RTCWEB Working Group   B. Burman
Internet-Draft   Ericsson
Intended status: Standards Track   M. Isomaki
Expires: April 30, 2015   Nokia
                                                            B. Aboba
                                                Microsoft Corporation
                                                    G. Martin-Cocher
                                                       BlackBerry Ltd
                                                          G. Mandyam
                                           Qualcomm Innovation Center
                                                          X. Marjou
                                                             Orange
                                                         C. Jennings
                                                        J. Rosenberg
                                                              Cisco
                                                          D. Singer
                                                             Apple
                                                   October 27, 2014

            H.264 as Mandatory to Implement Video Codec for WebRTC
                    draft-burman-rtcweb-h264-proposal-05

Abstract

   This document proposes that, and motivates why, H.264 should be a
   Mandatory To Implement video codec for WebRTC.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

   The selection of a Mandatory To Implement (MTI) video codec for
   WebRTC has been discussed for quite some time in the RTCWEB WG. This
   document proposes that the H.264 video codec should be mandatory to
   implement for WebRTC implementations and gives motivation to this
   proposal.

The core of the proposal is that:

> H.264 Constrained Baseline Profile Level 1.2 MUST be supported as
> Mandatory To Implement video codec.

To enable higher quality for devices capable of it:

> H.264 Constrained High Profile Level 1.3, logically extended to
> support 720p resolution at 30 Hz framerate is RECOMMENDED.

This draft discusses the advantages of H.264 as the authors of this
draft see them; a richness of implementations and hardware support,
well known licensing conditions, good performance, and well defined
handling of varying device capabilities.

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 BCP 14, RFC 2119
[RFC2119].

3. H.264 Overview

The video coding standard Advanced Video Coding (ITU-T H.264 | ISO/
IEC 14496-10 [H264]) has been around for almost ten years by now.
Developed jointly by MPEG and ITU-T in the Joint Video Team, it was
published in its first version in 2003 and amended with support for
higher-fidelity video in 2004. Other significant updates include
support for scalability (2007) and multiview (2009). The codec goes
under the names H.264, AVC and MPEG-4 Part10. In this memo the term
"H.264" will be used.

H.264 was from the start very successful and has become widely
adopted for (video) content as well as (video) communication services
worldwide.

H.264 is mandatory in mobile wireless standards for multimedia
telephony and packet switched streaming. It is also the leading de
facto standard for web video content delivered in HTML5 or other
technologies, and is supported in nearly all major web browsers,
mobile device platforms, and desktop operating systems.

4. Implementations

4.1. Software

There are many software implementations of the H.264 standard,
including royalty-free open source code from Cisco [OpenH264] and
Polycom [Woon], both of which support H.264/SVC (Annex G). Wikipedia
provides an illustration of the long list of other available
implementations [Implementations].

The Cisco OpenH264 implementation is notable for also providing
binaries for common platforms that can be downloaded directly to an
end-user's device or application, so distribution royalties are paid
by Cisco rather than the application developer. The latest Mozilla
Firefox browser release does just that - downloads an OpenH264 binary
- making it possible to use H.264 in WebRTC sessions [FF33].

Microsoft has also produced an H.264 prototype for use in browsers
[CURtcWeb]. Not only are there standalone implementations available,
including open source, but in addition recent Windows and Mac OS X
versions support H.264 encoding and decoding.

4.2. Hardware

Arguably, hardware or DSP acceleration for video encoding/decoding
would be mostly beneficial for devices that has relatively lower
capacity in terms of CPU and power (smaller batteries), and the most
common devices in this category are phones and tablets. There is a
long list of vendors offering hardware or DSP implementations of
H.264. In particular all vendors of platforms for mobile high-range
phones, smartphones, and tablets support H.264/AVC High Profile
encoding and decoding at least 1080p30, but those platforms are
currently in general not used for low- to mid-range devices. These
vendors are Qualcomm, TI, Nvidia, Renesas, Mediatek, Huawei
Hisilicon, Intel, Broadcom, Samsung. Those platforms all support
H.264/AVC codec with dedicated hardware or DSP. The majority of the
implementations also support low-delay real-time applications.

The WebM wiki [WEBM] shows only 8 (out of ~68) SoCs which support VP8
encode and decode. This only represents a fraction of deployed SoCs.
Almost all deployed SoCs, as well as future designs, support H.264
encode and decode, including desktop (Intel x86) chipsets.

