RFC5761]                          |
+-----+
+-----+
|a=rtcp-mux-only      | [I-D.ietf-mmusic-mux-exclusive]     |
+-----+
+-----+
|a=extmap:1 urn:ietf:params:rtp-hdext:ssrc-audio-level | [RFC6464]                          |
+-----+
+-----+
|a=extmap:2 urn:ietf:params:rtp-hdext:sdes:mid | [I-D.ietf-mmusic-sdp-bundle-negotiation] |
+-----+
+-----+

```

```

|a=candidate:0 1 UDP 2113667327 203.0.113.77 49203 typ host
| [I-D.ietf-mmusic-ice-sip-sdp] |
+-----+
|a=end-of-candidates
| [I-D.ietf-mmusic-trickle-ice] |
+-----+
|***** Video m=line *****
| ***** |
+-----+
|m=video 0 UDP/TLS/RTP/SAVPF 120
| [RFC4566] |
+-----+
|c=IN IP4 203.0.113.77
| [RFC4566] |
+-----+
|a=bundle-only
| [I-D.ietf-mmusic-sdp-bundle-negotiation] |
+-----+
|a=mid:video
| [RFC5888] |
+-----+
|a=msid:ma tb
| Identifies MediaStream ID (ma) and |
| MediaStreamTrack ID (tb) |
+-----+
|a=recvonly
| [RFC3264] |
+-----+

```

a=rtpmap:120 VP8/90000 [RFC7741]	
-----+-----	-----+-----
a=content:slides [RFC4796] - presentation stream	
-----+-----	-----+-----
a=rtcp-fb:120 nack [RFC4585]	
-----+-----	-----+-----
a=rtcp-fb:120 nack pli [RFC4585]	
-----+-----	-----+-----
a=rtcp-fb:120 ccm fir [RFC5104]	
-----+-----	-----+-----
a=extmap:2 urn:iETF:params:rtp-hdrext:sdes:mid [I-D.iETF-mmusic-sdp-bundle-negotiation]	
-----+-----	-----+-----

Table 14: 5.2.6 SDP Answer

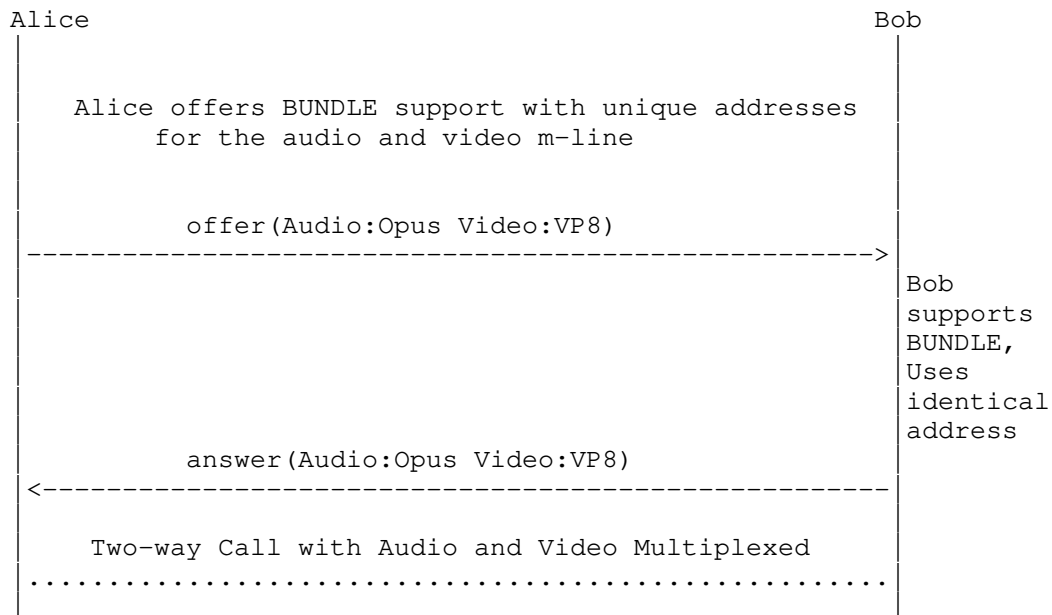
5.2.7. Audio, Video Session with BUNDLE Support Unknown

In this example, since Alice is unsure of the Bob's support of the BUNDLE framework, the following steps are performed in order to negotiate and set up a BUNDLE Address for the session.

- * An SDP offer, in which Alice assigns unique addresses to each "m=" line in the BUNDLE group, and requests the answerer to select the offerer's BUNDLE address.
- * An SDP answer, in which Bob indicates support for BUNDLE, selects the offerer's BUNDLE address, selects its own BUNDLE address and associates it with each BUNDLED m=line within the BUNDLE group.

Once the offer/answer exchange completes, both Alice and Bob each end up using a single RTP session for both of the media streams.

Two-way Secure Audio and Video with BUNDLE support unknown



Offer SDP Contents	RFC#/Notes
v=0	[RFC4566]
o=- 20518 0 IN IP4 0.0.0.0	[RFC4566]
s=-	[RFC4566]
t=0 0	[RFC4566]
a=group:BUNDLE audio video	[I-D.ietf-mmusic-sdp-bundle-negotiation]
	Alice supports grouping of m=lines under BUNDLE semantics
a=group:LS audio video	[RFC5888]

```

|a=ice-options:trickle
| [I-D.ietf-mmusic-trickle-ice] |
+-----+
|a=ice-options:ice2
| [RFC8445] |
+-----+
|***** Audio m=line *****
| ***** |
+-----+
|m=audio 54609 UDP/TLS/RTP/SAVPF 109
| [RFC4566] |

```

```
+-----+
+-----+
|c=IN IP4 203.0.113.141
| [RFC4566]
+-----+
+-----+
|a=mid:audio
| [RFC5888] Audio m=line part of BUNDLE
|
| group with a unique port number
+-----+
+-----+
|a=msid:ma ta
| Identifies MediaStream ID (ma) and
|
| MediaStreamTrack ID (ta)
+-----+
+-----+
|a=sendrecv
| [RFC3264]
+-----+
+-----+
|a=rtpmap:109 opus/48000/2
| [RFC7587]
+-----+
+-----+
|a=maxptime:120
| [RFC4566]
+-----+
+-----+
|a=ice-ufrag:074c6550
| [I-D.ietf-mmusic-ice-sip-sdp]
+-----+
+-----+
|a=ice-pwd:a28a397a4c3f31747dlee3474af08a068
| [I-D.ietf-mmusic-ice-sip-sdp]
+-----+
+-----+
|a=fingerprint:sha-256
| [RFC8122]
| 19:E2:1C:3B:4B:9F:81:E6:B8:5C:F4:A5:A8:D8:73:04:BB:05:2F:70:9F:04:A9:0E:05:E9
| :26:33:E8:70:88:A2
+-----+
+-----+
|a=setup:actpass
| [RFC5763] - Alice can act as DTLS client
|
| or server
+-----+
+-----+
|a=tls-id:89J2LRATQ3ULA24G9AHWVR31VJWSLB68
| [I-D.ietf-mmusic-dtls-sdp]
+-----+
+-----+
|a=rtcp-mux
| [RFC5761]
+-----+
+-----+
```

```

|a=rtcp:54610 IN IP4 203.0.113.141
|
| [RFC3605] - RTCP port different from RTP |
|
| Port |
+-----+
|a=rtcp-rsize
| [RFC5506] |
+-----+
|a=extmap:1 urn:iETF:params:rtp-hdext:ssrc-audio-level
| [RFC6464] |
+-----+
|a=extmap:2 urn:iETF:params:rtp-hdext:sdes:mid
| [I-D.iETF-mmusic-sdp-bundle-negotiation] |
+-----+
|a=candidate:0 1 UDP 2122194687 192.0.2.4 61665 typ host
| [I-D.iETF-mmusic-ice-sip-sdp] - RTP host |
|
| candidate |
+-----+
|a=candidate:1 1 UDP 1685987071 203.0.113.141 54609 typ srflx raddr 192.0.2.4
rport 61665 | [I-D.iETF-mmusic-ice-sip-sdp] - RTP |
|
| Server Reflexive candidate |
+-----+
|a=candidate:0 2 UDP 2122194687 192.0.2.4 61666 typ host
| [I-D.iETF-mmusic-ice-sip-sdp] - RTCP |
|
| host candidate |
+-----+
|a=candidate:1 2 UDP 1685987071 203.0.113.141 54610 typ srflx raddr 192.0.2.4
rport 61666 | [I-D.iETF-mmusic-ice-sip-sdp] - RTCP |

```

```

|
|           |Server Reflexive candidate           |
+-----+-----+
|***** Video m=line *****|
|*****|
+-----+-----+
|m=video 62537 UDP/TLS/RTP/SAVPF 120
|           |[RFC4566]           |
+-----+-----+
|c=IN IP4 203.0.113.141
|           |[RFC4566]           |
+-----+-----+
|a=mid:video
|           |[RFC5888] Video m=line part of the
|           |Bundle group with a unique port number
+-----+-----+
|a=msid:ma tb
|           |Identifies MediaStream ID (ma) and
|           |MediaStreamTrack ID (tb)
+-----+-----+
|a=sendrecv
|           |[RFC3264]           |
+-----+-----+
|a=rtpmap:120 VP8/90000
|           |[RFC7741]           |
+-----+-----+
|a=ice-ufrag:6550074c
|           |[I-D.ietf-mmusic-ice-sip-sdp]
+-----+-----+
|a=ice-pwd:74af08a068a28a397a4c3f31747dlee34
|           |[I-D.ietf-mmusic-ice-sip-sdp]
+-----+-----+
|a=fingerprint:sha-256
|           |[RFC8122]           |
|19:E2:1C:3B:4B:9F:81:E6:B8:5C:F4:A5:A8:D8:73:04:BB:05:2F:70:9F:04:A9:0E:05:E9
:26:33:E8:70:88:A2|
+-----+-----+
|a=setup:actpass
|           |[RFC5763] - Alice can act as DTLS client
|           |or server
+-----+-----+
|a=tls-id:UKA29UQLTF690JW4WNPNUO2Y0GF1FJOZ
|           |[I-D.ietf-mmusic-dtls-sdp]

```

```

-----
|a=rtcp-mux                               |
|                                           | [RFC5761] |
-----
|a=rtcp:62538 IN IP4 203.0.113.141       |
|                                           | [RFC3605] |
-----
|a=rtcp-rsize                              |
|                                           | [RFC5506] |
-----
|a=rtcp-fb:120 nack                       |
|                                           | [RFC4585] |
-----
|a=rtcp-fb:120 nack pli                  |
|                                           | [RFC4585] |
-----
|a=rtcp-fb:120 ccm fir                   |
|                                           | [RFC5104] |
-----
|a=extmap:2 urn:iETF:params:rtp-hdrext:sdes:mid
|                                           | [I-D.ietf-mmusic-sdp-bundle-negotiation] |
-----
|a=candidate:0 1 UDP 2122194687 192.0.2.4 61886 typ host
|                                           | [I-D.ietf-mmusic-ice-sip-sdp] - RTP Host |
|                                           | candidate |
-----
|a=candidate:1 1 UDP 1685987071 203.0.113.141 62537 typ srflx raddr 192.0.2.4
rport 61886                               | [I-D.ietf-mmusic-ice-sip-sdp] - RTP |

```

```

|
|           |Server Reflexive candidate           |
+-----+-----+
|a=candidate:0 2 UDP 2122194687 192.0.2.4 61888 typ host
|           |[I-D.ietf-mmusic-ice-sip-sdp] - RTCP   |
|           |host candidate                       |
+-----+-----+
|a=candidate:1 2 UDP 1685987071 203.0.113.141 62538 typ srflx raddr 192.0.2.4
rport 61888  |[I-D.ietf-mmusic-ice-sip-sdp] - RTCP   |
|           |Server Reflexive candidate           |
+-----+-----+

```

Table 15: 5.2.7 SDP Offer w/BUNDLE

```

+=====+
+-----+-----+
|Answer SDP Contents
|           |RFC#/Notes           |
+-----+-----+
|v=0
|           |[RFC4566]         |
+-----+-----+
|o=- 16833 0 IN IP4 0.0.0.0
|           |[RFC4566]         |
+-----+-----+
|s=-
|           |[RFC4566]         |
+-----+-----+
|t=0 0
|           |[RFC4566]         |
+-----+-----+
|a=group:BUNDLE audio video
|           |[I-D.ietf-mmusic-sdp-bundle-negotiation] |
|           |Bob supports BUNDLE semantics.           |
+-----+-----+
|a=group:LS audio video
|           |[RFC5888]         |
+-----+-----+
|a=ice-options:trickle
|           |[I-D.ietf-mmusic-trickle-ice]       |
+-----+-----+
|a=ice-options:ice2
|           |[RFC8445]         |
+-----+-----+

```

```

+-----+
| ***** Audio m=line ***** |
| ***** |
+-----+
|m=audio 49203 UDP/TLS/RTP/SAVPF 109 |
| [RFC4566] |
+-----+
|c=IN IP4 203.0.113.77 |
| [RFC4566] |
+-----+
|a=mid:audio |
| [RFC5888] Audio m=line part of the |
| BUNDLE group |
+-----+
|a=msid:ma ta |
| Identifies MediaStream ID (ma) and |
| MediaStreamTrack ID (ta) |
+-----+
|a=sendrecv |
| [RFC3264] |
+-----+
|a=rtpmap:109 opus/48000/2 |
| [RFC7587] |
+-----+

```



```

|a=maxptime:120
| [RFC4566]
+-----+
|a=ice-ufrag:c300d85b
| [I-D.ietf-mmusic-ice-sip-sdp]
+-----+
|a=ice-pwd:de4e99bd291c325921d5d47efbabd9a2
| [I-D.ietf-mmusic-ice-sip-sdp]
+-----+
|a=fingerprint:sha-256
| [RFC8122]
|6B:8B:F0:65:5F:78:E2:51:3B:AC:6F:F3:3F:46:1B:35:DC:B8:5F:64:1A:24:C2:43:F0:A1
:58:D0:A1:2C:19:08
+-----+
|a=setup:active
| [RFC5763] - Bob is the DTLS client
+-----+
|a=tls-id:CJ6FF9ZZMJW7MDRJIR7XVIQM48GE1G31
| [I-D.ietf-mmusic-dtls-sdp]
+-----+
|a=rtcp-mux
| [RFC5761]
+-----+
|a=rtcp-rsize
| [RFC5506]
+-----+
|a=extmap:1 urn:ietf:params:rtp-hdext:ssrc-audio-level
| [RFC6464]
+-----+
|a=extmap:2 urn:ietf:params:rtp-hdext:sdes:mid
| [I-D.ietf-mmusic-sdp-bundle-negotiation]
+-----+
|a=candidate:0 1 UDP 2122194687 198.51.100.7 49203 typ host
| [I-D.ietf-mmusic-ice-sip-sdp]
+-----+
|a=candidate:1 1 UDP 1685987071 203.0.113.77 51556 typ srflx raddr 198.51.100.
7 rport 49203
| [I-D.ietf-mmusic-ice-sip-sdp]
+-----+
|***** Video m=line *****
|*****
+-----+
|m=video 0 UDP/TLS/RTP/SAVPF 120
| [RFC4566]
+-----+

```

```

|c=IN IP4 203.0.113.77
| [RFC4566]
+-----+
|a=bundle-only
| [I-D.ietf-mmusic-sdp-bundle-negotiation]
+-----+
|a=mid:video
| [RFC5888] Video m=line part of the
|
| BUNDLE group with the port from audio
|
| line repeated
+-----+
|a=msid:ma tb
| Identifies MediaStream ID (ma) and
|
| MediaStreamTrack ID (tb)
+-----+
|a=sendrecv
| [RFC3264]
+-----+
|a=rtpmap:120 VP8/90000
| [RFC7741]
+-----+
|a=rtcp-fb:120 nack
| [RFC4585]
+-----+
|a=rtcp-fb:120 nack pli
| [RFC4585]
+-----+

```

```

|a=rtcp-fb:120 ccm fir
| [RFC5104]
+-----+
|a=extmap:2 urn:ietf:params:rtp-hdext:sdes:mid
| [I-D.ietf-mmusic-sdp-bundle-negotiation]
+-----+

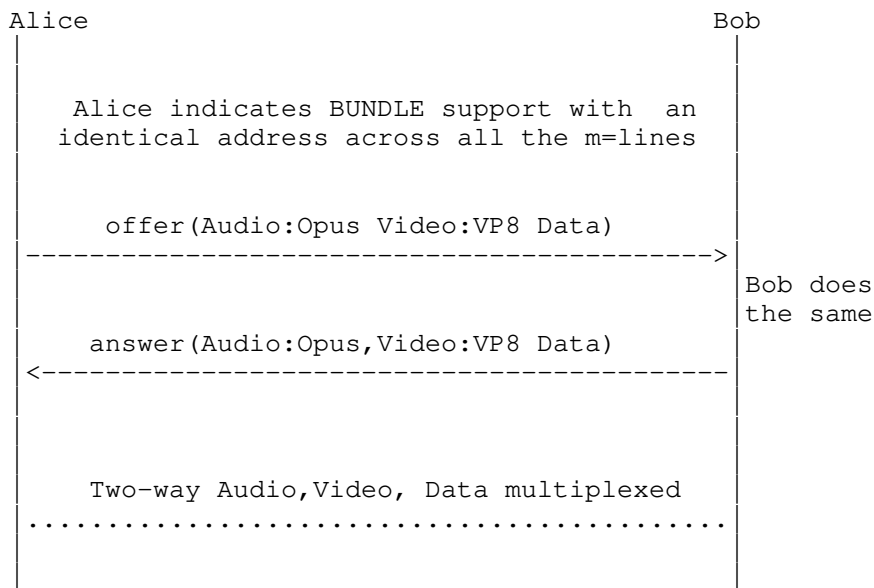
```

Table 16: 5.2.7 SDP Answer w/BUNDLE

5.2.8. Audio, Video, and Data Session

This example shows SDP for negotiating a session with Audio, Video and data streams between Alice and Bob with BUNDLE support known.

Audio, Video, and Data with BUNDLE support known



```

+=====+
|Offer SDP Contents
| RFC#/Notes
+=====+
|v=0
| [RFC4566]
+-----+
|o=- 20518 0 IN IP4 0.0.0.0
| [RFC4566]
+-----+
|s=-
| [RFC4566]
+-----+

```

```
|t=0 0  
| [RFC4566] |  
+-----+  
-----+-----+  
|a=group:BUNDLE audio video data  
| [I-D.ietf-mmusic-sdp-bundle-negotiation] |
```

```
+-----+
+-----+
|a=group:LS audio video
|           | [RFC5888]
+-----+
+-----+
|a=ice-options:trickle
|           | [I-D.ietf-mmusic-trickle-ice]
+-----+
+-----+
|a=ice-options:ice2
|           | [RFC8445]
+-----+
+-----+
|***** Audio m=line *****
|           | *****
+-----+
+-----+
|m=audio 54609 UDP/TLS/RTP/SAVPF 109
|           | [RFC4566]
+-----+
+-----+
|c=IN IP4 203.0.113.141
|           | [RFC4566]
+-----+
+-----+
|a=msid:ma ta
|           | Identifies MediaStream ID (ma) and
|           | MediaStreamTrack ID (ta)
+-----+
+-----+
|a=mid:audio
|           | [RFC5888]
+-----+
+-----+
|a=sendrecv
|           | [RFC3264]
+-----+
+-----+
|a=rtpmap:109 opus/48000/2
|           | [RFC7587]
+-----+
+-----+
|a=maxptime:120
|           | [RFC4566]
+-----+
+-----+
|a=ice-ufrag:074c6550
|           | [I-D.ietf-mmusic-ice-sip-sdp]
+-----+
+-----+
|a=ice-pwd:a28a397a4c3f31747dlee3474af08a068
|           | [I-D.ietf-mmusic-ice-sip-sdp]
+-----+
+-----+
|a=fingerprint:sha-256
|           | [RFC8122]
+-----+
```

```

|19:E2:1C:3B:4B:9F:81:E6:B8:5C:F4:A5:A8:D8:73:04:BB:05:2F:70:9F:04:A9:0E:05:E9
:26:33:E8:70:88:A2|
+-----+
|a=setup:actpass
| [RFC5763]
+-----+
|a=tls-id:89J2LRATQ3ULA24G9AHWVR31VJWSLB68
| [I-D.ietf-mmusic-dtls-sdp]
+-----+
|a=rtcp-mux
| [RFC5761]
+-----+
|a=rtcp-mux-only
| [I-D.ietf-mmusic-mux-exclusive]
+-----+
|a=rtcp-rsize
| [RFC5506]
+-----+
|a=extmap:1 urn:ietf:params:rtp-hdext:ssrc-audio-level
| [RFC6464]
+-----+
|a=extmap:2 urn:ietf:params:rtp-hdext:sdes:mid
| [I-D.ietf-mmusic-sdp-bundle-negotiation]
+-----+
|a=candidate:0 1 UDP 2122194687 192.0.2.4 61665 typ host
| [I-D.ietf-mmusic-ice-sip-sdp]
+-----+
|a=candidate:1 1 UDP 1685987071 203.0.113.141 54609 typ srflx raddr 192.0.2.4
rport 61665 | [I-D.ietf-mmusic-ice-sip-sdp]

```

```
+-----+
+-----+
|a=end-of-candidates
|           |[I-D.ietf-mmusic-trickle-ice]
+-----+
+-----+
|***** Video m=line *****
|           |*****
+-----+
+-----+
|m=video 0 UDP/TLS/RTP/SAVPF 120
|           |[RFC4566]
+-----+
+-----+
|c=IN IP4 203.0.113.141
|           |[RFC4566]
+-----+
+-----+
|a=bundle-only
|           |[I-D.ietf-mmusic-sdp-bundle-negotiation]
+-----+
+-----+
|a=mid:video
|           |[RFC5888]
+-----+
+-----+
|a=msid:ma tb
|           |Identifies MediaStream ID (ma) and
|           |MediaStreamTrack ID (tb)
+-----+
+-----+
|a=sendrecv
|           |[RFC3264]
+-----+
+-----+
|a=rtpmap:120 VP8/90000
|           |[RFC7741]
+-----+
+-----+
|a=rtcp-fb:120 nack
|           |[RFC4585]
+-----+
+-----+
|a=rtcp-fb:120 nack pli
|           |[RFC4585]
+-----+
+-----+
|a=rtcp-fb:120 ccm fir
|           |[RFC5104]
+-----+
+-----+
|a=extmap:2 urn:iETF:params:rtp-hdext:sdes:mid
|           |[I-D.ietf-mmusic-sdp-bundle-negotiation]
+-----+
+-----+
|***** Application m=line *****
|           |*****
+-----+
```

```

+-----+
|m=application 0 UDP/DTLS/SCTP webrtc-datachannel
|           | [I-D.ietf-rtcweb-data-channel] |
+-----+
|c=IN IP4 203.0.113.141
|           | [RFC4566] |
+-----+
|a=bundle-only
|           | [I-D.ietf-mmusic-sdp-bundle-negotiation] |
+-----+
|a=mid:data
|           | [RFC5888] |
+-----+
|a=sctp-port:5000
|           | [I-D.ietf-mmusic-sctp-sdp] |
+-----+
|a=max-message-size:100000
|           | [I-D.ietf-mmusic-sctp-sdp] |
+-----+
|a=sendrecv
|           | [RFC3264] |
+-----+

```

Table 17: 5.2.8 SDP Offer


```

=====
| Answer SDP Contents
|                               | RFC#/Notes |
=====
| v=0
|                               | [RFC4566] |
-----
| o=- 16833 0 IN IP4 0.0.0.0
|                               | [RFC4566] - Session Origin Information |
-----
| s=-
|                               | [RFC4566] |
-----
| t=0 0
|                               | [RFC4566] |
-----
| a=group:BUNDLE audio video data
|                               | [I-D.ietf-mmusic-sdp-bundle-negotiation] |
-----
| a=group:LS audio video
|                               | [RFC5888] |
-----
| a=ice-options:trickle
|                               | [I-D.ietf-mmusic-trickle-ice] |
-----
| a=ice-options:ice2
|                               | [RFC8445] |
-----
| ***** Audio m=line *****
|                               | ***** |
-----
| m=audio 49203 UDP/TLS/RTP/SAVPF 109
|                               | [RFC4566] |
-----
| c=IN IP4 203.0.113.77
|                               | [RFC4566] |
-----
| a=msid:ma ta
|                               | Identifies MediaStream ID (ma) and |
|                               | MediaStreamTrack ID (ta) |
-----
| a=mid:audio
|                               | [RFC5888] |

```

```

-----
|a=sendrecv                               |
|                                           | [RFC3264] |
-----
|a=rtpmap:109 opus/48000/2                |
|                                           | [RFC7587] |
-----
|a=maxptime:120                            |
|                                           | [RFC4566] |
-----
|a=ice-ufrag:c300d85b                     |
|                                           | [I-D.ietf-mmusic-ice-sip-sdp] |
-----
|a=ice-pwd:de4e99bd291c325921d5d47efbabd9a2 |
|                                           | [I-D.ietf-mmusic-ice-sip-sdp] |
-----
|a=fingerprint:sha-256                    |
|                                           | [RFC8122] |
|6B:8B:F0:65:5F:78:E2:51:3B:AC:6F:F3:3F:46:1B:35:DC:B8:5F:64:1A:24:C2:43:F0:A1 |
|:58:D0:A1:2C:19:08                       |
-----
|a=setup:active                            |
|                                           | [RFC5763] |
-----
|a=tls-id:CJ6FF9ZZMJW7MDRJIR7XVIQM48GE1G31 |
|                                           | [I-D.ietf-mmusic-dtls-sdp] |
-----
|a=rtcp-mux                                |
|                                           | [RFC5761] |

```

