

codec  
Internet-Draft  
Intended status: Informational  
Expires: January 28, 2012

JM. Valin  
Mozilla  
K. Vos  
Skype Technologies S.A.  
July 27, 2011

Requirements for an Internet Audio Codec  
draft-ietf-codec-requirements-05

Abstract

This document provides specific requirements for an Internet audio codec. These requirements address quality, sampling rate, bit-rate, and packet loss robustness, as well as other desirable properties.

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 January 28, 2012.

Copyright Notice

Copyright (c) 2011 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 . . . . .	3
2. Definitions . . . . .	4
3. Applications . . . . .	5
3.1. Point to point calls . . . . .	5
3.2. Conferencing . . . . .	5
3.3. Telepresence . . . . .	6
3.4. Teleoperation and Remote Software Services . . . . .	6
3.5. In-game voice chat . . . . .	7
3.6. Live distributed music performances / Internet music lessons . . . . .	7
3.7. Delay Tolerant Networking or Push-to-Talk Services . . . . .	8
3.8. Other applications . . . . .	8
4. Constraints Imposed by the Internet on the Codec . . . . .	9
5. Detailed Basic Requirements . . . . .	11
5.1. Operating space . . . . .	11
5.2. Quality and bit-rate . . . . .	11
5.3. Packet loss robustness . . . . .	12
5.4. Computational resources . . . . .	13
6. Additional considerations . . . . .	15
6.1. Low-complexity audio mixing . . . . .	15
6.2. Encoder side potential for improvement . . . . .	15
6.3. Layered bit-stream . . . . .	15
6.4. Partial redundancy . . . . .	16
6.5. Stereo support . . . . .	16
6.6. Bit error robustness . . . . .	16
6.7. Time stretching and shortening . . . . .	16
6.8. Input robustness . . . . .	17
6.9. Support of Audio forensics . . . . .	17
6.10. Legacy compatibility . . . . .	17
7. Security Considerations . . . . .	18
8. IANA Considerations . . . . .	19
9. Acknowledgments . . . . .	20
10. Informative References . . . . .	21
Authors' Addresses . . . . .	23

## 1. Introduction

This document provides requirements for an audio codec designed specifically for use over the Internet. The requirements attempt to address the needs of the most common Internet interactive audio transmission applications and to ensure good quality when operating in conditions that are typical for the Internet. These requirements address the quality, sampling rate, delay, bit-rate, and packet loss robustness. Other desirable codec properties are considered as well.

## 2. Definitions

Throughout this document, we will use the following conventions when referring to the sampling rate of a signal:

Narrowband: 8 kHz

Wideband: 16 kHz

Super-wideband: 24/32 kHz

Full-band: 44.1/48 kHz

Codec bit-rates in bits per second (b/s) will be considered without counting any overhead (IP/UDP/RTP headers, padding, ...). The codec delay is the total algorithmic delay when one adds the codec frame size to the "look-ahead". It is thus the minimum theoretically achievable end-to-end delay of a transmission system that uses the codec.

### 3. Applications

The following applications should be considered for Internet audio codecs, along with their requirements:

- o Point to point calls
- o Conferencing
- o Telepresence
- o Teleoperation
- o In-game voice chat
- o Live distributed music performances / Internet music lessons
- o Delay Tolerant Networking or Push-to-Talk Services
- o Other applications

#### 3.1. Point to point calls

Point to point calls are voice over IP (VoIP) calls from two "standard" (fixed or mobile) phones, and implemented in hardware or software. For these applications, a wideband codec is required, along with narrowband support for compatibility with legacy telephony equipment (PSTN). It is expected for the range of useful bit-rates to be 12 - 32 kb/s for wideband speech and 8 - 16 kb/s for narrowband speech. The codec delay must be less than 40 ms, but no more than 25 ms is desirable. Support for encoding music is not required, but it is desirable for the codec not to make background (on-hold) music excessively unpleasant to hear. Also, the codec should be robust to noise (produce intelligible speech and no annoying artifacts) even at lower bit-rates.

#### 3.2. Conferencing

Conferencing applications (which support multi-party calls) have additional requirements on top of the requirements for point-to-point calls. Conferencing systems often have higher-fidelity audio equipment and have greater network bandwidth available -- especially when video transmission is involved. For that reason, support for super-wideband audio becomes important, with useful bit-rates in the 32 - 64 kb/s range. The ability to vary the bit-rate (VBR) according to the "difficulty" of the audio signal is a desirable feature for the codec. This not only saves bandwidth "on average", but it can also help conference servers make more efficient use of the available

bandwidth by using more bandwidth for important audio streams and less bandwidth for less important ones (e.g. background noise).

Conferencing end-points often operate in hands-free conditions, which creates acoustic echo problems. For this reason lower delay is important, as it reduces the quality degradation due to any residual echo after acoustic echo cancellation (AEC). For this reason, the codec delay must be less than 30 ms for this application. An optional low-delay mode with less than 10 ms delay is desirable, but not required.

Most conferencing systems operate with a bridge that mixes some (or all) of the audio streams and sends them back to all the participants. In that case, it is important that the codec not produce annoying artefacts when two voices are present at the same time. Also, this mixing operation should be as easy as possible to perform. To make it easier to determine which streams have to be mixed (and which are noise/silence), it must be possible to measure (or estimate) the voice activity in a packet without having to fully decode the packet (saving most of the complexity when the packet need not be decoded). Also, the ability to save on the computational complexity when mixing is also desirable, but not required. For example, a transform codec may make it possible to mix the streams in the transform domain, without having to go back to time-domain. Low-complexity up-sampling and down-sampling within the codec is also a desirable feature when mixing streams with different sampling rates.

