Internet Engineering Task Force (IETF)
Request for Comments: 7742
Category: Standards Track
ISSN: 2070-1721

A.B. Roach
Mozilla

March 2016

WebRTC Video Processing and Codec Requirements

Abstract

This specification provides the requirements and considerations for WebRTC applications to send and receive video across a network. It specifies the video processing that is required as well as video codecs and their parameters.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc7742.

Table of Contents

1. Introduction

One of the major functions of WebRTC endpoints is the ability to send and receive interactive video. The video might come from a camera, a screen recording, a stored file, or some other source. This specification provides the requirements and considerations for WebRTC applications to send and receive video across a network. It specifies the video processing that is required as well as video codecs and their parameters.

Note that this document only discusses those issues dealing with video-codec handling. Issues that are related to transport of media streams across the network are specified in [WebRTC-RTP-USAGE].

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

The following definitions are used in this document:

- A WebRTC browser (also called a WebRTC User Agent or WebRTC UA) is something that conforms to both the protocol specification and the Javascript API (see [RTCWEB-OVERVIEW]).

- A WebRTC non-browser is something that conforms to the protocol specification, but it does not claim to implement the Javascript API. This can also be called a "WebRTC device" or "WebRTC native application".

- A WebRTC endpoint is either a WebRTC browser or a WebRTC non-browser. It conforms to the protocol specification.

- A WebRTC-compatible endpoint is an endpoint that is able to successfully communicate with a WebRTC endpoint but may fail to meet some requirements of a WebRTC endpoint. This may limit where in the network such an endpoint can be attached, or it may limit the security guarantees that it offers to others. It is not constrained by this specification; when it is mentioned at all, it is to note the implications on WebRTC-compatible endpoints of the requirements placed on WebRTC endpoints.

These definitions are also found in [RTCWEB-OVERVIEW] and that document should be consulted for additional information.

3. Pre- and Post-Processing

This section provides guidance on pre- and post-processing of video streams.

Unless specified otherwise by the Session Description Protocol (SDP) or codec, the color space SHOULD be sRGB [SRGB]. For clarity, this is the color space indicated by codepoint 1 from "ColourPrimaries" as defined in [IEC23001-8].

Unless specified otherwise by the SDP or codec, the video scan pattern for video codecs is Y’CbCr 4:2:0.

3.1. Camera-Source Video

This document imposes no normative requirements on camera capture; however, implementors are encouraged to take advantage of the following features, if feasible for their platform:

- Automatic focus, if applicable for the camera in use

- Automatic white balance

- Automatic light-level control

- Dynamic frame rate for video capture based on actual encoding in use (e.g., if encoding at 15 fps due to bandwidth constraints, low light conditions, or application settings, the camera will ideally capture at 15 fps rather than a higher rate).

3.2. Screen-Source Video

If the video source is some portion of a computer screen (e.g., desktop or application sharing), then the considerations in this section also apply.

Because screen-sourced video can change resolution (due to, e.g., window resizing and similar operations), WebRTC-video recipients MUST be prepared to handle midstream resolution changes in a way that preserves their utility. Precise handling (e.g., resizing the element a video is rendered in versus scaling down the received stream; decisions around letter/pillarboxing) is left to the discretion of the application.

Note that the default video-scan format (Y'CbCr 4:2:0) is known to be less than optimal for the representation of screen content produced by most systems in use at the time of this document's writing, which generally use RGB with at least 24 bits per sample. In the future, it may be advisable to use video codecs optimized for screen content for the representation of this type of content.

Additionally, attention is drawn to the requirements in Section 5.2 of [WebRTC-SEC-ARCH] and the considerations in Section 4.1.1. of [WebRTC-SEC].

4. Stream Orientation

In some circumstances -- and notably those involving mobile devices -- the orientation of the camera may not match the orientation used by the encoder. Of more importance, the orientation may change over the course of a call, requiring the receiver to change the orientation in which it renders the stream.

While the sender may elect to simply change the pre-encoding orientation of frames, this may not be practical or efficient (in particular, in cases where the interface to the camera returns pre-compressed video frames). Note that the potential for this behavior adds another set of circumstances under which the resolution of a screen might change in the middle of a video stream, in addition to those mentioned in Section 3.2.

To accommodate these circumstances, WebRTC implementations that can generate media in orientations other than the default MUST support generating the R0 and R1 bits of the Coordination of Video Orientation (CVO) mechanism described in Section 7.4.5 of [TS26.114] and MUST send them for all orientations when the peer indicates support for the mechanism. They MAY support sending the other bits in the CVO extension, including the higher-resolution rotation bits. All implementations SHOULD support interpretation of the R0 and R1 bits and MAY support the other CVO bits.