```

+-----+
+-----+
|a=rtcp-mux-only
|                                     | [I-D.ietf-mmusic-mux-exclusive] |
+-----+
+-----+
|a=rtcp-rsize
|                                     | [RFC5506] |
+-----+
+-----+
|a=extmap:1 urn:ietf:params:rtp-hdext:ssrc-audio-level
|                                     | [RFC6464] |
+-----+
+-----+
|a=extmap:2 urn:ietf:params:rtp-hdext:sdes:mid
|                                     | [I-D.ietf-mmusic-sdp-bundle-negotiation] |
+-----+
+-----+
|a=candidate:0 1 UDP 2122194687 198.51.100.7 51556 typ host
|                                     | [I-D.ietf-mmusic-ice-sip-sdp] |
+-----+
+-----+
|a=candidate:1 1 UDP 1685987071 203.0.113.77 49203 typ srflx raddr 198.51.100.
7 rport 51556
|                                     | [I-D.ietf-mmusic-ice-sip-sdp] |
+-----+
+-----+
|a=end-of-candidates
|                                     | [I-D.ietf-mmusic-trickle-ice] |
+-----+
+-----+
|***** Video m=line *****
|*****
+-----+
+-----+
|m=video 0 UDP/TLS/RTP/SAVPF 120
|                                     | [RFC4566] |
+-----+
+-----+
|c=IN IP4 203.0.113.77
|                                     | [RFC4566] |
+-----+
+-----+
|a=bundle-only
|                                     | [I-D.ietf-mmusic-sdp-bundle-negotiation] |
+-----+
+-----+
|a=mid:video
|                                     | [RFC5888] |
+-----+
+-----+
|a=msid:ma tb
|                                     | Identifies MediaStream ID (ma) and |
|                                     | MediaStreamTrack ID (tb) |
+-----+
+-----+
|a=sendrecv
|                                     | [RFC3264] |

```

```

+-----+
|a=rtpmap:120 VP8/90000
| [RFC7741]
+-----+
|a=rtcp-fb:120 nack
| [RFC4585]
+-----+
|a=rtcp-fb:120 nack pli
| [RFC4585]
+-----+
|a=rtcp-fb:120 ccm fir
| [RFC5104]
+-----+
|a=extmap:2 urn:ietf:params:rtp-hdext:sdes:mid
| [I-D.ietf-mmusic-sdp-bundle-negotiation]
+-----+
|***** Application m=line *****
|*****
+-----+
|m=application 0 UDP/DTLS/SCTP webrtc-datachannel
| [I-D.ietf-mmusic-sctp-sdp]
+-----+
|c=IN IP4 203.0.113.77
| [RFC4566]
+-----+
|a=bundle-only
| [I-D.ietf-mmusic-sdp-bundle-negotiation]
+-----+

```

a=mid:data	[RFC5888]	
+-----+-----+		
a=sctp-port:5000	[I-D.ietf-mmusic-sctp-sdp]	
+-----+-----+		
a=max-message-size:100000	[I-D.ietf-mmusic-sctp-sdp]	
+-----+-----+		
a=sendrecv	[RFC3264]	
+-----+-----+		

Table 18: 5.2.8 SDP Answer

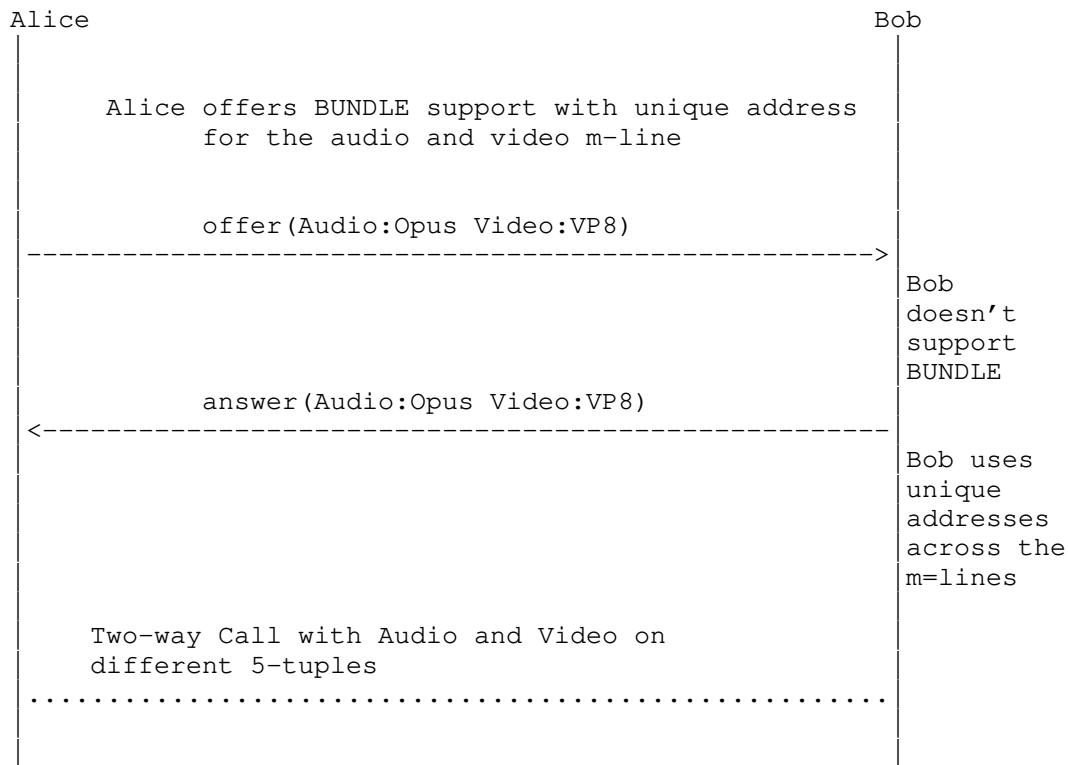
5.2.9. Audio and Video Session with BUNDLE Unsupported

This use case illustrates SDP offer/answer exchange where the far-end (Bob) either doesn't support media bundling or doesn't want to group m=lines over a single 5-tuple.

This is indicated by dropping the "a=group:BUNDLE" line and BUNDLE RTP header extension in the answer SDP.

On successful offer/answer exchange, Alice and Bob each end up using unique 5-tuple for audio and video media streams respectively.

Two-way Secure Audio and Video with BUNDLE Unsupported



Offer SDP Contents		RFC#/Notes
v=0		[RFC4566]
o=- 20518 0 IN IP4 0.0.0.0		[RFC4566]
s=-		[RFC4566]
t=0 0		[RFC4566]
a=group:BUNDLE audio video	[[I-D.ietf-mmusic-sdp-bundle-negotiation]]	
	Alice supports grouping of m=lines under	
	BUNDLE semantics	

```
-----+-----+
|a=group:LS audio video
|          |[RFC5888]          |
+-----+-----+
|a=ice-options:trickle
|          |[I-D.ietf-mmusic-trickle-ice]          |
+-----+-----+
-----+-----+
```



```

|a=ice-options:ice2
| [RFC8445]
+-----+
+-----+
|***** Audio m=line *****
| *****
+-----+
+-----+
|m=audio 54609 UDP/TLS/RTP/SAVPF 109
| [RFC4566]
+-----+
+-----+
|c=IN IP4 203.0.113.141
| [RFC4566]
+-----+
+-----+
|a=mid:audio
| [RFC5888] Audio m=line part of BUNDLE
|
| group with a unique port number
+-----+
+-----+
|a=msid:ma ta
| Identifies MediaStream ID (ma) and
|
| MediaStreamTrack ID (ta)
+-----+
+-----+
|a=sendrecv
| [RFC3264]
+-----+
+-----+
|a=rtpmap:109 opus/48000/2
| [RFC7587]
+-----+
+-----+
|a=maxptime:120
| [RFC4566]
+-----+
+-----+
|a=ice-ufrag:074c6550
| [I-D.ietf-mmusic-ice-sip-sdp]
+-----+
+-----+
|a=ice-pwd:a28a397a4c3f31747dlee3474af08a068
| [I-D.ietf-mmusic-ice-sip-sdp]
+-----+
+-----+
|a=fingerprint:sha-256
| [RFC8122]
| 19:E2:1C:3B:4B:9F:81:E6:B8:5C:F4:A5:A8:D8:73:04:BB:05:2F:70:9F:04:A9:0E:05:E9
:26:33:E8:70:88:A2
+-----+
+-----+
|a=setup:actpass
| [RFC5763] - Alice can act as DTLS client
|
| or server

```

```

-----
|a=tls-id:89J2LRATQ3ULA24G9AHWVR31VJWSLB68
| [I-D.ietf-mmusic-dtls-sdp]
-----
|a=rtcp-mux
| [RFC5761]
-----
|a=rtcp:55232 IN IP4 203.0.113.141
| [RFC3605] - RTCP port different from RTP
|
| port
-----
|a=rtcp-rsize
| [RFC5506]
-----
|a=extmap:1 urn:ietf:params:rtp-hdext:ssrc-audio-level
| [RFC6464]
-----
|a=extmap:2 urn:ietf:params:rtp-hdext:sdes:mid
| [I-D.ietf-mmusic-sdp-bundle-negotiation]
-----
|a=candidate:0 1 UDP 2122194687 192.0.2.4 61665 typ host
| [I-D.ietf-mmusic-ice-sip-sdp]
-----
|a=candidate:1 1 UDP 1685987071 203.0.113.141 54609 typ srflx raddr 192.0.2.4
rport 61665 | [I-D.ietf-mmusic-ice-sip-sdp]
-----
|a=candidate:0 2 UDP 2122194687 192.0.2.4 61666 typ host
| [I-D.ietf-mmusic-ice-sip-sdp]
-----

```

```
+-----+
+-----+
|a=candidate:1 2 UDP 1685987071 203.0.113.141 55232 typ srflx raddr 192.0.2.4
rport 61666 | [I-D.ietf-mmusic-ice-sip-sdp] |
+-----+
+-----+
|a=end-of-candidates
| | [I-D.ietf-mmusic-trickle-ice] |
+-----+
+-----+
|***** Video m=line *****
|***** |
+-----+
+-----+
|m=video 54332 UDP/TLS/RTP/SAVPF 120
| | [RFC4566] |
+-----+
+-----+
|c=IN IP4 203.0.113.141
| | [RFC4566] |
+-----+
+-----+
|a=mid:video
| | [RFC5888] Video m=line part of the |
| | |
| | BUNDLE group with a unique port number |
+-----+
+-----+
|a=msid:ma tb
| | Identifies MediaStream ID (ma) and |
| | |
| | MediaStreamTrack ID (tb) |
+-----+
+-----+
|a=sendrecv
| | [RFC3264] |
+-----+
+-----+
|a=rtpmap:120 VP8/90000
| | [RFC7741] |
+-----+
+-----+
|a=ice-ufrag:7872093
| | [I-D.ietf-mmusic-ice-sip-sdp] |
+-----+
+-----+
|a=ice-pwd:ee3474af08a068a28a397a4c3f31747d1
| | [I-D.ietf-mmusic-ice-sip-sdp] |
+-----+
+-----+
|a=fingerprint:sha-256
| | [RFC8122] |
|19:E2:1C:3B:4B:9F:81:E6:B8:5C:F4:A5:A8:D8:73:04:BB:05:2F:70:9F:04:A9:0E:05:E9
:26:33:E8:70:88:A2|
+-----+
+-----+
|a=setup:actpass
| | [RFC5763] - Alice can act as DTLS client |
```

```

|
|                               |or server                               |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|a=tls-id:UKA29UQLTF690JW4WNPNUO2Y0GF1FJOZ
|                               |[I-D.ietf-mmusic-dtls-sdp]           |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|a=rtcp-mux
|                               |[RFC5761]                           |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|a=rtcp:60052 IN IP4 203.0.113.141
|                               |[RFC3605]                           |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|a=rtcp-rsize
|                               |[RFC5506]                           |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|a=rtcp-fb:120 nack
|                               |[RFC4585]                           |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|a=rtcp-fb:120 nack pli
|                               |[RFC4585]                           |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|a=rtcp-fb:120 ccm fir
|                               |[RFC5104]                           |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|a=extmap:2 urn:ietf:params:rtp-hdrext:sdes:mid
|                               |[I-D.ietf-mmusic-sdp-bundle-negotiation]|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|a=candidate:0 1 UDP 2122194687 192.0.2.4 71775 typ host
|                               |[I-D.ietf-mmusic-ice-sip-sdp]       |

```

```

+-----+
+-----+
|a=candidate:1 1 UDP 1685987071 203.0.113.141 54332 typ srflx raddr 192.0.2.4
rport 71775 | [I-D.ietf-mmusic-ice-sip-sdp] |
+-----+
+-----+
|a=candidate:0 2 UDP 2122194687 192.0.2.4 71776 typ host
| [I-D.ietf-mmusic-ice-sip-sdp] |
+-----+
+-----+
|a=candidate:1 2 UDP 1685987071 203.0.113.141 60052 typ srflx raddr 192.0.2.4
rport 71776 | [I-D.ietf-mmusic-ice-sip-sdp] |
+-----+
+-----+
|a=end-of-candidates
| [I-D.ietf-mmusic-trickle-ice] |
+-----+
+-----+

```

Table 19: 5.2.9 SDP Offer w/BUNDLE

```

+=====+
+=====+
|Answer SDP Contents
|RFC#/Notes |
+=====+
+=====+
|v=0
| [RFC4566] |
+-----+
+-----+
|o=- 16833 0 IN IP4 0.0.0.0
| [RFC4566] |
+-----+
+-----+
|s=-
| [RFC4566] |
+-----+
+-----+
|t=0 0
| [RFC4566] |
+-----+
+-----+
|a=group:LS audio video
| [RFC5888] |
+-----+
+-----+
|a=ice-options:trickle
| [I-D.ietf-mmusic-trickle-ice] |
+-----+
+-----+
|a=ice-options:ice2
| [RFC8445] |
+-----+
+-----+
|***** Audio m=line *****
|*****|

```

```

+-----+
|m=audio 53214 UDP/TLS/RTP/SAVPF 109
| [RFC4566]
+-----+
|c=IN IP4 203.0.113.77
| [RFC4566]
+-----+
|a=mid:audio
| [RFC5888]
+-----+
|a=msid:ma ta
| Identifies MediaStream ID
| (ma) and MediaStreamTrack ID
| (ta)
+-----+
|a=sendrecv
| [RFC3264]
+-----+
|a=rtpmap:109 opus/48000/2
| [RFC7587]
+-----+
|a=maxptime:120
| [RFC4566]
+-----+

```

```
|a=ice-ufrag:c300d85b
| [I-D.ietf-mmusic-ice-sip-sdp] |
+-----+
+-----+
|a=ice-pwd:de4e99bd291c325921d5d47efbabd9a2
| [I-D.ietf-mmusic-ice-sip-sdp] |
+-----+
+-----+
|a=fingerprint:sha-256
| [RFC8122] |
|6B:8B:F0:65:5F:78:E2:51:3B:AC:6F:F3:3F:46:1B:35:DC:B8:5F:64:1A:24:C2:43:F0:A1
:58:D0:A1:2C:19:08|
+-----+
+-----+
|a=setup:active
| [RFC5763] - Bob is the DTLS |
| client |
+-----+
+-----+
|a=tls-id:CJ6FF9ZZMJW7MDRJIR7XVIQM48GE1G31
| [I-D.ietf-mmusic-dtls-sdp] |
+-----+
+-----+
|a=rtcp-mux
| [RFC5761] |
+-----+
+-----+
|a=rtcp-rsize
| [RFC5506] |
+-----+
+-----+
|a=extmap:1 urn:ietf:params:rtp-hdext:ssrc-audio-level
| [RFC6464] |
+-----+
+-----+
|a=candidate:0 1 UDP 2122194687 198.51.100.7 51556 typ host
| [I-D.ietf-mmusic-ice-sip-sdp] |
+-----+
+-----+
|a=candidate:1 1 UDP 1685987071 203.0.113.77 53214 typ srflx raddr 198.51.100.
7 rport 51556 | [I-D.ietf-mmusic-ice-sip-sdp] |
+-----+
+-----+
|a=candidate:0 2 UDP 2122194687 198.51.100.7 51558 typ host
| [I-D.ietf-mmusic-ice-sip-sdp] |
+-----+
+-----+
|a=candidate:1 2 UDP 1685987071 203.0.113.77 60065 typ srflx raddr 198.51.100.
7 rport 51558 | [I-D.ietf-mmusic-ice-sip-sdp] |
+-----+
+-----+
|***** Video m=line *****
| ***** |
+-----+
+-----+
|m=video 58679 UDP/TLS/RTP/SAVPF 120
| [RFC4566] |
```

```

-----+-----
|c=IN IP4 203.0.113.77
| [RFC4566]
-----+-----
|a=mid:video
| [RFC5888]
-----+-----
|a=msid:ma tb
| Identifies MediaStream ID
|
| (ma) and MediaStreamTrack ID
|
| (tb)
-----+-----
|a=sendrecv
| [RFC3264]
-----+-----
|a=rtpmap:120 VP8/90000
| [RFC7741]
-----+-----
|a=ice-ufrag:85bC300
| [I-D.ietf-mmusic-ice-sip-sdp]
-----+-----
|a=ice-pwd:325921d5d47efbabd9a2de4e99bd291c
| [I-D.ietf-mmusic-ice-sip-sdp]
-----+-----
|a=fingerprint:sha-256 6B:8B:F0:65:5F:78:E2:51:3B:AC:6F:F3:3F:46:1B:35
| [RFC8122]
|:DC:B8:5F:64:1A:24:C2:43:F0:A1:58:D0:A1:2C:19:08
|

```



```

+-----+
+-----+
|a=setup:active
|           |[RFC5763] - Bob is the DTLS |
|           |client                       |
+-----+
+-----+
|a=tls-id:9AIFS8AQ009IXF5D6QQUJ7P8BXPEZJ8G
|           |[I-D.ietf-mmusic-dtls-sdp] |
+-----+
+-----+
|a=rtcp-mux
|           |[RFC5761]                   |
+-----+
+-----+
|a=rtcp-rsize
|           |[RFC5506]                   |
+-----+
+-----+
|a=rtcp-fb:120 nack
|           |[RFC4585]                   |
+-----+
+-----+
|a=rtcp-fb:120 nack pli
|           |[RFC4585]                   |
+-----+
+-----+
|a=rtcp-fb:120 ccm fir
|           |[RFC5104]                   |
+-----+
+-----+
|a=candidate:0 1 UDP 2122194687 198.51.100.7 61556 typ host
|           |[I-D.ietf-mmusic-ice-sip-sdp]|
+-----+
+-----+
|a=candidate:1 1 UDP 1685987071 203.0.113.77 58679 typ srflx raddr 198.51.100.
7 rport 61556 |           |[I-D.ietf-mmusic-ice-sip-sdp]|
+-----+
+-----+
|a=end-of-candidates
|           |[I-D.ietf-mmusic-trickle-ice]|
+-----+
+-----+

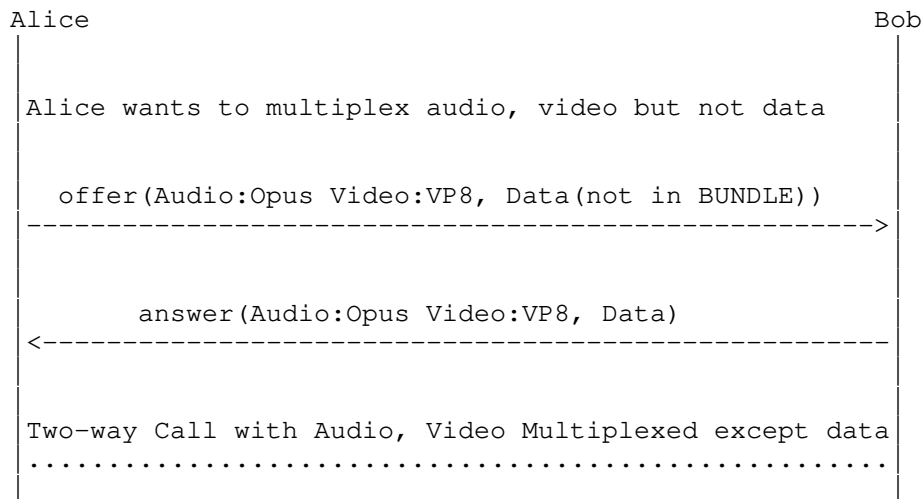
```

Table 20: 5.2.9 SDP Answer without BUNDLE

5.2.10. Audio, Video BUNDLED, but Data Not BUNDLED

This example showcases SDP for negotiating a session with audio, video, and the data streams between Alice and Bob with data stream not being part of the BUNDLE group. This is shown by assigning a unique port for the data media section and not adding the "mid" identification tag to the BUNDLE group.

Audio, Video, with Data (Not in BUNDLE)



Offer SDP Contents	RFC#/Notes
v=0	[RFC4566]
o=- 20518 0 IN IP4 0.0.0.0	[RFC4566]
s=-	[RFC4566]
t=0 0	[RFC4566]
a=group:BUNDLE audio video	[I-D.ietf-mmusic-sdp-bundle-negotiation]
	Alice wants to BUNDLE only audio and
	video media.
a=group:LS audio video	[RFC5888]
a=ice-options:trickle	[I-D.ietf-mmusic-trickle-ice]

```

|a=ice-options:ice2
| [RFC8445]
+-----+
|***** Audio m=line *****
|*****
+-----+
|m=audio 54609 UDP/TLS/RTP/SAVPF 109
| [RFC4566]
+-----+
|c=IN IP4 203.0.113.141
| [RFC4566]
+-----+
|a=mid:audio
| [RFC5888]

```

```

+-----+
+-----+
|a=msid:ma ta          | Identifies MediaStream ID (ma) and |
|                      | MediaStreamTrack ID (ta)          |
+-----+
+-----+
|a=sendrecv            | [RFC3264]                          |
+-----+
+-----+
|a=rtpmap:109 opus/48000/2 | [RFC7587]                          |
+-----+
+-----+
|a=maxptime:120        | [RFC4566]                          |
+-----+
+-----+
|a=ice-ufrag:074c6550   | [I-D.ietf-mmusic-ice-sip-sdp]      |
+-----+
+-----+
|a=ice-pwd:a28a397a4c3f31747dlee3474af08a068 | [I-D.ietf-mmusic-ice-sip-sdp]      |
+-----+
+-----+
|a=fingerprint:sha-256  | [RFC8122]                          |
|19:E2:1C:3B:4B:9F:81:E6:B8:5C:F4:A5:A8:D8:73:04:BB:05:2F:70:9F:04:A9:0E:05:E9 |
|:26:33:E8:70:88:A2    |                                         |
+-----+
+-----+
|a=setup:actpass       | [RFC5763]                          |
+-----+
+-----+
|a=tls-id:89J2LRATQ3ULA24G9AHWVR31VJWSLB68  | [I-D.ietf-mmusic-dtls-sdp]        |
+-----+
+-----+
|a=rtcp-mux            | [RFC5761]                          |
+-----+
+-----+
|a=rtcp-mux-only      | [I-D.ietf-mmusic-mux-exclusive]    |
+-----+
+-----+
|a=rtcp-rsize         | [RFC5506]                          |
+-----+
+-----+
|a=extmap:1 urn:ietf:params:rtp-hdext:ssrc-audio-level | [RFC6464]                          |
+-----+
+-----+

```

```

|a=extmap:2 urn:iETF:params:rtp-hdext:sdes:mid
| [I-D.iETF-mmusic-sdp-bundle-negotiation] |
+-----+
|a=candidate:0 1 UDP 2113667327 192.0.2.4 54609 typ host
| [I-D.iETF-mmusic-ice-sip-sdp] |
+-----+
|a=end-of-candidates
| [I-D.iETF-mmusic-trickle-ice] |
+-----+
|***** Video m=line *****
| ***** |
+-----+
|m=video 0 UDP/TLS/RTP/SAVPF 120
| [RFC4566] |
+-----+
|c=IN IP4 203.0.113.141
| [RFC4566] |
+-----+
|a=bundle-only
| [I-D.iETF-mmusic-sdp-bundle-negotiation] |
+-----+
|a=mid:video
| [RFC5888] |
+-----+
|a=msid:ma tb
| Identifies MediaStream ID (ma) and |
| |
| MediaStreamTrack ID (tb) |
+-----+

```



```

+-----+
|a=ice-ufrag:89819013
| [I-D.ietf-mmusic-ice-sip-sdp]
+-----+
|a=ice-pwd:1747d1ee3474af08a068a28a397a4c3f3
| [I-D.ietf-mmusic-ice-sip-sdp]
+-----+
|a=fingerprint:sha-256
| [RFC8122]
|29:E2:1C:3B:4B:9F:81:E6:B8:5C:F4:A5:A8:D8:73:04:BB:05:2F:70:9F:04:A9:0E:05:E9
:26:33:E8:70:88:A2|
+-----+
|a=candidate:0 1 UDP 2113667327 192.0.2.4 10000 typ host
| [I-D.ietf-mmusic-ice-sip-sdp]
+-----+
|a=end-of-candidates
| [I-D.ietf-mmusic-trickle-ice]
+-----+

```

Table 21: 5.2.10 SDP Offer


