Beta Release v1.3

J. Spittka
H. Astrom
K. Vos
Skype
August 16, 2010

RTP Payload Format and File Storage Format for SILK Speech and Audio Codec

Status of this Memo

This document may not be modified, and derivative works of it may not be created. Note that SILK is still in a beta phase and that SILK and the RTP payload format as well as the file storage format for SILK may still be changed. Please direct all comments to silksupport@skype.net.

Copyright Notice

Abstract

This document defines the Real-time Transport Protocol (RTP) payload format and file storage format for packetization of SILK encoded speech and audio data that is essential to implement SILK in the most compatible way. Further, media type registrations are described for the RTP payload format and the file storage format.

Table of Contents

1. Introduction

SILK is a speech and audio codec developed internally at Skype which is used as the default codec for all Skype to Skype calls. It is highly scalable in terms of audio bandwidth, network bit rate, and complexity, making it the codec of choice for multiple modes and applications.

Skype encourages 3rd party partners to adopt SILK for applications that may or may not be able to inter-operate with the Skype network. Therefore, this document defines the Real-time Transport Protocol (RTP) [3] payload format and file storage format for packetization of SILK encoded speech and audio data that is essential to implement SILK in the most compatible way. Further, media type registrations are described for the RTP payload format and the file storage format.

More information about SILK can be obtained at https://developer.skype.com/silk.

2. Conventions, Definitions and Acronyms used in this document

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 RFC 2119 [1].

Definitions and Acronyms:

CPU - Central Processing Unit

IP - Internet Protocol

MTU - Maximum Transmission Unit

PSTN - Public Switched Telephone Network

Samples - Speech or audio samples

SDP - Session Description Protocol

3. SILK Codec

The SILK speech and audio codec is highly scalable in terms of audio bandwidth, network bit rate, and complexity.

SILK supports four different audio bandwidths, narrowband at 8000 Hz sampling frequency, mediumband at 12000 Hz sampling frequency, wideband at 16000 Hz sampling frequency, and super wideband at 24000 Hz sampling frequency. Narrowband mode SHOULD only be used to interface to PSTN networks or on low end devices that do not support greater than 8000 Hz sampling frequency. Mediumband mode SHOULD be used for lower end devices that do not support greater than 12000 Hz sampling frequency or are under severe network bandwidth constrains (e.g. wireless devices). Wideband mode SHOULD be used for all-IP platforms that do not support greater than 16000 Hz sampling frequency. Super wideband mode SHOULD be used on all platforms that support 24000 Hz and greater sampling frequency.

The average network bit rate target is adaptive within the range specified in Table 1 for corresponding audio bandwidths. The average network bit rate target can be defined and modified in real-time while the actual bit rate will be dependent on the input signal and change over time. The actual bit rate may be higher or lower than the target adaptive bit rates specified in Table 1. The upper limits of the bit rate ranges in this table are recommended values.

	fs (Hz)	BR (kbps)
Narrowband	8000	5 - 20
Mediumband	12000	7 - 25
Wideband	16000	8 - 30
Super Wideband	24000	20 - 40

Table 1: fs specifies the audio sampling frequency in Hertz (Hz); BR specifies the adaptive bit rate target range in kilobits per second (kbps).

Complexity can be scaled to optimize for CPU resources in real-time, mostly in trade-off to network bit rate.

The internal frame size of SILK is 20 ms. The SILK encoder can be set to bundle up to five internal frames into a single frame output, allowing for 20, 40, 60, 80, or 100 ms frames of encoded speech or audio data. Table 2 below shows the number of samples contained in

one frame of speech or audio, for the various frame sizes and sampling rates.

Frame size	20 ms	40 ms	60 ms	80 ms	100 ms
Narrowband samples	160	320	480	640	800
Mediumband samples	240	480	720	960	1200
Wideband samples	320	640	960	1280	1600
Super wideband samples	480	960	1440	1920	2400

Table 2: Samples contained in one frame, for different frame sizes and sampling rates.

