Internet Engineering Task Force
Internet-Draft
Updates: 6716 (if approved)
Intended status: Standards Track
Expires: 24 January 2026

J. Buethe, Ed.
Meta
J.-M. Valin
Google

23 July 2025

Integration of Speech Codec Enhancement Methods into the Opus Codec
draft-ietf-mlcodec-opus-speech-coding-enhancement-02

Abstract

This document proposes a set of requirements for integrating a speech codec enhancement method into the Opus codec [RFC6716]

Status of This Memo

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

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

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

This Internet-Draft will expire on 24 January 2026.

Copyright Notice

Table of Contents

1. Introduction

Since the specification of the original Opus codec [RFC6716], new data-driven speech codec enhancement methods emerged which outperform classical enhancement methods by a large margin. Using such enhancement methods to improve the quality of the Opus speech codec SILK requires an update of [RFC6716] since SILK is an embedded coding mode and changing the output of the SILK decoder will lead to a violation of the Opus conformance criteria. The purpose of this document is hence to update [RFC6716] to enable the use of a speech codec enhancement algorihm. Specifically, this document defines the notion of a SILK enhancement algorithm and sets forth a list of requirements, some mandatory, some optional, that aim to ensure

(1) consistent performance of the enhancement method itself,

(2) preservation of decoder performance (e.g. seamless mode switching), and

(3) preservation of basic interoperability when tuning the Opus encoder for use with an enhanced decoder.

While the first two objectives target the Opus decoder alone, the third objective introduces new restrictions on the Opus encoder. However, these are not expected to interfer with any existing

implementation of an Opus encoder since they target potential interoperability issues arising from new incentives connected to the possibility to enhance the Opus decoder.

The approach of specifying requirements instead of specifying the enhancement algorithm itself has the advantage of allowing the Opus decoder to benefit from future improvements in a field that currently sees rapid development. Still, a description of the linear-adaptive coding enhancer (LACE) and its integration into the Opus decoder is included as an illustrative example for a SILK enhancement method.

1.1. Requirements Language

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

2. An Illustrative Example

We use the linear-adaptive coding enhancer (LACE) [lace-paper] as an illustrative example to highlight the specific challenges of integrating a speech codec enhancement method into the Opus decoder. LACE is trained to enhance the output signal of the SILK decoder, the speech coding mode of Opus, and Figure 1 depicts a high-level overview of the Opus decoder with LACE added as an enhancement algorithm.

The first requirement for a speech coding enhancement method concerns the performance of the method itself. In this example it relates to the question how the SILK decoder output compares to the LACE output. In [lace-paper] this has been evaluated on clean speech samples using a P.808 listening test [p.808] as well as the objective method PESQ, which showed consistent improvement for all tested bitrates. For a general enhancement method it will be necessary to specify testing material and performance criteria to prevent unintended quality degradation of the Opus codec.

The second requirement concerns performance of the Opus decoder as a whole. Depending on the bitstream the decoder may have to perform mode switching, e.g. between SILK and CELT, or it may combine the SILK and CELT outputs when the codec operates in hybrid mode. Changes to the SILK output signal by an enhancement method, such as added delay, phase shifts, or level alterations can therefore negatively impact the performance of the Opus decoder even if the first requirement is met. LACE solves this problem by adding no delay and by being approximately phase and level preserving.

However, since many enhancement methods are non causal and non phase preserving, these requirements may be too strict for a general enhancement method.

The third requirement concerns interoperability. The Opus specification provides significant freedom for tuning the encoder and the presence of an enhancement method in the decoder may change the optimal encoding choices significantly. In the present example encoding e.g. wideband content at 6 kb/s still leads to fair-to-good quality when using then LACE-enhanced decoder while the quality of a legacy decoder is significantly worse. To make full use of these new enhancement methods, such encoder tunings should be allowed but basic interoperability with legacy decoders or other enhanced decoders needs to be ensured.