```

-----+-----
|a=sendrecv                               |
| [RFC3264]                               |
-----+-----
|a=rtpmap:109 opus/48000/2                |
| [RFC7587]                               |
-----+-----
|a=maxptime:120                           |
| [RFC4566]                               |
-----+-----
|a=ice-ufrag:c300d85b                    |
| [I-D.ietf-mmusic-ice-sip-sdp]         |
-----+-----
|a=ice-pwd:de4e99bd291c325921d5d47efbabd9a2
| [I-D.ietf-mmusic-ice-sip-sdp]         |
-----+-----
|a=fingerprint:sha-256                   |
| [RFC8122]                               |
|6B:8B:F0:65:5F:78:E2:51:3B:AC:6F:F3:3F:46:1B:35:DC:B8:5F:64:1A:24:C2:43:F0:A1
:58:D0:A1:2C:19:08|
-----+-----
|a=setup:active                           |
| [RFC5763]                               |
-----+-----
|a=tls-id:CJ6FF9ZZMJW7MDRJIR7XVIQM48GE1G31
| [I-D.ietf-mmusic-dtls-sdp]           |
-----+-----
|a=rtcp-mux                               |
| [RFC5761]                               |

```

```

+-----+
+-----+
|a=rtcp-mux-only
|                               |[I-D.ietf-mmusic-mux-exclusive]
+-----+
+-----+
|a=rtcp-rsize
|                               |[RFC5506]
+-----+
+-----+
|a=extmap:1 urn:ietf:params:rtp-hdext:ssrc-audio-level
|                               |[RFC6464]
+-----+
+-----+
|a=extmap:2 urn:ietf:params:rtp-hdext:sdes:mid
|                               |[I-D.ietf-mmusic-sdp-bundle-negotiation]
+-----+
+-----+
|a=candidate:0 1 UDP 2113667327 198.51.100.7 49203 typ host
|                               |[I-D.ietf-mmusic-ice-sip-sdp]
+-----+
+-----+
|a=end-of-candidates
|                               |[I-D.ietf-mmusic-trickle-ice]
+-----+
+-----+
|***** Video m=line *****
|                               |*****
+-----+
+-----+
|m=video 0 UDP/TLS/RTP/SAVPF 120
|                               |[RFC4566]
+-----+
+-----+
|c=IN IP4 203.0.113.77
|                               |[RFC4566]
+-----+
+-----+
|a=bundle-only
|                               |[I-D.ietf-mmusic-sdp-bundle-negotiation]
+-----+
+-----+
|a=mid:video
|                               |[RFC5888]
+-----+
+-----+
|a=msid:ma tb
|                               |Identifies MediaStream ID (ma) and
|                               |MediaStreamTrack ID (tb)
+-----+
+-----+
|a=sendrecv
|                               |[RFC3264]
+-----+
+-----+
|a=rtpmap:120 VP8/90000
|                               |[RFC7741]

```

```

+-----+
|a=rtcp-fb:120 nack
| [RFC4585]
+-----+
|a=rtcp-fb:120 nack pli
| [RFC4585]
+-----+
|a=rtcp-fb:120 ccm fir
| [RFC5104]
+-----+
|a=extmap:2 urn:ietf:params:rtp-hdrext:sdes:mid
| [I-D.ietf-mmusic-sdp-bundle-negotiation]
+-----+
|***** Application m=line *****
|*****
+-----+
|m=application 20000 UDP/DTLS/SCTP webrtc-datachannel
| [I-D.ietf-mmusic-sctp-sdp]
+-----+
|c=IN IP4 203.0.113.77
| [RFC4566]
+-----+
|a=mid:data
| [RFC5888]
+-----+
|a=sctp-port:5000
| [I-D.ietf-mmusic-sctp-sdp]
+-----+

```

```

|a=max-message-size:100000
| [I-D.ietf-mmusic-sctp-sdp] |
+-----+
|a=setup:active
| [RFC5763] |
+-----+
|a=tls-id:9AIFS8AQ009IXF5D6QQUJ7P8BXPEZJ8G
| [I-D.ietf-mmusic-dtls-sdp] |
+-----+
|a=sendrecv
| [RFC3264] |
+-----+
|a=ice-ufrag:991Ca2a5e
| [I-D.ietf-mmusic-ice-sip-sdp] |
+-----+
|a=ice-pwd:921d5d47efbabd9a2de4e99bd291c325
| [I-D.ietf-mmusic-ice-sip-sdp] |
+-----+
|a=fingerprint:sha-256
| [RFC8122] |
|7B:8B:F0:65:5F:78:E2:51:3B:AC:6F:F3:3F:46:1B:35:DC:B8:5F:64:1A:24:C2:43:F0:A1
:58:D0:A1:2C:19:08| |
+-----+
|a=candidate:0 1 UDP 2113667327 198.51.100.7 20000 typ host
| [I-D.ietf-mmusic-ice-sip-sdp] |
+-----+
|a=end-of-candidates
| [I-D.ietf-mmusic-trickle-ice] |
+-----+

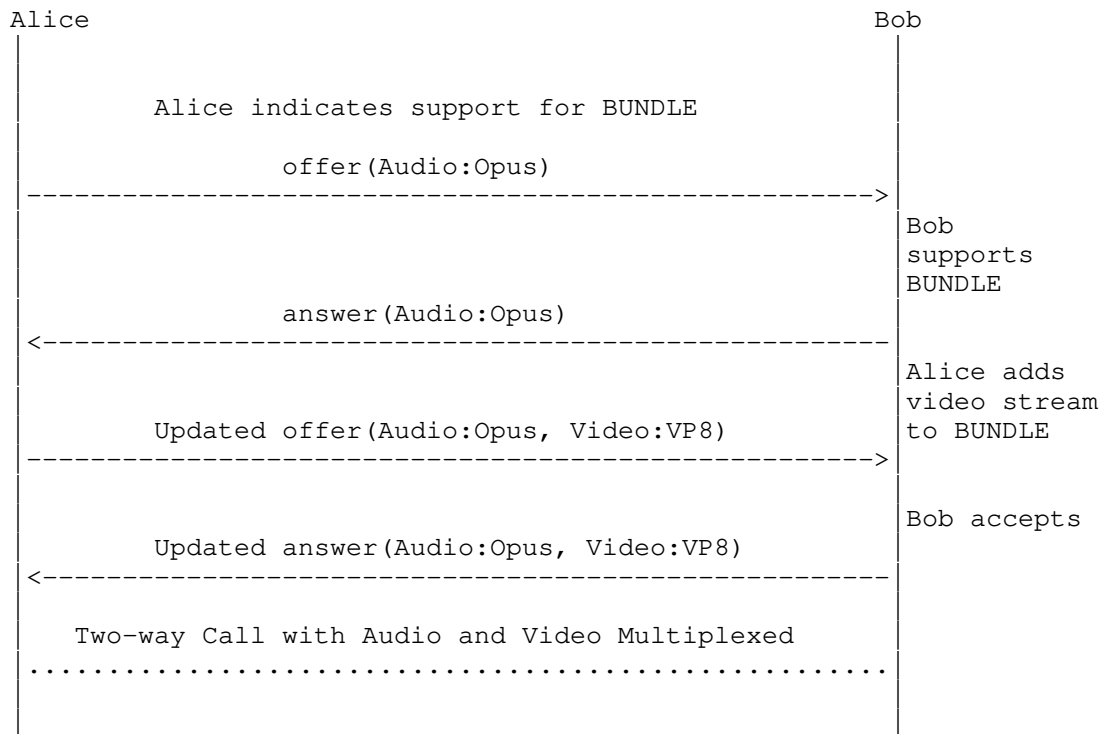
```

Table 22: 5.2.10 SDP Answer

5.2.11. Audio Only, Add Video to BUNDLE

This example involves 2 offer/answer exchanges. The first one is used to negotiate and set up BUNDLE support for audio-only session followed by an updated offer/answer exchange to add video stream to the ongoing session. Also the newly added video stream is BUNDLED with the audio stream.

Audio, Add Video and BUNDLE



Offer SDP Contents	
	RFC#/Notes
v=0	[RFC4566]
o=- 20518 0 IN IP4 0.0.0.0	[RFC4566]
s=-	[RFC4566]
t=0 0	[RFC4566]
a=group:BUNDLE audio	[I-D.ietf-mmusic-sdp-bundle-negotiation]
	Alice adds audio m=line to the BUNDLE
	group

```
|a=ice-options:trickle
| [I-D.ietf-mmusic-trickle-ice] |
+-----+
|a=ice-options:ice2
| [RFC8445] |
+-----+
|***** Audio m=line *****
| ***** |
```



```
+-----+
+-----+
|m=audio 54609 UDP/TLS/RTP/SAVPF 109
|      | [RFC4566]
+-----+
+-----+
|c=IN IP4 203.0.113.141
|      | [RFC4566]
+-----+
+-----+
|a=mid:audio
|      | [RFC5888]
+-----+
+-----+
|a=msid:ma ta
|      | Identifies MediaStream ID (ma) and
|      | MediaStreamTrack ID (ta)
+-----+
+-----+
|a=sendrecv
|      | [RFC3264]
+-----+
+-----+
|a=rtpmap:109 opus/48000/2
|      | [RFC7587]
+-----+
+-----+
|a=maxptime:120
|      | [RFC4566]
+-----+
+-----+
|a=ice-ufrag:074c6550
|      | [I-D.ietf-mmusic-ice-sip-sdp]
+-----+
+-----+
|a=ice-pwd:a28a397a4c3f31747dlee3474af08a068
|      | [I-D.ietf-mmusic-ice-sip-sdp]
+-----+
+-----+
|a=fingerprint:sha-256
|      | [RFC8122]
|19:E2:1C:3B:4B:9F:81:E6:B8:5C:F4:A5:A8:D8:73:04:BB:05:2F:70:9F:04:A9:0E:05:E9
:26:33:E8:70:88:A2|
+-----+
+-----+
|a=setup:actpass
|      | [RFC5763]
+-----+
+-----+
|a=tls-id:89J2LRATQ3ULA24G9AHWVR31VJWSLB68
|      | [I-D.ietf-mmusic-dtls-sdp]
+-----+
+-----+
|a=rtcp-mux
|      | [RFC5761]
+-----+
+-----+
```

```

|a=rtcp-mux-only
| [I-D.ietf-mmusic-mux-exclusive]
+-----+
|a=rtcp-rsize
| [RFC5506]
+-----+
|a=extmap:1 urn:ietf:params:rtp-hdext:ssrc-audio-level
| [RFC6464]
+-----+
|a=extmap:2 urn:ietf:params:rtp-hdext:sdes:mid
| [I-D.ietf-mmusic-sdp-bundle-negotiation]
+-----+
|a=candidate:0 1 UDP 2113667327 192.0.2.4 61665 typ host
| [I-D.ietf-mmusic-ice-sip-sdp]
+-----+
|a=candidate:1 1 UDP 694302207 203.0.113.141 54609 typ srflx raddr 192.0.2.4 r
port 61665 | [I-D.ietf-mmusic-ice-sip-sdp]
+-----+
|a=end-of-candidates
| [I-D.ietf-mmusic-trickle-ice]
+-----+

```

Table 23: 5.2.11 SDP Offer

```

=====
| Answer SDP Contents
| RFC#/Notes
|
=====
| v=0
| [RFC4566]
|
-----
| o=- 16833 0 IN IP4 0.0.0.0
| [RFC4566] - Session Origin Information
|
-----
| s=-
| [RFC4566]
|
-----
| t=0 0
| [RFC4566]
|
-----
| a=group:BUNDLE audio
| [I-D.ietf-mmusic-sdp-bundle-negotiation]
|
-----
| a=ice-options:trickle
| [I-D.ietf-mmusic-trickle-ice]
|
-----
| a=ice-options:ice2
| [RFC8445]
|
-----
| ***** Audio m=line *****
| *****
|
-----
| m=audio 49203 UDP/TLS/RTP/SAVPF 109
| [RFC4566]
|
-----
| c=IN IP4 203.0.113.77
| [RFC4566]
|
-----
| a=mid:audio
| [RFC5888]
|
-----
| a=msid:ma ta
| Identifies MediaStream ID (ma) and
|
| MediaStreamTrack ID (ta)
|
-----
| a=sendrecv
| [RFC3264]
|

```

```

-----+
|a=rtpmap:109 opus/48000/2
| [RFC7587] |
-----+
|a=maxptime:120
| [RFC4566] |
-----+
|a=ice-ufrag:c300d85b
| [I-D.ietf-mmusic-ice-sip-sdp] |
-----+
|a=ice-pwd:de4e99bd291c325921d5d47efbabd9a2
| [I-D.ietf-mmusic-ice-sip-sdp] |
-----+
|a=fingerprint:sha-256
| [RFC8122] |
|6B:8B:F0:65:5F:78:E2:51:3B:AC:6F:F3:3F:46:1B:35:DC:B8:5F:64:1A:24:C2:43:F0:A1
:58:D0:A1:2C:19:08|
-----+
|a=setup:active
| [RFC5763] |
-----+
|a=tls-id:CJ6FF9ZZMJW7MDRJIR7XVIQM48GE1G31
| [I-D.ietf-mmusic-dtls-sdp] |
-----+
|a=rtcp-mux
| [RFC5761] |
-----+
|a=rtcp-mux-only
| [I-D.ietf-mmusic-mux-exclusive] |

```

```

+-----+
+-----+
|a=rtcp-rsize          | [RFC5506]          |
+-----+
+-----+
|a=extmap:1 urn:ietf:params:rtp-hdext:ssrc-audio-level      |
|                   | [RFC6464]          |
+-----+
+-----+
|a=extmap:2 urn:ietf:params:rtp-hdext:sdes:mid              |
|                   | [I-D.ietf-mmusic-sdp-bundle-negotiation] |
+-----+
+-----+
|a=candidate:0 1 UDP 2113667327 198.51.100.7 51556 typ host |
|                   | [I-D.ietf-mmusic-ice-sip-sdp]          |
+-----+
+-----+
|a=candidate:1 1 UDP 1694302207 203.0.113.77 49203 typ srflx raddr 198.51.100.7 rport 51556 |
|                   | [I-D.ietf-mmusic-ice-sip-sdp]          |
+-----+
+-----+
|a=end-of-candidates  | [I-D.ietf-mmusic-trickle-ice] |
+-----+
+-----+

```

Table 24: 5.2.10 SDP Answer

```

=====
+-----+
|Updated Offer SDP Contents |
|                   | RFC#/Notes          |
+-----+
+-----+
|v=0                    | [RFC4566]          |
+-----+
+-----+
|o=- 20518 1 IN IP4 0.0.0.0 |
|                   | Version number incremented [RFC4566] |
+-----+
+-----+
|s=-                    | [RFC4566]          |
+-----+
+-----+
|t=0 0                  | [RFC4566]          |
+-----+
+-----+
|a=group:BUNDLE audio video |
|                   | [I-D.ietf-mmusic-sdp-bundle-negotiation] |
+-----+
+-----+
|a=group:LS audio video   |
|                   | [RFC5888]          |
+-----+

```

```

+-----+
|a=ice-options:trickle
|                               |[I-D.ietf-mmusic-trickle-ice]
+-----+
|a=ice-options:ice2
|                               |[RFC8445]
+-----+
|***** Audio m=line *****
|*****
+-----+
|m=audio 54609 UDP/TLS/RTP/SAVPF 109
|                               |[RFC4566]
+-----+
|c=IN IP4 203.0.113.141
|                               |[RFC4566]
+-----+
|a=mid:audio
|                               |[RFC5888]
+-----+
|a=msid:ma ta
|                               |Identifies MediaStream ID (ma) and
|                               |
|                               |MediaStreamTrack ID (ta)
+-----+
|a=sendrecv
|                               |[RFC3264]
+-----+

```

```
+-----+
+-----+
|a=rtpmap:109 opus/48000/2
|                                     | [RFC7587] |
+-----+
+-----+
|a=maxptime:120
|                                     | [RFC4566] |
+-----+
+-----+
|a=ice-ufrag:074c6550
|                                     | [I-D.ietf-mmusic-ice-sip-sdp] |
+-----+
+-----+
|a=ice-pwd:a28a397a4c3f31747dlee3474af08a068
|                                     | [I-D.ietf-mmusic-ice-sip-sdp] |
+-----+
+-----+
|a=fingerprint:sha-256
|                                     | [RFC8122] |
|19:E2:1C:3B:4B:9F:81:E6:B8:5C:F4:A5:A8:D8:73:04:BB:05:2F:70:9F:04:A9:0E:05:E9
:26:33:E8:70:88:A2|
+-----+
+-----+
|a=setup:actpass
|                                     | [RFC5763] |
+-----+
+-----+
|a=tls-id:89J2LRATQ3ULA24G9AHWVR31VJWSLB68
|                                     | [I-D.ietf-mmusic-dtls-sdp]Alice wants to|
|                                     | use the same DTLS association |
+-----+
+-----+
|a=rtcp-mux
|                                     | [RFC5761] |
+-----+
+-----+
|a=rtcp-mux-only
|                                     | [I-D.ietf-mmusic-mux-exclusive] |
+-----+
+-----+
|a=rtcp-rsize
|                                     | [RFC5506] |
+-----+
+-----+
|a=extmap:1 urn:ietf:params:rtp-hdext:ssrc-audio-level
|                                     | [RFC6464] |
+-----+
+-----+
|a=extmap:2 urn:ietf:params:rtp-hdext:sdes:mid
|                                     | [I-D.ietf-mmusic-sdp-bundle-negotiation] |
+-----+
+-----+
|a=candidate:0 1 UDP 2113667327 192.0.2.4 61665 typ host
|                                     | [I-D.ietf-mmusic-ice-sip-sdp] |
+-----+
+-----+
```

```

|a=candidate:1 1 UDP 694302207 203.0.113.141 54609 typ srflx raddr 192.0.2.4 r
port 61665 | [I-D.ietf-mmusic-ice-sip-sdp] |
+-----+
|a=end-of-candidates | [I-D.ietf-mmusic-trickle-ice] |
+-----+
|***** Video m=line ***** | ***** |
+-----+
|m=video 0 UDP/TLS/RTP/SAVPF 120 | [RFC4566] |
+-----+
|c=IN IP4 203.0.113.141 | [RFC4566] |
+-----+
|a=bundle-only | [I-D.ietf-mmusic-sdp-bundle-negotiation] |
+-----+
|a=mid:video | [RFC5888] |
+-----+
|a=msid:ma tb | Identifies MediaStream ID (ma) and |
| | MediaStreamTrack ID (tb) |
+-----+
|a=sendrecv | [RFC3264] |
+-----+

```


a=rtpmap:120 VP8/90000	[RFC7741]	
-----	-----	-----
a=rtcp-fb:120 nack	[RFC4585]	
-----	-----	-----
a=rtcp-fb:120 nack pli	[RFC4585]	
-----	-----	-----
a=rtcp-fb:120 ccm fir	[RFC5104]	
-----	-----	-----
a=extmap:2 urn:iETF:params:rtp-hdrext:sdes:mid	[I-D.iETF-mmusic-sdp-bundle-negotiation]	
-----	-----	-----

Table 25: 5.2.11 SDP Updated Offer

=====		
Updated Answer SDP Contents	RFC#/Notes	
=====		
v=0	[RFC4566] Version number incremented	
-----	-----	-----
o=- 16833 1 IN IP4 0.0.0.0	[RFC4566] - Session Origin Information	
-----	-----	-----
s=-	[RFC4566]	
-----	-----	-----
t=0 0	[RFC4566]	
-----	-----	-----
a=group:BUNDLE audio video	[I-D.iETF-mmusic-sdp-bundle-negotiation]	
-----	-----	-----
a=group:LS audio video	[RFC5888]	
-----	-----	-----
a=ice-options:trickle	[I-D.iETF-mmusic-trickle-ice]	
-----	-----	-----

```

|a=ice-options:ice2
| [RFC8445]
+-----+
|***** Audio m=line *****
|*****
+-----+
|m=audio 49203 UDP/TLS/RTP/SAVPF 109
| [RFC4566]
+-----+
|c=IN IP4 203.0.113.77
| [RFC4566]
+-----+
|a=mid:audio
| [RFC5888]
+-----+
|a=msid:ma ta
| Identifies MediaStream ID (ma) and
|
| MediaStreamTrack ID (ta)
+-----+
|a=sendrecv
| [RFC3264]
+-----+
|a=rtpmap:109 opus/48000/2
| [RFC7587]
+-----+

```

```
|a=maxptime:120
| [RFC4566]
+-----+
|a=ice-ufrag:c300d85b
| [I-D.ietf-mmusic-ice-sip-sdp]
+-----+
|a=ice-pwd:de4e99bd291c325921d5d47efbabd9a2
| [I-D.ietf-mmusic-ice-sip-sdp]
+-----+
|a=fingerprint:sha-256
| [RFC8122]
| 6B:8B:F0:65:5F:78:E2:51:3B:AC:6F:F3:3F:46:1B:35:DC:B8:5F:64:1A:24:C2:43:F0:A1
:58:D0:A1:2C:19:08
+-----+
|a=setup:active
| [RFC5763]
+-----+
|a=tls-id:CJ6FF9ZZMJW7MDRJIR7XVIQM48GE1G31
| [I-D.ietf-mmusic-dtls-sdp] - Bob agrees
|
| to use the same DTLS association
+-----+
|a=rtcp-mux
| [RFC5761]
+-----+
|a=rtcp-mux-only
| [I-D.ietf-mmusic-mux-exclusive]
+-----+
|a=rtcp-rsize
| [RFC5506]
+-----+
|a=extmap:1 urn:ietf:params:rtp-hdext:ssrc-audio-level
| [RFC6464]
+-----+
|a=extmap:2 urn:ietf:params:rtp-hdext:sdes:mid
| [I-D.ietf-mmusic-sdp-bundle-negotiation]
+-----+
|a=candidate:0 1 UDP 2113667327 198.51.100.7 51556 typ host
| [I-D.ietf-mmusic-ice-sip-sdp]
+-----+
|a=candidate:1 1 UDP 1694302207 203.0.113.77 49203 typ srflx raddr 198.51.100.
7 rport 51556
| [I-D.ietf-mmusic-ice-sip-sdp]
+-----+
|a=end-of-candidates
| [I-D.ietf-mmusic-trickle-ice]
```

```

+-----+
| ***** Video m=line ***** |
| ***** |
+-----+
|m=video 0 UDP/TLS/RTP/SAVPF 120 |
| [RFC4566] |
+-----+
|c=IN IP4 203.0.113.77 |
| [RFC4566] |
+-----+
|a=bundle-only |
| [I-D.ietf-mmusic-sdp-bundle-negotiation] |
+-----+
|a=mid:video |
| [RFC5888] |
+-----+
|a=msid:ma tb |
| Identifies MediaStream ID (ma) and |
| |
| MediaStreamTrack ID (tb) |
+-----+
|a=sendrecv |
| [RFC3264] |
+-----+
|a=rtpmap:120 VP8/90000 |
| [RFC7741] |
+-----+
|a=rtcp-fb:120 nack |
| [RFC4585] |

```

```

+-----+
+-----+
|a=rtcp-fb:120 nack pli
| [RFC4585]
+-----+
+-----+
|a=rtcp-fb:120 ccm fir
| [RFC5104]
+-----+
+-----+
|a=extmap:2 urn:ietf:params:rtp-hdext:sdes:mid
| [I-D.ietf-mmusic-sdp-bundle-negotiation]
+-----+
+-----+

```

Table 26: 5.2.11 SDP Updated Answer

5.3. MultiResolution, RTX, FEC Examples

This section provides examples related to multi-source, multi-stream negotiation such as layered coding and simulcast. Further included are a few examples that cover techniques to deal with providing robustness against transmission errors such as FEC and RTX. Also, mechanisms such as FEC and RTX could be envisioned in the above basic scenarios as well.

5.3.1. Send-only Simulcast Session with 2 Cameras and 2 Encodings per Camera

The SDP example below shows an offer/answer exchange with one audio and two video sources (say 2 video cameras). Each of the video sources can be sent at two different resolutions.

One video source corresponds to VP8 encoding, while the other corresponds to H.264 encoding.

The [I-D.ietf-mmusic-rid] framework is used to further constrain the media format encodings and map the payload types (PT) to the 'rid' identifiers.

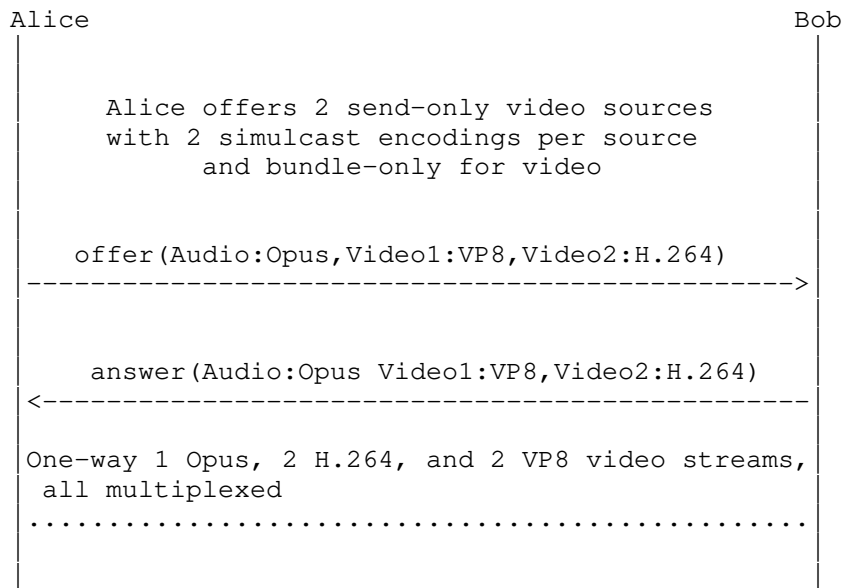
The [I-D.ietf-mmusic-sdp-simulcast] framework identifies the simulcast streams via their 'rid' identifiers.

The bundle-only attribute is used for the video sources in the offer to ensure enabling video sources solely in the context of BUNDLE alone.

The BUNDLE grouping framework enables multiplexing of all 5 Source RTP Streams (1 audio stream + 4 video streams) over a single RTP session.

Also, the audio and one video source RTP stream form a lip sync group while the other video source RTP stream represents non-interactive media data.

One-way Successful Simulcast w/BUNDLE



Offer SDP Contents	RFC#/Notes
v=0	[RFC4566]
o=- 20519 0 IN IP4 0.0.0.0	[RFC4566]
s=-	[RFC4566]
t=0 0	[RFC4566]
a=group:BUNDLE m0 m1 m2 [I-D.ietf-mmusic-sdp-bundle-negotiation] Alice supports grouping of m=lines under BUNDLE semantics	
a=group:LS m0 m1	[RFC5888]
a=ice-options:trickle	