SILK operates at a very low algorithmic delay, consisting of packetization delay, i.e. 20, 40, 60, 80, or 100 ms, plus 5 ms look-ahead delay.

3.1. Adaptive Sampling Frequency

The internal sampling frequency of the encoded speech or audio signal of SILK may change during the duration of a transmission. This can happen for two reasons.

First, SILK provides an internal logic that may decide to automatically adapt the internal sampling frequency of the encoded speech or audio signal to the most efficient sampling frequency dependent on the input signal and information about the capacity of the channel. This allows support for congestion control and network load management.

Further, SILK provides API functionality for setting the maximum internal sampling frequency of the encoded speech or audio signal manually. This maximum internal sampling frequency MUST NOT be set higher than the sampling frequency agreed upon during call setup negotiation. The reason for this is that earlier versions of SILK cannot decode signals with higher internal sampling rate than the decoder output sampling rate (decoder API sampling rate).

In both cases, only the internal sampling frequency of the speech or audio signal that is encoded into the bit stream is switched while the sampling frequencies of the input speech or audio signal to the encoder and the output speech or audio signal of the decoder of SILK will not be affected and can be set separately.

At the time a session is set up, the decoding side SHOULD signal to the encoding side all sampling frequencies (rate parameter) that the system can take advantage of.

3.2. Adaptive Network Bit Rate

The SILK encoder can be set to output encoded speech or audio data at a defined average bit rate target. Since the achieved bit rate for each frame varies with the perceptual importance of the input audio or speech signal, the specified average bit rate target is for an active, i.e. non-silent, signal. The average bit rate target can be adjusted on a per frame basis. This allows support for congestion control and network load management.

To do this efficiently, information about the capacity of a channel or storage device has to be available. There are various methods to obtain this information that are outside the scope of this document.

When no information about the capacity of a channel is available, SILK can be run with a fixed average bit rate target. The fixed average bit rate target must be chosen with care since exceeding the capacity of the channel may lead to extensive latency and loss of speech frames. Unless limitations of the channel are known, for most broadband network access technologies, it is recommended to use the maximum average bit rate target provided in Table 1 to maximize speech quality.

3.3. Discontinuous Transmission (DTX)

The SILK codec is, as described in Section 3.2 of this document, a codec with adaptive bit rate. The bit rate will automatically be reduced for certain input signals like periods of silence. During continuous transmission mode the bit rate will be reduced, when the input signal allows the encoder to do so, but the transmission to the receiver itself is not interrupted. Therefore, the received signal will maintain the same high level of quality over the full duration of a transmission while minimizing the average bit rate over time.

In cases where the average bit rate of SILK needs to be reduced even further, the SILK encoder may be set to use a discontinuous transmission mode (DTX), where parts of the encoded signal that correspond to periods of silence in the input speech or audio signal are not transmitted to the receiver.

On the receiving side, the non-transmitted parts will be handled by a frame loss concealment unit in the SILK decoder which generates a

comfort noise signal to replace the non transmitted parts of the speech or audio signal.

The DTX mode of SILK will have a slightly lower speech or audio quality than the continuous mode. Therefore, it is RECOMMENDED to use SILK in the continuous mode unless restraints of network bandwidth are severe.

3.4. Forward Error Correction (FEC)

The SILK codec allows for “in-band” forward error correction (FEC) data to be embedded into the bit stream of SILK. This FEC scheme adds redundant information about the previous frame (n-1) or the frame that is two frames back in time (n-2) to the current output frame n. For each frame, the encoder decides whether to use FEC based on (1) an externally provided estimate of the channel's packet loss rate; (2) an externally provided estimate of the channel's capacity; (3) the sensitivity of the audio or speech signal to packet loss; (4) whether the receiving decoder has indicated it can take advantage of “in-band” FEC information. The decision to send “in-band” FEC information is entirely controlled by the encoder and therefore no special precautions for the payload or storage format have to be taken.