The benefits of hardware encoder and decoder implementations
typically have an order of magnitude or more performance advantage
(e.g., 1080p versus 360p becomes achievable) and power savings (e.g.,
tens of milliwatts versus many hundreds of milliwatts or even watts
are consumed just by the encoder and decoder). While VP8 proponents
have argued codec power is not a major concern relative to displays,

this neglects the advances in display technology that put the central
processor back near the top power consumers.

The availability of hardware codecs for real-time communication to
developers through public APIs is increasing. As of iOS8, Apple has
provided API for access to the hardware H.264 encoder and decoder on
the iOS platforms. The APIs can be found in the Video Toolbox
[AppleVideoToolbox]. BlackBerry recently released the API
[BlackBerryAPI] to the hardware H.264 codec via OpenMAX-AL [OpenMAX].
Android has provided the MediaCodec API [MediaCodec] to the hardware
H.264 codec since version 4.1 (API 16), as well as enhancements and a
Compatibility Test Suite in 4.3 (API 18).

4.3. Standards

There are also other standards and specifications that support H.264.
One notable area is wireless display standards, where H.264 support
is pervasive among all the following leading standards:

o AirPlay (Apple) [AirPlay].

o WiDi (Intel) [WiDi].

o Miracast (Wi-Fi Alliance) [Miracast].

o Google Cast (Google) [GoogleCast].

o DLNA (Sony) [DLNA].

GSMA [GSMA] has defined the following services for use in 3GPP
[ThreeGPP] IP Multimedia Subsystem (IMS), which use H.264 Constrained
Baseline Profile as MTI video codec:

IR.94 IMS Profile for Conversational Video Service [IR94]

IR.39 IMS Profile for High Definition Video Conference (HDVC)
   Service [IR39]

5. Deployment

Today, the Internet runs on H.264 for real-time video communications.
Though not yet on the web, video communications is in widespread
usage on the Internet. It is supported in consumer applications both
on the desktop and in mobile apps, provided by many players like
Skype and Tango. It is in widespread usage for business
communications, in many applications like Webex, Citrix Go-To-
Meeting, Tandberg and Polycom telepresence systems, and many more.

All of these are in widespread deployment and widespread usage, and
are based on H.264.

Today, every single GSM/WCDMA mobile device, mobile operator network
and mobile operating system supports the H.264 Constrained Baseline
Profile video codec [GSMA-Codec-WP].

If we want WebRTC to be successful, we must make sure it is something
that can be adopted by the application providers who deploy real-time
communications on the Internet. WebRTC needs to be for the
developers - the people who are building applications. And a
critical target customer base are the ones who are already doing
voice and video communications - the ones with the network effect and
user bases which need to be tapped to make this technology
successful. If WebRTC does not embrace H.264, it will be at the risk
of ignoring the needs of one of its most important set of potential
adopters - the ones most eager to use it - the ones already in the
market for real-time communications.

It may be argued that clients can be upgraded to support any new
codec. Opus is mandatory despite no deployment. However, G.711 is
also mandatory to ensure broad adoption. Likewise, H.264 should be
mandatory to ensure broad video adoption, since it is as widely
adopted in video as G.711 in voice. Also, video is more processing
intensive than voice, and therefore often implemented in hardware
that is not easily upgradeable. Other video systems use desktop
software which can also be difficult to broadly upgrade. Still
others provide SDKs and toolkits to third parties which cannot easily
be upgraded. Others have mobile apps which users cannot be
forcefully made to upgrade.

It may be argued that clients must be upgraded anyway to support ICE,
DTLS-SRTP and other WebRTC requirements. Some will, some won't. For
the latter, application providers will need to build server side
gateways. While that adds cost and complexity, the need to transcode
video would greatly escalate costs, perhaps making them prohibitive.
The CPU cost for transcoding, and the corresponding impact on quality
due to recoding and increased delays, are substantially larger
compared to just transport-level gateway functions. Perhaps enough
to make it impractical at scale. This view is supported by the
discussion on transcoding in a GSMA whitepaper [GSMA-Codec-WP], where
it is concluded that "...to preserve end-user experience, transcoding
must be avoided altogether".