### 3.3. Telepresence

Most telepresence applications can be considered to be essentially very high-quality video-conferencing environments, so all of the conferencing requirements also apply to telepresence. In addition, telepresence applications require super-wideband and full-band audio capability with useful bit-rates in the 32 - 80 kb/s range. While voice is still the most important signal to be encoded, it must be possible to obtain good quality (even if not transparent) music.

Most telepresence applications require more than one audio channel, so support for stereo and multi-channel is important. While this can always be accomplished by encoding multiple single-channel streams, it is preferable to take advantage of the redundancy that exists between channels.

### 3.4. Teleoperation and Remote Software Services

Teleoperation applications are similar to telepresence, with the exception that they involve remote physical interactions. For example, the user may be controlling a robot while receiving real-

time audio feedback from that robot. For these applications, the delay has to be less than 10 ms. The other requirements of telepresence (quality, bit-rate, multi-channel) apply to teleoperation as well. The only exception is that mixing is not an important issue for teleoperation.

The requirements for remote software services are similar to those of teleoperation. These applications include remote desktop applications, remote virtualization, and interactive media application being rendered remotely (e.g. video games rendered on central servers). For all these applications, full-band audio with an algorithmic delay below 10 ms are important.

### 3.5. In-game voice chat

An increasing number of computer/console games make use of VoIP to allow players to communicate in real-time. The requirements for gaming are similar to those of conferencing, with the main difference being that narrowband compatibility is not necessary. While for most applications a codec delay up to 30 ms is acceptable, a low-delay (< 10 ms) option is highly desirable, especially for games with rapid interactions. The ability to use VBR (with a maximum allowed bitrate) is also highly desirable because it can significantly reduce the bandwidth requirement for a game server.

### 3.6. Live distributed music performances / Internet music lessons

Live music over the Internet requires extremely low end-to-end delay and is one of the most demanding application for interactive audio transmission. It has been observed that for most scenarios, total end-to-end delays up to 25 ms could be tolerated by musicians, with the absolute limit (where none of the scenarios are possible) being around 50 ms [carot09]. In order to achieve this low delay on the Internet -- either in the same city or a nearby city -- the network propagation time must be taken into account. When also subtracting the delay of the audio buffer, jitter buffer, and acoustic path, that leaves around 2 ms to 10 ms for the total delay of the codec. Considering the speed of light in fiber, every 1 ms reduction in the codec delay increases the range over which synchronization is possible by approximately 200 km.

Acoustic echo is expected to be an even more important issue for network music than it is in conferencing, especially considering that the music quality requirements essentially forbid the use of a "nonlinear processor" (NLP) with the AEC. This is another reason why very low delay is essential.

Considering that the application is music, the full audio bandwidth

(44.1 or 48 kHz sampling rate) must be transmitted with a bit-rate that is sufficient to provide near-transparent to transparent quality. With the current audio coding technology, this corresponds to approximately 64 kb/s to 128 kb/s per channel. As for telepresence, support for two or more channels is often desired, so it would be useful for a codec to be able to take advantage of the redundancy that is often present between audio channels.

### 3.7. Delay Tolerant Networking or Push-to-Talk Services

Internet transmissions are subjected to interruptions of connectivity that severely disturb a phone call. This may happen in cases of route changes, handovers, slow fading, or device failures. To overcome this distortion, the phone call can be halted and resumed after the connectivity has been reestablished again.

Also, if transmission capacity is lower than the minimal coding rate, switching to a push-to-talk mode still allows for effective communication. In that situation, voice is transmitted at slower-than-real-time bitrate and conversations are interrupted until the speech has been transmitted.

These modes require interrupting the audio playout and continuing after a pause of arbitrary duration.

### 3.8. Other applications

The above list is by no means a complete list of all applications involving interactive audio transmission on the Internet. However, it is believed that meeting the needs of all these different applications should be sufficient to ensure that most applications not listed will also be met.



#### 4. Constraints Imposed by the Internet on the Codec

Packet losses are inevitable on the Internet and dealing with those is one of the most fundamental requirements for an Internet audio codec. While any audio codec can be combined with a good packet loss concealment (PLC) algorithm, the important aspect is what happens on the first packets received after the loss. More specifically, this means that:

- o it should be possible to interpret the contents of any received packet, irrespective of previous losses as specified in BCP 36 [PAYLOADS]; and
- o the decoder should re-synchronize as quickly as possible (i.e. the output should quickly converge to the output that would have been obtained if no-loss had occurred).

The constraint of being able to decode any packet implies the following considerations for an audio codec:

- o The size of a compressed frame must be kept smaller than the MTU to avoid fragmentation;
- o The interpretation of any parameter encoded in the bit-stream must not depend on information contained in other packets. For example, it is not acceptable for a codec to allow signaling a mode change in one packet and assume that subsequent frames will be decoded according to that mode.

Although the interpretation of parameters cannot depend on other packets, it is still reasonable to use some amount of prediction across frames, provided that the predictors can resynchronize quickly in case of a lost packet. In this case, it is important to use the best compromise between the gain in coding efficiency and the loss in packet loss robustness due to the use of inter-frame prediction. It is a desirable property for the codec to allow some real-time control of that trade-off so that it can take advantage of more prediction when the loss rate is small, while being more robust to losses when the loss rate is high.

To improve the robustness to packet loss, it would be desirable for the codec to allow an adaptive (data- and network-dependent) amount of side information to help improve audio quality when losses occur. For example, this side information may include the retransmission of certain parameters encoded in the previous frame(s).

To ensure freedom of implementation, decoder-side only error concealment does not need to be specified, although a functional PLC

algorithm is desirable as part of the codec reference implementation. Obviously, any information signaled in the bitstream intended to aid PLC needs to be specified.