On the receiving side, the decoder can take advantage of this additional information when, in case of a frame loss, future frames are available. In order to use the FEC data, the jitter buffer needs to provide access to payloads with those future SILK frames and information about the offset to the last decoded SILK frame. A special SILK API function allows searching for available FEC data that, in case of a successful search, can be provided to the decoder as a replacement for the current lost frame.

If the FEC scheme is not implemented on the receiving side, FEC SHOULD NOT be used, as it leads to an inefficient usage of network bandwidth. Decoder support for FEC SHOULD be indicated at the time a session is set up.

4. SILK RTP Payload Format

The payload format for SILK consists of the RTP header and SILK payload data.

4.1. RTP Header Usage

The format of the RTP header is specified in RFC 3550 [3]. The SILK payload format uses the fields of the RTP header consistent with this specification.

The payload length of SILK is a multiple number of octets and therefore no padding is required. The payload MAY be padded by an integer number of octets according to RFC 3550 [3].

The marker bit (M) of the RTP header has no function in combination with SILK and MAY be ignored.

The RTP payload type for SILK has not been assigned statically and is expected to be assigned dynamically.

The receiving side MUST be prepared to receive duplicates of RTP packets. Only one of those payloads MUST be provided to the SILK decoder for decoding and others MUST be discarded.

Depending on what mode of maximum internal sampling frequency is set for SILK; 8000, 12000, 16000, or 24000 Hz, the RTP timestamp clock frequency has to be adjusted accordingly and is the same as the sampling frequency. Note, that this sampling frequency only corresponds to the maximum internal sampling frequency specified manually through the API and does not correspond to automatic adaptations of the sampling frequency during runtime which are not directly visible to the outside. The unit for the timestamp is samples. The RTP timestamp corresponds to the sample time of the first encoded sample in the encoded frame. Therefore, the timestamp is increased by the number of samples provided in Table 2, depending on the sampling frequency and frame size.

4.2. Payload Structure

The SILK encoder can be set to output encoded frames representing 20, 40, 60, 80, or 100 ms of speech or audio data. Only one frame output from the encoder MUST be used as the payload. Figure 1 shows the structure combined with the RTP header.

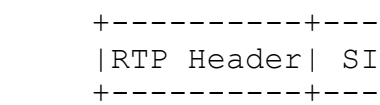

Figure 1 Payload Structure with RTP header

5. SILK Storage Format

The SILK storage format allows to store SILK encoded data into e.g. a file or an email attachment. The storage format consists of a header and a series of blocks containing encoded speech or audio frames. The storage format closely mimics the real-time payload format.

Figure 2 shows an example of a SILK encoded file. Note that due to the adaptive bit rate and therefore variable frame length of SILK no fixed block size can be defined for blocks containing encoded data.

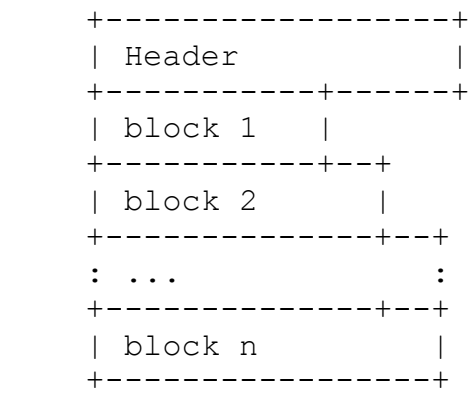

Figure 2 Example of SILK file storage format showing different block lengths due to adaptive bit rate of SILK

5.1. Storage Header Structure

A SILK storage header contains the following ASCII character string as a magic number:

"#!SILK\n" (hexadecimal: 0x23 0x21 0x53 0x49 0x4C 0x4B 0x0A)

5.2. Storage Block Structure

Following the storage header, blocks of encoded data are stored in consecutive order in time according to Figure 2. Each block contains a block header followed by a payload according to Figure 3.

The block header contains information that, for an RTP-based session, can be derived from the IP and RTP headers: SILK sample rate, the number of octets contained in the subsequent payload and the RTP time stamp.

