Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 9, 2011

JM. Valin
Octasic Inc.
K. Vos
Skype Technologies S.A.

July 8, 2010

Prototype for IETF Interactive Audio Codec
draft-valin-codec-prototype-01

Abstract

This document provides a quick overview of a prototype codec combining a linear prediction layer (SILK) with an MDCT-based layer (CELT). These codecs are used because of the authors' familiarity with the source code, but it does not prevent inclusion of code from other codecs as well. This is a work in progress.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

We propose a hybrid codec based on a linear prediction layer (LP) and an MDCT-based enhancement layer. The main idea behind the proposal is that the speech low frequencies are usually more efficiently coded using linear prediction codecs (such as CELP variants), while the higher frequencies are more efficiently coded in the transform domain (e.g. MDCT). For low sampling rates, the MDCT layer is not useful and only the LP-based layer is used. On the other hand, non-speech signals are not always adequately coded using linear prediction, so for music only the MDCT-based layer is used.

In this proposed prototype, the LP layer is based on the SILK [1] codec [SILK] and the MDCT layer is based on the CELT [2] codec [CELT]. These codecs are used because of the authors' familiarity with the source code, but it does not prevent inclusion of code from other codecs in the future.

This is a work in progress.

2. Hybrid Codec

In hybrid mode, each frame is coded first by the LP layer and then by the MDCT layer. In the current prototype, the cutoff frequency is 8 kHz. In the MDCT layer, all bands below 8 kHz are discarded, such that there is no coding redundancy between the two layers. Also both layers use the same instance of the range coder to encode the signal, which ensures that no "padding bits" are wasted. The hybrid approach makes it easy to support both constant bit-rate (CBR) and varaible bit-rate (VBR) coding. Although the SILK layer used is VBR, it is easy to make the bit allocation of the CELT layer produce a final stream that is CBR by using all the bits left unused by the SILK layer.

The implementation of SILK-based LP layer is similar to the description in the SILK Internet-Draft [SILK] with the main exception that SILK was modified to use the same range coder as CELT. The implementation of the CELT-based MDCT layer is available from the CELT website and is a more recent version (0.8.1) of the CELT Internet-Draft [CELT]. The main changes include better support for 20 ms frames as well as the ability to encode only the higher bands using a range coder partially filled by the SILK layer.

In addition to their frame size, the SILK and CELT codecs require a look-ahead of 5.2 ms and 2.5 ms, respectively. SILK's look-ahead is due to noise shaping estimation (5 ms) and the internal resampling (0.2 ms), while CELT's look-ahead is due to the overlapping MDCT windows. To compensate for the difference, the CELT encoder input is delayed by 2.7 ms. This ensures that low frequencies and high frequencies arrive at the same time.

2.1. Source Code

The source code is currently available in a Git repository [3] which references two other repositories (for SILK and CELT). Some snapshots are provided for convenience at <http://people.xiph.org/~jm/ietfcodec/> along with sample files. Although the build system is very primitive, some instructions are provided in the toplevel README file. This is very early development so both the quality and feature set should greatly improve over time. In the current version, only 48 kHz audio is supported, but support for all configurations listed in Section 3 is planned.

3. Codec Modes

There are three possible operating modes for the proposed prototype:

1. A linear prediction (LP) mode for use in low bit-rate connections with up to 8 kHz audio bandwidth (16 kHz sampling rate)

2. A hybrid (LP+MDCT) mode for full-bandwidth speech at medium bitrates

3. An MDCT-only mode for very low delay speech transmission as well as music transmission.

Each of these modes supports a number of difference frame sizes and sampling rates. In order to distinguish between the various modes and configurations, we need to define a simple header that can used in the transport layer (e.g RTP) to signal this information. The following describes the proposed header.

The LP mode supports the following configurations (numbered from 00000...01011 in binary):

- 16 kHz: 10, 20, 40, 60 ms (00000...00011)

- 12 kHz: 10, 20, 40, 60 ms (00100...00111)

- 8 kHz: 10, 20, 40, 60 ms (01000...01011)

for a total of 12 configurations.