Another important property of the Internet is that it is mostly a best-effort network, with no guaranteed bandwidth. This means that the codec has to be able to vary its output bit-rate dynamically (in real-time), without requiring an out-of-band signaling mechanism, and without causing audible artifacts at the bit-rate change boundaries. Additional desirable features are:

- o Having the possibility to use smooth bit-rate changes with one byte/frame resolution;
- o Making it possible for a codec to adapt its bit-rate based on the source signal being encoded (source-controlled VBR) to maximize the quality for a certain average bit-rate.

Because the Internet transmits data in bytes, a codec should produce compressed data in integer numbers of bytes. In general, the codec design should take into consideration explicit congestion notification (ECN) and may include features that would improve the quality of an ECN implementation.

The IETF has defined a set of application-layer protocols to be used for transmitting real-time transport of multimedia data, including voice. It is thus important for the resulting codec to be easy to use with these protocols. For example, it must be possible to create an [RTP] payload format that conforms to BCP 36 [PAYLOADS]. If any codec parameters need to be negotiated between end-points, the negotiation should be as easy as possible to carry over SIP [RFC3261]/SDP [RFC4566] or alternatively over XMPP [RFC6120]/Jingle [XEP-0167].

## 5. Detailed Basic Requirements

This section summarizes all the constraints imposed by the target applications and by the Internet into a set of actual requirements for codec development.

### 5.1. Operating space

The operating space for the target applications can be divided in terms of delay: most applications require a "medium delay" (20-30 ms), while a few require a "very low delay" (< 10 ms). It makes sense to divide the space based on delay because lowering the delay has a cost in terms of quality vs bit-rate.

For medium delay, the resulting codec must be able to efficiently operate within the following range of bit-rates (per channel):

- o Narrowband: 8 kb/s to 16 kb/s
- o Wideband: 12 to 32 kb/s
- o Super-wideband: 24 to 64 kb/s
- o Full-band: 32 to 80 kb/s

Obviously, a lower-delay codec that can operate in the above range is also acceptable.

For very low delay, the resulting codec will need to operate within the following range of bit-rates (per channel):

- o Super-wideband: 32 to 80 kb/s
- o Full-band: 48 to 128 kb/s
- o (Narrowband and wideband not required)

### 5.2. Quality and bit-rate

The quality of a codec is directly linked to the bit-rate, so these two must be considered jointly. When comparing the bit-rate of codecs, the overhead of IP/UDP/RTP headers should not be considered, but any additional bits required in the RTP payload format after the header (e.g. required signalling) should be considered. In terms of quality vs bit-rate, the codec to be developed must be better than the following codecs, that are generally considered as royalty-free:

- o For narrowband: Speex (NB) [Speex], and iLBC(\*) [RFC3951]
- o For wideband: Speex (WB) [Speex], G.722.1(\*) [ITU.G722.1]
- o For super-wideband/fullband: G.722.1C(\*) [ITU.G722.1]

The codecs marked with (\*) have additional licensing restrictions, but the codec to be developed should still not perform significantly worse. In addition to the quality targets listed above, a desirable objective is for the codec quality to be no worse than AMB-NB and AMR-WB, for narrowband and wideband, respectively. Quality should be measured for multiple languages, including tonal languages. The case of multiple simultaneous voices (as sometimes happens in conferencing) should be evaluated as well.

The comparison with the above codecs assumes that the codecs being compared have similar delay characteristics. The bit-rate required for a certain level of quality may be higher than the referenced codecs in cases where a much lower delay is required. In that case, the increase in bit-rate must be less than the ratio between the delays.

It is desirable for the codecs to support source-controlled variable bit-rate (VBR) to take advantage from the fact that different inputs require a different bitrate to achieve the same quality. However, it should still be possible to use the codec at truly constant bit-rate to ensure that no information leak is possible when using an encrypted channel.

### 5.3. Packet loss robustness

Robustness to packet loss is a very important aspect of any codec to be used on the Internet. Codecs must maintain acceptable quality at loss rates up to 5% and maintain good intelligibility up to 15% loss rate. At any sampling rate, bit-rate, and packet loss rate, the quality must be no less than the quality obtained with the Speex codec or the GSM-FR codec in the same conditions. The actual packet loss "patterns" to be used in testing must be obtained from real packet loss traces collected on the Internet, rather than from loss models. These traces should be representative of the typical environments in which the applications of Section 3 operate. For example, traces related to VoIP calls should consider the loss patterns observed for typical home broadband and corporate connections.

#### 5.4. Computational resources

The resulting codec should be implementable on a wide range of devices, so there should be a fixed-point implementation or at least assurance that a reasonable fixed-point is possible. The computational resources figures listed below are meant to be upper bounds. Even below these bounds, resources should still be minimized. Any proposed increase in computational resources consumption (e.g. to increase quality) should be carefully evaluated even if the resulting resource consumption is below the upper bound. Having variable complexity would be useful (but not required) in achieving that goal as it would allow trading quality/bit-rate for lower complexity.

The computational requirements for real-time encoding and decoding of a mono signal on one core of a recent x86 CPU (as measured with the unix "time" utility or equivalent) are as follows:

- o Narrowband: 40 MHz (2% of a 2 GHz CPU core)
- o Wideband: 80 MHz (4% of a 2 GHz CPU core)
- o Superwideband/fullband: 200 MHz (10% of a 2 GHz CPU core)

It is a desirable objective that the MHz values listed above also be achievable on fixed-point digital signal processors that are capable of single-cycle multiply-accumulate operations (16x16 multiplication accumulated into 32 bits).