The sample rate is specified by three bits with the following bit convention:

000: SILK Narrowband 8000 Hz

001: SILK Mediumband 12000 Hz

010: SILK Wideband 16000 Hz

011: SILK Super Wideband 24000 Hz

Other values are reserved for future use and blocks where these appear MUST be discarded.

Further, the number of octets in the payload is represented by 13 bits and the timestamp is specified by 32 bits. For the first block, the timestamp MAY be a random number. For the following blocks, the timestamp MUST be incremented according to the way timestamps are incremented when SILK payloads are transmitted over RTP.

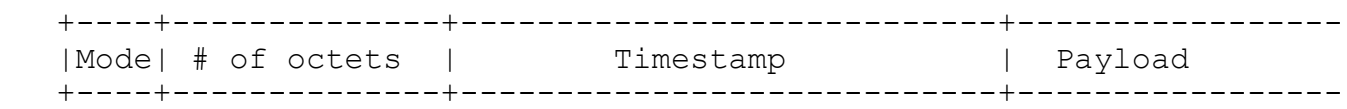

Figure 3 Storage block header structure

The payload of each block in Figure 2 represents one frame of SILK encoded data representing 20, 40, 60, 80, or 100 ms speech or audio data.

During the usage of DTX no blocks are stored when the channel is inactive. Timestamps MUST be used to reassemble the decoded signal in a time-aligned way.

6. Congestion Control

The adaptive nature of the SILK codec allows for an efficient congestion control.

The average bit rate of SILK is dependent on the input signal and will especially decrease during silent periods. The average bit rate can be controlled on a per frame basis and therefore the amount of payload data can be controlled.

Furthermore, 20, 40, 60, 80, or 100 ms of speech or audio data can be combined in a single RTP payload, and the transmission rate is inversely proportional to these frame sizes. A lower packet transmission rate reduces the amount of header overhead but at the same time increases latency and error sensitivity and should be done with care.

While manual adaptation of the internal sampling frequency of SILK allows adjusting the overall network bandwidth usage it SHOULD NOT be used for congestion control during the duration of a call. By controlling the average bit rate target parameter of SILK, SILK will automatically adapt the internal sampling frequency and find the best trade-off between audio quality and bit rate.

It is RECOMMENDED that congestion control is applied during the transmission of SILK encoded data.

7. Security Considerations

All RTP packets using the payload format defined in this specification are subject to the general security considerations discussed in the RTP specification RFC 3550 [3] and any profile from e.g. RFC 3711 [4] or RFC 3551 [5].

This payload format transports SILK encoded speech or audio data, hence, security issues include confidentiality, integrity protection, and authentication of the speech or audio itself. The SILK payload format does not have any built-in security mechanisms. Any suitable external mechanisms, such as SRTP RFC 3711 [4], MAY be used.

This payload format and the SILK encoding do not exhibit any significant non-uniformity in the receiver-end computational load and thus are unlikely to pose a denial-of-service threat due to the receipt of pathological datagrams.

8. IANA Considerations

One media subtype (audio/SILK) has been defined and registered as described in the following section.

8.1. SILK Media Type Registration

Media type registration is done according to RFC 4288 [6] and RFC 4855 [7].

Type name: audio

Subtype name: SILK

Required parameters:

rate: RTP timestamp clock rate that is equal to the sampling frequency in Hertz (Hz) of the represented media in a packet. Possible values are 8000, 12000, 16000, and 24000.