Further, some codecs support in-band signaling of orientation (for example, the SEI "Display Orientation" messages in H.264 and H.265 [H265]). If CVO has been negotiated, then the sender MUST NOT make use of such codec-specific mechanisms. However, when support for CVO is not signaled in the SDP, then such implementations MAY make use of the codec-specific mechanisms instead.

5. Mandatory-to-Implement Video Codec

For the definitions of "WebRTC browser", "WebRTC non-browser", and "WebRTC-compatible endpoint" as they are used in this section, please refer to Section 2.

WebRTC Browsers MUST implement the VP8 video codec as described in [RFC6386] and H.264 Constrained Baseline as described in [H264].

WebRTC Non-Browsers that support transmitting and/or receiving video MUST implement the VP8 video codec as described in [RFC6386] and H.264 Constrained Baseline as described in [H264].

> NOTE: To promote the use of non-royalty-bearing video codecs, participants in the RTCWEB working group, and any successor working groups in the IETF, intend to monitor the evolving licensing landscape as it pertains to the two mandatory-to-implement codecs. If compelling evidence arises that one of the codecs is available for use on a royalty-free basis, the working group plans to revisit the question of which codecs are required for Non-Browsers, with the intention being that the royalty-free codec will remain mandatory to implement and the other will become optional.
>
> These provisions apply to WebRTC Non-Browsers only. There is no plan to revisit the codecs required for WebRTC Browsers.

"WebRTC-compatible endpoints" are free to implement any video codecs they see fit. This follows logically from the definition of "WebRTC-compatible endpoint". It is, of course, advisable to implement at least one of the video codecs that is mandated for WebRTC browsers, and implementors are encouraged to do so.

6. Codec-Specific Considerations

SDP allows for codec-independent indication of preferred video resolutions using the mechanism described in [RFC6236]. WebRTC endpoints MAY send an "a=imageattr" attribute to indicate the maximum resolution they wish to receive. Senders SHOULD interpret and honor this attribute by limiting the encoded resolution to the indicated maximum size, as the receiver may not be capable of handling higher resolutions.

Additionally, codecs may include codec-specific means of signaling maximum receiver abilities with regard to resolution, frame rate, and bitrate.

Unless otherwise signaled in SDP, recipients of video streams MUST be able to decode video at a rate of at least 20 fps at a resolution of at least 320 pixels by 240 pixels. These values are selected based on the recommendations in [HSUP1].

Encoders are encouraged to support encoding media with at least the same resolution and frame rates cited above.

6.1. VP8

For the VP8 codec, defined in [RFC6386], endpoints MUST support the payload formats defined in [RFC7741].

In addition to the [RFC6236] mechanism, VP8 encoders MUST limit the streams they send to conform to the values indicated by receivers in the corresponding max-fr and max-fs SDP attributes.

Unless otherwise signaled, implementations that use VP8 MUST encode and decode pixels with an implied 1:1 (square) aspect ratio.

6.2. H.264

For the [H264] codec, endpoints MUST support the payload formats defined in [RFC6184]. In addition, they MUST support Constrained Baseline Profile Level 1.2 and SHOULD support H.264 Constrained High Profile Level 1.3.

Implementations of the H.264 codec have utilized a wide variety of optional parameters. To improve interoperability, the following parameter settings are specified:

packetization-mode: Packetization-mode 1 MUST be supported. Other modes MAY be negotiated and used.

profile-level-id: Implementations MUST include this parameter within SDP and MUST interpret it when receiving it.

max-mbps, max-smbps, max-fs, max-cpb, max-dpb, and max-br:

These parameters allow the implementation to specify that they can support certain features of H.264 at higher rates and values than those signaled by their level (set with profile-level-id). Implementations MAY include these parameters in their SDP, but they SHOULD interpret them when receiving them, allowing them to send the highest quality of video possible.

sprop-parameter-sets: H.264 allows sequence and picture information to be sent both in-band and out-of-band. WebRTC implementations MUST signal this information in-band. This means that WebRTC implementations MUST NOT include this parameter in the SDP they generate.

H.264 codecs MAY send and MUST support proper interpretation of Supplemental Enhancement Information (SEI) "filler payload" and "full frame freeze" messages. The "full frame freeze" messages are used in video-switching MCUs, to ensure a stable decoded displayed picture while switching among various input streams.