For applications that require mixing (e.g. conferencing), it should be possible to estimate the energy and/or the voice activity status of the decoded signal with less than 10% of the complexity figures listed above.

It is the intent to maximize the range of devices on which a codec can be implemented. For this reasons, the reference implementation must not depend on special hardware features or instructions to be present in order to meet the complexity requirement. However, it may be desirable to take advantage of such hardware when available, (e.g., hardware accelerators for operations like fast Fourier transforms and convolutions). A codec should also minimize the use of saturating arithmetic so as to be implementable on architectures that do not provide hardware saturation (e.g. ARMv4).

The combined codec size and data ROM should be small enough not to cause significant implementation problems on typical embedded devices. The codec context/state size required should be no more than  $2 \cdot R \cdot C$  bytes in floating-point, where  $R$  is the sampling rate and

C is the number of channels. For fixed-point, that size should be less than  $R \cdot C$ . The scratch space required should also be less than  $2 \cdot R \cdot C$  bytes for floating point or less than  $R \cdot C$  bytes for fixed-point.

## 6. Additional considerations

There are additional features or characteristics that may be desirable under some circumstances, but should not be part of the strict requirements. The benefit of meeting these considerations should be weighted against the associated cost.

### 6.1. Low-complexity audio mixing

In many applications that require a mixing server (e.g. conferencing, games), it is important to minimize the computational cost of the mixing. As much as possible, it should be possible to perform the mixing with fewer computations than it would take to decode all the streams, mix them, and re-encode the result. Properties that reduce the complexity of the mixing process include:

- o the ability to derive sufficient parameters, such as loudness and/or spectral envelope, for estimating voice activity of a compressed frame without fully decoding that frame;
- o the ability to mix the streams in an intermediate representation (e.g. transform domain), rather than having to fully decode the signals before the mixing;
- o the use of bit-stream layers (Section 6.3) by aggregating a small number of active streams at lower quality.

For conferencing applications, the total complexity of the decoding, VAD and mixing should be considered when evaluating proposals.

### 6.2. Encoder side potential for improvement

In many codecs, it is possible to improve the quality by improving the encoder without breaking compatibility (i.e. without changing the decoder). Potential for improvement varies from one codec to another. It is generally low for PCM or ADPCM codecs and higher for perceptual transform codecs. All things being equal, being able to improve a codec after the bit-stream is a desirable property. However, this should not be done at the expense of quality in the reference encoder. Other potential improvements include signal-adaptive frame size selection and improved discontinuous transmission (DTX) algorithms that take advantage of predicting the decoder sides packet loss concealment (PLC) algorithms.

### 6.3. Layered bit-stream

A layered codec makes it possible to transmit only a certain subset of the bits and still obtain a valid bit-stream with a quality that

is equivalent to the quality that would be obtained from encoding at the corresponding rate. While this is not a necessary feature for most applications, it can be desirable for cases where a "mixing server" needs to handle a large number of streams with limited computational resources.

#### 6.4. Partial redundancy

One possible way of increasing robustness to packet loss is to include partial redundancy within packets. This can be achieved either by including the base layer of the previous frame (for a layered codec) or by transmitting other parameters from the previous frame(s) to assist the PLC algorithm in case of loss. The ability to include partial redundancy for high-loss scenarios is desirable, provided that the feature can be dynamically turned on or off (so that no bandwidth is wasted in case of loss-free transmission).

#### 6.5. Stereo support

It is highly desirable for the codec to have stereo support. At a minimum, the codec should be able to encode two channels independently without causing significant stereo image artefacts. It is also desirable for the codec to take advantage of the inter-channel redundancy in stereo audio to reduce the bitrate (for an equivalent quality) of stereo audio compared to coding channels independently.

#### 6.6. Bit error robustness

The vast majority of Internet-based applications do not need to be robust to bit errors because packets either arrive unaltered, or do not arrive at all. Considering that, the emphasis should be on packet loss robustness and packet loss concealment. That being said, it is often the case that extra robustness to bit errors can be achieved at no cost at all (i.e. no increase in size, complexity or bit-rate, no decrease in quality or packet loss robustness, ...). In those cases then it is useful to make a change that increases the robustness to bit errors. This can be useful for applications that use UDP Lite transmission (e.g. over a wireless LAN). Robustness to packet loss should *\*never\** be sacrificed to achieve higher bit error robustness.

#### 6.7. Time stretching and shortening

When adaptive jitter buffers are used it is often necessary to stretch or shorten the audio signal to allow changes in buffering. While this operation can be performed directly on the decoder's output, it is often more computationally efficient to stretch or



shorten the signal directly within the decoder. It is desirable for the reference implementation to provide a time stretching/shortening implementation, although it should not be normative.

#### 6.8. Input robustness

The systems providing input to the encoder and receiving output from the decoder may be far from ideal in actual use. Input and output audio streams may be corrupted by compounding non-linear artifacts from analog hardware and digital processing. The codecs to be developed should be tested to ensure that they degrade gracefully under adverse audio input conditions. Types of digital corruption that may be tested include tandeming, transcoding, low-quality resampling, and digital clipping. Types of analog corruption that may be tested include microphones with substantial background noise, analog clipping, and loudspeaker distortion. No specific end-to-end quality requirements are mandated for use with the proposed codec. It is advisable, however, that several typical in-situ environments/processing chains be specified for the purpose of benchmarking end-to-end quality with the proposed codec.

#### 6.9. Support of Audio forensics

Emergency calls can be analyzed using audio forensics if the context and situation of the caller has to be identified. Thus, it is important to transmit not only the voice of the callees well but also to transmit background noise at high quality. In these situations, sounds or noises of low volume should also not be compressed or dropped. For this reason, the encoder must allow DTX to be disabled when required (e.g. for emergency calls).