Optional parameters:

maxptime: the decoder's maximum length of time in milliseconds (ms) represented by the media in a packet that can be encapsulated in a received packet according to Section 6 of RFC 4566 [8]. Possible values are 60, 80, and 100 as defined in Section 4 and 5 of this document. If no value is specified, 100 is assumed as default. The receiving endpoint MUST be able to receive packets that represent 20, 40, and 60 ms of media.

ptime: the decoder's recommended length of time in milliseconds (ms) represented by the media in a packet according to Section 6 of RFC 4566 [8]. Possible values are 20, 40, 60, 80, or 100 as defined in Section 4 and 5 of this document. If no value is specified, 20 is assumed as default. If ptime is greater than maxptime, ptime MUST be ignored. This parameter MAY be changed during a session.

maxaveragebitrate: specifies the maximum average receive bit rate of a session in bits per second (bps). The actual value of the bit rate may vary as it is dependent on the characteristics of the media in a frame. Note that the maximum average bit rate target MAY be modified dynamically during a session. A value greater or equal to the lower limit of the average bit rate target specified in Table 1 MUST be provided.

useinbandfec: specifies that SILK in-band FEC is supported by the decoder and MAY be used during a session. Possible values are 1 and 0. It is RECOMMENDED to provide 0 in case FEC is not implemented on the receiving side. If no value is specified, useinbandfec is assumed to be 1.

usedtx: specifies if the decoder prefers the use of DTX. Possible values are 1 and 0. If no value is specified, usedtx is assumed to be 0.

Encoding considerations:

SILK media type is framed and consists of binary data according to Section 4.8 in RFC 4288 [6].

Security considerations:

See Section 7 of this document.

Interoperability considerations: none

Published specification: none

Applications that use this media type:

Any application that requires the transport or storage of speech or audio data may use this media type. Some examples are, but not limited to, audio and video conferencing, Voice over IP, voice recording, media streaming, voice messaging.

Additional information:

For storage transfer methods the following applies:

Magic number:

"#!SILK\n" (hexadecimal: 0x23 0x21 0x53 0x49 0x4C 0x4B 0x0A)

File extension(s): sil, SIL

Macintosh file type code(s): "silk"

Person & email address to contact for further information:

SILK Support <silksupport@skype.net>

Intended usage: COMMON

Restrictions on usage:

For transfer over RTP, the RTP payload format (Section 4 of this document) SHALL be used. For storage usage, the storage format (Section 5 of this document) SHALL be used.

Author:

Julian Spittka <julian.spittka@skype.net>

Henrik Astrom <henrik.astrom@skype.net>

Koen Vos <koen.vos@skype.net>

Change controller:

Skype

8.2. Mapping to SDP Parameters

The information described in the media type specification has a specific mapping to fields in the Session Description Protocol (SDP) RFC 4566 [8], which are commonly used to describe RTP sessions. When SDP is used to specify sessions employing the SILK codec, the mapping is as follows:

- The media type ("audio") goes in SDP "m=" as the media name.
- The media subtype ("SILK") goes in SDP "a=rtpmap" as the encoding name. The RTP clock rate in "a=rtpmap" MUST be mapped to the required media type parameter "rate".
- The optional media type parameters "ptime" and "maxptime" are mapped to "a=ptime" and "a=maxptime" attributes, respectively, in the SDP.
- All remaining media type parameters are mapped to the "a=fmtp" attribute in the SDP by copying them directly from the media type parameter string as a semicolon-separated list of parameter=value pairs (e.g. maxaveragebitrate=20000).

Below are some examples of SDP session descriptions for SILK:

Example 1: Standard session with 12000 Hz clock rate

```
m=audio 54312 RTP/AVP 101

a=rtpmap:101 SILK/12000
```


Example 2: 16000 Hz clock rate, maximum packet size of 60 ms, recommended packet size of 40 ms, maximum average bit rate of 20000 bps, FEC is allowed, DTX is not allowed

```
m=audio 54312 RTP/AVP 101

a=rtpmap:101 SILK/16000

a=fmtp:101 maxaveragebitrate=20000; \

  useinbandfec=1; usedtx=0

a=ptime:40

a=maxptime:60
```

8.2.1. Offer-Answer Model Considerations for SILK

When using the offer-answer procedure described in RFC 3264 [9] to negotiate the use of SILK, the following considerations apply:

- SILK supports several clock rates. Every supported clock rate MUST be announced separately in the "m=audio" line. It is RECOMMENDED to list the highest clock rate with highest priority and lower clock rates with lower priority in decreasing order. The answer will only keep the payload types that are supported by the answerer and the conversation will be performed with the payload type of the first, and, thus, highest common clock rate.