```

+-----+
| [I-D.ietf-mmusic-trickle-ice] |
+-----+
| a=ice-options:ice2 |
| [RFC8445] |
+-----+
| ***** Audio m=line ***** |
| ***** |
+-----+
| m=audio 54609 UDP/TLS/RTP/SAVPF 109 |
| [RFC4566] |
+-----+

```



```
|c=IN IP4 203.0.113.141
| [RFC4566]
+-----+
|a=mid:m0
| [RFC5888]
+-----+
|a=msid:ma ta
| Identifies MediaStream ID (ma) and
| MediaStreamTrack ID (ta)
+-----+
|a=sendonly
| [RFC3264]
+-----+
|a=rtpmap:109 opus/48000/2
| [RFC7587]
+-----+
|a=maxptime:120
| [RFC4566]
+-----+
|a=ice-ufraq:074c6550
| [I-D.ietf-mmusic-ice-sip-sdp]
+-----+
|a=ice-pwd:a28a397a4c3f31747dlee3474af08a068
| [I-D.ietf-mmusic-ice-sip-sdp]
+-----+
|a=fingerprint:sha-256
| [RFC8122]
| 19:E2:1C:3B:4B:9F:81:E6:B8:5C:F4:A5:A8:D8:73:04:BB:05:2F:70:9F:04:A9:0E:05:E9
:26:33:E8:70:88:A2
+-----+
|a=setup:actpass
| [RFC5763]
+-----+
|a=tls-id:89J2LRATQ3ULA24G9AHWVR31VJWSLB68
| [I-D.ietf-mmusic-dtls-sdp]
+-----+
|a=rtcp-mux
| [RFC5761]
+-----+
|a=rtcp-rsize
| [RFC5506]
+-----+
|a=extmap:1 urn:ietf:params:rtp-hdext:ssrc-audio-level
| [RFC6464]
```

```

-----
|a=extmap:2 urn:ietf:params:rtp-hdext:sdes:mid
| [I-D.ietf-mmusic-sdp-bundle-negotiation] |
-----
|a=candidate:0 1 UDP 2113667327 192.0.2.4 61665 typ host
| [I-D.ietf-mmusic-ice-sip-sdp] |
-----
|a=candidate:1 1 UDP 694302207 203.0.113.141 54609 typ srflx raddr 192.0.2.4 r
port 61665 | [I-D.ietf-mmusic-ice-sip-sdp] |
-----
|a=end-of-candidates
| [I-D.ietf-mmusic-trickle-ice] |
-----
|***** Video-1 m=line *****
| ***** |
-----
|m=video 0 UDP/TLS/RTP/SAVPF 98 100
| bundle-only video line with port number |
|
| set to zero |
-----
|c=IN IP4 203.0.113.141
| [RFC4566] |
-----
|a=bundle-only
| [I-D.ietf-mmusic-sdp-bundle-negotiation] |
-----
|a=mid:m1
| [RFC5888] Video m=line part of BUNDLE |

```

	group	
+-----+		+-----+
a=msid:ma tb	Identifies MediaStream ID (ma) and	
	MediaStreamTrack ID (tb)	
+-----+		+-----+
a=sendonly	[RFC3264] - Send-only video stream	
+-----+		+-----+
a=rtpmap:98 VP8/90000	[RFC7741]	
+-----+		+-----+
a=fmtp:98 max-fr=30	[RFC4566]	
+-----+		+-----+
a=rtpmap:100 VP8/90000	[RFC7741]	
+-----+		+-----+
a=fmtp:100 max-fr=15	[RFC4566]	
+-----+		+-----+
a=rtcp-fb:* nack	[RFC4585]	
+-----+		+-----+
a=rtcp-fb:* nack pli	[RFC4585]	
+-----+		+-----+
a=rtcp-fb:* ccm fir	[RFC5104]	
+-----+		+-----+
a=extmap:2 urn:ietf:params:rtp-hdext:sdes:mid	[I-D.ietf-mmusic-sdp-bundle-negotiation]	
+-----+		+-----+
a=extmap:3 urn:ietf:params:rtp-hdext:sdes:rtp-stream-id	[I-D.ietf-avtext-rid]	
+-----+		+-----+
a=rid:1 send pt=98;max-width=1280;max-height=720	[I-D.ietf-mmusic-rid] 1:1 rid mapping to	
	payload type and specify resolution	
	constraints	
+-----+		+-----+

```

|a=rid:2 send pt=100;max-width=640;max-height=480
|           |[I-D.ietf-mmusic-rid] 1:1 rid mapping to|
|           |payload type and specify resolution   |
|           |constraints                           |
+-----+
|a=simulcast:send 1;~2
|           |[I-D.ietf-mmusic-sdp-simulcast] Alice |
|           |can send 2 resolutions identified by the|
|           |'rid' identifiers Also, the second    |
|           |stream is initially paused.          |
+-----+
|***** Video-2 m=line *****
|*****
+-----+
|m=video 0 UDP/TLS/RTP/SAVPF 101 102
|           |bundle-only video line with port number |
|           |set to zero                             |
+-----+
|c=IN IP4 203.0.113.141
|           |[RFC4566]                               |
+-----+
|a=bundle-only
|           |[I-D.ietf-mmusic-sdp-bundle-negotiation]|
+-----+
|a=mid:m2
|           |[RFC5888] Video m=line part of BUNDLE  |

```

	group	
+-----+		+-----+
a=msid:ma tc	Identifies MediaStream ID (ma) and	
	MediaStreamTrack ID (tc)	
+-----+		+-----+
a=sendonly	[RFC3264] - Send only video stream	
+-----+		+-----+
a=rtpmap:101 H264/90000	[RFC6184]	
+-----+		+-----+
a=rtpmap:102 H264/90000	[RFC6184]	
+-----+		+-----+
a=fmtp:101 profile-level-id=42401f;packetization-mode=0	[RFC6184]Camera-2,Encoding-1	
+-----+		+-----+
a=fmtp:102 profile-level-id=42401f;packetization-mode=1	[RFC6184]Camera-2,Encoding-2	
+-----+		+-----+
a=rtcp-fb:* nack	[RFC4585]	
+-----+		+-----+
a=rtcp-fb:* nack pli	[RFC4585]	
+-----+		+-----+
a=rtcp-fb:* ccm fir	[RFC5104]	
+-----+		+-----+
a=extmap:2 urn:iETF:params:rtp-hdext:sdes:mid	[I-D.iETF-mmusic-sdp-bundle-negotiation]	
+-----+		+-----+
a=extmap:3 urn:iETF:params:rtp-hdext:sdes:rtp-stream-id	[I-D.iETF-avtext-rid]	
+-----+		+-----+
a=rid:3 send pt=101;max-width=1280;max-height=720;max-fr=30	[I-D.iETF-mmusic-rid] 1:1 rid mapping to	
	payload type and specify resolution	
	constraints	
+-----+		+-----+

a=rid:4 send pt=102;max-width=640;max-height=360;max-fr=15	
	[I-D.ietf-mmusic-rid] 1:1 rid mapping to
	payload type and specify resolution
	constraints
+-----+-----+	
a=simulcast:send 3;4	
	[I-D.ietf-mmusic-sdp-simulcast] Alice
	can send 2 resolutions identified by the
	'rid' identifiers
+-----+-----+	

Table 27: 5.3.1 SDP Offer

+=====+	
Answer SDP Contents	
	RFC#/Notes
+=====+	
v=0	
	[RFC4566]
+-----+-----+	
o=- 20519 0 IN IP4 0.0.0.0	
	[RFC4566]
+-----+-----+	


```

|a=maxptime:120
| [RFC4566]
+-----+
|a=ice-ufrag:c300d85b
| [I-D.ietf-mmusic-ice-sip-sdp]
+-----+
|a=ice-pwd:de4e99bd291c325921d5d47efbabd9a2
| [I-D.ietf-mmusic-ice-sip-sdp]
+-----+
|a=fingerprint:sha-256
| [RFC8122]
|6B:8B:F0:65:5F:78:E2:51:3B:AC:6F:F3:3F:46:1B:35:DC:B8:5F:64:1A:24:C2:43:F0:A1
:58:D0:A1:2C:19:08|
+-----+
|a=setup:active
| [RFC5763]
+-----+
|a=tls-id:CJ6FF9ZZMJW7MDRJIR7XVIQM48GE1G31
| [I-D.ietf-mmusic-dtls-sdp]
+-----+
|a=rtcp-mux
| [RFC5761]
+-----+
|a=rtcp-rsize
| [RFC5506]
+-----+
|a=extmap:1 urn:ietf:params:rtp-hdext:ssrc-audio-level
| [RFC6464]
+-----+

```



```

|a=extmap:2 urn:ietf:params:rtp-hdext:sdes:mid
| [I-D.ietf-mmusic-sdp-bundle-negotiation] |
+-----+
+-----+
|a=candidate:0 1 UDP 2113667327 198.51.100.7 61665 typ host
| [I-D.ietf-mmusic-ice-sip-sdp] |
+-----+
+-----+
|a=candidate:1 1 UDP 694302207 203.0.113.77 49203 typ srflx raddr 198.51.100.7
rport 61665 | [I-D.ietf-mmusic-ice-sip-sdp] |
+-----+
+-----+
|a=end-of-candidates
| [I-D.ietf-mmusic-trickle-ice] |
+-----+
+-----+
|***** Video-1 m=line *****
| ***** |
+-----+
+-----+
|m=video 0 UDP/TLS/RTP/SAVPF 98 100
| BUNDLE accepted |
+-----+
+-----+
|c=IN IP4 203.0.113.77
| [RFC4566] |
+-----+
+-----+
|a=bundle-only
| [I-D.ietf-mmusic-sdp-bundle-negotiation] |
+-----+
+-----+
|a=mid:m1
| [RFC5888] Video m=line part of BUNDLE |
| group |
+-----+
+-----+
|a=msid:ma tb
| Identifies MediaStream ID (ma) and |
| MediaStreamTrack ID (tb) |
+-----+
+-----+
|a=recvonly
| [RFC3264] - receive only video stream |
+-----+
+-----+
|a=rtpmap:98 VP8/90000
| [RFC7741] |
+-----+
+-----+
|a=rtpmap:100 VP8/90000
| [RFC7741] |
+-----+
+-----+
|a=fmtp:98 max-fr=30
| [RFC4566] |

```

```

+-----+
|a=fmtp:100 max-fr=15
| [RFC4566]
+-----+
|a=rtcp-fb:* nack
| [RFC4585]
+-----+
|a=rtcp-fb:* nack pli
| [RFC4585]
+-----+
|a=rtcp-fb:* ccm fir
| [RFC5104]
+-----+
|a=extmap:2 urn:ietf:params:rtp-hdext:sdes:mid
| [I-D.ietf-mmusic-sdp-bundle-negotiation]
+-----+
|a=extmap:3 urn:ietf:params:rtp-hdext:sdes:rtp-stream-id
| [I-D.ietf-avtext-rid]
+-----+
|a=rid:1 recv pt=98;max-width=1280;max-height=720
| [I-D.ietf-mmusic-rid] Bob accepts the
|
| offered payload format constraints
+-----+
|a=rid:2 recv pt=100;max-width=640;max-height=480
| [I-D.ietf-mmusic-rid] Bob accepts the
|
| offered payload format constraints
+-----+

```

```

|a=simulcast:recv 1;2
|           |[I-D.ietf-mmusic-sdp-simulcast] Bob |
|           |accepts the offered simulcast streams |
|           |and removes the paused state of stream |
|           |with 'rid' value 2. |
+-----+
|***** Video-2 m=line *****
|*****
+-----+
|m=video 0 UDP/TLS/RTP/SAVPF 101 102
|           |BUNDLE accepted |
+-----+
|c=IN IP4 203.0.113.77
|           |[RFC4566] |
+-----+
|a=bundle-only
|           |[I-D.ietf-mmusic-sdp-bundle-negotiation] |
+-----+
|a=mid:m2
|           |[RFC5888] Video m=line part of BUNDLE |
|           |group |
+-----+
|a=msid:ma tc
|           |Identifies MediaStream ID (ma) and |
|           |MediaStreamTrack ID (tc) |
+-----+
|a=recvonly
|           |[RFC3264] |
+-----+
|a=rtpmap:101 H264/90000
|           |[RFC6184] |
+-----+
|a=rtpmap:102 H264/90000
|           |[RFC6184] |
+-----+
|a=fmtp:101 profile-level-id=42401f;packetization-mode=1
|           |[RFC6184] |
+-----+
|a=fmtp:102 profile-level-id=42401f;packetization-mode=1
|           |[RFC6184] |
+-----+

```

```

|a=rtcp-fb:* nack
| [RFC4585]
+-----+
|a=rtcp-fb:* nack pli
| [RFC4585]
+-----+
|a=rtcp-fb:* ccm fir
| [RFC5104]
+-----+
|a=extmap:2 urn:ietf:params:rtp-hdext:sdes:mid
| [I-D.ietf-mmusic-sdp-bundle-negotiation]
+-----+
|a=extmap:3 urn:ietf:params:rtp-hdext:sdes:rtp-stream-id
| [I-D.ietf-avtext-rid]
+-----+
|a=rid:3 recv pt=101;max-width=1280;max-height=720;max-fr=30
| [I-D.ietf-mmusic-rid] Bob accepts the
|
| offered payload format constraints
+-----+
|a=rid:4 recv pt=102;max-width=640;max-height=360;max-fr=15
| [I-D.ietf-mmusic-rid] Bob accepts the
|
| offered payload format constraints
+-----+
|a=simulcast:recv 3;4
| [I-D.ietf-mmusic-sdp-simulcast] Bob
|
| accepts the offered simulcast streams.
+-----+

```

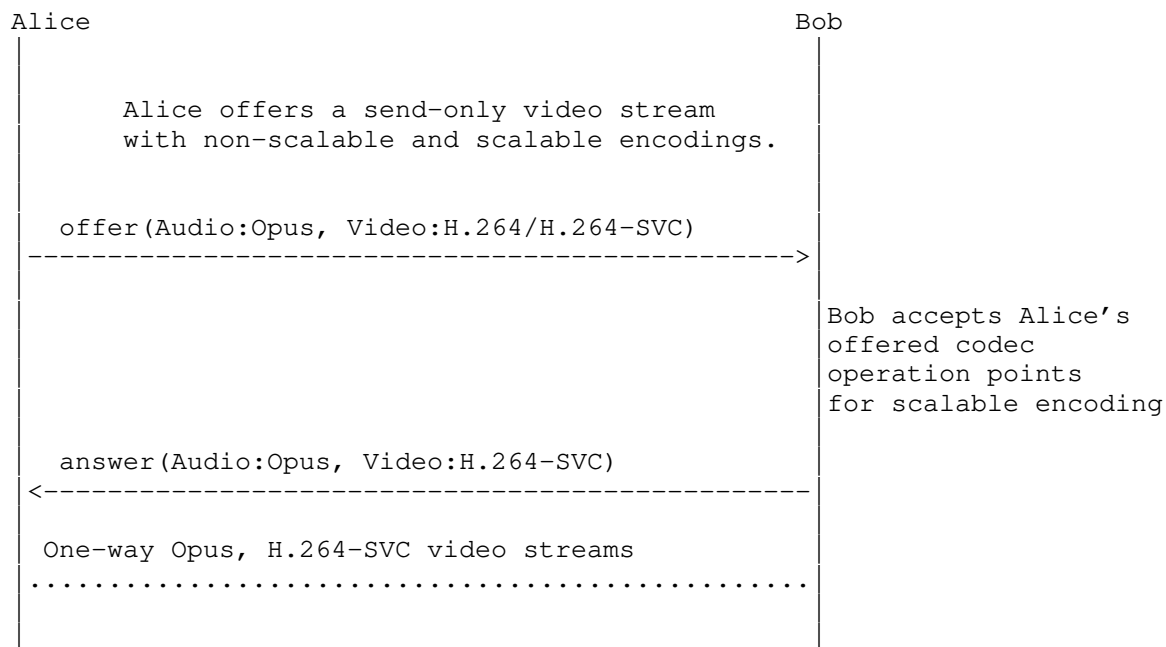
Table 28: 5.3.1 SDP Answer

5.3.2. Successful SVC Video Session

This section shows an SDP offer/answer for a session with an audio and a single video source. The video source is being encoded both as non-scalable and scalable H.264-SVC RTP streams (in the SST mode).

The answerer picks the payload type corresponding to scalable encoding.

SVC Session - 3 Layers w/BUNDLE



Offer SDP Contents	
	RFC#/Notes
v=0	[RFC4566]
o=- 20519 0 IN IP4 0.0.0.0	[RFC4566]
s=-	[RFC4566]
t=0 0	[RFC4566]

-----+-----+

Nandakumar & Jennings

Expires 20 June 2021

[Page 78]


```

-----
|a=ice-pwd:a28a397a4c3f31747d1ee3474af08a068
| [I-D.ietf-mmusic-ice-sip-sdp]
-----
|a=fingerprint:sha-256
| [RFC8122]
|19:E2:1C:3B:4B:9F:81:E6:B8:5C:F4:A5:A8:D8:73:04:BB:05:2F:70:9F:04:A9:0E:05:E9
:26:33:E8:70:88:A2|
-----
|a=setup:actpass
| [RFC5763]
-----
|a=tls-id:89J2LRATQ3ULA24G9AHWVR31VJWSLB68
| [I-D.ietf-mmusic-dtls-sdp]
-----
|a=rtcp-mux
| [RFC5761]
-----
|a=rtcp-rsize
| [RFC5506]
-----
|a=extmap:1 urn:ietf:params:rtp-hdext:ssrc-audio-level
| [RFC6464]
-----
|a=extmap:2 urn:ietf:params:rtp-hdext:sdes:mid
| [I-D.ietf-mmusic-sdp-bundle-negotiation]
-----
|a=candidate:0 1 UDP 2113667327 192.0.2.4 61665 typ host
| [I-D.ietf-mmusic-ice-sip-sdp]
-----

```



```

+-----+
+-----+
|a=candidate:1 1 UDP 694302207 203.0.113.141 54609 typ srflx raddr 192.0.2.4 r
port 61665      |[I-D.ietf-mmusic-ice-sip-sdp]      |
+-----+
+-----+
|a=end-of-candidates
                |[I-D.ietf-mmusic-trickle-ice]    |
+-----+
+-----+
|***** Video m=line *****
                |*****
+-----+
+-----+
|m=video 0 UDP/TLS/RTP/SAVPF 96 100
                |bundle-only video line with port number |
                |set to zero                               |
+-----+
+-----+
|c=IN IP4 203.0.113.141
                |[RFC4566]
+-----+
+-----+
|a=bundle-only
                |[I-D.ietf-mmusic-sdp-bundle-negotiation] |
+-----+
+-----+
|a=mid:m1
                |[RFC5888] Video m=line part of BUNDLE   |
                |group
+-----+
+-----+
|a=msid:ma tb
                |Identifies MediaStream ID (ma) and      |
                |MediaStreamTrack ID (tb)
+-----+
+-----+
|a=sendonly
                |[RFC3264] - Send only video stream      |
+-----+
+-----+
|a=rtpmap:96 H264/90000
                |[RFC6184]
+-----+
+-----+
|a=fmtp:96 profile-level-id=4d0028; packetization-mode=1;max-fs=8040
                |[RFC6184]H.264 Non Scalable
+-----+
+-----+
|a=rtpmap:100 H264-SVC/90000
                |[RFC6190]
+-----+
+-----+
|a=fmtp:100 profile-level-id=53001f;packetization-mode=0
                |[RFC6190] H.264 Scalable Encoding

```

a=rtcp-fb:* nack	[RFC4585]	
a=rtcp-fb:* nack pli	[RFC4585]	
a=rtcp-fb:* ccm fir	[RFC5104]	
a=extmap:2 urn:ietf:params:rtp-hdext:sdes:mid	[I-D.ietf-mmusic-sdp-bundle-negotiation]	

Table 29: 5.3.2 SDP Offer with SVC

Answer SDP Contents	RFC#/Notes	
v=0	[RFC4566]	
o=- 20519 0 IN IP4 0.0.0.0	[RFC4566]	

```
+-----+
+-----+
|s=-
|           | [RFC4566]
+-----+
+-----+
|t=0 0
|           | [RFC4566]
+-----+
+-----+
|a=group:BUNDLE m0 m1
|           | [I-D.ietf-mmusic-sdp-bundle-negotiation]
+-----+
+-----+
|a=group:LS m0 m1
|           | [RFC5888]
+-----+
+-----+
|a=ice-options:trickle
|           | [I-D.ietf-mmusic-trickle-ice]
+-----+
+-----+
|a=ice-options:ice2
|           | [RFC8445]
+-----+
+-----+
|***** Audio m=line *****
|           | *****
+-----+
+-----+
|m=audio 49203 UDP/TLS/RTP/SAVPF 109
|           | [RFC4566]
+-----+
+-----+
|c=IN IP4 203.0.113.77
|           | [RFC4566]
+-----+
+-----+
|a=mid:m0
|           | [RFC5888]
+-----+
+-----+
|a=msid:ma ta
|           | Identifies MediaStream ID (ma) and
|           |
|           | MediaStreamTrack ID (ta)
+-----+
+-----+
|a=recvonly
|           | [RFC3264]
+-----+
+-----+
|a=rtpmap:109 opus/48000/2
|           | [RFC7587]
+-----+
+-----+
|a=maxptime:120
|           | [RFC4566]
+-----+
```

```

-----+-----
|a=ice-ufrag:074c6550
| [I-D.ietf-mmusic-ice-sip-sdp]
-----+-----
|a=ice-pwd:a28a397a4c3f31747dlee3474af08a068
| [I-D.ietf-mmusic-ice-sip-sdp]
-----+-----
|a=fingerprint:sha-256
| [RFC8122]
|6B:8B:F0:65:5F:78:E2:51:3B:AC:6F:F3:3F:46:1B:35:DC:B8:5F:64:1A:24:C2:43:F0:A1
:58:D0:A1:2C:19:08|
-----+-----
|a=setup:active
| [RFC5763]
-----+-----
|a=tls-id:CJ6FF9ZZMJW7MDRJIR7XVIQM48GE1G31
| [I-D.ietf-mmusic-dtls-sdp]
-----+-----
|a=rtcp-mux
| [RFC5761]
-----+-----
|a=rtcp-rsize
| [RFC5506]
-----+-----
|a=extmap:1 urn:ietf:params:rtp-hdext:ssrc-audio-level
| [RFC6464]
-----+-----
|a=extmap:2 urn:ietf:params:rtp-hdext:sdes:mid
| [I-D.ietf-mmusic-sdp-bundle-negotiation]

```

```

+-----+
+-----+
|a=candidate:0 1 UDP 2113667326 198.51.100.7 51556 typ host
| [I-D.ietf-mmusic-ice-sip-sdp] |
+-----+
+-----+
|a=candidate:1 1 UDP 1694302206 203.0.113.77 49203 typ srflx raddr 198.51.100.
7 rport 51556 | [I-D.ietf-mmusic-ice-sip-sdp] |
+-----+
+-----+
|a=end-of-candidates
| [I-D.ietf-mmusic-trickle-ice] |
+-----+
+-----+
|***** Video m=line *****
| ***** |
+-----+
+-----+
|m=video 0 UDP/TLS/RTP/SAVPF 100
| BUNDLE accepted. |
+-----+
+-----+
|c=IN IP4 203.0.113.77
| [RFC4566] |
+-----+
+-----+
|a=bundle-only
| [I-D.ietf-mmusic-sdp-bundle-negotiation] |
+-----+
+-----+
|a=mid:m1
| [RFC5888] Video m=line part of BUNDLE |
| group |
+-----+
+-----+
|a=msid:ma tb
| Identifies MediaStream ID (ma) and |
| MediaStreamTrack ID (tb) |
+-----+
+-----+
|a=recvonly
| [RFC3264] - Receive only video stream |
+-----+
+-----+
|a=rtpmap:100 H264-SVC/90000
| [RFC6190] |
+-----+
+-----+
|a=fmtp:100 profile-level-id=53001f;packetization-mode=0
| [RFC6190] |
+-----+
+-----+
|a=rtcp-fb:* nack
| [RFC4585] |
+-----+
+-----+

```