It may be argued that deployed video systems and applications are
insignificant compared to the larger number of web browsers that will
support WebRTC. This misses a key point. Real-time communications
exists amongst a set of users that can talk to each other, typically

because they are customers of the same service. Skype users can talk
to each other. Tango users can talk to each other. There is, to
date, relatively little federation for video between these providers,
a problem which WebRTC is unlikely to remedy, as its causes have
little to do with media stacks, and everything to do with business.
Enabling real-time communications in the browser does not immediately
create a connected user base that is the size of the web. WebRTC is
just a media stack; the namespace is provided by the application
provider, as is the size of the communications network to which that
user can connect. Existing communications providers greatly value
their user bases, and those user bases define the reachable
communications network. When viewed in that lens, the most important
thing for allowing a WebRTC user to reach a massive network, is
enabling WebRTC to be usable by those which have existing networks of
users. Of those, many are asking for H.264.

It may be argued that WebRTC should build for the future, and not be
constrained by the past. This is reminiscent of the arguments made
by those who advocated against IETF doing work on NAT or making NAT
friendly protocols. The hope was the same - that IETF could, through
standards, dictate the future as we wished it - that by designing
protocols which didn't work through NAT, we would force the industry
to move away from NAT and embrace IPv6. That strategy failed. The
Internet is a living, breathing thing, constantly evolving. Those
technologies which are successful are actually those which work for
the Internet as it is today, not the Internet as we wish it could be.
Those then allow the Internet to take a baby step forward, and from
there, another step forward. Successful technologies require
consideration for transition, as it is more important than the
target. Just like NAT was, and still is, a reality on the Internet
today, so too is H.264 a reality of the Internet today. Just like we
could not upgrade the routers and switches to eliminate NAT, so too
are we unable to upgrade many of the Internet endpoints today to
instantly move away from H.264. We should learn from the past and
define a WebRTC which can work with the applications in existence
today, otherwise we significantly hinder the success and growth of
WebRTC.

6. Licensing

6.1. Royalty Free for Innovation, Low-volume Shipments

MPEG-LA released their AVC Patent Portfolio License already in 2004
and in 2010 they announced that H.264 encoded Internet video is free
to end users will never be charged royalties [MPEGLA]. Real-time
generated content, the content most applicable to WebRTC, was free
already from the establishment of the MPEG-LA license
[MPEGLA-License]. License fees for the distribution of products that

decode and encode H.264 video remain though. Those fees
[MPEGLA-Terms] are, and will very likely continue to be for the
lifetime of MPEG-LA pool, $0.20 per codec or less.

To paraphrase, the MPEG LA license does allow up to 100K units per
year, per legal entity/company (type "a" sublicensees in MPEG LA's
definition), to be shipped for zero ($0) royalty cost. This should
be adequate for many WebRTC innovators or start-ups to try out new
implementations on a large set of users before incurring any patent
royalty costs, a benefit to selecting a H.264/AVC profile as the
mandatory codec.

6.2. Higher H.264/AVC Profile Tools Bundled

It should be noted that when one licenses the MPEG LA H.264/AVC pool,
patents for higher profile tools - such as CABAC, 8x8 - are bundled
in with those required for the Constrained Baseline Profile. Thus,
these could optionally be used by WebRTC implementers to achieve even
greater performance or efficiencies than using H.264 Constrained
Baseline Profile alone.

It can also be noted that for MPEG-LA, since one license covers both
an encoder and decoder, there is no additional cost of using an
encoder to an implementation that supports decoding of H.264.

6.3. Licensing Stability

H.264 is a mature codec with a mature and well-known licensing model.

It is a well-established fact that not all H.264 right holders are
MPEG-LA pool members. H.264 is however an ITU/ISO/IEC international
standard, developed under their respective patent policies, and all
contributors must license their patents under Reasonable And Non-
Discriminatory (RAND) terms. In the field of video coding, most
major research groups interested in patents do contribute to the
ITU/ISO/IEC standards process and are therefore bound by those terms.

VP8 is a much younger codec than H.264 and it is fair to say that the
licensing situation is less clear than for H.264. Google has
provided their patent rights on VP8, including patents owned by 11
patent holders [MpegLaVp8], under a open source friendly license with
very restrictive reciprocity conditions.