The hybrid mode supports the following configurations (numbered from 01100...01111):

- 48 kHz: 10, 20 ms (01100...01101)

- 32 kHz: 10, 20 ms (01110...01111)

for a total of 4 configurations.

The MDCT-only mode supports the following configurations (numbered from 10000...11101):

- 48 kHz: 2.5, 5, 10, 20 ms (10000...10011)

- 32 kHz: 2.5, 5, 10, 20 ms (10100...10111)

- 16 kHz: 2.5, 5, 10, 20 ms (11000...11011)

- 8 kHz: 2.5, 5, 10, 20 ms (11100...11111)

for a total of 16 configurations.

There is thus a total of 32 configurations, so 5 bits are necessary to indicate the mode, frame size and sampling rate (MFS). This leaves 3 bits for the number of frames per packets (codes 0 to 7):

- 0-2: 1-3 frames in the packet, each with equal compressed size

- 3: arbitrary number of frames in the packet, each with equal compressed size (one size needs to be encoded)

- 4-5: 2-3 frames in the packet, with different compressed sizes, which need to be encoded (except the last one)

- 6: arbitrary number of frames in the packet, with different compressed sizes, each of which needs to be encoded

- 7: The first frame has this MFS, but others have different MFS. Each compressed size needs to be encoded.

When code 7 is used and the last frames of a packet have the same MFS, it is allowed to switch to another code for them.

The compressed size of the frames (if needed) is indicated -- usually -- with one byte, with the following meaning:

- 0: No frame (DTX or lost packet)

- 1-251: Size of the frame in bytes

- 252-255: A second byte is needed. The total size is (size[1]*4)+(size[0]%4)+252

The maximum size representable is 255*4+3+252=1275 bytes. For 20 ms frames, that represents a bit-rate of 510 kb/s, which is really the highest rate anyone would want to use in stereo mode (beyond that point, lossless codecs would be more appropriate).

3.1. Examples

Simplest case: one packet

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |   MFS   |0|0|0|               compressed data...              |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Four frames of the same compressed size:

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |   MFS   |0|1|1|               compressed data...              |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Two frames of different compressed size:

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |   MFS   |1|0|1|   frame size  |        compressed data...     |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Three frames of different _durations_:

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | 1st MFS |1|1|1|   frame size  | 2nd MFS |1|1|1|   frame size  |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | 3rd MFS |1|1|1|   frame size  |      compressed data...       |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

4. Security Considerations

The codec needs to take appropriate security considerations into account, as outlined in [DOS] and [SECGUIDE]. It is extremely important for the decoder to be robust against malicious payloads. Malicious payloads must not cause the decoder to overrun its allocated memory or to take much more resources to decode. Although problems in encoders are typically rarer, the same applies to the encoder. Malicious audio stream must not cause the encoder to misbehave because this would allow an attacker to attack transcoding gateways.

In its current version, this prototype codec likely does NOT meet these security considerations, so it should be used with caution.

5. IANA Considerations

This document has no actions for IANA.

6. Acknowledgments

Thanks to all other developers, including Soeren Skak Jensen, Gregory Maxwell, Christopher Montgomery, Karsten Vandborg Soerensen, and Timothy Terriberry.

7. Informative References

[SILK] Vos, K., Jensen, S., and K. Soerensen, "SILK Speech Codec", draft-vos-silk-01 (work in progress), March 2010.

[CELT] Valin, J-M., Terriberry, T., Maxwell, G., and C. Montgomery, "Constrained-Energy Lapped Transform (CELT) Codec", draft-valin-celt-codec-02 (work in progress), July 2010.

[DOS] Handley, M., Rescorla, E., and IAB, "Internet Denial-of-Service Considerations", RFC 4732, December 2006.

[SECGUIDE]
Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", BCP 72, RFC 3552, July 2003.

[1] <http://developer.skype.com/silk>

[2] <http://www.celt-codec.org/>

[3] <git://git.xiph.org/users/jm/ietfcodec.git>

Authors' Addresses

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

Phone: +1 514 282-8858
Email: jean-marc.valin@octasic.com

Koen Vos
Skype Technologies S.A.
Stadsgaarden 6
Stockholm,   11645
SE

Phone: +46 855 921 989
Email: koen.vos@skype.net