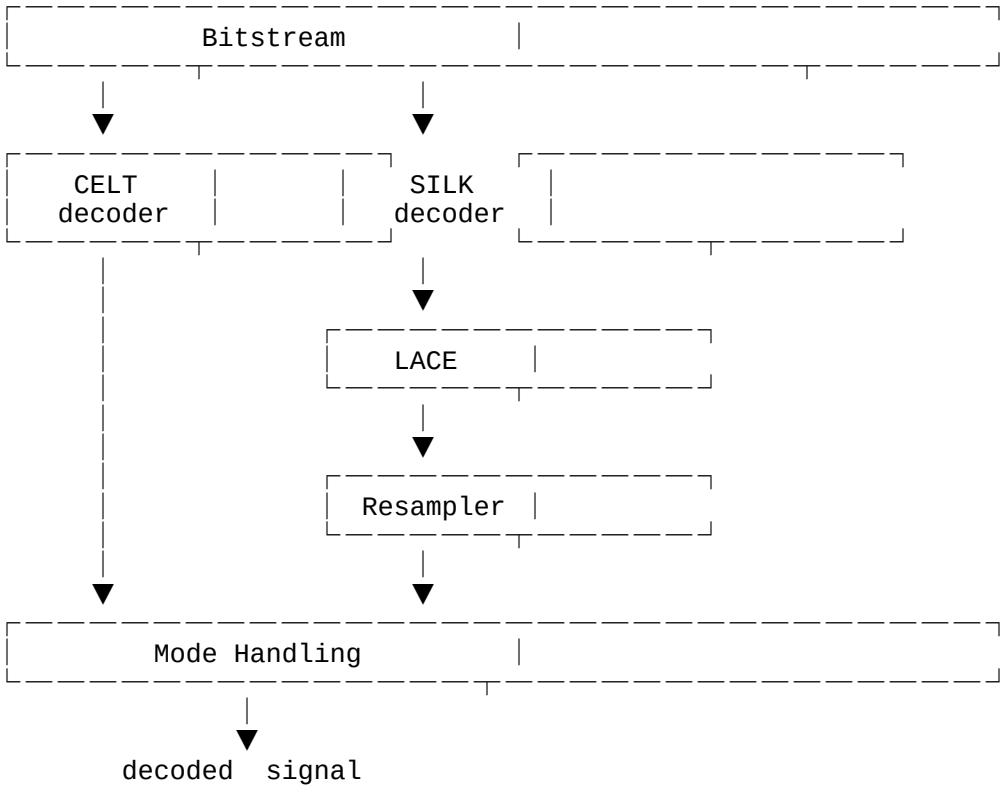

Figure 1: A simplified Opus decoder diagram including LACE as enhancement module

LACE has meanwhile been superceded by the Non-Linear adaptive coding enhancer (NoLACE) [nolace-paper] which shares all basic properties of LACE outlined above but provides higher quality. This stresses the advantage of specifying requirements for an enhancement method over specifying the method itself.

3. Definition of a SILK enhancement algorithm

A SILK enhancement algorithm denotes any algorithm that modifies or replaces the output of the SILK decoder an example of which is depicted in Figure 1. If the decoder sampling rate allows for a higher bandwidth than the encoded bandwidth, a SILK enhancement algorithm may also increase the bandwidth of the output signal, replacing the resampler in Figure 1, or it may modify the combination of a SILK-decoded wideband signal with a CELT decoded highband signal in hybrid mode. However, it may not modify the output of pure CELT frames. A SILK enhancement algorithm that extends the bandwidth of the input signal will be referred to as extending, whereas a SILK enhancement algorithm preserving the bandwidth of the input signal will be referred to as non-extending. Furthermore, an Opus decoder including a SILK enhancement algorithm will be referred to as enhanced decoder. Note however, that simply resampling the signal to a higher sampling rate is neither considered enhancement nor extending.