VP8 in Video Coding for Browsers in MPEG is at the time of writing in
Draft International Standard ballot until January 2015, which is the
next-to-last step in becoming an MPEG standard. As such, it will
have to follow the ISO/IEC/ITU common patent policy
[IsoIecItuPolicy], before becoming International Standard. IPR

statements in MPEG or in the ISO/IEC database [IEC-Declarations],
received so far, contain royalty free (option 1), "Fair, Reasonable
And Non-Discriminatory" (FRAND, option 2), and "Unwilling to grant
license" (option 3). Potential IPR owners that do not participate in
this MPEG work are under no obligation to offer any license at all.
This indicates that the licensing situation for VP8 has still not
settled but tends toward a non-RF situation.

7. Performance

Comparing video quality is difficult. Practically no modern video
encoding method includes any bit-exact encoding where a given (video)
input produces a specified encoded output bitstream. Instead, the
encoded bitstream syntax and semantics are specified such that a
decoder can correctly interpret it and produce a known output. This
is true both for H.264 and VP8. Significant freedom is left to the
encoder implementation to choose how to represent the encoded video,
for example given a specific targeted bitrate. Thus it cannot in
general be expected that any encoded video bitstream represents the
best possible or most efficient representation, given the defined
bitstream syntax elements available to that codec. The actually
achieved quality for a certain bitstream, how close it is to the
optimally possible with available syntax, at any given bitrate rather
depends on the performance of the individual encoder implementation.

Also, not only is the resulting experienced video quality subjective,
but also depends on the source material, on the point of operation
and a number of other considerations. In addition, performance can
be measured vs. bitrate, but also vs. e.g. complexity - and here
another can of worms can be opened because complexity depends on
hardware used (some platforms have video codec accelerations), SW
platform (and how efficient it can use the hardware) and so on. On
top of this comes that different implementations can have different
performance, and can be operated in different ways (e.g. tradeoffs
between complexity and quality can be made). Regardless of how a
performance evaluation is carried out it can always be said that it
is not "fair". This section nevertheless attempts to shed some light
on this subject, and specifically the performance (measured against
bitrate) of H.264 compared to VP8.

A number of studies [H264perf1][H264perf2][H264perf3] have been made
to compare the compression efficiency performance between H.264 and
VP8. These studies show that H.264 is in general performing better
than VP8 but the studies are not specifically targeting video
conferencing. While constituting an independent test material
providing some indications, those tests however do not use exactly
the proposed profiles and levels, which calls for performing a set of
more targeted tests.

Google made a comparison test between VP8 and H.264 [GooglePSNR], providing a set of test scripts [GoogleScripts]. That test includes the use of rate control for both codecs. We believe this to be a comparison problem since rate control is part of the encoder, which as said above is typically not specified in video codec standards but left up to individual implementations. The quantization parameter (qp) level affects the rate/distortion tradeoff in video coding. Comparing using fixed qp-levels is what has typically been used when benchmarking new codecs, for example when benchmarking HEVC [H265] against H.264 in the JCT-VC [JCT-VC] standardization. We are going to select a codec (essentially bit stream format), not a rate control mechanism; once the codec is selected you can choose whatever rate control mechanism you wish that best suits your specific application. Therefore, we propose to compare the codecs with rate control off, using fixed quantization parameter (qp) levels.

Ericsson made a comparison using Google's published test scripts as baseline and changed the parameter settings in order to make it possible to measure using fixed qp. The focus of that test was to evaluate the best compression efficiency that could be achieved with both codecs since it was believed to be harder to make a fair comparison trying to use complexity constraints. We used the same eleven sequences as in the previous Google test, but limited them to the first 10 seconds since they varied from 10 seconds to minutes; this also eased computation time. The used video resolutions are 640x360 @ 30 fps, 640x480 @ 30 fps, 1280x720 @ 30 fps and 1280x720 @ 50 fps.

We used two H.264 encoder implementations:

o X264, which is an open-source codec that can operate in everything
   from real-time to slow

o JM, which is the (Joint Model) reference implementation that was
   used to develop H.264, and is very slow but attempts to be very
   efficient in terms of bits per quality

This is a summary of the results (complete scripts and results available here [H264VP8Tests]):

```
+-------------------------------+-------------------------------+
| Test                          | Resulting bitrate at          |
|                               | equivalent quality            |
+-------------------------------+-------------------------------+
| X264 Constrained Baseline vs VP8 | H.264 wins with 1%         |
| JM Constrained Baseline vs VP8   | H.264 wins with 4%         |
| X264 Constrained High vs VP8     | H.264 wins with 25%        |
| JM Constrained High vs VP8       | H.264 wins with 24%        |
+-------------------------------+-------------------------------+
```

Table 1: Performance Comparison Results

It is interesting to note that the measurements are more stable in
this test; the variance of the percentages for the different
sequences is now around 70, down from around 700 in Google's test.
We believe this is due to the removal of the rate controller, which
acts as noise on the measurements.