When the use of the video orientation (CVO) RTP header extension is not signaled as part of the SDP, H.264 implementations MAY send and SHOULD support proper interpretation of Display Orientation SEI messages.

Implementations MAY send and act upon "User data registered by Rec. ITU-T T.35" and "User data unregistered" messages. Even if they do not act on them, implementations MUST be prepared to receive such messages without any ill effects.

Unless otherwise signaled, implementations that use H.264 MUST encode and decode pixels with an implied 1:1 (square) aspect ratio.

7. Security Considerations

This specification does not introduce any new mechanisms or security concerns beyond what is in the other documents it references. In WebRTC, video is protected using Datagram Transport Layer Security (DTLS) / Secure Real-time Transport Protocol (SRTP). A complete discussion of the security considerations can be found in [WebRTC-SEC] and [WebRTC-SEC-ARCH]. Implementors should consider whether the use of variable bitrate video codecs are appropriate for their application, keeping in mind that the degree of inter-frame change (and, by inference, the amount of motion in the frame) may be deduced by an eavesdropper based on the video stream's bitrate.

Implementors making use of H.264 are also advised to take careful note of the "Security Considerations" section of [RFC6184], paying special regard to the normative requirement pertaining to SEI messages.

8. References

8.1. Normative References

[H264] ITU-T, "Advanced video coding for generic audiovisual services (V9)", ITU-T Recommendation H.264, February 2014, <http://www.itu.int/rec/T-REC-H.264>.

[HSUP1] ITU-T, "Application profile - Sign language and lip-reading real-time conversation using low bit rate video communication", ITU-T Recommendation H.Sup1, May 1999, <http://www.itu.int/rec/T-REC-H.Sup1>.

[IEC23001-8] ISO/IEC, "Coding independent media description code points", ISO/IEC 23001-8:2013/DCOR1, 2013, <http://standards.iso.org/ittf/PubliclyAvailableStandards/c062088_ISO_IEC_23001-8_2013.zip>.

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

[RFC6184] Wang, Y., Even, R., Kristensen, T., and R. Jesup, "RTP Payload Format for H.264 Video", RFC 6184, DOI 10.17487/RFC6184, May 2011, <http://www.rfc-editor.org/info/rfc6184>.

[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, <http://www.rfc-editor.org/info/rfc6236>.

[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, <http://www.rfc-editor.org/info/rfc6386>.

[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, <http://www.rfc-editor.org/info/rfc7741>.

[SRGB] IEC, "Multimedia systems and equipment - Colour measurement and management - Part 2-1: Colour management - Default RGB colour space - sRGB.", IEC 61966-2-1, October 1999, <https://webstore.iec.ch/publication/6169>.

[TS26.114] 3GPP, "IP Multimedia Subsystem (IMS); Multimedia Telephony; Media handling and interaction", TS 26.114, Version 13.2.0, December 2015, <http://www.3gpp.org/DynaReport/26114.htm>.

8.2. Informative References

[H265] ITU-T, "High efficiency video coding", ITU-T Recommendation H.265, April 2015, <http://www.itu.int/rec/T-REC-H.265>.

[RTCWEB-OVERVIEW] Alvestrand, H., "Overview: Real Time Protocols for Browser-based Applications", Work in Progress, draft-ietf-rtcweb-overview-14, June 2015.

[WebRTC-RTP-USAGE] Perkins, C., Westerlund, M., and J. Ott, "Web Real-Time Communication (WebRTC): Media Transport and Use of RTP", Work in Progress, draft-ietf-rtcweb-rtp-usage-25, June 2015.

[WebRTC-SEC] Rescorla, E., "Security Considerations for WebRTC", Work in Progress, draft-ietf-rtcweb-security-08, February 2015.

[WebRTC-SEC-ARCH]
Rescorla, E., "WebRTC Security Architecture", Work in Progress, draft-ietf-rtcweb-security-arch-11, March 2015.

Acknowledgements

The author would like to thank Gaelle Martin-Cocher, Stephan Wenger, and Bernard Aboba for their detailed feedback and assistance with this document. Thanks to Cullen Jennings for providing text and review and to Russ Housley for a careful final review. This document includes text that originally appeared in "WebRTC Codec and Media Processing Requirements" (March 2012).

Author’s Address

Adam Roach
Mozilla
Dallas
United States

Phone: +1 650 903 0800 x863
Email: adam@nostrum.com