4. Qualification of a SILK enhancement algorithm

4.1. General requirements

4.1.1. Subjective Evaluation

Objective metrics for quality evaluation have often proved unreliable especially for evaluating completely new algorithms for processing speech or audio signals. Therefore, any SILK enhancement algorithm SHOULD undergo subjective evaluation before integration into the Opus decoder. For genuinely new algorithms, it is RECOMMENDED to perform either an absolute category rating (ACR) or degradation category rating (DCR) listening test according to [p.800] or [p.808], where the test conditions SHOULD cover a relevant range of bitrates. For modifications of previously tested algorithms, e.g. changing the size of a LACE model or adding small tunings for quality improvement or complexity reduction, at least an informal subjective evaluation SHOULD be carried out. Any enhancement method SHOULD significantly improve quality for at least one encoder operating point while showing no significant degradation for other operating points.

4.1.2. Delay and Phase considerations

SILK is approximately phase preserving and to avoid additional delay and maintain usability for applications relying on phase information, any SILK enhancement algorithm SHOULD also be approximately phase preserving.

4.1.3. Encoder Requirements

The Opus specification [RFC6716] provides much freedom for encoding an audio signal and the presence of a powerful enhancement method can provide an incentive to use that freedom to produce bitstreams that, when decoded with a legacy Opus decoder, do not result in a reproduction of the input signal anymore. To prevent this, the following requirement is added for an Opus encoder that is designed to be used with an enhanced Opus decoder: if an Opus encoder produces a bitstream that can be decoded into a human-recognizable reproduction of the encoded signal with an enhanced Opus decoder, then that bitstream MUST also result in a human-recognizable reproduction of the encoded signal when decoded with a legacy Opus decoder.

4.2. Requirements specific to non-extending SILK enhancement algorithms for wideband speech

4.2.1. Objective Evaluation

Every non-extending SILK enhancement algorithm for SILK decoded wideband speech signals MUST pass all objective tests put forth in this section. This collection of tests is designed to uncover major failure points of the tested algorithm that could be due to improper design or training data, or due to improper integration into the opus decoder. It is not designed to (and cannot) assess the quality of a particular enhancement method.

The tests are based on comparing a degradation score for audio samples decoded from a list of bitstreams contained in https://media.xiph.org/opus/ietf/osce_testvectors_v0.zip (FIXME: find final location) to a reference degradation score computed from audio decoded with a reference decoder. The exact reference decoder is TBD. Each test corresponds to an encoder operating point and the test names follow the scheme

osce_test_BITRATE_BITRATEMODE_FRAMESIZEms_BANDWIDTH_cCOMPLEXITY_MODE

where

(1) BITRATE is either a number specifying the encoder bitrate in bits per second or the string "SWITCHING" indicating the bitrate has been switched during encoding,

(2) BITRATEMODE is either vbr or cbr indicating variable bitrate or constant bitrate encoding,

(3) FRAMESIZE is either 10 or 20,

(4) BANDWIDTH specifies the maximal bandwidth and is always WB for this test (note however that the actual bandwidth can be lower),

(5) COMPLEXITY is a number from 0 to 10 and specifies the encoder complexity,

(6) MODE refers to the coding mode and is either "native" or "celtswitching". In "native" mode, the encoder decision whether to use SILK or CELT is based on signal classification whereas in "celtswitching" mode the encoder has been forced to switch between SILK and CELT at a fix rate.