```

|a=rtcp-fb:* nack pli
| [RFC4585] |
+-----+
|a=rtcp-fb:* ccm fir
| [RFC5104] |
+-----+
|a=extmap:2 urn:ietf:params:rtp-hdrext:sdes:mid
| [I-D.ietf-mmusic-sdp-bundle-negotiation] |
+-----+

```

Table 30: 5.3.2 SDP Answer with SVC

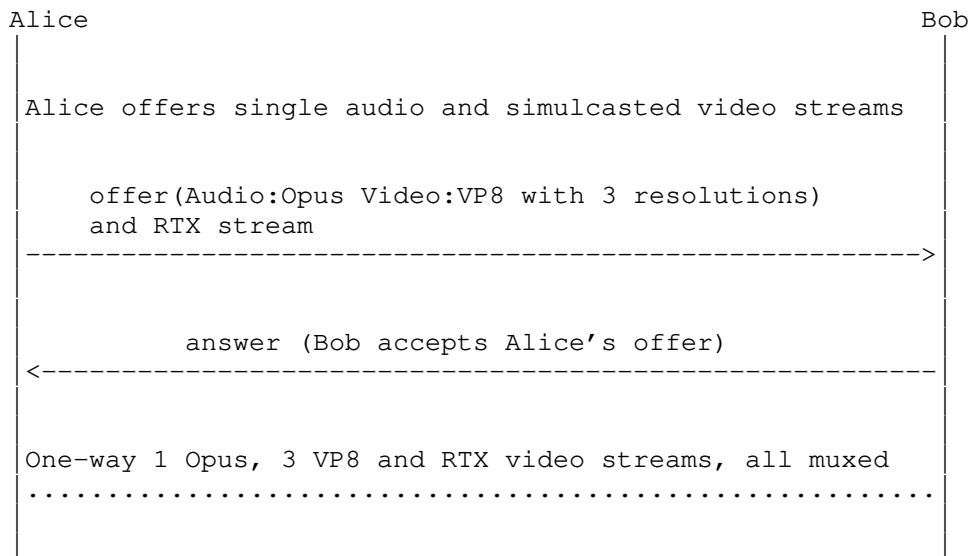
5.3.3. Successful Simulcast Video Session with Retransmission

This section shows an SDP offer/answer exchange for a simulcast scenario with 3 resolutions and has [RFC4588]-style retransmission flows.

The [I-D.ietf-mmusic-rid] framework is used to specify all the (3) resolution constraints mapped to a single Payload Type (98).

The [I-D.ietf-mmusic-sdp-simulcast] framework identifies the simulcast streams via their 'rid' identifiers.

Simulcast Streams with Retransmission



Offer SDP Contents	RFC#/Notes
v=0	[RFC4566]
o=- 20519 0 IN IP4 0.0.0.0	[RFC4566]
s=-	[RFC4566]
t=0 0	[RFC4566]
a=group:BUNDLE m0 m1	[I-D.ietf-mmusic-sdp-bundle-negotiation]
	Alice supports grouping of m=lines under
	BUNDLE semantics
a=group:LS m0 m1	[RFC5888]

```
-----+-----+
|a=ice-options:trickle
|      |[I-D.ietf-mmusic-trickle-ice]      |
+-----+-----+
|a=ice-options:ice2
|      |[RFC8445]                          |
+-----+-----+
|***** Audio m=line *****
|*****
+-----+-----+
-----+-----+
```



```
|m=audio 54609 UDP/TLS/RTP/SAVPF 109
|   | [RFC4566]
+-----+
|c=IN IP4 203.0.113.141
|   | [RFC4566]
+-----+
|a=mid:m0
|   | [RFC5888] Audio m=line part of BUNDLE
|   | group with a unique port number
+-----+
|a=msid:ma ta
|   | Identifies MediaStream ID (ma) and
|   | MediaStreamTrack ID (ta)
+-----+
|a=sendonly
|   | [RFC3264]
+-----+
|a=rtpmap:109 opus/48000/2
|   | [RFC7587]
+-----+
|a=maxptime:120
|   | [RFC4566]
+-----+
|a=ice-ufrag:074c6550
|   | [I-D.ietf-mmusic-ice-sip-sdp]
+-----+
|a=ice-pwd:a28a397a4c3f31747dlee3474af08a068
|   | [I-D.ietf-mmusic-ice-sip-sdp]
+-----+
|a=fingerprint:sha-256
|   | [RFC8122]
|19:E2:1C:3B:4B:9F:81:E6:B8:5C:F4:A5:A8:D8:73:04:BB:05:2F:70:9F:04:A9:0E:05:E9
:26:33:E8:70:88:A2|
+-----+
|a=setup:actpass
|   | [RFC5763]
+-----+
|a=tls-id:89J2LRATQ3ULA24G9AHWVR31VJWSLB68
|   | [I-D.ietf-mmusic-dtls-sdp]
+-----+
|a=rtcp-mux
|   | [RFC5761]
+-----+
```

```

|a=rtcp-rsize
| [RFC5506]
+-----+
|a=extmap:1 urn:ietf:params:rtp-hdext:ssrc-audio-level
| [RFC6464]
+-----+
|a=extmap:2 urn:ietf:params:rtp-hdext:sdes:mid
| [I-D.ietf-mmusic-sdp-bundle-negotiation]
+-----+
|a=candidate:0 1 UDP 2113667327 192.0.2.4 61665 typ host
| [I-D.ietf-mmusic-ice-sip-sdp]
+-----+
|a=candidate:1 1 UDP 694302207 203.0.113.141 54609 typ srflx raddr 192.0.2.4 r
port 61665 | [I-D.ietf-mmusic-ice-sip-sdp]
+-----+
|a=end-of-candidates
| [I-D.ietf-mmusic-trickle-ice]
+-----+
|***** Video m=line *****
| *****
+-----+
|m=video 0 UDP/TLS/RTP/SAVPF 98 103
| bundle-only video line with port number
|
| set to zero
+-----+
|c=IN IP4 203.0.113.141
| [RFC4566]
+-----+

```

a=bundle-only	[I-D.ietf-mmusic-sdp-bundle-negotiation]
a=mid:m1	[RFC5888]
a=msid:ma tb	Identifies MediaStream ID (ma) and MediaStreamTrack ID (tb)
a=sendonly	[RFC3264]
a=rtpmap:98 VP8/90000	[RFC7741]
a=fmtp:98 max-fr=30	[RFC4566]
a=rtpmap:103 rtx/90000	[RFC4588]
a=fmtp:103 apt=98;rtx-time=200	[RFC4588]
a=rtcp-fb:* nack	[RFC4585]
a=rtcp-fb:* nack pli	[RFC4585]
a=rtcp-fb:* ccm fir	[RFC5104]
a=extmap:2 urn:ietf:params:rtp-hdext:sdes:mid	[I-D.ietf-mmusic-sdp-bundle-negotiation]
a=extmap:3 urn:ietf:params:rtp-hdext:sdes:rtp-stream-id	[I-D.ietf-avtext-rid]
a=extmap:4 urn:ietf:params:rtp-hdext:sdes:repaired-rtp-stream-id	[I-D.ietf-avtext-rid]

```

|a=rid:1 send pt=98;max-fs=921600;max-fr=30
| [I-D.ietf-mmusic-rid] |
+-----+
|a=rid:2 send pt=98;max-fs=614400;max-fr=15
| [I-D.ietf-mmusic-rid] |
+-----+
|a=rid:3 send pt=98;max-fs=230400;max-fr=30
| [I-D.ietf-mmusic-rid] |
+-----+
|a=simulcast:send 1;2;3
| [I-D.ietf-mmusic-sdp-simulcast] Alice |
| |
| | can send all the simulcast streams |
+-----+

```

Table 31: 5.3.3 SDP Offer w/Simulcast, RTX

```

=====
| Answer SDP Contents
| RFC#/Notes |
+-----+
| v=0
| [RFC4566] |
+-----+
| o=- 20519 0 IN IP4 0.0.0.0
| [RFC4566] |

```

```

+-----+
+-----+
|s=-
|           | [RFC4566]
+-----+
+-----+
|t=0 0
|           | [RFC4566]
+-----+
+-----+
|a=group:BUNDLE m0 m1
|           | [I-D.ietf-mmusic-sdp-bundle-negotiation]
|           |
|           | Bob supports grouping of m=lines under
|           |
|           | BUNDLE semantics
+-----+
+-----+
|a=group:LS m0 m1
|           | [RFC5888]
+-----+
+-----+
|a=ice-options:trickle
|           | [I-D.ietf-mmusic-trickle-ice]
+-----+
+-----+
|a=ice-options:ice2
|           | [RFC8445]
+-----+
+-----+
|***** Audio m=line *****
|           | *****
+-----+
+-----+
|m=audio 49203 UDP/TLS/RTP/SAVPF 109
|           | [RFC4566]
+-----+
+-----+
|c=IN IP4 203.0.113.77
|           | [RFC4566]
+-----+
+-----+
|a=mid:m0
|           | [RFC5888]
+-----+
+-----+
|a=msid:ma ta
|           | Identifies MediaStream ID (ma) and
|           |
|           | MediaStreamTrack ID (ta)
+-----+
+-----+
|a=recvonly
|           | [RFC3264]
+-----+
+-----+
|a=rtpmap:109 opus/48000/2
|           | [RFC7587]

```

```

-----+
|a=maxptime:120
| [RFC4566] |
-----+
|a=ice-ufrag:074c6550
| [I-D.ietf-mmusic-ice-sip-sdp] |
-----+
|a=ice-pwd:a28a397a4c3f31747d1ee3474af08a068
| [I-D.ietf-mmusic-ice-sip-sdp] |
-----+
|a=fingerprint:sha-256
| [RFC8122] |
|6B:8B:F0:65:5F:78:E2:51:3B:AC:6F:F3:3F:46:1B:35:DC:B8:5F:64:1A:24:C2:43:F0:A1
:58:D0:A1:2C:19:08|
-----+
|a=setup:active
| [RFC5763] |
-----+
|a=tls-id:CJ6FF9ZZMJW7MDRJIR7XVIQM48GE1G31
| [I-D.ietf-mmusic-dtls-sdp] |
-----+
|a=rtcp-mux
| [RFC5761] |
-----+
|a=rtcp-rsize
| [RFC5506] |
-----+
|a=extmap:1 urn:ietf:params:rtp-hdext:ssrc-audio-level
| [RFC6464] |

```

```

+-----+
+-----+
|a=extmap:2 urn:ietf:params:rtp-hdext:sdes:mid
| [I-D.ietf-mmusic-sdp-bundle-negotiation] |
+-----+
+-----+
|a=candidate:0 1 UDP 2113667326 198.51.100.7 51556 typ host
| [I-D.ietf-mmusic-ice-sip-sdp] |
+-----+
+-----+
|a=candidate:1 1 UDP 1694302206 203.0.113.77 49203 typ srflx raddr 198.51.100.
7 rport 51556 | [I-D.ietf-mmusic-ice-sip-sdp] |
+-----+
+-----+
|a=end-of-candidates
| [I-D.ietf-mmusic-trickle-ice] |
+-----+
+-----+
|***** Video m=line *****
| ***** |
+-----+
+-----+
|m=video 0 UDP/TLS/RTP/SAVPF 98 103
| BUNDLE accepted |
+-----+
+-----+
|c=IN IP4 203.0.113.77
| [RFC4566] |
+-----+
+-----+
|a=bundle-only
| [I-D.ietf-mmusic-sdp-bundle-negotiation] |
+-----+
+-----+
|a=mid:m1
| [RFC5888] Video m=line part of BUNDLE |
|
| group |
+-----+
+-----+
|a=msid:ma tb
| Identifies MediaStream ID (ma) and |
|
| MediaStreamTrack ID (tb) |
+-----+
+-----+
|a=recvonly
| [RFC3264] |
+-----+
+-----+
|a=rtpmap:98 VP8/90000
| [RFC7741] |
+-----+
+-----+
|a=fmtp:98 max-fr=30
| [RFC4566] |
+-----+
+-----+

```

```

|a=rtpmap:103 rtx/90000
| [RFC4588]
+-----+
|a=fmtp:103 apt=98;rtx-time=200
| [RFC4588]
+-----+
|a=rtcp-fb:* nack
| [RFC4585]
+-----+
|a=rtcp-fb:* nack pli
| [RFC4585]
+-----+
|a=rtcp-fb:* ccm fir
| [RFC5104]
+-----+
|a=extmap:2 urn:ietf:params:rtp-hdext:sdes:mid
| [I-D.ietf-mmusic-sdp-bundle-negotiation]
+-----+
|a=extmap:3 urn:ietf:params:rtp-hdext:sdes:rtp-stream-id
| [I-D.ietf-avtext-rid]
+-----+
|a=extmap:4 urn:ietf:params:rtp-hdext:sdes:repaired-rtp-stream-id
| [I-D.ietf-avtext-rid]
+-----+
|a=rid:1 recv pt=98;max-fs=921600;max-fr=30
| [I-D.ietf-mmusic-rid]
+-----+
|a=rid:2 recv pt=98;max-fs=614400;max-fr=15
| [I-D.ietf-mmusic-rid]

```


<pre> a=rid:3 recv pt=98;max-fs=230400;max-fr=30 [I-D.ietf-mmusic-rid] </pre>
<pre> a=simulcast:recv 1;2;3 [I-D.ietf-mmusic-sdp-simulcast] Bob accepts the offered simulcast streams </pre>

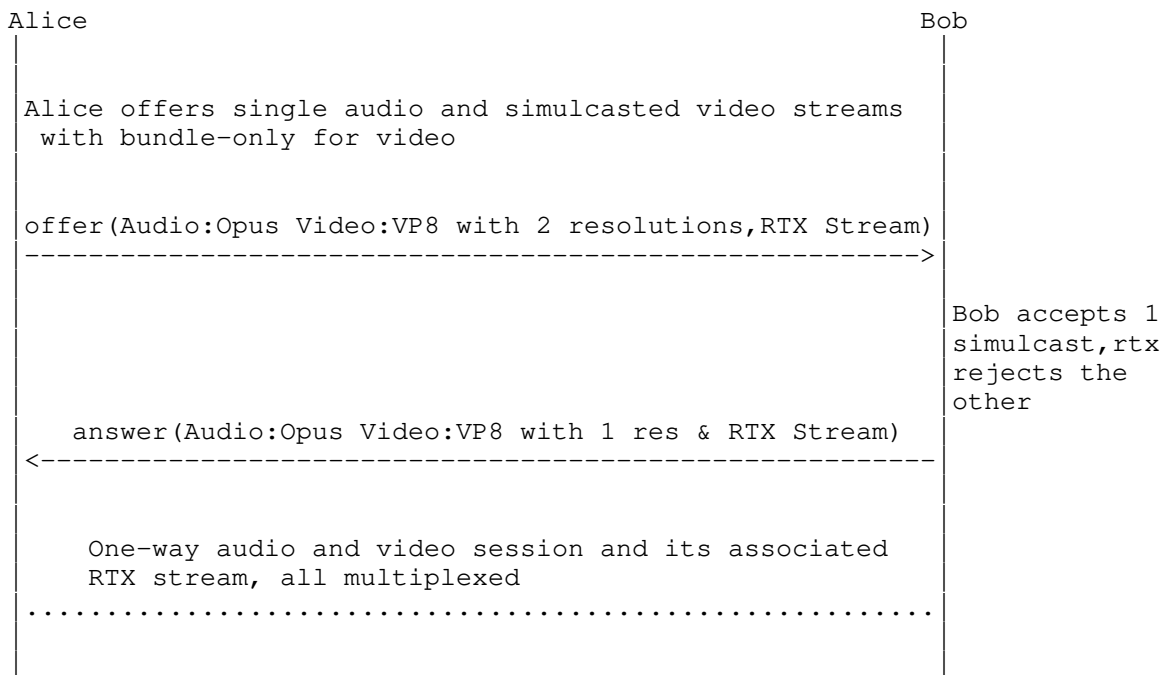
Table 32: 5.3.3 SDP Answer w/Simulcast, RTX

5.3.4. Successful One-way Simulcast Session with 2 resolutions and RTX
 - One resolution rejected

This section shows an SDP offer/answer exchange for a simulcast scenario with two resolutions.

It also showcases where Bob rejects one of the Simulcast Video streams, which results in the rejection of the associated repair stream implicitly.

Simulcast Streams with Retransmission Rejected




```

=====
| Offer SDP Contents
| RFC#/Notes
|
=====
| v=0
| [RFC4566]
|-----
| o=- 20519 0 IN IP4 0.0.0.0
| [RFC4566]
|-----
| s=-
| [RFC4566]
|-----
| t=0 0
| [RFC4566]
|-----
| a=group:BUNDLE m0 m1
| [I-D.ietf-mmusic-sdp-bundle-negotiation]
| Alice supports grouping of m=lines under
| BUNDLE semantics
|-----
| a=group:LS m0 m1
| [RFC5888]
|-----
| a=ice-options:trickle
| [I-D.ietf-mmusic-trickle-ice]
|-----
| a=ice-options:ice2
| [RFC8445]
|-----
| ***** Audio m=line *****
| *****
|-----
| m=audio 54609 UDP/TLS/RTP/SAVPF 109
| [RFC4566]
|-----
| c=IN IP4 203.0.113.141
| [RFC4566]
|-----
| a=mid:m0
| [RFC5888]
|-----

```

a=msid:ma ta	Identifies MediaStream ID (ma) and	
	MediaStreamTrack ID (ta)	
+-----+		
a=sendonly	[RFC3264]	
+-----+		
a=rtpmap:109 opus/48000/2	[RFC7587]	
+-----+		
a=maxptime:120	[RFC4566]	
+-----+		
a=ice-ufrag:074c6550	[I-D.ietf-mmusic-ice-sip-sdp]	
+-----+		
a=ice-pwd:a28a397a4c3f31747d1ee3474af08a068	[I-D.ietf-mmusic-ice-sip-sdp]	
+-----+		
a=fingerprint:sha-256	[RFC8122]	
19:E2:1C:3B:4B:9F:81:E6:B8:5C:F4:A5:A8:D8:73:04:BB:05:2F:70:9F:04:A9:0E:05:E9		
:26:33:E8:70:88:A2		
+-----+		
a=setup:actpass	[RFC5763]	
+-----+		
a=tls-id:89J2LRATQ3ULA24G9AHWVR31VJWSLB68	[I-D.ietf-mmusic-dtls-sdp]	

```

+-----+
+-----+
|a=rtcp-mux                               |
|                                           |
| [RFC5761]                               |
+-----+
+-----+
|a=rtcp-rsize                             |
|                                           |
| [RFC5506]                               |
+-----+
+-----+
|a=extmap:1 urn:iETF:params:rtp-hdext:ssrc-audio-level |
|                                           |
| [RFC6464]                               |
+-----+
+-----+
|a=extmap:2 urn:iETF:params:rtp-hdext:sdes:mid      |
|                                           |
| [I-D.iETF-mmusic-sdp-bundle-negotiation] |
+-----+
+-----+
|a=candidate:0 1 UDP 2113667327 192.0.2.4 61665 typ host |
|                                           |
| [I-D.iETF-mmusic-ice-sip-sdp]          |
+-----+
+-----+
|a=candidate:1 1 UDP 694302207 203.0.113.141 54609 typ srflx raddr 192.0.2.4 r |
port 61665                               |
|                                           |
| [I-D.iETF-mmusic-ice-sip-sdp]          |
+-----+
+-----+
|a=end-of-candidates                       |
|                                           |
| [I-D.iETF-mmusic-trickle-ice]         |
+-----+
+-----+
|***** Video m=line *****             |
|*****                                   |
+-----+
+-----+
|m=video 0 UDP/TLS/RTP/SAVPF 98 100 101 103 |
|                                           |
| bundle-only video line with port number |
|                                           |
| set to zero                             |
+-----+
+-----+
|c=IN IP4 203.0.113.141                    |
|                                           |
| [RFC4566]                               |
+-----+
+-----+
|a=bundle-only                             |
|                                           |
| [I-D.iETF-mmusic-sdp-bundle-negotiation] |
+-----+
+-----+
|a=mid:m1                                   |
|                                           |
| [RFC5888]                               |
+-----+
+-----+
|a=msid:ma tb                              |
|                                           |
| Identifies MediaStream ID (ma) and     |
|                                           |
| MediaStreamTrack ID (tb)               |
+-----+
+-----+

```

```

|a=sendonly
| [RFC3264]
+-----+
|a=rtpmap:98 VP8/90000
| [RFC7741]
+-----+
|a=rtpmap:100 VP8/90000
| [RFC7741]
+-----+
|a=rtpmap:101 rtx/90000
| [RFC4588]
+-----+
|a=rtpmap:103 rtx/90000
| [RFC4588]
+-----+
|a=fmtp:98 max-fr=30;max-fs=8040
| [RFC4566]
+-----+
|a=fmtp:100 max-fr=15;max-fs=1200
| [RFC4566]
+-----+
|a=fmtp:101 apt=98;rtx-time=200
| [RFC4588]
+-----+
|a=fmtp:103 apt=100;rtx-time=200
| [RFC4588]
+-----+
|a=rtcp-fb:* nack
| [RFC4585]

```

a=rtcp-fb:* nack pli	[RFC4585]	
a=rtcp-fb:* ccm fir	[RFC5104]	
a=extmap:2 urn:ietf:params:rtp-hdext:sdes:mid	[I-D.ietf-mmusic-sdp-bundle-negotiation]	
a=extmap:3 urn:ietf:params:rtp-hdext:sdes:rtp-stream-id	[I-D.ietf-avtext-rid]	
a=extmap:4 urn:ietf:params:rtp-hdext:sdes:repaired-rtp-stream-id	[I-D.ietf-avtext-rid]	
a=rid:1 send pt=98	[I-D.ietf-mmusic-rid] 1:1 mapping	
	between the PT and the 'rid' identifier	
a=rid:2 send pt=100	[I-D.ietf-mmusic-rid] 1:1 mapping	
	between the PT and the 'rid' identifier	
a=simulcast:send 1;2	[I-D.ietf-mmusic-sdp-simulcast]	

Table 33: 5.3.4 SDP Offer w/Simulcast, RTX

=====		
Answer SDP Contents	RFC#/Notes	
v=0	[RFC4566]	
o=- 20519 0 IN IP4 0.0.0.0	[RFC4566]	
s=-	[RFC4566]	

```

+-----+
|t=0 0                                     |
|                                     | [RFC4566] |
+-----+
|a=group:BUNDLE m0 m1                    |
|                                     | [I-D.ietf-mmusic-sdp-bundle-negotiation] |
|                                     | Bob supports grouping of m=lines under |
|                                     | BUNDLE semantics |
+-----+
|a=group:LS m0 m1                         |
|                                     | [RFC5888] |
+-----+
|a=ice-options:trickle                    |
|                                     | [I-D.ietf-mmusic-trickle-ice] |
+-----+
|a=ice-options:ice2                       |
|                                     | [RFC8445] |
+-----+
|***** Audio m=line *****            |
|*****                                  |
+-----+
|m=audio 49203 UDP/TLS/RTP/SAVPF 109     |
|                                     | [RFC4566] |
+-----+

```



```

|c=IN IP4 203.0.113.77
| [RFC4566]
+-----+
|a=mid:m0
| [RFC5888]
+-----+
|a=msid:ma ta
| Identifies MediaStream ID (ma) and
| MediaStreamTrack ID (ta)
+-----+
|a=recvonly
| [RFC3264]
+-----+
|a=rtpmap:109 opus/48000/2
| [RFC7587]
+-----+
|a=maxptime:120
| [RFC4566]
+-----+
|a=ice-ufraq:074c6550
| [I-D.ietf-mmusic-ice-sip-sdp]
+-----+
|a=ice-pwd:a28a397a4c3f31747dlee3474af08a068
| [I-D.ietf-mmusic-ice-sip-sdp]
+-----+
|a=fingerprint:sha-256
| [RFC8122]
| 6B:8B:F0:65:5F:78:E2:51:3B:AC:6F:F3:3F:46:1B:35:DC:B8:5F:64:1A:24:C2:43:F0:A1
:58:D0:A1:2C:19:08
+-----+
|a=setup:active
| [RFC5763]
+-----+
|a=tls-id:CJ6FF9ZZMJW7MDRJIR7XVIQM48GE1G31
| [I-D.ietf-mmusic-dtls-sdp]
+-----+
|a=rtcp-mux
| [RFC5761]
+-----+
|a=rtcp-rsize
| [RFC5506]
+-----+
|a=extmap:1 urn:ietf:params:rtp-hdext:ssrc-audio-level
| [RFC6464]

```