#### 6.10. Legacy compatibility

In order to create the best possible codec for the Internet, there is no requirement for compatibility with legacy Internet codecs.

## 7. Security Considerations

Although this document itself does not have security considerations, this section describes the security requirements for the codec.

Just like for any protocol to be used over the Internet, security is a very important aspect to consider. This goes beyond the obvious considerations of preventing buffer overflows and similar attacks that can lead to denial-of-service or remote code execution. One very important security aspect is to make sure that the decoders have a bounded and reasonable worst-case complexity. This prevents an attacker from causing a DoS by sending packets that are specially crafted to take a very long (or infinite) time to decode.

A more subtle aspect is the information leak that can occur when the codec is used over an encrypted channel (e.g. [SRTP]). For example, it was suggested [wright08] [whitell] that use of source-controlled VBR may reveal some information about a conversation through the size of the compressed packets. For that reason, it should be possible to use the codec at truly constant bit-rate if needed.

## 8. IANA Considerations

This document has no actions for IANA.

## 9. Acknowledgments

The original authors of this document are: Jean-Marc Valin, Slava Borilin, Koen Vos, Christopher Montgomery and Raymond (Juin-Hwey) Chen. We would like to thank all the other people who contributed directly or indirectly to this document, including Jason Fischl, Gregory Maxwell, Alan Duric, Jonathan Christensen, Julian Spittka, Michael Knappe, Christian Hoene, and Henry Sinnreich. We also like to thank Cullen Jennings and Gregory Lebovitz for their advice.

## 10. Informative References

- [RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.
- [RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006.
- [RFC6120] Saint-Andre, P., "Extensible Messaging and Presence Protocol (XMPP): Core", RFC 6120, March 2011.
- [XEP-0167] Ludwig, S., Saint-Andre, P., Egan, S., McQueen, R., and D. Cionoiu, "Jingle RTP Sessions", XSF XEP 0167, December 2009.
- [RFC3951] Andersen, S., Duric, A., Astrom, H., Hagen, R., Kleijn, W., and J. Linden, "Internet Low Bit Rate Codec (iLBC)", RFC 3951, December 2004.
- [ITU.G722.1] International Telecommunications Union, "Low-complexity coding at 24 and 32 kbit/s for hands-free operation in systems with low frame loss", ITU-T Recommendation G.722.1, May 2005.
- [Speex] Xiph.Org Foundation, "Speex: <http://www.speex.org/>", 2003.
- [carot09] Carot, A., Werner, C., and T. Fischinger, "Towards a Comprehensive Cognitive Analysis of Delay-Influenced Rhythmical Interaction: <http://www.carot.de/icmc2009.pdf>", 2009.
- [PAYLOADS] Handley, M. and C. Perkins, "Guidelines for Writers of RTP Payload Format Specifications", RFC 2736, BCP 36.
- [RTP] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for real-time applications", RFC 3550.
- [SRTP] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, March 2004.
- [wright08]

Wright, C., Ballard, L., Coull, S., Monroe, F., and G. Masson, "Spot me if you can: Uncovering spoken phrases in encrypted VoIP conversations:  
<http://www.cs.jhu.edu/~cwright/oakland08.pdf>", 2008.

[white11] White, A., Matthews, A., Snow, K., and F. Monroe, "Phonotactic Reconstruction of Encrypted VoIP Conversations: Hookt on fon-iks  
<http://www.cs.unc.edu/~fabian/papers/foniks-oak11.pdf>", 2011.

Authors' Addresses

Jean-Marc Valin  
Mozilla  
650 Castro Street  
Mountain View, CA 94041  
USA

Email: [jmvalin@jmvalin.ca](mailto:jmvalin@jmvalin.ca)

Koen Vos  
Skype Technologies S.A.  
Stadsgarden 6  
Stockholm, 11645  
Sweden

Email: [koen.vos@skype.net](mailto:koen.vos@skype.net)





Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: April 28, 2011

JM. Valin  
Octasic Inc.  
S. Borilin  
SPIRIT DSP  
K. Vos  
Skype Technologies S.A.  
C. Montgomery  
Xiph.Org Foundation  
R. Chen  
Broadcom Corporation  
October 25, 2010

Guidelines for the Codec Development Within the IETF  
draft-valin-codec-guidelines-08

Abstract

This document provides general guidelines for work on developing and specifying a codec within the IETF. These guidelines cover the development process, evaluation, requirements conformance, and intellectual property issues.

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 April 28, 2011.

Copyright Notice

Copyright (c) 2010 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 . . . . .	3
2. Development Process . . . . .	4
3. Evaluation, Testing, and Characterization . . . . .	7
4. Requirements Conformance . . . . .	8
5. Intellectual Property . . . . .	10
6. Relationship with Other SDOs . . . . .	12
7. Security Considerations . . . . .	14
8. IANA Considerations . . . . .	15
9. Acknowledgments . . . . .	16
10. References . . . . .	17
10.1. Normative References . . . . .	17
10.2. Informative References . . . . .	17
Authors' Addresses . . . . .	19

## 1. Introduction

This document describes a suggested process for work at the IETF on standardization of a codec that is optimized for use in interactive Internet applications and that can be widely implemented and easily distributed among application developers, service operators, and end users.

## 2. Development Process

The process outlined here is intended to make the work on an audio codec within the IETF transparent, predictable, and well organized. Such work might involve development of a completely new codec, adaptation of an existing codec to meet the requirements, or integration between two or more existing codecs that results in an improved codec combining the best aspects of each codec. To enable such procedural transparency, the contributor of an existing codec must be willing to cede change control to the IETF and should have sufficient knowledge of the codec to assist in the work of adapting it or applying some of its technology to the development or improvement of other codecs. Furthermore, contributors need to be aware that any codec that results from work within the IETF is likely to be different from any existing codec that was contributed to the Internet Standards Process.