The testvectors are further divided into groups, where each group contains either speech samples from the same language or dialect, or music content. Each group GROUP is tested separately and the test is passed if it is passed for every group. The bitstreams in TESTNAME/GROUP follow the naming pattern CLIPNAME_TESTNAME which associates each bitstream uniquely with a reference signal reference_clips/CLIPNAME.s16. For every CLIPNAME in GROUP let REFMOC(CLIPNAME) denote the reference degradation score stored in the YAML [RFC9512] file TESTNAME/reference_scores_TESTNAME.yml under GROUP as primary key and CLIPNAME as secondary key. Furthermore, let CLIPNAME_test.s16 denote the signal decoded with the enhanced decoder under test at a sampling frequency of 16 kHz after delay compensation. The degradation for the test signal CLIPNAME_test.s16 is calculated using the moc.py tool https://gitlab.xiph.org/xiph/opus/-/blob/osce-testing/dnn/torch/osce/stndrd/qualification/moc.py (FIXME: moc should be implemented in C and PLC will require masking) with reference signal path as first argument and test signal path as second argument. The resulting degradation score will be referred to as TESTMOC(CLIPNAME).

From the reference degradation score REFMOC(CLIPNAME) and the test degradation score TESTMOC(CLIPNAME) a difference score is calculated according to

$$D(CLIPNAME) = \frac{REFMOC(CLIPNAME) - TESTMOC(CLIPNAME)}{0.1 + REFMOC(CLIPNAME)^{0.5}}$$

To pass the test for group GROUP, the following two criteria MUST be met:

(1) D(CLIPNAME) is larger than A for every CLIPNAME in GROUP,

(2) The average of D(CLIPNAME) over GROUP is larger than B.

The exact thresholds A and B are TBD. A test is passed if it is passed for all groups in that test.

4.2.2. Requirements specific to extending SILK enhancement algorithms for wideband speech

Requirements for SILK enhancement algorithms extending the bandwidth of wideband speech are TBD.

5. IANA Considerations

The decoder should be able to signal the presence of an enhancement method to the encoder over SDP. The exact mechanism is TBD and the following options are open for discussion.

(1) update audio/opus media type registration [RFC7587] to include a parameter speech_enhancement with possible values 0 and 1

(2) assign an extension ID, e.g. 33, from the registry defined in [opus-extension] to implement speech coding enhancement. This has the advantage of a double use, meaning the extension ID can both be used to signal the decoder capability to the encoder and for transmitting side information to guide a speech enhancment method from the encoder to the decoder. However, it needs to be proven that side information is useful.

(3) update [opus-extension] to include extension IDs beyond 127 for data-less extensions

6. Security Considerations

TBD

7. References

7.1. Normative References

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

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

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

[RFC7587] Spittka, J., Vos, K., and JM. Valin, "RTP Payload Format for the Opus Speech and Audio Codec", RFC 7587, DOI 10.17487/RFC7587, June 2015, <https://www.rfc-editor.org/info/rfc7587>.

[RFC9512] Polli, R., Wilde, E., and E. Aro, "YAML Media Type", RFC 9512, DOI 10.17487/RFC9512, February 2024, <https://www.rfc-editor.org/info/rfc9512>.

[opus-extension] Valin, J.-M., "Extension Formatting for the Opus Codec (draft-valin-opus-extension)", April 2023.

7.2. Informative References

[lace-paper] Buethe, J., Valin, J.-M., and A. Mustafa, "LACE: A light-weight, causal Model for enhancing coded Speech through Adaptive Convolutions", 2023.

[nolace-paper] Buethe, J., Mustafa, A., Valin, J.-M., Helwani, K., and M. Goodwin, "NoLACE: Improving Low-Complexity Speech Codec Enhancement Through Adaptive Temporal Shaping", 2024.

[p.800] ITU-T, "P.800 : Methods for subjective determination of transmission quality", August 1996, <https://www.itu.int/rec/T-REC-P.800-199608-I>.

[p.808] ITU-T, "P.808 : Subjective evaluation of speech quality with a crowdsourcing approach", June 2021, <https://www.itu.int/rec/T-REC-P.808-202106-I/en>.

Authors' Addresses

Jan (editor)
Meta
United States of America
Email: jan.buethe@googlemail.com

Jean-Marc
Google
Canada
Email: jmvalin@jmvalin.ca