```

-----+-----
|a=extmap:2 urn:iETF:params:rtp-hdext:sdes:mid
| [I-D.iETF-mmusic-sdp-bundle-negotiation] |
-----+-----
|a=candidate:0 1 UDP 2113667326 198.51.100.7 51556 typ host
| [I-D.iETF-mmusic-ice-sip-sdp] |
-----+-----
|a=candidate:1 1 UDP 1694302206 203.0.113.77 49203 typ srflx raddr 198.51.100.
7 rport 51556 | [I-D.iETF-mmusic-ice-sip-sdp] |
-----+-----
|a=end-of-candidates
| [I-D.iETF-mmusic-trickle-ice] |
-----+-----
|***** Video m=line *****
| ***** |
-----+-----
|m=video 0 UDP/TLS/RTP/SAVPF 98 101
| BUNDLE accepted |
-----+-----
|c=IN IP4 203.0.113.77
| [RFC4566] |
-----+-----
|a=bundle-only
| [I-D.iETF-mmusic-sdp-bundle-negotiation] |
-----+-----
|a=mid:m1
| [RFC5888] |
-----+-----

```

a=msid:ma tb	Identifies MediaStream ID (ma) and	
	MediaStreamTrack ID (tb)	
+-----+		
a=recvonly	[RFC3264]	
+-----+		
a=rtpmap:98 VP8/90000	[RFC7741]	
+-----+		
a=rtpmap:101 rtx/90000	[RFC4588]	
+-----+		
a=fmtp:101 apt=98;rtx-time=200	[RFC4588]	
+-----+		
a=rtcp-fb:* nack	[RFC4585]	
+-----+		
a=rtcp-fb:* nack pli	[RFC4585]	
+-----+		
a=rtcp-fb:* ccm fir	[RFC5104]	
+-----+		
a=extmap:2 urn:ietf:params:rtp-hdext:sdes:mid	[I-D.ietf-mmusic-sdp-bundle-negotiation]	
+-----+		
a=extmap:3 urn:ietf:params:rtp-hdext:sdes:rtp-stream-id	[I-D.ietf-avtext-rid]	
+-----+		
a=extmap:4 urn:ietf:params:rtp-hdext:sdes:repaired-rtp-stream-id	[I-D.ietf-avtext-rid]	
+-----+		
a=rid:1 recv pt=98	[I-D.ietf-mmusic-rid]	
+-----+		
a=simulcast:recv 1	[I-D.ietf-mmusic-sdp-simulcast] Bob	
	rejects the second simulcast stream and	
	the associated rtx stream.	
+-----+		

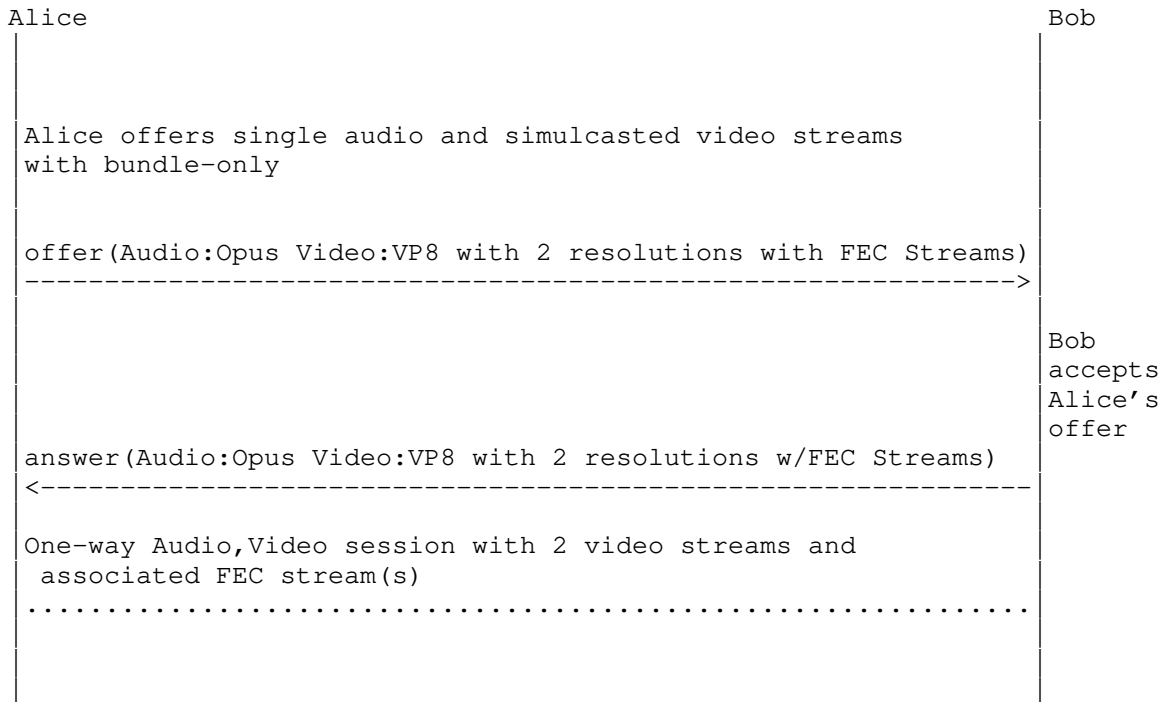
Table 34: 5.3.4 SDP Answer (one Simulcast Rejected)

5.3.5. Simulcast Video Session with Forward Error Correction

This section shows an SDP offer/answer exchange for a simulcast video stream at two resolutions with [RFC8627] based forward error correction mechanisms.

On completion of the offer/answer exchange mechanism, we end up with one audio stream, 2 simulcast video streams, and associated FEC stream(s), all of which are sent over a single 5-tuple.

Simulcast Streams with Forward Error Correction



Offer SDP Contents	RFC#/Notes
v=0	[RFC4566]
o=- 20519 0 IN IP4 0.0.0.0	[RFC4566]
s=-	[RFC4566]
t=0 0	[RFC4566]
a=group:BUNDLE m0 m1	[I-D.ietf-mmusic-sdp-bundle-negotiation]
	Alice supports grouping of m=lines under
	BUNDLE semantics
a=group:LS m0 m1	

```

+-----| [RFC5888] |-----+
|a=ice-options:trickle
| [I-D.ietf-mmusic-trickle-ice] |
+-----+-----+
|a=ice-options:ice2
| [RFC8445] |
+-----+-----+

```

```
|***** Audio m=line *****|
|*****|
+-----+
|m=audio 54609 UDP/TLS/RTP/SAVPF 109|
| [RFC4566]|
+-----+
|c=IN IP4 203.0.113.141|
| [RFC4566]|
+-----+
|a=mid:m0|
| [RFC5888]|
+-----+
|a=msid:ma ta|
| Identifies MediaStream ID (ma) and|
| |
| MediaStreamTrack ID (ta)|
+-----+
|a=sendonly|
| [RFC3264]|
+-----+
|a=rtpmap:109 opus/48000/2|
| [RFC7587]|
+-----+
|a=maxptime:120|
| [RFC4566]|
+-----+
|a=ice-ufrag:074c6550|
| [I-D.ietf-mmusic-ice-sip-sdp]|
+-----+
|a=ice-pwd:a28a397a4c3f31747dlee3474af08a068|
| [I-D.ietf-mmusic-ice-sip-sdp]|
+-----+
|a=fingerprint:sha-256|
| [RFC8122]|
| 19:E2:1C:3B:4B:9F:81:E6:B8:5C:F4:A5:A8:D8:73:04:BB:05:2F:70:9F:04:A9:0E:05:E9|
| :26:33:E8:70:88:A2|
+-----+
|a=setup:actpass|
| [RFC5763]|
+-----+
|a=rtcp-mux|
| [RFC5761]|
+-----+
|a=tls-id:89J2LRATQ3ULA24G9AHWVR31VJWSLB68|
| [I-D.ietf-mmusic-dtls-sdp]|
```

```

-----
|a=rtcp-rsize                               |
|                                             | [RFC5506]
-----
|a=extmap:1 urn:iETF:params:rtp-hdext:ssrc-audio-level
|                                             | [RFC6464]
-----
|a=extmap:2 urn:iETF:params:rtp-hdext:sdes:mid
|                                             | [I-D.iETF-mmusic-sdp-bundle-negotiation]
-----
|a=candidate:0 1 UDP 2113667327 192.0.2.4 61665 typ host
|                                             | [I-D.iETF-mmusic-ice-sip-sdp]
-----
|a=candidate:1 1 UDP 694302207 203.0.113.141 54609 typ srflx raddr 192.0.2.4 r
port 61665                                   | [I-D.iETF-mmusic-ice-sip-sdp]
-----
|a=end-of-candidates
|                                             | [I-D.iETF-mmusic-trickle-ice]
-----
|***** Video m=line *****
|*****
-----
|m=video 0 UDP/TLS/RTP/SAVPF 98 100 101
|                                             | bundle-only video line with port number
|
|                                             | set to zero
-----
|c=IN IP4 203.0.113.141
|                                             | [RFC4566]
-----

```



```
+-----+
+-----+
|a=bundle-only          |[I-D.ietf-mmusic-sdp-bundle-negotiation]|
+-----+
+-----+
|a=mid:m1              |[RFC5888] Video m=line part of BUNDLE |
|                      |group                    |
+-----+
+-----+
|a=msid:ma tb         |Identifies MediaStream ID (ma) and   |
|                      |MediaStreamTrack ID (tb)                 |
+-----+
+-----+
|a=sendonly            |[RFC3264]                    |
+-----+
+-----+
|a=rtpmap:98 VP8/90000 |[RFC7741]                    |
+-----+
+-----+
|a=rtpmap:100 VP8/90000|[RFC7741]                    |
+-----+
+-----+
|a=rtpmap:101 flexfec/90000|[RFC8627]                    |
+-----+
+-----+
|a=fmtp:98 max-fr=30;max-fs=8040|[RFC4566]                    |
+-----+
+-----+
|a=fmtp:100 max-fr=15;max-fs=1200|[RFC4566]                    |
+-----+
+-----+
|a=fmtp:101 repair-window=200000|[RFC8627]                    |
+-----+
+-----+
|a=rtcp-fb:* nack pli |[RFC4585]                    |
+-----+
+-----+
|a=rtcp-fb:* ccm fir   |[RFC5104]                    |
+-----+
+-----+
|a=extmap:2 urn:ietf:params:rtp-hdext:sdes:mid|[I-D.ietf-mmusic-sdp-bundle-negotiation]|
+-----+
+-----+
```

a=extmap:3 urn:ietf:params:rtp-hdext:sdes:rtp-stream-id	
[I-D.ietf-avtext-rid]	
+-----+	
a=extmap:4 urn:ietf:params:rtp-hdext:sdes:repaired-rtp-stream-id	
[I-D.ietf-avtext-rid]	
+-----+	
a=rid:1 send pt=98	
[I-D.ietf-mmusic-rid] 1:1 mapping	
between the PT and the 'rid' identifier	
+-----+	
a=rid:2 send pt=100	
[I-D.ietf-mmusic-rid] 1:1 mapping	
between the PT and the 'rid' identifier	
+-----+	
a=simulcast:send 1;2	
[I-D.ietf-mmusic-sdp-simulcast]	
+-----+	

Table 35: 5.3.5 SDP Offer

Answer SDP Contents	RFC#/Notes
v=0	[RFC4566]
o=- 20519 0 IN IP4 0.0.0.0	[RFC4566]
s=-	[RFC4566]
t=0 0	[RFC4566]
a=group:BUNDLE m0 m1	[I-D.ietf-mmusic-sdp-bundle-negotiation]
a=group:LS m0 m1	[RFC5888]
a=ice-options:trickle	[I-D.ietf-mmusic-trickle-ice]
a=ice-options:ice2	[RFC8445]
***** Audio m=line *****	*****
m=audio 49203 UDP/TLS/RTP/SAVPF 109	[RFC4566]
c=IN IP4 203.0.113.77	[RFC4566]
a=mid:m0	[RFC5888] Audio m=line part of BUNDLE group with a unique port number
a=msid:ma ta	Identifies MediaStream ID (ma) and

```

|
|                               |MediaStreamTrack ID (ta)                               |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|a=recvonly                       | [RFC3264]                               |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|a=rtpmap:109 opus/48000/2         | [RFC7587]                               |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|a=maxptime:120                   | [RFC4566]                               |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|a=ice-ufrag:074c6550             | [I-D.ietf-mmusic-ice-sip-sdp]          |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|a=ice-pwd:a28a397a4c3f31747dlee3474af08a068 | [I-D.ietf-mmusic-ice-sip-sdp]          |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|a=fingerprint:sha-256            | [RFC8122]                               |
|6B:8B:F0:65:5F:78:E2:51:3B:AC:6F:F3:3F:46:1B:35:DC:B8:5F:64:1A:24:C2:43:F0:A1
:58:D0:A1:2C:19:08|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|a=setup:active                   | [RFC5763]                               |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|a=tls-id:CJ6FF9ZZMJW7MDRJIR7XVIQM48GE1G31 | [I-D.ietf-mmusic-dtls-sdp]            |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```

```

|a=rtcp-mux                               |
|-----|-----|
+-----+-----+
|a=rtcp-rsize                             |
|-----|-----|
+-----+-----+
|a=extmap:1 urn:iETF:params:rtp-hdext:ssrc-audio-level
|-----|-----|
|                                     | [RFC6464] |
+-----+-----+
|a=extmap:2 urn:iETF:params:rtp-hdext:sdes:mid
|-----|-----|
|                                     | [I-D.iETF-mmusic-sdp-bundle-negotiation] |
+-----+-----+
|a=candidate:0 1 UDP 2113667326 198.51.100.7 51556 typ host
|-----|-----|
|                                     | [I-D.iETF-mmusic-ice-sip-sdp] |
+-----+-----+
|a=candidate:1 1 UDP 1694302206 203.0.113.77 49203 typ srflx raddr 198.51.100.
7 rport 51556 |-----|-----|
|                                     | [I-D.iETF-mmusic-ice-sip-sdp] |
+-----+-----+
|a=end-of-candidates
|-----|-----|
|                                     | [I-D.iETF-mmusic-trickle-ice] |
+-----+-----+
|***** Video m=line *****
|-----|-----|
|                                     | ***** |
+-----+-----+
|m=video 0 UDP/TLS/RTP/SAVPF 98 100 101
|-----|-----|
|                                     | BUNDLE accepted. |
+-----+-----+
|c=IN IP4 203.0.113.77
|-----|-----|
|                                     | [RFC4566] |
+-----+-----+
|a=bundle-only
|-----|-----|
|                                     | [I-D.iETF-mmusic-sdp-bundle-negotiation] |
+-----+-----+
|a=mid:m1
|-----|-----|
|                                     | [RFC5888] Video m=line part of BUNDLE |
|-----|-----|
|                                     | group |
+-----+-----+
|a=msid:ma tb
|-----|-----|
|                                     | Identifies MediaStream ID (ma) and |
|-----|-----|
|                                     | MediaStreamTrack ID (tb) |
+-----+-----+
|a=recvonly
|-----|-----|
|                                     | [RFC3264] |

```

```

-----+-----
|a=rtpmap:98 VP8/90000
| [RFC7741]
-----+-----
|a=rtpmap:100 VP8/90000
| [RFC7741]
-----+-----
|a=rtpmap:101 flexfec/90000
| [RFC8627]
-----+-----
|a=fmtp:98 max-fr=30;max-fs=8040
| [RFC4566]
-----+-----
|a=fmtp:100 max-fr=15;max-fs=1200
| [RFC4566]
-----+-----
|a=fmtp:101 repair-window=200000
| [RFC8627]
-----+-----
|a=rtcp-fb:* nack pli
| [RFC4585]
-----+-----
|a=rtcp-fb:* ccm fir
| [RFC5104]
-----+-----
|a=extmap:2 urn:ietf:params:rtp-hdext:sdes:mid
| [I-D.ietf-mmusic-sdp-bundle-negotiation]
-----+-----

```

a=extmap:3 urn:ietf:params:rtp-hdext:sdes:rtp-stream-id	
[I-D.ietf-avtext-rid]	
a=extmap:4 urn:ietf:params:rtp-hdext:sdes:repaired-rtp-stream-id	
[I-D.ietf-avtext-rid]	
a=rid:1 recv pt=98	
[I-D.ietf-mmusic-rid]	
a=rid:2 recv pt=100	
[I-D.ietf-mmusic-rid]	
a=simulcast:recv 1;2	
[I-D.ietf-mmusic-sdp-simulcast]	

Table 36: 5.3.5 SDP Answer

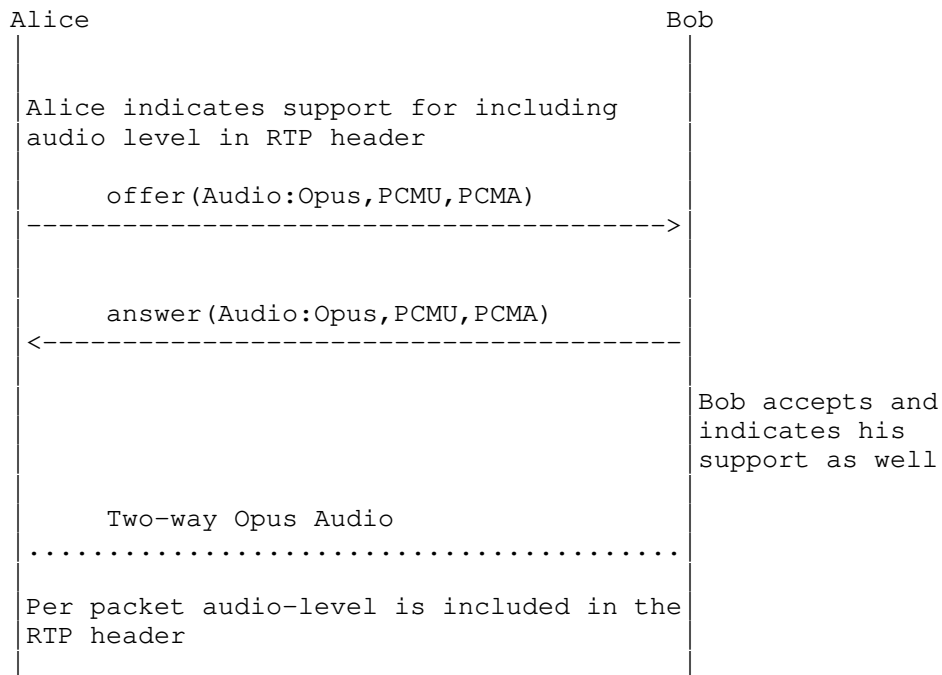
5.4. Others

The examples in this section provide SDP offer/answer exchanges for a variety of scenarios related to RTP Header extensions for conference usages, Legacy Interop scenarios, and more.

5.4.1. Audio Session - Voice Activity Detection

This example shows Alice indicating the support of the RTP header extension to include the audio level of the audio sample carried in the RTP packet.

Two-way Audio with VAD



Offer SDP Contents	
	RFC#/Notes
v=0	[RFC4566]
o=- 20518 0 IN IP4 0.0.0.0	[RFC4566]
s=-	[RFC4566]
t=0 0	[RFC4566]
a=group:BUNDLE audio	[I-D.ietf-mmusic-sdp-bundle-negotiation]
a=ice-options:trickle	[I-D.ietf-mmusic-trickle-ice]
a=ice-options:ice2	[RFC8445]

```
+-----+
|***** Audio m=line *****|
|*****|
+-----+
|m=audio 54609 UDP/TLS/RTP/SAVPF 109 0 8|
| [RFC4566]|
+-----+
|c=IN IP4 203.0.113.141|
| [RFC4566]|
+-----+
```



```

|a=rtcp-rsize          | [RFC5506]          |
+-----+-----+
|a=rtcp-fb:* nack     | [RFC4585]          |
+-----+-----+
|a=extmap:1 urn:iETF:params:rtp-hdext:ssrc-audio-level      |
| [RFC6464]          |
+-----+-----+
|a=extmap:2 urn:iETF:params:rtp-hdext:sdes:mid              |
| [I-D.iETF-mmusic-sdp-bundle-negotiation] |
+-----+-----+
|a=candidate:0 1 UDP 2113667327 192.0.2.4 61665 typ host    |
| [I-D.iETF-mmusic-ice-sip-sdp]          |
+-----+-----+
|a=candidate:1 1 UDP 694302207 203.0.113.141 54609 typ srflx raddr 192.0.2.4 r
port 61665          | [I-D.iETF-mmusic-ice-sip-sdp]          |
+-----+-----+
|a=end-of-candidates  | [I-D.iETF-mmusic-trickle-ice]          |
+-----+-----+

```

Table 37: 5.4.1 SDP Offer

```

=====
| Answer SDP Contents
|                               | RFC#/Notes |
=====
| v=0
|                               | [RFC4566] |
-----
| o=- 16833 0 IN IP4 0.0.0.0
|                               | [RFC4566] |
-----
| s=-
|                               | [RFC4566] |
-----
| t=0 0
|                               | [RFC4566] |
-----
| a=group:BUNDLE audio
|                               | [I-D.ietf-mmusic-sdp-bundle-negotiation] |
-----
| a=ice-options:trickle
|                               | [I-D.ietf-mmusic-trickle-ice] |
-----
| a=ice-options:ice2
|                               | [RFC8445] |
-----
| ***** Audio m=line *****
|                               | ***** |
-----
| m=audio 49203 UDP/TLS/RTP/SAVPF 109 0 8
|                               | [RFC4566] |
-----
| c=IN IP4 203.0.113.77
|                               | [RFC4566] |
-----
| a=mid:audio
|                               | [RFC5888] |
-----
| a=msid:ma ta
|                               | Identifies MediaStream ID (ma) and |
|                               | MediaStreamTrack ID (ta) |
-----
| a=sendrecv
|                               | [RFC3264] - Bob can send and recv audio |

```

```

-----+-----
|a=rtpmap:109 opus/48000/2
| [RFC7587] - Bob accepts only Opus Codec |
-----+-----
|a=rtpmap:0 PCMU/8000
| [RFC3551] PCMU Audio Codec |
-----+-----
|a=rtpmap:8 PCMA/8000
| [RFC3551] PCMA Audio Codec |
-----+-----
|a=maxptime:120
| [RFC4566] |
-----+-----
|a=ice-frag:c300d85b
| [I-D.ietf-mmusic-ice-sip-sdp] |
-----+-----
|a=ice-pwd:de4e99bd291c325921d5d47efbabd9a2
| [I-D.ietf-mmusic-ice-sip-sdp] |
-----+-----
|a=fingerprint:sha-256
| [RFC8122] |
|6B:8B:F0:65:5F:78:E2:51:3B:AC:6F:F3:3F:46:1B:35:DC:B8:5F:64:1A:24:C2:43:F0:A1
:58:D0:A1:2C:19:08|
-----+-----
|a=setup:active
| [RFC5763] |
-----+-----
|a=tls-id:CJ6FF9ZZMJW7MDRJIR7XVIQM48GE1G31
| [I-D.ietf-mmusic-dtls-sdp] |

```

```

+-----+
+-----+
|a=rtcp-mux          | [RFC5761] - Bob can perform RTP/RTCP |
|                   | Muxing on port 49203                |
+-----+
+-----+
|a=rtcp-rsize       | [RFC5506]                               |
+-----+
+-----+
|a=rtcp-fb:* nack   | [RFC4585]                               |
+-----+
+-----+
|a=extmap:1 urn:ietf:params:rtp-hdext:ssrc-audio-level |
|                   | [RFC6464]                               |
+-----+
+-----+
|a=extmap:2 urn:ietf:params:rtp-hdext:sdes:mid         |
|                   | [I-D.ietf-mmusic-sdp-bundle-negotiation] |
+-----+
+-----+
|a=candidate:0 1 UDP 2113667327 198.51.100.7 51556 typ host |
|                   | [I-D.ietf-mmusic-ice-sip-sdp]           |
+-----+
+-----+
|a=candidate:1 1 UDP 1694302207 203.0.113.77 49203 typ srflx raddr 198.51.100. |
7 rport 51556      | [I-D.ietf-mmusic-ice-sip-sdp]           |
+-----+
+-----+
|a=end-of-candidates | [I-D.ietf-mmusic-trickle-ice]           |
+-----+
+-----+

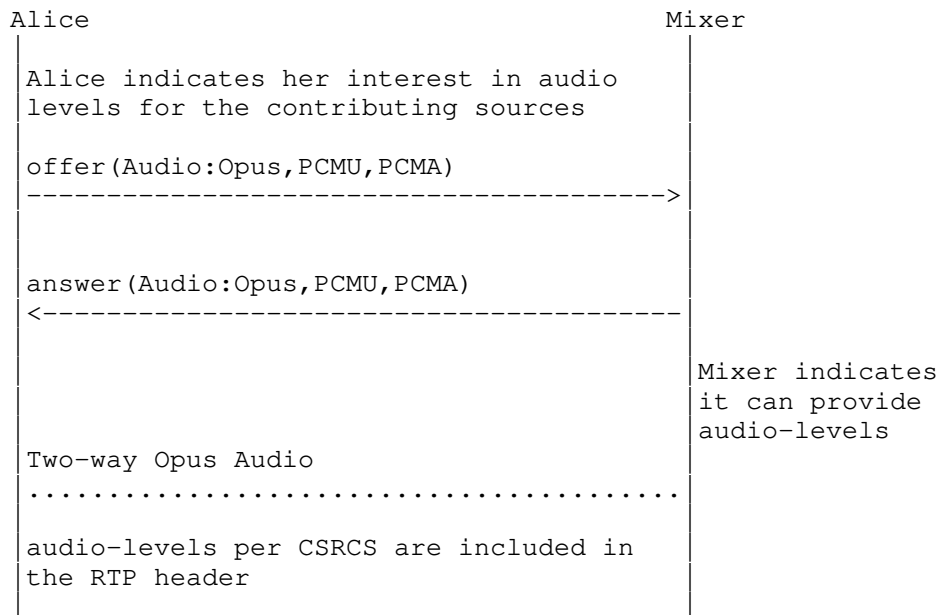
```

Table 38: 5.4.1 SDP Answer

5.4.2. Audio Conference - Voice Activity Detection

This example of SDP highlights the RTP header RTP header extension that allows RTP-level mixers in audio conferences to deliver information about the audio level of individual participants.