Work on codec development is expected to proceed as follows:

1. IETF participants will identify the requirements to be met by an Internet codec, in the form of an Internet-Draft.
2. Interested parties are encouraged to make contributions proposing existing or new codecs, or elements thereof, to the codec WG as long as these contributions are within the scope of the WG. Ideally, these contributions should be in the form of Internet Drafts, although other forms of contributions are also possible as discussed in [PROCESS] and in the IETF's Note Well. As always, contributions to the IETF are subject, among other process oriented RFCs, to [PROCESS], [TRUST], and [IPR]. Considering the field of technology, IPR transparency may be particularly high on the priority list of many codec WG participants. Accordingly, contributors are specifically reminded of their IPR disclosure requirement, and all participants are reminded of the solicitation of the disclosure of third party IPR, both as codified in [IPR].
3. As contributions are received and discussed within the working group, the group should gain a clearer understanding of what is achievable within the design space. As a result, the authors of the requirements document should iteratively clarify and improve their document to reflect the emerging working group consensus. This is likely to involve collaboration with IETF working groups in other areas, such as collaboration with working groups in the Transport area to identify important aspects of packet transmission over the Internet and to understand the degree of rate adaptation desirable, and with working groups in the RAI area to ensure that information about and negotiation of the

codec can be easily represented at the signalling layer. In parallel with this work, interested parties should evaluate the contributions at a higher level to see which requirements might be met by each codec.

4. Once a sufficient number of proposals has been received, the interested parties will identify the strengths, weaknesses, and innovative aspects of the contributed codecs. This step will consider not only the codecs as a whole, but also key features of the individual algorithms (predictors, quantizers, transforms, etc.).
5. It is expected that none of the contributed codecs will meet all of the defined requirements. Therefore, it is expected that IETF participants will accept a baseline codec as a WG item to facilitate the development process. This baseline codec will meet as many of the requirements as possible, but probably will need to be adjusted through an iterative development process in order to meet all of the requirements (or as many requirements as possible). The baseline codec might be one of the contributed codecs (especially if it is the only codec that meets most of the requirements), a combination of two or more of the contributed codecs, or an entirely new codec. None of the decisions taken at this step will be definitive. In particular, IETF participants will not provide a "rubber stamp" for any contributed codec.
6. IETF participants should then attempt to iteratively improve each component of the baseline codec reference implementation, where by "component" we mean individual algorithms such as predictors, transforms, quantizers, and entropy coders. The participants should proceed by trying new designs, applying ideas from the contributed codecs, evaluating "proof of concept" ideas, and using their expertise in codec development to improve the baseline codec. Any aspect of the baseline codec might be changed (even the fundamental principles of the codec) or the participants might start over entirely by scrapping the baseline codec and designing a completely new one. The overriding goal shall be to design a codec that will meet the requirements defined in the requirements document. Given the IETF's open standards process, any interested party will be able to contribute to this work, whether or not they submitted an Internet-Draft for one of the contributed codecs. The codec itself should be normatively specified with code in an Internet-Draft.
7. In parallel with work on the codec reference implementation, developers and other interested parties should perform evaluation of the codec as described under Section 3, IETF participants

should define (within the AVT Working Group) the codec's payload format for use with the Real-time Transport Protocol [RTP], and application developers should start testing the codec by implementing it in code and deploying it in actual Internet applications to identify any potential problems.

8. Once IETF participants agree that the codec being developed meets the requirements, IETF participants can begin the task of characterizing the codec. The characterization process is described under Section 3.

### 3. Evaluation, Testing, and Characterization

Lab evaluation of the codec being developed should happen throughout the development process because it will help ensure that progress is being made toward fulfillment of the requirements. There are many ways in which continuous evaluation can be performed. For minor, uncontroversial changes to the codec it should usually be sufficient to use objective measurements (e.g., PESQ, PEAQ, and SegSNR) validated by informal subjective evaluation. For more complex changes (e.g., when psychoacoustic aspects are involved) or for controversial issues, internal testing should be performed. An example of internal testing would be to have individual participants rate the decoded samples using one of the established testing methodologies, such as ITU-R BS.1534 (MUSHRA).

Throughout the process, it will be important to make use of the Internet community at large for real-world distributed testing. This will enable many different people with different equipment and use cases to test the codec and report any problems they experience. In the same way, third-party software developers will be encouraged to integrate the codec (with a warning about the bit-stream not being final) and provide feedback on its performance in real-world use cases.

Characterization of the final codec must be based on the reference implementation only (and not on any "private implementation"). This can be performed by independent testing labs or, if this is not possible, using the testing labs of the organizations that contribute to the Internet Standards Process. Packet loss robustness should be evaluated using actual loss patterns collected from use over the Internet, rather than theoretical models. The goals of the characterization phase are to:

- o ensure that the requirements have been fulfilled
- o guide the IESG in its evaluation of the resulting work
- o assist application developers in understanding whether the codec is suitable for a particular application

The exact methodology for the characterization phase is still subject to discussion within the working group.