It can also be noted that the Google method of calculating the rate
differences does not give exactly the same numbers as the JCT-VC way
of calculating Bjontegaard Delta bitrate (BD-rate) [PSNRdiff]. The
main difference is that the JM score for Constrained High in the
table above (Table 1) is around 29% better than VP8 if the JCT-VC way
of calculating BD-rate is used.

A rough complexity estimate can be obtained from the total running
times for the tests:

o X264: 1 hour 3 minutes

o VP8: 2 hours 0 minutes

o JM: An order of magnitude slower

Again, video quality is difficult to compare. The authors however
believe that the data provided in this section shows that H.264
Constrained Baseline is at least on par with VP8, while H.264
Constrained High seems to have a clear quality advantage. As a final
note, the new H.265/HEVC standard [H265] clearly outperforms all
three, but the authors think it is premature to mandate HEVC for
WebRTC.

8. Profile/level

H.264/AVC [H264] has a large number of encoding tools, grouped in
functionally reasonable toolsets by codec profiles, and a wide range
of possible implementation capability and complexity, specified by
codec levels. It is typically not reasonable for H.264 encoders and

decoders to implement maximum complexity capability for all of the available tools. Thus, any H.264 decoder implementation is typically not able to receive all possible H.264 streams. Which streams can be received is described by what profile and level the decoder conforms to. Any video stream produced by an H.264 encoder must keep within the limits defined by the intended receiving decoder's profile and level to ensure that the video stream can be correctly decoded.

Profiles can be "ranked" in terms of the amount of tools included, such that some profiles with few tools are "lower" than profiles with more tools. However, profiles are typically not strictly supersets or subsets of each other in terms of which tools are used, so a strict ranking cannot be defined. It is also in some cases possible to express compliance to the common subset of tools between two different profiles. This is fairly well described in [RFC6184].

When choosing a Mandatory To Implement codec, it is desirable to use a profile and level that is as widely supported as possible. Therefore, H.264 Constrained Baseline Profile Level 1.2 MUST be supported as Mandatory To Implement video codec. This is possible to support with significant margin in hardware devices (Section 4) and should likely also not cause performance problems for software-only implementations. All Level definitions (Annex A of [H264]) include a maximum framesize in macroblocks (16*16 pixels) as well as a maximum processing requirement in macroblocks per second. That number of macroblocks per second can be almost freely distributed between framesize and framerate. The maximum framesize for Level 1.2 corresponds to 352*288 pixels (CIF). Examples of allowed framesize and framerate combinations for Level 1.2 are CIF (352*288 pixels) at 15 Hz, QVGA (320*240 pixels) at 20 Hz, and QCIF (176*144 pixels) at 60 Hz.

Recognizing that while the above profile and level will likely be possible to implement in any device, it is also likely not sufficient for applications that require higher quality. Therefore, it is RECOMMENDED that devices and implementations that can meet the additional requirements also implement at least H.264 Constrained High Profile Level 1.3, logically extended to support 720p resolution at 30 Hz framerate, but in formal specification text it would have to be expressed as a restriction on a higher level.

Note that the lowest non-extended Level that support 720p30 is Level 3.1, but fully supporting Level 3.1 also requires fairly high bitrate, large buffers, and other encoding parameters included in that Level definition that are likely not reasonable for the targeted communication scenario. This method of extending a lower level in SDP (Section 9) with a smaller set of applicable parameters is fully

in line with [RFC6184], and is already used by some video
conferencing vendors.

When considering the main WebRTC use case, real-time communication,
the lack of need to support interlaced image format in that context,
the limited use of bi-predictive (B) pictures, and the added
implementation and computation complexity that comes with interlace
and B-picture handling suggests that Constrained High Profile should
be preferred over High Profile as optional codec. Note also that
while Constrained High Profile is currently less supported in devices
than High Profile, any High Profile decoder will be capable of
decoding a Constrained High Profile bitstream since it is a subset of
High Profile. To make a High Profile encoder support Constrained
High Profile encoding, it will have to turn off interlace encoding
and turn off the use of bi-prediction.