Audio Conference with VAD Support



Offer SDP Contents	
	RFC#/Notes
v=0	[RFC4566]
o=- 20518 0 IN IP4 0.0.0.0	[RFC4566] - Session Origin Information
s=-	[RFC4566]
t=0 0	[RFC4566]
a=group:BUNDLE audio	[I-D.ietf-mmusic-sdp-bundle-negotiation]
a=ice-options:trickle	[I-D.ietf-mmusic-trickle-ice]
a=ice-options:ice2	[RFC8445]

```
|***** Audio m=line *****  
|*****  
+-----+  
|m=audio 54609 UDP/TLS/RTP/SAVPF 109 0 8  
| [RFC4566]  
+-----+  
|c=IN IP4 203.0.113.141  
| [RFC4566]  
+-----+  
|a=mid:audio  
| [RFC5888]  
|
```

a=msid:ma ta	Identifies MediaStream ID (ma) and MediaStreamTrack ID (ta)	
a=sendrecv	[RFC3264] - Alice can send and recv audio	
a=rtpmap:109 opus/48000/2	[RFC7587]	
a=rtpmap:0 PCMU/8000	[RFC3551] PCMU Audio Codec	
a=rtpmap:8 PCMA/8000	[RFC3551] PCMA Audio Codec	
a=maxptime:120	[RFC4566]	
a=ice-ufrag:074c6550	[I-D.ietf-mmusic-ice-sip-sdp]	
a=ice-pwd:a28a397a4c3f31747d1ee3474af08a068	[I-D.ietf-mmusic-ice-sip-sdp]	
a=fingerprint:sha-256	[RFC8122]	
19:E2:1C:3B:4B:9F:81:E6:B8:5C:F4:A5:A8:D8:73:04:BB:05:2F:70:9F:04:A9:0E:05:E9:26:33:E8:70:88:A2		
a=setup:actpass	[RFC5763]	
a=tls-id:89J2LRATQ3ULA24G9AHWVR31VJWSLB68	[I-D.ietf-mmusic-dtls-sdp]	
a=rtcp-mux	[RFC5761]	
a=rtcp-rsize	[RFC5506]	

```

+-----+
|a=rtcp-fb:* nack
| [RFC4585]
+-----+
|a=extmap:1/recvonly urn:ietf:params:rtp-hdext:csrc-audio-level
| [RFC6465]
+-----+
|a=extmap:2 urn:ietf:params:rtp-hdext:ssrc-audio-level
| [RFC6464]
+-----+
|a=extmap:3 urn:ietf:params:rtp-hdext:sdes:mid
| [I-D.ietf-mmusic-sdp-bundle-negotiation]
+-----+
|a=candidate:0 1 UDP 2113667327 192.0.2.4 61665 typ host
| [I-D.ietf-mmusic-ice-sip-sdp]
+-----+
|a=candidate:1 1 UDP 694302207 203.0.113.141 54609 typ srflx raddr 192.0.2.4 r
port 61665 | [I-D.ietf-mmusic-ice-sip-sdp]
+-----+
|a=end-of-candidates
| [I-D.ietf-mmusic-trickle-ice]
+-----+

```

Table 39: 5.4.2 SDP Offer

```

=====
| Answer SDP Contents
|                               | RFC#/Notes |
=====
| v=0
|                               | [RFC4566] |
-----
| o=- 16833 0 IN IP4 0.0.0.0
|                               | [RFC4566] - Session Origin Information |
-----
| s=-
|                               | [RFC4566] |
-----
| t=0 0
|                               | [RFC4566] |
-----
| a=group:BUNDLE audio
|                               | [I-D.ietf-mmusic-sdp-bundle-negotiation] |
-----
| a=ice-options:trickle
|                               | [I-D.ietf-mmusic-trickle-ice] |
-----
| a=ice-options:ice2
|                               | [RFC8445] |
-----
| ***** Audio m=line *****
|                               | ***** |
-----
| m=audio 49203 UDP/TLS/RTP/SAVPF 109 0 8
|                               | [RFC4566] |
-----
| c=IN IP4 203.0.113.77
|                               | [RFC4566] |
-----
| a=mid:audio
|                               | [RFC5888] |
-----
| a=msid:ma ta
|                               | Identifies MediaStream ID (ma) and |
|                               | MediaStreamTrack ID (ta) |
-----
| a=sendrecv
|                               | [RFC3264] |

```

```

-----+-----
|a=rtpmap:109 opus/48000/2
|          | [RFC7587]          |
-----+-----
|a=rtpmap:0 PCMU/8000
|          | [RFC3551] PCMU Audio Codec |
-----+-----
|a=rtpmap:8 PCMA/8000
|          | [RFC3551] PCMA Audio Codec |
-----+-----
|a=maxptime:120
|          | [RFC4566]          |
-----+-----
|a=ice-ufrag:c300d85b
|          | [I-D.ietf-mmusic-ice-sip-sdp] |
-----+-----
|a=ice-pwd:de4e99bd291c325921d5d47efbabd9a2
|          | [I-D.ietf-mmusic-ice-sip-sdp] |
-----+-----
|a=fingerprint:sha-256
|          | [RFC8122]          |
|6B:8B:F0:65:5F:78:E2:51:3B:AC:6F:F3:3F:46:1B:35:DC:B8:5F:64:1A:24:C2:43:F0:A1
:58:D0:A1:2C:19:08|          |
-----+-----
|a=setup:active
|          | [RFC5763]          |
-----+-----
|a=tls-id:CJ6FF9ZZMJW7MDRJIR7XVIQM48GE1G31
|          | [I-D.ietf-mmusic-dtls-sdp] |

```

```

+-----+
+-----+
|a=rtcp-mux                               |
|                                     | [RFC5761] |
+-----+
+-----+
|a=rtcp-rsize                             |
|                                     | [RFC5506] |
+-----+
+-----+
|a=rtcp-fb:* nack                         |
|                                     | [RFC4585] |
+-----+
+-----+
|a=extmap:1/sendonly urn:ietf:params:rtp-hdrext:csrc-audio-level
|                                     | [RFC6465] |
+-----+
+-----+
|a=extmap:2 urn:ietf:params:rtp-hdrext:sdes:mid
|                                     | [I-D.ietf-mmusic-sdp-bundle-negotiation] |
+-----+
+-----+
|a=candidate:0 1 UDP 2113667327 198.51.100.7 51556 typ host
|                                     | [I-D.ietf-mmusic-ice-sip-sdp] |
+-----+
+-----+
|a=candidate:1 1 UDP 1694302207 203.0.113.77 49203 typ srflx raddr 198.51.100.
7 rport 51556 | [I-D.ietf-mmusic-ice-sip-sdp] |
+-----+
+-----+
|a=end-of-candidates                     |
|                                     | [I-D.ietf-mmusic-trickle-ice] |
+-----+
+-----+

```

Table 40: 5.4.2 SDP Answer

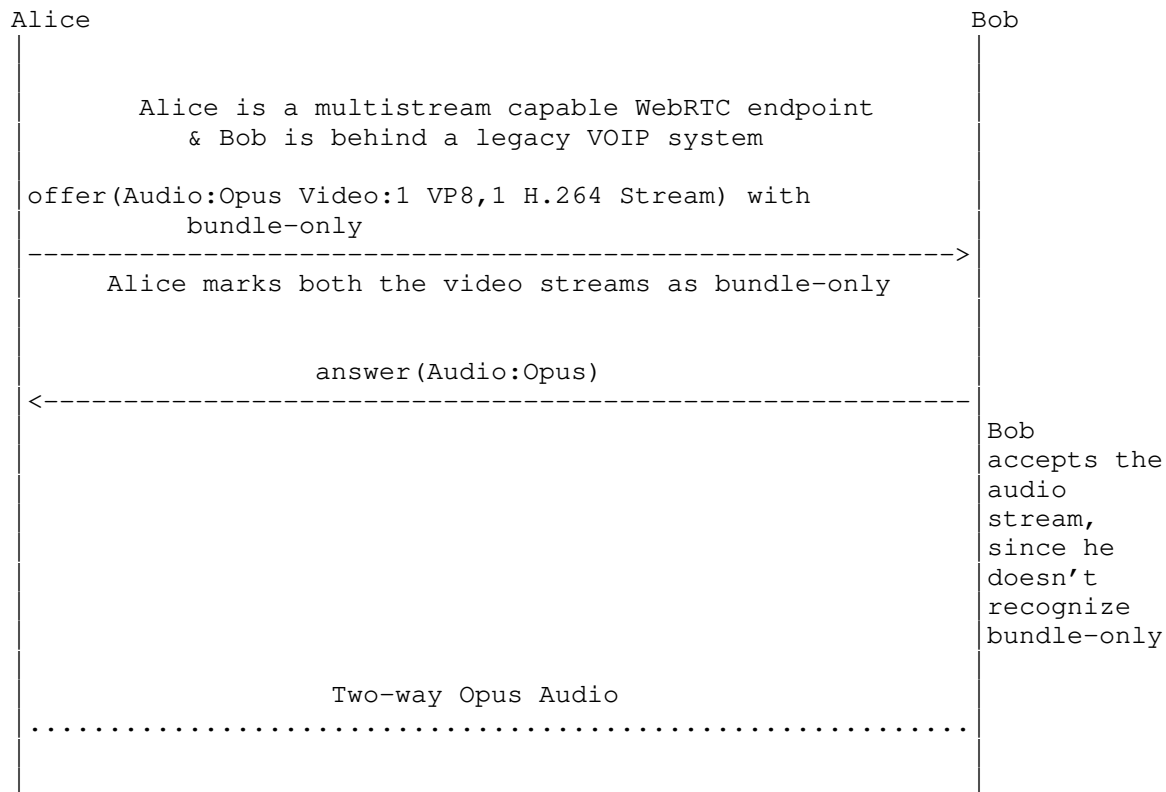
5.4.3. Successful Legacy Interop Fallback with bundle-only

In the scenario described below, Alice is a multi-stream capable WebRTC endpoint, while Bob is a legacy VOIP endpoint. The SDP offer/answer exchange demonstrates successful session setup with fallback to an audio-only stream negotiated via the bundle-only framework between the endpoints. Specifically:

- * The offer from Alice describes 2 cameras via 2 video m=lines with both marked as bundle-only.
- * Bob doesn't recognize the BUNDLE mechanism and since Alice has marked both the video m=lines with port 0, Bob accepts just the audio stream from Alice.

NOTE: Alice is unaware of whether Bob supports the BUNDLE framework. Alice includes separate RTP/RTCP ports and candidate information.

Successful Two-way WebRTC <-> VOIP Interop



Offer SDP Contents		RFC#/Notes
v=0	[RFC4566]	
o=- 20519 0 IN IP4 0.0.0.0	[RFC4566]	
s=-	[RFC4566]	
t=0 0	[RFC4566]	
a=group:BUNDLE m0 m1 m2	[I-D.ietf-mmusic-sdp-bundle-negotiation]	
	Alice supports grouping of m=lines under	
	BUNDLE semantics	

```
+-----+
|a=group:LS m0 m1
| [RFC5888]
+-----+
|a=ice-options:trickle
| [I-D.ietf-mmusic-trickle-ice]
+-----+
```

```

+-----+
+-----+
|a=ice-options:ice2
|           | [RFC8445]
+-----+
+-----+
|***** Audio m=line *****
|           | *****
+-----+
+-----+
|m=audio 54609 UDP/TLS/RTP/SAVPF 109
|           | [RFC4566]
+-----+
+-----+
|c=IN IP4 203.0.113.141
|           | [RFC4566]
+-----+
+-----+
|a=mid:m0
|           | [RFC5888] Audio m=line part of BUNDLE
|           |
|           | group with a unique port number
+-----+
+-----+
|a=msid:ma ta
|           | Identifies MediaStream ID (ma) and
|           |
|           | MediaStreamTrack ID (ta)
+-----+
+-----+
|a=sendrecv
|           | [RFC3264]
+-----+
+-----+
|a=rtpmap:109 opus/48000/2
|           | [RFC7587]
+-----+
+-----+
|a=maxptime:120
|           | [RFC4566]
+-----+
+-----+
|a=ice-ufrag:074c6550
|           | [I-D.ietf-mmusic-ice-sip-sdp]
+-----+
+-----+
|a=ice-pwd:a28a397a4c3f31747dlee3474af08a068
|           | [I-D.ietf-mmusic-ice-sip-sdp]
+-----+
+-----+
|a=fingerprint:sha-256
|           | [RFC8122]
|19:E2:1C:3B:4B:9F:81:E6:B8:5C:F4:A5:A8:D8:73:04:BB:05:2F:70:9F:04:A9:0E:05:E9
|:26:33:E8:70:88:A2
+-----+
+-----+
|a=setup:actpass
|           | [RFC5763]

```

```

-----
|a=tls-id:89J2LRATQ3ULA24G9AHWVR31VJWSLB68
| [I-D.ietf-mmusic-dtls-sdp]
-----
|a=rtcp-mux
| [RFC5761]
-----
|a=rtcp:64678 IN IP4 203.0.113.141
| [RFC3605]
-----
|a=rtcp-rsize
| [RFC5506]
-----
|a=extmap:1 urn:ietf:params:rtp-hdext:ssrc-audio-level
| [RFC6464]
-----
|a=extmap:2 urn:ietf:params:rtp-hdext:sdes:mid
| [I-D.ietf-mmusic-sdp-bundle-negotiation]
-----
|a=candidate:0 1 UDP 2113667327 192.0.2.4 61665 typ host
| [I-D.ietf-mmusic-ice-sip-sdp]
-----
|a=candidate:1 1 UDP 694302207 203.0.113.141 54609 typ srflx raddr 192.0.2.4 r
port 61665 | [I-D.ietf-mmusic-ice-sip-sdp]
-----
|a=candidate:0 1 UDP 2113667326 192.0.2.4 61667 typ host
| [I-D.ietf-mmusic-ice-sip-sdp]
-----

```

```

|a=candidate:1 1 UDP 1694302206 203.0.113.141 64678 typ srflx raddr 192.0.2.4
rport 61667 | [I-D.ietf-mmusic-ice-sip-sdp] |
+-----+
|***** Video-1 m=line *****
|*****|
+-----+
|m=video 0 UDP/TLS/RTP/SAVPF 98
| bundle-only video line with port number |
| set to zero |
+-----+
|c=IN IP4 203.0.113.141
| [RFC4566] |
+-----+
|a=bundle-only
| [I-D.ietf-mmusic-sdp-bundle-negotiation] |
+-----+
|a=mid:m1
| [RFC5888] Video m=line part of BUNDLE |
| group |
+-----+
|a=msid:ma tb
| Identifies MediaStream ID (ma) and |
| MediaStreamTrack ID (tb) |
+-----+
|a=sendrecv
| [RFC3264] |
+-----+
|a=rtpmap:98 VP8/90000
| [RFC7741] |
+-----+
|a=imageattr:98 [x=1280,y=720]
| [RFC6236] |
+-----+
|a=fmtp:98 max-fr=30
| [RFC4566] |
+-----+
|a=rtcp-fb:* nack
| [RFC4585] |
+-----+
|a=rtcp-fb:* nack pli
| [RFC4585] |
+-----+

```

```

|a=rtcp-fb:* ccm fir
| [RFC5104]
+-----+
|a=extmap:2 urn:ietf:params:rtp-hdext:sdes:mid
| [I-D.ietf-mmusic-sdp-bundle-negotiation]
+-----+
|***** Video-2 m=line *****
| *****
+-----+
|m=video 0 UDP/TLS/RTP/SAVPF 101
| bundle-only video line with port number
|
| set to zero
+-----+
|c=IN IP4 203.0.113.141
| [RFC4566]
+-----+
|a=bundle-only
| [I-D.ietf-mmusic-sdp-bundle-negotiation]
+-----+
|a=mid:m2
| [RFC5888] Video m=line part of BUNDLE
|
| group
+-----+
|a=msid:ma tc
| Identifies MediaStream ID (ma) and
|
| MediaStreamTrack ID (tc)
+-----+

```

a=sendrecv	[RFC3264]	
+-----+-----+		
a=rtpmap:101 H264/90000	[RFC6184]	
+-----+-----+		
a=fmtp:101 profile-level-id=4d0028;packetization-mode=1	[RFC6184]Camera-2,Encoding-1 Resolution	
+-----+-----+		
a=rtcp-fb:* nack	[RFC4585]	
+-----+-----+		
a=rtcp-fb:* nack pli	[RFC4585]	
+-----+-----+		
a=rtcp-fb:* ccm fir	[RFC5104]	
+-----+-----+		
a=extmap:2 urn:ietf:params:rtp-hdext:sdes:mid	[I-D.ietf-mmusic-sdp-bundle-negotiation]	
+-----+-----+		

Table 41: 5.4.3 SDP Simulcast bundle-only

+=====+		
Answer SDP Contents	RFC#/Notes	
+=====+		
v=0	[RFC4566]	
+-----+-----+		
o=- 20519 0 IN IP4 0.0.0.0	[RFC4566]	
+-----+-----+		
s=-	[RFC4566]	
+-----+-----+		
t=0 0	[RFC4566]	
+-----+-----+		
***** Audio m=line *****	*****	
+-----+-----+		

```
|m=audio 49203 UDP/TLS/RTP/SAVPF 109
| [RFC4566]
+-----+
|c=IN IP4 203.0.113.141
| [RFC4566]
+-----+
|a=rtcp:60065 IN IP4 203.0.113.141
| [RFC3605]
+-----+
|a=sendrecv
| [RFC3264]
+-----+
|a=rtpmap:109 opus/48000/2
| [RFC7587]
+-----+
|a=maxptime:120
| [RFC4566]
+-----+
|a=ice-ufrag:c300d85b
| [I-D.ietf-mmusic-ice-sip-sdp]
+-----+
|a=ice-pwd:de4e99bd291c325921d5d47efbabd9a2
| [I-D.ietf-mmusic-ice-sip-sdp]
+-----+
|a=fingerprint:sha-256
| [RFC8122]
```



```

|6B:8B:F0:65:5F:78:E2:51:3B:AC:6F:F3:3F:46:1B:35:DC:B8:5F:64:1A:24:C2:43:F0:A1
:58:D0:A1:2C:19:08|
+-----+
|a=setup:active|
|               |[RFC5763]|
+-----+
|a=rtcp-rsize  |
|               |[RFC5506]|
+-----+
|a=extmap:1 urn:ietf:params:rtp-hdext:ssrc-audio-level
|               |[RFC6464]|
+-----+
|a=candidate:0 1 UDP 2113667327 198.51.100.7 51556 typ host
|               |[I-D.ietf-mmusic-ice-sip-sdp]|
+-----+
|a=candidate:1 1 UDP 694302207 203.0.113.77 49203 typ srflx raddr 198.51.100.7
rport 51556     |[I-D.ietf-mmusic-ice-sip-sdp]|
+-----+
|a=candidate:0 2 UDP 2113667326 198.51.100.7 51558 typ host
|               |[I-D.ietf-mmusic-ice-sip-sdp]|
+-----+
|a=candidate:1 2 UDP 1694302206 203.0.113.77 60065 typ srflx raddr 198.51.100.
7 rport 51558   |[I-D.ietf-mmusic-ice-sip-sdp]|
+-----+
|***** Video m=line *****
|*****|
+-----+
|m=video 0 UDP/TLS/RTP/SAVPF 98
|      |Bob doesn't recognize bundle-|
|      |
|      |only and hence the m=line is |
|      |
|      |rejected implicitly due to  |
|      |
|      |port 0                       |
+-----+
|***** Video m=line *****
|*****|
+-----+
|m=video 0 UDP/TLS/RTP/SAVPF 101
|      |Bob doesn't recognize bundle-|
|      |
|      |only and hence the m=line is |
|      |
|      |rejected implicitly due to  |
|      |
|      |port 0                       |

```

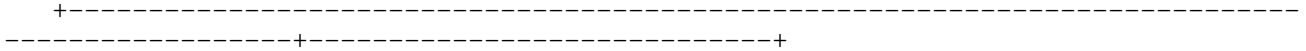


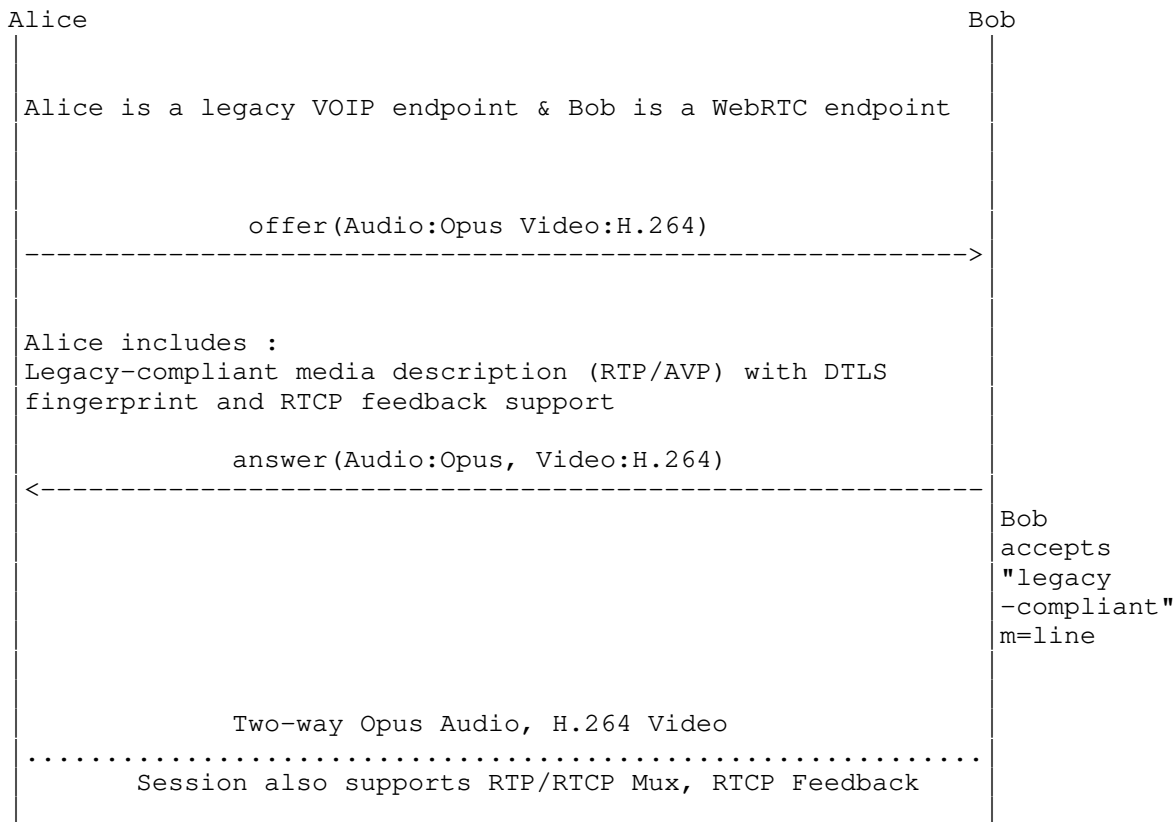
Table 42: 5.4.3 SDP Answer

5.4.4. Legacy Interop with RTP/AVP profile

In the scenario described below, Alice is a legacy endpoint that sends an [RFC3264] offer with RTP/AVP-based audio and video descriptions along with a DTLS fingerprint and RTCP feedback information.

On the other hand, Bob, being a WebRTC endpoint, follows the procedures in section 5.1.2 of [I-D.ietf-rtcweb-jsep] and accepts the Alice's offer for a DTLS-SRTP-based session with RTCP feedback.