#### 4. Requirements Conformance

It is the responsibility of the working group to define criteria for evaluating conformance, including but not limited to comparison tools and test vectors. The following text provides suggestions for consideration by the working group:

1. Any codec specified by the IETF must include source code for a normative C89 implementation, documented in an Internet Draft destined for standards track RFC. This implementation will be used to verify conformance of an implementation. Although a text description of the algorithm should be provided, its use should be limited to helping the reader in understanding the source code. Should the description contradict the source code, the latter shall take precedence. For convenience, the source code may be provided in compressed form, with base64 encoding.
2. Because of the size of the codec's source code, it is possible that even after publishing the RFC, bugs would be found from time to time. An errata of the RFC and its software description should be maintained, along with a public software repository containing the current reference implementation.
3. It is the intention of the group to allow the greatest possible choice of freedom in implementing the specification. Accordingly, the number of binding RFC2119 keywords is going to be the minimum still allowing for interoperable implementations. In practice this generally means that only the decoder needs to be normative, so that the encoder can improve over time. This also enables different tradeoffs between quality and complexity.
4. To reduce the risk of bias towards certain CPU/DSP architectures, ideally the decoder specification should not require "bit-exact" conformance with the reference implementation. The output of a decoder implementation should only be "close enough" to the output of the reference decoder. A comparison tool should be provided along with the codec to verify objectively that the output of a decoder is likely to be perceptually indistinguishable from that of the reference decoder. However, an implementation may still wish to produce an output that is bit-exact with the reference implementation to simplify the testing procedure.
5. To ensure freedom of implementation, decoder-side only error concealment does not need to be specified, although the reference implementation should include the same PLC algorithm as used in the testing phase. Is it up to the working group to decide whether minimum requirements on PLC quality will be required for



compliance with the specification. Obviously, any information signaled in the bitstream intended to aid PLC needs to be specified.

6. An encoder implementation should not be required to make use of all the "features" (tools) in the bit-stream definition. However, the codec specification may require that an encoder implementation be able to generate any possible bit-rate. Unless a particular "profile" is defined in the specification, the decoder must be able to decode all features of the bit-stream. The decoder must also be able to handle any combination of bits, even combinations that cannot be generated by the reference encoder. It is recommended that the decoder specification shall define exactly how the decoder should react to "impossible" packets. However, an encoder must never generate such packets that do not conform to the bit-stream definition.
7. Compressed test vectors should be provided as a means to verify conformance with the decoder specification. These test vectors should exercise all paths in the decoder (100% code coverage).
8. While the exact encoder will not be specified, it is recommended to specify objective measurement targets for an encoder, below which use of a particular encoder implementation is not recommended. For example, one such specification could be: "the use of an encoder whose PESQ MOS is less than 0.1 below the reference encoder in the following conditions is not recommended".

## 5. Intellectual Property

Producing an unencumbered codec is desirable for the following reasons:

- o It is the experience of a wide variety of application developers and service providers that encumbrances such as licensing and royalties make it difficult to implement, deploy, and distribute audio applications for use by the Internet community.
- o It is beneficial to have low-cost options whenever possible because standalone voice services are being commoditized and small, innovative development teams often cannot afford to pay per-channel licensing fees and royalties.
- o Many market segments are moving away from selling hard-coded hardware devices and toward freely distributing end-user software; this is true of numerous large application providers and even telcos themselves.
- o Compatibility with the licensing of typical open source applications implies the need to avoid encumbrances, including even the requirement to obtain a license for implementation, deployment, or use (even if the license does not require the payment of a fee).

Therefore, a codec that can be widely implemented and easily distributed among application developers, service operators, and end users is preferred. Many existing codecs that might fulfill some or most of the technical attributes listed above are encumbered in various ways. For example, patent holders might require that those wishing to implement the codec in software, deploy the codec in a service, or distribute the codec in software or hardware need to request a license, enter into a business agreement, pay licensing fees or royalties, or adhere to other special conditions or restrictions. Because such encumbrances have made it difficult to widely implement and easily distribute high-quality audio codecs across the entire Internet community, the working group prefers unencumbered technologies in a way that is consistent with BCP 78 and BCP 79. In particular, the working group shall heed the preference stated in BCP 79: "In general, IETF working groups prefer technologies with no known IPR claims or, for technologies with claims against them, an offer of royalty-free licensing." Although this preference cannot guarantee that the working group will produce an unencumbered codec, the working group shall follow BCP 79, and adhere to the spirit of BCP 79. The working group cannot explicitly rule out the possibility of adopting encumbered technologies; however, the working group will try to avoid encumbered technologies

that require royalties or other encumbrances that would prevent such technologies from being easy to redistribute and use.

The following guidelines will help to maximize the odds that the codec will be unencumbered:

1. In accordance with BCP 79 [IPR], contributed codecs should preferably use technologies with no known IPR claims or technologies with an offer of royalty-free (RF) licensing.
2. Whenever possible, the working group should use technologies that are perceived by the participants to be safer with regard to IPR issues.
3. Contributors must disclose IPR as specified in BCP 79.
4. In cases where no RF license can be obtained regarding a patent, the group should consider alternative algorithms or methods, even if they result in lower quality, higher complexity, or otherwise less desirable characteristics (in most cases, the degradation will likely be small once the best alternative has been identified).
5. In accordance with BCP 78 [TRUST], the source code for the reference implementation must be made available under a BSD-style license (or whatever license is defined as acceptable by the IETF Trust when the Internet-Draft defining the reference implementation is published).

IETF participants should be aware that, given the way patents work in most countries, the resulting codec can never be guaranteed to be free of patent claims because some patents may not be known to the contributors, some patent applications may not be disclosed at the time the codec is developed, and only courts of law can determine the validity and breadth of patent claims. However, these observations are no different within the Internet Standards Process than they are for standardization of codecs within other SDOs (or development of codecs outside the context of any SDO), and furthermore are no different for codecs than for other technologies worked on within the IETF. In all these cases, the best approach is to minimize the risk of unknowingly incurring encumbrance on existing patents. Despite these precautions, participants need to understand that, practically speaking, it is nearly impossible to guarantee that implementors will not incur encumbrance on existing patents.