The below table summarizes the H.264 video encoding features used by
Constrained Baseline Profile (CBP) and Constrained High Profile
(CHP). For more information on the listed features, see
[WikipediaAVC].

Feature	CBP	CHP
Bit depth per sample	8	8
Chroma formats	4:2:0	4:2:0
Flexible Macroblock Ordering (FMO)	No	No
Arbitrary Slice Ordering (ASO)	No	No
Redundant Slices	No	No
Data Partitioning	No	No
SI and SP slices	No	No
Interlaced coding	No	No
B slices	No	No
CABAC entropy coding	No	Yes
Monochrome 4:0:0	No	Yes
8x8 vs. 4x4 transform adaptivity	No	Yes
Quantization scaling matrices	No	Yes
Separate color QP control	No	Yes
Separate color plane coding	No	No
Predictive lossless coding	No	No
Weighted prediction	No	Yes

9. Negotiation

Given that there exist a fairly large set of defined profiles and
levels (Section 8) in the H.264 specification, the probability is
rather low that randomly chosen H.264 encoder and decoder

implementations have exactly matching capabilities. In any
communication scenario, there is therefore a need for a decoder to be
able to convey its maximum supported profile and level that the
encoder must not exceed.

In addition and depending on the wanted use case and the conditions
that apply at a certain communication instance, there may also be a
need to describe the currently wanted profile and level at the start
of the communication session, which may be lower than the maximum
supported by the implementation. In this scenario it may also be of
interest to communicate from the encoder to the decoder both which
profile and level that will actually be used and what is the maximum
supported profile and level. The reason to communicate not only the
starting point but also the maximum assumes that communication
conditions may change during the conditions, maybe multiple times,
possibly making another profile and level be a more appropriate
choice.

Communication of maximum supported profile and level is the only
mandatory SDP [RFC4566] parameter in the H.264 payload format
[RFC6184], which also includes a large set of optional parameters,
describing available use (decoder) and intended use (encoder) of
those parameters for a specific offered [RFC3264] stream.

If the above mentioned (Section 8) capability for 720p30 is supported
as an extension to Constrained High Profile Level 1.3 (or higher),
the logical level extension SHOULD be signaled in SDP using the
following parameters as defined in section 8.1 of [RFC6184]:

o profile-level-id=640c0d (or corresponding to a higher Level of
   Constrained High profile)

o max-fs=3600 (or greater)

o max-mbps=108000 (or greater)

o max-br=768 (or greater, whatever the device implementation can
   support)

10. Summary

H.264 is widely adopted and used for a large set of video services.
This in turn is because H.264 offers great performance, reasonable
licensing terms (and manageable risks). As a consequence of its
adoption for many services, a multitude implementations in software
and hardware are available. Another result of the widespread
adoption is that all associated technologies, such as payload
formats, negotiation mechanisms and so on are well defined and

standardized. In addition, using H.264 enables interoperability with
many other services without video transcoding.

We therefore propose to the WG that H.264 shall be mandatory to
implement for all WebRTC endpoints that support video, according to
the details described in Section 8 and Section 9.

11. IANA Considerations

This document makes no request of IANA.

Note to RFC Editor: this section may be removed on publication as an
RFC.

12. Security Considerations

No specific considerations apply to the information in this document.

13. Acknowledgements

All that provided valuable descriptions, comments and insights about
the H.264 codec on the IETF mailing lists.

14. References

14.1. Normative References

[H264]   ITU-T Recommendation H.264, "Advanced video coding for
            generic audiovisual services", April 2013,
            <http://www.itu.int/rec/T-REC-H.264-201304-I>.

[RFC2119]   Bradner, S., "Key words for use in RFCs to Indicate
            Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3264]   Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model
            with Session Description Protocol (SDP)", RFC 3264, June
            2002.

[RFC4566]   Handley, M., Jacobson, V., and C. Perkins, "SDP: Session
            Description Protocol", RFC 4566, July 2006.

[RFC6184]   Wang, Y., Even, R., Kristensen, T., and R. Jesup, "RTP
            Payload Format for H.264 Video", RFC 6184, May 2011.