Successful Two-way WebRTC <-> VOIP Interop



```

+=====+
|=====+=====+
| Offer SDP Contents
|                               |
|                               | RFC#/Notes
+=====+=====+
|v=0
|                               | [RFC4566]
+-----+-----+
|o=- 20518 0 IN IP4 0.0.0.0
|                               | [RFC4566]
+-----+-----+
|s=-
|                               | [RFC4566]
+-----+-----+
|t=0 0
|                               | [RFC4566]
+-----+-----+
|a=ice-ufrag:074c6550
|                               | [I-D.ietf-mmusic-ice-sip-sdp]
+-----+-----+
|a=ice-pwd:a28a397a4c3f31747dlee3474af08a068
  
```

```

+-----| [I-D.ietf-mmusic-ice-sip-sdp] |-----+
|a=rtcp-rsize| [RFC5506] |
+-----+-----+

```

```

|***** Audio m=line *****|
|*****|
+-----+
|m=audio 54732 RTP/AVP 109|
|      [RFC4566]Alice includes RTP/|
|      |
|      AVP audio stream description|
+-----+
|c=IN IP4 203.0.113.141|
|      [RFC4566]|
+-----+
|a=fingerprint:sha-256|
|      [RFC8122]|
|19:E2:1C:3B:4B:9F:81:E6:B8:5C:F4:A5:A8:D8:73:04:BB:05:2F:70:9F:04:A9:0E:05:E9|
|:26:33:E8:70:88:A2|
+-----+
|a=rtpmap:109 opus/48000|
|      |
+-----+
|a=ptime:20|
|      |
+-----+
|a=sendrecv|
|      [RFC3264]|
+-----+
|a=rtcp-mux|
|      [RFC5761]Alice still includes|
|      |
|      RTP/RTCP Mux support|
+-----+
|a=rtcp:64678 IN IP4 203.0.113.141|
|      [RFC3605]|
+-----+
|a=candidate:0 1 UDP 2113667327 192.0.2.4 54732 typ host|
|      [I-D.ietf-mmusic-ice-sip-sdp]|
+-----+
|a=candidate:1 1 UDP 694302207 203.0.113.141 54732 typ srflx raddr 192.0.2.4 r|
|port 54732|
|      [I-D.ietf-mmusic-ice-sip-sdp]|
+-----+
|a=candidate:0 2 UDP 2113667326 192.0.2.4 64678 typ host|
|      [I-D.ietf-mmusic-ice-sip-sdp]|
+-----+
|a=candidate:1 2 UDP 1694302206 203.0.113.141 64678 typ srflx raddr 192.0.2.4|
|rport 64678|
|      [I-D.ietf-mmusic-ice-sip-sdp]|
+-----+

```

```

|***** Video m=line *****
|*****|
+-----+
|m=video 62445 RTP/AVP 120
|      |[RFC4566]Alice includes RTP/
|
|      |AVP video stream description |
+-----+
|c=IN IP4 203.0.113.141
|      |[RFC4566]
+-----+
|a=fingerprint:sha-256
|      |[RFC8122]
|DC:B8:5F:64:1A:24:C2:43:F0:A1:58:D0:A1:2C:19:08:6B:8B:F0:65:5F:78:E2:51:3B:AC
:6F:F3:3F:46:1B:35|
+-----+
|a=rtpmap:120 VP8/90000
|      |[RFC7741]
+-----+
|a=sendrecv
|      |[RFC3264]
+-----+
|a=rtcp-mux
|      |[RFC5761]Alice intends to
|
|      |perform RTP/RTCP Mux
+-----+
|a=rtcp:54721 IN IP4 203.0.113.141
|      |[RFC3605]
+-----+

```

a=candidate:0 1 UDP 2113667327 192.0.2.4 62445 typ host	[I-D.ietf-mmusic-ice-sip-sdp]
+-----+-----+	
a=candidate:1 1 UDP 1694302207 203.0.113.141 62537 typ srflx raddr 192.0.2.4 rport 62445	[I-D.ietf-mmusic-ice-sip-sdp]
+-----+-----+	
a=candidate:0 2 UDP 2113667326 192.0.2.4 54721 typ host	[I-D.ietf-mmusic-ice-sip-sdp]
+-----+-----+	
a=candidate:1 2 UDP 1694302206 203.0.113.141 54721 typ srflx raddr 192.0.2.4 rport 54721	[I-D.ietf-mmusic-ice-sip-sdp]
+-----+-----+	
a=rtcp-fb:120 nack pli	[RFC4585] Alice indicates
	support for Picture Loss
	Indication and NACK RTCP
	feedback
+-----+-----+	
a=rtcp-fb:120 ccm fir	[RFC5104]
+-----+-----+	

Table 43: 5.4.5 SDP Offer

+=====+	
Answer SDP Contents	RFC#/Notes
+=====+	
v=0	[RFC4566]
+-----+-----+	
o=- 16833 0 IN IP4 0.0.0.0	[RFC4566]
+-----+-----+	
s=-	[RFC4566]
+-----+-----+	
t=0 0	[RFC4566]
+-----+-----+	
***** Audio m=line *****	*****

```

+-----+
|m=audio 49203 RTP/AVP 109
|          | [RFC4566] Bob accepts RTP/
|          | AVP-based audio stream
+-----+
|c=IN IP4 203.0.113.77
|          | [RFC4566]
+-----+
|a=rtpmap:109 opus/48000
|          |
+-----+
|a=ptime:20
|          |
+-----+
|a=sendrecv
|          | [RFC3264]
+-----+
|a=ice-ufrag:c300d85b
|          | [I-D.ietf-mmusic-ice-sip-sdp]
+-----+
|a=ice-pwd:de4e99bd291c325921d5d47efbabd9a2
|          | [I-D.ietf-mmusic-ice-sip-sdp]
+-----+
|a=fingerprint:sha-256
|          | [RFC8122]

```



```

|BB:05:2F:70:9F:04:A9:0E:05:E9:26:33:E8:70:88:A2:19:E2:1C:3B:4B:9F:81:E6:B8:5C
:F4:A5:A8:D8:73:04|
+-----+
|a=rtcp-mux|[RFC5761]|
+-----+
|a=candidate:0 1 UDP 2113667327 198.51.100.7 49203 typ host
|[I-D.ietf-mmusic-ice-sip-sdp]|
+-----+
|a=candidate:1 1 UDP 1694302207 203.0.113.77 49203 typ srflx raddr 198.51.100.
7 rport 49203|[I-D.ietf-mmusic-ice-sip-sdp]|
+-----+
|***** Video m=line *****
|*****|
+-----+
|m=video 63130 RTP/AVP 120
|[RFC4566] Bob accepts RTP/
|
|AVP-based video stram|
+-----+
|c=IN IP4 203.0.113.77
|[RFC4566]|
+-----+
|a=rtpmap:120 VP8/90000
|[RFC7741]|
+-----+
|a=sendrecv|[RFC3264]|
+-----+
|a=ice-ufraq:e39091na|[I-D.ietf-mmusic-ice-sip-sdp]|
+-----+
|a=ice-pwd:dbc325921d5dd29e4e99147efbabd9a2|[I-D.ietf-mmusic-ice-sip-sdp]|
+-----+
|a=fingerprint:sha-256|[RFC8122]|
|BB:0A:0E:05:E9:26:33:E8:70:88:A2:2F:70:9F:04:19:E2:1C:3B:4B:9F:81:56:2F:70:9F
:04:F4:A5:A8:D8|
+-----+
|a=rtcp-mux|[RFC5761]|
+-----+
|a=candidate:0 1 UDP 2113667327 198.51.100.7 63130 typ host
|[I-D.ietf-mmusic-ice-sip-sdp]|

```

```

+-----+
|a=candidate:1 1 UDP 1694302207 203.0.113.77 63130 typ srflx raddr 198.51.100.
7 rport 63130 | [I-D.ietf-mmusic-ice-sip-sdp] |
+-----+
|a=rtcp-fb:120 nack pli
| [RFC4585] |
+-----+
|a=rtcp-fb:120 ccm fir
| [RFC5104] |
+-----+

```

Table 44: 5.4.5 SDP Answer

6. IANA Considerations

This document requires no actions from IANA.

7. Security Considerations

The IETF has published separate documents [I-D.ietf-rtcweb-security-arch] [I-D.ietf-rtcweb-security] describing the security architecture for WebRTC as a whole.

In addition, since the SDP offer and answer messages can contain private information about addresses and sessions to be established between parties, if this information needs to be kept private, some security mechanism (using the TLS transport, for example) in the protocol used to carry the offers and answers must be used.

8. Acknowledgments

We would like to thank Justin Uberti, Chris Flo, Paul Kyzivat, Nils Ohlmeier, Flemming Andreason, Magnus Westerlund for their detailed review and inputs. Thanks to Adam Roach for providing a syntax validation script to help highlight syntax and formatting errors.

Thanks to experts at IESG for careful review and feedback.

Thanks to Peter Yee for a detailed Genart review and suggestions.

9. Change Log

[RFC EDITOR NOTE: Please remove this section when publishing]

Changes from draft-ietf-rtcweb-sdp-11

- * Address Magnus Westerlund Review comments.
- * Remove reference to RFC5245 and use ice-sip-sdp instead.
- * Add an ice2 example for RFC8445.

Changes from draft-ietf-rtcweb-sdp-09

- * Review feedback incorporated from Flemming Andreason in Section 5.4

Changes from draft-ietf-rtcweb-sdp-08

- * Fixed formatting and syntax issues pointed out by Adam Roach's validator script. List of issues are here:
<https://github.com/fluffy/ietf/issues?q=is%3Aissue+is%3Aclosed>
- * Align examples to match latest bundle specification as pointed by Christer Holmberg

Changes from draft-ietf-rtcweb-sdp-07

- * Incorporate review from Nils.

Changes from draft-ietf-rtcweb-sdp-06

- * Keep Alive Version.

Changes from draft-ietf-rtcweb-sdp-05

- * Title change.

Changes from draft-ietf-rtcweb-sdp-04

- * Add IPv6 Example.
- * Add a=rtcp-mux-only and fix a=rtcp in examples.
- * Fix Idnits.
- * Add Security Considerations section.

Changes from draft-ietf-rtcweb-sdp-02 to draft-ietf-rtcweb-sdp-04

- * Alignment with JSEP-19.
- * Added a=identity example.
- * Added a=dtls-id, a=group:LS in the examples.
- * Added Appendix section to capture list of checklists for the attributes.
- * Removed SSRC lines to match JSEP-19.
- * Closed open issues on a=fingerprint, a=rtcp and a=rtcp-mux-only from ietf96 to reflect JSEP-19.
- * Simplified Inter-op example

Changes from draft-ietf-rtcweb-sdp-02

- * Version increment to avoid expiry

Changes from draft-ietf-rtcweb-sdp-01

- * Complete face-lift

- * Added visual markers around m=lines to indicate their type, added spacing between tables for aiding readers
- * Updated table names to indicate offer vs answer
- * Attempted to align to latest versions of SCTP, BUNDLE, MSID drafts
- * Added mid header extensions to all the lines
- * Harmonized BUNDLE semantics and conventions updated.

Changes from draft-ietf-rtcweb-sdp-00

- * Updated Simulcast/FEC/RTX examples to use RID framework
- * Fixed BUNDLE references for a=bundle-only

Changes from draft-nandakumar-rtcweb-sdp-08

- * Fixed typos
- * Moved to a WG version

Changes from draft-nandakumar-rtcweb-sdp-06 and draft-nandakumar-rtcweb-sdp-07

- * Added clarification on Call-Flow diagram usage
- * More cleanups

Changes from draft-nandakumar-rtcweb-sdp-05

- * Added Ascii chart for all the SDP Eaxamples
- * Improved text and updated SDP Examples for Simulcast and FEC
- * Fixed MediaStream ID Semantics SDP Errors

Changes from draft-nandakumar-rtcweb-sdp-04

- * Interim version of the draft to avert expiry
- * Corrected placement of c= line as per RFC4566
- * Updated simulcast SDP to reflect draft-westerlund-avtcore-rtp-simulcast-04

Changes from draft-nandakumar-rtcweb-sdp-03

- * Aligned more closely with JSEP version -05
- * Added Conventions to help readability
- * Add more examples to clarify BUNDLE use cases

Changes from draft-nandakumar-rtcweb-sdp-02

- * Major refactoring was done to group the examples in to categories
- * SDP was updated through out to reflect JSEP-04 style of defining attributes per m=line than at the session level.
- * Added 8 new examples.
- * Updated references for Trickle, Unified Plan
- * Add section to explain the syntax conventions followed in the examples.

Changes from draft-nandakumar-rtcweb-sdp-01

- * Updated references to OPUS RTP Payload Specification.
- * Updated BUNDLE examples based on the latest draft-ietf-mmusic-sdp-bundle-negotiation.
- * Added examples for multiple audio and video flows based on Unified Plan.
- * Added new examples for RTX and FEC streams
- * Updated Simulcast and SVC examples

Changes from draft-nandakumar-rtcweb-sdp-00

- * Fixed editorial comments on the mailing list.
- * Updated Data-channel SDP information based on draft-ietf-mmusic-sctp-sdp.
- * Updated BUNDLE examples based on draft-ietf-mmusic-sdp-bundle-negotiation.
- * Added examples for few more BUNDLE variants
- * Added new examples for Simulcast and SVC

10. Informative References

- [RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, DOI 10.17487/RFC3264, June 2002, <<https://www.rfc-editor.org/info/rfc3264>>.
- [RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, DOI 10.17487/RFC4566, July 2006, <<https://www.rfc-editor.org/info/rfc4566>>.
- [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>>.
- [RFC5506] Johansson, I. and M. Westerlund, "Support for Reduced-Size Real-Time Transport Control Protocol (RTCP): Opportunities and Consequences", RFC 5506, DOI 10.17487/RFC5506, April 2009, <<https://www.rfc-editor.org/info/rfc5506>>.
- [RFC3551] Schulzrinne, H. and S. Casner, "RTP Profile for Audio and Video Conferences with Minimal Control", STD 65, RFC 3551, DOI 10.17487/RFC3551, July 2003, <<https://www.rfc-editor.org/info/rfc3551>>.
- [RFC4796] Hautakorpi, J. and G. Camarillo, "The Session Description Protocol (SDP) Content Attribute", RFC 4796, DOI 10.17487/RFC4796, February 2007, <<https://www.rfc-editor.org/info/rfc4796>>.
- [RFC5761] Perkins, C. and M. Westerlund, "Multiplexing RTP Data and Control Packets on a Single Port", RFC 5761, DOI 10.17487/RFC5761, April 2010, <<https://www.rfc-editor.org/info/rfc5761>>.
- [RFC5763] Fischl, J., Tschofenig, H., and E. Rescorla, "Framework for Establishing a Secure Real-time Transport Protocol (SRTP) Security Context Using Datagram Transport Layer Security (DTLS)", RFC 5763, DOI 10.17487/RFC5763, May 2010, <<https://www.rfc-editor.org/info/rfc5763>>.
- [RFC5104] Wenger, S., Chandra, U., Westerlund, M., and B. Burman, "Codec Control Messages in the RTP Audio-Visual Profile with Feedback (AVPF)", RFC 5104, DOI 10.17487/RFC5104, February 2008, <<https://www.rfc-editor.org/info/rfc5104>>.

- [RFC4585] Ott, J., Wenger, S., Sato, N., Burmeister, C., and J. Rey, "Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/AVPF)", RFC 4585, DOI 10.17487/RFC4585, July 2006, <<https://www.rfc-editor.org/info/rfc4585>>.
- [RFC4588] Rey, J., Leon, D., Miyazaki, A., Varsa, V., and R. Hakenberg, "RTP Retransmission Payload Format", RFC 4588, DOI 10.17487/RFC4588, July 2006, <<https://www.rfc-editor.org/info/rfc4588>>.
- [RFC5888] Camarillo, G. and H. Schulzrinne, "The Session Description Protocol (SDP) Grouping Framework", RFC 5888, DOI 10.17487/RFC5888, June 2010, <<https://www.rfc-editor.org/info/rfc5888>>.
- [RFC6236] Johansson, I. and K. Jung, "Negotiation of Generic Image Attributes in the Session Description Protocol (SDP)", RFC 6236, DOI 10.17487/RFC6236, May 2011, <<https://www.rfc-editor.org/info/rfc6236>>.
- [RFC6184] Wang, Y.-K., Even, R., Kristensen, T., and R. Jesup, "RTP Payload Format for H.264 Video", RFC 6184, DOI 10.17487/RFC6184, May 2011, <<https://www.rfc-editor.org/info/rfc6184>>.
- [RFC6190] Wenger, S., Wang, Y.-K., Schierl, T., and A. Eleftheriadis, "RTP Payload Format for Scalable Video Coding", RFC 6190, DOI 10.17487/RFC6190, May 2011, <<https://www.rfc-editor.org/info/rfc6190>>.
- [RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, DOI 10.17487/RFC3550, July 2003, <<https://www.rfc-editor.org/info/rfc3550>>.
- [RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, DOI 10.17487/RFC3261, June 2002, <<https://www.rfc-editor.org/info/rfc3261>>.
- [RFC3605] Huitema, C., "Real Time Control Protocol (RTCP) attribute in Session Description Protocol (SDP)", RFC 3605, DOI 10.17487/RFC3605, October 2003, <<https://www.rfc-editor.org/info/rfc3605>>.

- [RFC4733] Schulzrinne, H. and T. Taylor, "RTP Payload for DTMF Digits, Telephony Tones, and Telephony Signals", RFC 4733, DOI 10.17487/RFC4733, December 2006, <<https://www.rfc-editor.org/info/rfc4733>>.
- [RFC6464] Lennox, J., Ed., Iovov, E., and E. Marocco, "A Real-time Transport Protocol (RTP) Header Extension for Client-to-Mixer Audio Level Indication", RFC 6464, DOI 10.17487/RFC6464, December 2011, <<https://www.rfc-editor.org/info/rfc6464>>.
- [RFC6465] Iovov, E., Ed., Marocco, E., Ed., and J. Lennox, "A Real-time Transport Protocol (RTP) Header Extension for Mixer-to-Client Audio Level Indication", RFC 6465, DOI 10.17487/RFC6465, December 2011, <<https://www.rfc-editor.org/info/rfc6465>>.
- [RFC7587] Spittka, J., Vos, K., and JM. Valin, "RTP Payload Format for the Opus Speech and Audio Codec", RFC 7587, DOI 10.17487/RFC7587, June 2015, <<https://www.rfc-editor.org/info/rfc7587>>.
- [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>>.
- [RFC7741] Westin, P., Lundin, H., Glover, M., Uberti, J., and F. Galligan, "RTP Payload Format for VP8 Video", RFC 7741, DOI 10.17487/RFC7741, March 2016, <<https://www.rfc-editor.org/info/rfc7741>>.
- [RFC7826] Schulzrinne, H., Rao, A., Lanphier, R., Westerlund, M., and M. Stiemerling, Ed., "Real-Time Streaming Protocol Version 2.0", RFC 7826, DOI 10.17487/RFC7826, December 2016, <<https://www.rfc-editor.org/info/rfc7826>>.
- [RFC8122] Lennox, J. and C. Holmberg, "Connection-Oriented Media Transport over the Transport Layer Security (TLS) Protocol in the Session Description Protocol (SDP)", RFC 8122, DOI 10.17487/RFC8122, March 2017, <<https://www.rfc-editor.org/info/rfc8122>>.
- [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>>.
- [RFC8627] Zanaty, M., Singh, V., Bege, A., and G. Mandyam, "RTP Payload Format for Flexible Forward Error Correction (FEC)", RFC 8627, DOI 10.17487/RFC8627, July 2019, <<https://www.rfc-editor.org/info/rfc8627>>.
- [I-D.ietf-mmusic-sdp-bundle-negotiation]
Holmberg, C., Alvestrand, H., and C. Jennings,
"Negotiating Media Multiplexing Using the Session
Description Protocol (SDP)", Work in Progress, Internet-
Draft, draft-ietf-mmusic-sdp-bundle-negotiation-54, 14
December 2018, <<http://www.ietf.org/internet-drafts/draft-ietf-mmusic-sdp-bundle-negotiation-54.txt>>.
- [I-D.ietf-mmusic-sdp-simulcast]
Burman, B., Westerlund, M., Nandakumar, S., and M. Zanaty,
"Using Simulcast in SDP and RTP Sessions", Work in
Progress, Internet-Draft, draft-ietf-mmusic-sdp-simulcast-
14, 5 March 2019, <<http://www.ietf.org/internet-drafts/draft-ietf-mmusic-sdp-simulcast-14.txt>>.
- [I-D.ietf-mmusic-rid]
Roach, A., "RTP Payload Format Restrictions", Work in
Progress, Internet-Draft, draft-ietf-mmusic-rid-15, 15 May
2018, <<http://www.ietf.org/internet-drafts/draft-ietf-mmusic-rid-15.txt>>.
- [I-D.ietf-mmusic-ice-sip-sdp]
Petit-Huguenin, M., Nandakumar, S., Holmberg, C., Keranen,
A., and R. Shpount, "Session Description Protocol (SDP)
Offer/Answer procedures for Interactive Connectivity
Establishment (ICE)", Work in Progress, Internet-Draft,
draft-ietf-mmusic-ice-sip-sdp-39, 13 August 2019,
<<http://www.ietf.org/internet-drafts/draft-ietf-mmusic-ice-sip-sdp-39.txt>>.
- [I-D.ietf-rtcweb-jsep]
Uberti, J., Jennings, C., and E. Rescorla, "JavaScript
Session Establishment Protocol", Work in Progress,
Internet-Draft, draft-ietf-rtcweb-jsep-26, 27 February
2019, <<http://www.ietf.org/internet-drafts/draft-ietf-rtcweb-jsep-26.txt>>.

[I-D.ietf-mmusic-trickle-ice]

Ivov, E., Rescorla, E., and J. Uberti, "Trickle ICE: Incremental Provisioning of Candidates for the Interactive Connectivity Establishment (ICE) Protocol", Work in Progress, Internet-Draft, draft-ietf-mmusic-trickle-ice-02, 15 January 2015, <<http://www.ietf.org/internet-drafts/draft-ietf-mmusic-trickle-ice-02.txt>>.

[I-D.ietf-mmusic-msid]

Alvestrand, H., "WebRTC MediaStream Identification in the Session Description Protocol", Work in Progress, Internet-Draft, draft-ietf-mmusic-msid-17, 13 December 2018, <<http://www.ietf.org/internet-drafts/draft-ietf-mmusic-msid-17.txt>>.

[I-D.ietf-mmusic-sctp-sdp]

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.", Work in Progress, Internet-Draft, draft-ietf-mmusic-sctp-sdp-26, 20 April 2017, <<http://www.ietf.org/internet-drafts/draft-ietf-mmusic-sctp-sdp-26.txt>>.

[I-D.ietf-rtcweb-data-channel]

Jesup, R., Loreto, S., and M. Tuexen, "WebRTC Data Channels", Work in Progress, Internet-Draft, draft-ietf-rtcweb-data-channel-13, 4 January 2015, <<http://www.ietf.org/internet-drafts/draft-ietf-rtcweb-data-channel-13.txt>>.

[I-D.ietf-mmusic-mux-exclusive]

Holmberg, C., "Indicating Exclusive Support of RTP/RTCP Multiplexing using SDP", Work in Progress, Internet-Draft, draft-ietf-mmusic-mux-exclusive-12, 5 May 2017, <<http://www.ietf.org/internet-drafts/draft-ietf-mmusic-mux-exclusive-12.txt>>.

[I-D.ietf-mmusic-dtls-sdp]

Holmberg, C. and R. Shpount, "Session Description Protocol (SDP) Offer/Answer Considerations for Datagram Transport Layer Security (DTLS) and Transport Layer Security (TLS)", Work in Progress, Internet-Draft, draft-ietf-mmusic-dtls-sdp-32, 29 October 2017, <<http://www.ietf.org/internet-drafts/draft-ietf-mmusic-dtls-sdp-32.txt>>.

- [I-D.ietf-rtcweb-security-arch]
Rescorla, E., "WebRTC Security Architecture", Work in Progress, Internet-Draft, draft-ietf-rtcweb-security-arch-20, 22 July 2019, <<http://www.ietf.org/internet-drafts/draft-ietf-rtcweb-security-arch-20.txt>>.
- [I-D.ietf-rtcweb-security]
Rescorla, E., "Security Considerations for WebRTC", Work in Progress, Internet-Draft, draft-ietf-rtcweb-security-12, 5 July 2019, <<http://www.ietf.org/internet-drafts/draft-ietf-rtcweb-security-12.txt>>.
- [I-D.ietf-avtext-rid]
Roach, A., Nandakumar, S., and P. Thatcher, "RTP Stream Identifier Source Description (SDS)", Work in Progress, Internet-Draft, draft-ietf-avtext-rid-09, 6 October 2016, <<http://www.ietf.org/internet-drafts/draft-ietf-avtext-rid-09.txt>>.
- [WebRTC] W3C, "WebRTC 1.0: Real-time Communication Between Browsers", <<http://dev.w3.org/2011/webrtc/editor/webrtc.html>> .

Appendix A. Appendix

A.1. JSEP SDP Attributes Checklist

This section compiles a high-level checklist of the required SDP attributes to be verified against the examples defined in this specification. The goal here is to ensure that the examples are compliant with the rules defined in section 5 of the [I-D.ietf-rtcweb-jsep] specification.

A.1.1. Common Checklist

This subsection lists SDP attributes that mostly apply at the session level.

- * v=0 MUST be the first SDP line.
- * o= line MUST follow with values '-' for username, a 64-bit value for session id and dummy values for 'nettype', 'addrtype', and 'unicast-address' (for example: IN IP4 0.0.0.0).
- * o= line MUST have the session version incremented in the case of subsequent offers.
- * s= MUST be the third line and have the value of '-'.

- * t= line MUST follow with the values for 'start-time' and 'stop-time' set to zeroes.
- * a=identity line MUST be included at the session level if WebRTC Identity mechanism is being used.
- * a=ice-options:trickle MUST be present at the session level in all offers and answers when supported.
- * a=ice-options:ice3 MUST be present at the session level in all offers and answers when supported.

A.1.2. RTP Media Description Checklist

The following set of checklist items apply to RTP audio and video media descriptions.

- * The media description's port value MUST either be set to the dummy value of '9' or MUST use the port from the default candidate, if available.
- * The media description's proto value MUST be 'UDP/TLS/RTP/SAVPF' for JSEP offers.
- * The JSEP answerer MUST support any combination of "RTP/[S]AVP[F]" for interoperability scenarios as defined in section 5 of [I-D.ietf-rtcweb-jsep]
- * c= line MUST be the first line in a media description. A dummy value of 'IN IP 0.0.0.0' is set if there are no candidates gathered or its value MUST match the default candidate.
- * a=mid attribute MUST be included.
- * One of a=sendrecv/a=sendonly/a=recvonly/a=inactive SDP direction attributes MUST be present.
- * a=rtpmap and a=fmtp attributes per primary, retransmission, and forward error correction media format MUST be included.
- * a=rtcp-fb lines for each supported feedback mechanism MUST be included when using RTP with feedback.
- * a=imageattr can be optionally present for video media descriptions.

- * a=msid line MUST be included for all the media senders identifying the MediaStreamTrack (i.e., when a=sendonly/a=sendrecv attribute is present).
- * a=extmap line identifying the RTP MID header extension" MUST be present.
- * a=extmap line for other supported RTP header extensions MUST be included.
- * a=rid line 'per encoding' with the direction of 'send' MUST be included when further constraining the media format or multiple encodings per media format is needed.
- * a=simulcast line MUST be present if there exists more than one 'a=rid' lines for the media senders.
- * a=bundle-only attribute MUST be present for media descriptions that are impacted by various bundle policies (such as max-bundle/balanced)
- * For media descriptions that aren't "a=bundle-only" and that have unique address, the following attributes MUST be present:
 - a=ice-ufrag and a=ice-pwd
 - a=fingerprint
 - a=setup with value 'actpass' in the offers and a value of 'active'/'passive' in the answerer.
 - a=tls-id
 - a=rtcp
 - a=rtcp-mux
 - For offerers requiring RTCP to be multiplexed, 'a=rtcp-mux-only' line
 - a=rtcp-rsize
- * a=group:BUNDLE line with all the 'mid' identifiers part of the BUNDLE group is included at the session-level.
- * a=group:LS session level attribute MUST be included with the 'mid' identifiers that are part of the same lip sync group.

A.1.3. Data Channel Media Description checklist

If a data channel is required, an 'application' type media description MUST be included with the following properties:

- * Media description's proto value MUST be 'UDP/DTLS/SCTP' in the JSEP offers.
- * A JSEP answerer MUST support reception of 'UDP/DTLS/SCTP'/'TCP/DTLS/SCTP'/'DTLS/SCTP' for backward compatibility reasons.
- * A value of 'webrtc-datachannel' MUST be used for the media description 'fmt' value.
- * a=mid line MUST be present.
- * a=sctp-port with an SCTP port number MUST be included.
- * a=max-message-size MAY be included, if appropriate.

Authors' Addresses

Suhas Nandakumar
Cisco
170 West Tasman Drive
San Jose, CA 95134
United States of America

Email: snandaku@cisco.com

Cullen Jennings
Cisco
170 West Tasman Drive
San Jose, CA 95134
United States of America

Phone: +1 408 421-9990
Email: fluffy@cisco.com