## 6. Relationship with Other SDOs

It is understood that other SDOs are also involved in the codec development and standardization, including but not necessarily limited to:

- o The Telecommunication Standardization Sector (ITU-T) of the International Telecommunication Union (ITU), in particular Study Group 16
- o The Moving Picture Experts Group (MPEG)
- o The European Telecommunications Standards Institute (ETSI)
- o The 3rd Generation Partnership Project (3GPP)
- o The 3rd Generation Partnership Project 2 (3GPP2)

It is important to ensure that such work does not constitute uncoordinated protocol development, of the kind described in [UNCOORD] in the following principle:

[T]he IAB considers an essential principle of the protocol development process that only one SDO maintains design authority for a given protocol, with that SDO having ultimate authority over the allocation of protocol parameter code-points; defining the intended semantics, interpretation, and actions associated with those code-points.

The work envisioned by this guidelines document is not "uncoordinated" in the sense described in the foregoing quote, for the following reasons:

- o Internet signalling technologies are designed to enable the negotiation of any codecs that are supported in a particular application (such signalling technologies include the Session Initiation Protocol [SIP], Session Description Protocol [SDP], and the Extensible Messaging and Presence Protocol [XMPP] extensions for media negotiation as specified in [Jingle]).
- o Internet transport technologies such as the Real-time Transport Protocol [RTP] (including secure transport as described in [SRTP]) are designed to support any codec for which RTP packetization rules have been defined.
- o The IETF codec working group will focus on issues that are specific to the Internet, including robustness to packet loss and other aspects of packet transmission over the Internet. Issues

that are specific to non-Internet transports (e.g., radio communication and circuit-switched networks) are specifically out of scope.

Although there is already sufficient codec expertise available among IETF participants to complete the envisioned work, additional contributions are welcome within the framework of the Internet Standards Process, in the following ways:

- o Individuals who are technical contributors to codec work within other SDOs can participate directly in codec work within the IETF.
- o Other SDOs can contribute their expertise (e.g., codec characterization and evaluation techniques) and thus facilitate the testing of a codec produced by the IETF.
- o Any SDO can provide input to IETF work through liaison statements.

However, it is important to note that final responsibility for the development process and the resulting codec will remain with the IETF as governed by BCP 9 [PROCESS].

Finally, there is precedent for the contribution of codecs developed elsewhere to the ITU-T (e.g., AMR Wideband was standardized originally within 3GPP). This is a model to explore as the IETF coordinates further with the ITU-T in accordance with the collaboration guidelines defined in [COLLAB].

## 7. Security Considerations

The procedural guidelines for codec development do not have security considerations. However, the resulting codec needs to take appropriate security considerations into account, for example as outlined in [DOS] and [SECGUIDE].

## 8. IANA Considerations

This document has no actions for IANA.

## 9. Acknowledgments

We would like to thank all the other people who contributed directly or indirectly to this document, including Jason Fischl, Gregory Maxwell, Alan Duric, Jonathan Christensen, Julian Spittka, Michael Knappe, Timothy Terriberry, Christian Hoene, Stephan Wenger and Henry Sinnreich. We also like to thank Cullen Jennings and Gregory Lebovitz for their advice. Special thanks to Peter Saint-Andre, who originally co-authored this document.



## 10. References

### 10.1. Normative References

- [IPR] Bradner, S., "Intellectual Property Rights in IETF Technology", BCP 79, RFC 3979, March 2005.
- [PROCESS] Bradner, S., "The Internet Standards Process -- Revision 3", BCP 9, RFC 2026, October 1996.
- [TRUST] Bradner, S. and J. Contreras, "Rights Contributors Provide to the IETF Trust", BCP 78, RFC 5378, November 2008.

### 10.2. Informative References

- [COLLAB] Fishman, G. and S. Bradner, "Internet Engineering Task Force and International Telecommunication Union - Telecommunications Standardization Sector Collaboration Guidelines", RFC 3356, August 2002.
- [DOS] Handley, M., Rescorla, E., and IAB, "Internet Denial-of-Service Considerations", RFC 4732, December 2006.
- [Jingle] Ludwig, S., Saint-Andre, P., Egan, S., McQueen, R., and D. Cionoiu, "Jingle RTP Sessions", XSF XEP 0167, June 2009.
- [RTP] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.
- [SDP] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006.
- [SECGUIDE] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", BCP 72, RFC 3552, July 2003.
- [SIP] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.
- [SRTP] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, March 2004.
- [UNCOORD] Bryant, S. and M. Morrow, "Uncoordinated Protocol

Development Considered Harmful", RFC 5704, November 2009.

[XMPP] Saint-Andre, P., Ed., "Extensible Messaging and Presence Protocol (XMPP): Core", RFC 3920, October 2004.

Authors' Addresses

Jean-Marc Valin  
Octasic Inc.  
4101, Molson Street  
Montreal, Quebec  
Canada

Email: [jean-marc.valin@octasic.com](mailto:jean-marc.valin@octasic.com)

Slava Borilin  
SPIRIT DSP

Email: [borilin@spiritdsp.net](mailto:borilin@spiritdsp.net)

Koen Vos  
Skype Technologies S.A.  
Stadsgarden 6  
Stockholm, 11645  
Sweden

Email: [koen.vos@skype.net](mailto:koen.vos@skype.net)

Christopher Montgomery  
Xiph.Org Foundation

Email: [xiphmont@xiph.org](mailto:xiphmont@xiph.org)

Raymond (Juin-Hwey) Chen  
Broadcom Corporation

Email: [rchen@broadcom.com](mailto:rchen@broadcom.com)