[14.2](#). **Informative References**

   [AirPlay] Apple Inc, "AirPlay Overview: About AirPlay", September
              2012, <https://developer.apple.com/library/ios/documentati
              on/AudioVideo/Conceptual/AirPlayGuide/Introduction/
              Introduction.html>.

   [AppleVideoToolbox]
              Apple Inc., "AV Foundation Programming Guide", March 2014,
              <https://developer.apple.com/library/ios/documentation/
              AudioVideo/Conceptual/AVFoundationPG>.

   [BlackBerryAPI]
              BlackBerry Limited, "Supported codecs - BlackBerry
              Native", September 2014,
              <http://developer.blackberry.com/native/documentation/
              core/openmax_supported_codecs.html>.

   [CURtcWeb]
              Microsoft Open Technologies, Inc., "CU-RTC-Web-Video",
              July 2013,
              <http://html5labs.interoperabilitybridges.com/prototypes/
              cu-rtc-web-video/cu-rtc-web-video/info>.

   [DLNA]   DLNA(R), "Technical Overview", 2013, <http://www.dlna.org/
              dlna-for-industry/digital-living/how-it-works/
              technical-overview>.

   [FF33]   "Cisco's OpenH264 Now Part of Firefox", October 2014,
              <http://blogs.cisco.com/collaboration/
              ciscos-openh264-now-part-of-firefox/>.

   [GSMA]   "GSM Association", 2014, <http://www.gsma.com/>.

   [GSMA-Codec-WP]
              GSM Association, "WebRTC Codecs DRAFT v1.3", September
              2014,
              <http://www.gsma.com/newsroom/webrtc-codecs-draft-v1-3/>.

   [GoogleCast]
              Google, "Supported Media Types - Google Cast", October
              2013, <https://developers.google.com/cast/
              supported_media_types>.

   [GooglePSNR]
              The WebM Project, "VP8 Results", April 2013,
              <http://downloads.webmproject.org/ietf_tests/
              vp8_vs_h264_quality.html>.

[GoogleScripts]
          The WebM Project, "VP8 vs H.264 Test Scripts", April 2013,
          <http://downloads.webmproject.org/ietf_tests/
          vp8_vs_h264.tar.xz>.

[H264VP8Tests]
          Ericsson, "More H.264 vs VP8 tests", June 2013,
          <http://www.ietf.org/mail-archive/web/rtcweb/current/
          zipDGJUJ9JZ8n.zip>.

[H264perf1]
          Vatolin, D., "MPEG-4 AVC/H.264 Video Codecs Comparison
          2010 - Appendixes", , May 2010,
          <http://compression.graphicon.ru/video/codec_comparison/
          h264_2010/appendixes.html#Appendix_8>.

[H264perf2]
          Shah, K., "Implementation, performance analysis and
          comparison of VP8 and H.264.", University of Texas at
          Arlington Department of Electrical Engineering, 2011,
          <http://www-
          ee.uta.edu/Dip/Courses/EE5359/2011SpringFinalReportPPT/
          Shah_EE5359Spring2011FinalPPT.pdf>.

[H264perf3]
          De Simone, F., Goldmann, L., Lee, J., and T. Ebrahimi,
          "Performance analysis of VP8 image and video compression
          based on subjective evaluations", Ecole Polytechnique
          F'd'rale de Lausanne (EPFL) , Aug 2011,
          <http://infoscience.epfl.ch/record/168259/files/
          article.pdf>.

[H265]   ITU-T Recommendation H.265, "High Efficiency Video
          Coding", April 2013,
          <http://www.itu.int/rec/T-REC-H.265-201304-I>.

[IEC-Declarations]
          International Electrotechnical Commission, "List of IEC
          patent declarations received by IEC", October 2014,
          <http://patents.iec.ch/>.

[IR39]   GSM Association, "IMS Profile for High Definition Video
          Conference (HDVC)", May 2013,
          <http://www.gsma.com/newsroom/ir-39-v2-1-ims-profile-for-
          high-definition-video-conference-hdvc-service/>.

[IR94]   GSM Association, "IMS Profile for Conversational Video
            Service", May 2013, <http://www.gsma.com/newsroom/
            official-document-ir-94-ims-profile-for-conversational-
            video-service-2/>.

[Implementations]
            Wikipedia, "H.264/MPEG-4 AVC products and
            implementations", September 2014,
            <http://en.wikipedia.org/wiki/H.264/
            MPEG-4_AVC_products_and_implementations>.