  Once a rate has been agreed on through this procedure, the rate SHOULD NOT be changed for the duration of a call. SILK allows to use the encoder API to change the maximum internal sampling rate in-band rather than through signaling. The timestamp will continue to be increased with the initially agreed upon rate throughout the call.

  An example is shown below:

```
m=audio 54312 RTP/AVP 100 101 102 103

a=rtpmap:100 SILK/24000

a=rtpmap:101 SILK/16000

a=rtpmap:102 SILK/12000

a=rtpmap:103 SILK/8000
```

- The parameters "ptime" and "maxptime" are unidirectional receive-only parameters and typically will not compromise interoperability; however, dependent on the set values of the parameters the performance of the application may suffer. RFC 3264 [9] defines the SDP offer-answer handling of the "ptime" parameter. The "maxptime" parameter MUST be handled in the same way.

- The parameter "maxaveragebitrate" is a unidirectional receive-only parameter that reflects limitations of the local receiver. The sender of the other side MUST NOT send with an average bit rate higher than "maxaveragebitrate" as it might overload the network and/or receiver. The parameter "maxaveragebitrate" typically will not compromise interoperability; however, dependent on the set value of the parameter the performance of the application may suffer and should be set with care.

- If the parameter "maxaveragebitrate" is below the range specified in Table 1 the session MUST be rejected.

- The parameter "useinbandfec" is a unidirectional receive-only parameter.

- The parameter "usedtx" is a unidirectional receive-only parameter.

- Any unknown parameter in an offer MUST be ignored by the receiver and MUST be removed from the answer.

8.2.2. Declarative SDP Considerations for SILK

For declarative use of SDP such as in Session Announcement Protocol (SAP), RFC 2974 [10], and RTSP, RFC 2326 [11], for SILK, the following needs to be considered:

- The values for "maxptime", "ptime", and "maxaveragebitrate" should be selected carefully to ensure that a reasonable performance can be achieved for the participants of a session.

- All parameters of the payload format configuration are declarative and a participant MUST use the configurations that are provided for the session. More than one configuration may be provided if necessary by declaring multiple RTP payload types; however, the number of types should be kept small.

9. References

9.1. Normative References

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

[2] Crocker, D. and Overell, P.(Editors), "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, Internet Mail Consortium and Demon Internet Ltd., November 1997.

[3] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.

[4] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, March 2004.

[5] Schulzrinne, H. and S. Casner, "RTP Profile for Audio and Video Conferences with Minimal Control", STD 65, RFC 3551, July 2003.

[6] Freed, N. and J. Klensin, "Media Type Specifications and Registration Procedures", BCP 13, RFC 4288, December 2005.

[7] Casner, S., "Media Type Registration of RTP Payload Formats", RFC 4855, February 2007.

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

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

[10] Handley, M., Perkins, C., and E. Whelan, "Session Announcement Protocol", RFC 2974, October 2000.

[11] Schulzrinne, H., Rao, A., and R. Lanphier, "Real Time Streaming Protocol (RTSP)", RFC 2326, April 1998.

9.2. Informative References

Casner, S. and P. Hoschka, "MIME Type Registration of RTP Payload Formats", RFC 3555, July 2003.

10. Acknowledgments

The authors like to thank Soren Skak Jensen and Jason Fischl for their invaluable input.

Authors' Addresses

Julian Spittka
Skype
2145 Hamilton Avenue
San Jose, CA 95125

Email: julian.spittka@skype.net

Henrik Astrom
Skype
2145 Hamilton Avenue
San Jose, CA 95125

Email: henrik.astrom@skype.net

Koen Vos
Skype
2145 Hamilton Avenue
San Jose, CA 95125

Email: koen.vos@skype.net