[IsoIecItuPolicy]
            ISO, "ISO/IEC/ITU common patent policy", April 2007,
            <http://isotc.iso.org/livelink/livelink/
            fetch/2000/2122/3770791/Common_Policy.htm>.

[JCT-VC]   ITU-T, "JCT-VC - Joint Collaborative Team on Video
            Coding", <http://www.itu.int/en/ITU-T/studygroups/2013-
            2016/16/Pages/video/jctvc.aspx>.

[MPEGLA]   MPEG LA, "MPEG LAs AVC License Will Not Charge Royalties
            for Internet Video that is Free to End Users through Life
            of License", MPEGLA News Release, August 2010,
            <www.mpegla.com/Lists/MPEG%20LA%20News%20List/
            Attachments/231/n-10-08-26.pdf>.

[MPEGLA-License]
            MPEG LA, "AVC Patent Portfolio License Briefing", May
            2009, <http://www.mpegla.com/main/programs/avc/Documents/
            avcweb.pdf>.

[MPEGLA-Terms]
            MPEG LA, "SUMMARY OF AVC/H.264 LICENSE TERMS",
            <http://www.mpegla.com/main/programs/avc/Documents/
            AVC_TermsSummary.pdf>.

[MediaCodec]
            Android, "MediaCodec | Android Developers", October 2014,
            <http://developer.android.com/reference/android/media/
            MediaCodec.html>.

[Miracast]
            Wi-Fi Alliance(R), "What formats does Miracast support?",
            2013, <http://www.wi-fi.org/knowledge-center/faq/
            what-formats-does-miracast-support>.

[MpegLaVp8]
          O'Reilly, T., "Google and MPEG LA Announce Agreement
          Covering VP8 Video Format", March 2013,
          <http://www.mpegla.com/Lists/MPEG%20LA%20News%20List/
          Attachments/88/n-13-03-07.pdf>.

[OpenH264]
          "OpenH264", 2014, <http://www.openh264.org/>.

[OpenMAX] Khronos, "OpenMAX - The Standard for Media Library
          Portability", 2014, <https://www.khronos.org/openmax/>.

[PSNRdiff]
          Bjontegaard, G., "Calculation of Average PSNR Differences
          between RD-Curves", ITU-T SG16 Q.6 Document VCEG-M33,
          April 2001.

[ThreeGPP]
          "3rd Generation Partnership Project",
          <http://www.3gpp.org/>.

[WEBM]   The WebM Project, "SoCs Supporting VP8/VP9", October 2014,
          <http://wiki.webmproject.org/hardware/socs>.

[WiDi]   Intel Corporation, "Intel(R) Wireless Display and Intel(R)
          Pro Wireless Display", October 2013,
          <http://www.intel.com/content/www/us/en/architecture-and-
          technology/intel-wireless-display.html>.

[WikipediaAVC]
          Wikipedia, "H.264/MPEG-4 AVC", October 2013,
          <http://en.wikipedia.org/wiki/H.264/MPEG-4_AVC>.

[Woon]   Polycom, "Polycom Delivers Open Standards-Based Scalable
          Video Coding (SVC) Technology, Royalty-Free to Industry",
          October 2012,
          <http://www.polycom.com/content/www/en/company/news/
          press-releases/2012/20121004.html>.

Authors' Addresses

   Bo Burman
   Ericsson
   Farogatan 6
   Stockholm 16480
   Sweden

   Email: bo.burman@ericsson.com

Markus Isomaki
Nokia
Keilalahdentie 2-4
Espoo FI-02150
Finland

Email: markus.isomaki@nokia.com

Bernard Aboba
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052
US

Email: bernard_aboba@hotmail.com

Gaelle Martin-Cocher
BlackBerry Ltd
1875 Buckhorn Gate
Mississauga, ON L4W 5P1
Canada

Email: gmartincocher@blackberry.com

Giri Mandyam
Qualcomm Innovation Center

Email: mandyam@quicinc.com

Xavier Marjou
Orange
2, avenue Pierre Marzin
Lannion 22307
France

Email: xavier.marjou@orange.com

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

Email: fluffy@cisco.com

Jonathan Rosenberg
Cisco
170 West Tasman Drive
San Jose, CA 95134
USA

Email: jdrosen@cisco.com

David Singer
Apple

Email: singer@apple.com