RTP/RTCP Reception Hint Tracks for Video Call Recording and Playback

Joni Räsänen, Marko Viitanen, Jarno Vanne, and Timo D. Hämäläinen
Tampere University of Technology, Finland

Miska M. Hannuksela and Vinod K. Malamal Vadakital
Nokia Technologies, Tampere, Finland

***Abstract*— This paper proposes to use RTP (Real-time Transport Protocol) Reception Hint Tracks for convenient recording and playback of a video call in MP4 format. The feasibility of RTP Reception Hint Tracks is validated as a part of an implemented end-to-end video call system. The proposed approach records a bidirectional Linphone video call and multiplexes it as MP4 RTP Reception Hint Tracks with the L-SMASH software library. It also stores RTCP (RTP Control Protocol) Reception Hint Tracks for additional timing information. Playback of the MP4 file is performed with VLC Media Player that is made compatible with RTP Reception Hint Tracks. The proposed proof-of-concept setup meets particularly well the needs of multi-codec solutions where different audio and video codecs can be used for a video call recording and playback. According to our analysis, recording RTP reception Hint Tracks increases the Linphone CPU time by under 1% and the bitrate by under 2% over the bare bitrate of the recorded RTP packets.**

Keywords—Video call, Real-time Transport Protocol (RTP), RTP Control Protocol (RTCP), RTP/RTCP Reception Hint Tracks, MP4

I. INTRODUCTION

Real-time Transport Protocol (*RTP*) [1] is developed by *Internet Engineering Task Force* (*IETF*) for real-time data transfer over wired and wireless networks. The quality of an RTP stream is monitored by *RTP Control Protocol* (*RTCP*) [1]. RTP and RTCP are commonly used in *Voice over Internet protocol* (*VOIP*) applications which can also support video communication. A typical video call encompasses video and audio transmitted over a network in order to enable live conversation between participants. However, these applications tend to miss recording and playback functionalities. Skype [2], e.g., requires external software or an add-on for recording [3].

MP4 (*MPEG-4 Part 14*) is a container format developed by the *Moving Picture Expert Group* (*MPEG*). The format is an extension of *ISO Base Media File Format* (*ISOBMFF*) [4]. MP4 is able to encapsulate various media contents such as video, audio, or subtitles into a single container. ISOBMFF was further extended with a feature called RTP Reception Hint Tracks. This extension is used to record RTP stream to a file with additional information that would be lost if recording was done using traditional media tracks.

This paper examines the usage of RTP Reception Hint Tracks in recording and playback of a two-way video call. RTP and RTCP Reception Hint Tracks provide many benefits, including the following:

- An original phone call can be played back from an MP4 file or re-sent to a new destination without losing synchronization information;
- More information (IP addresses, ports used, etc.) can be retained in a file to improve tracking of the media sources;
- Desynchronization caused by clock drift can be corrected during playback;
- Packet losses can be detected during playback;
- The recording operation does not require reconstruction of a correct media bit stream from the packet payloads, which can be challenging particularly in the case of packet losses.

Despite these advantages, none of the known VOIP applications or their add-ons [3], [5] uses RTP Reception Hint Tracks.

In this work, an end-to-end video call system has been used as a proof of concept to illustrate and validate the benefits of RTP and RTCP Reception Hint Tracks. The proposed system has been built on three open-source software tools:

1) Linphone version 3.8.2 [5] which has been extended to support RTP/RTCP Reception Hint Track recording for input streams and media track recording for output streams;
2) L-SMASH library version 2.11.2 [6] which is used to multiplex RTP Reception Hint Tracks into MP4 file; and
3) VLC media player version 2.2.2 [7] which has been modified to demultiplex an MP4 file containing Reception Hint Tracks and playing its content.

All these tools are available for Windows and Linux.

This paper is organized as follows. Section 2 specifies the RTP and RTCP Reception Hint Tracks in more detail. Section 3 describes our extensions to Linphone and L-SMASH for recording RTP/RTCP Reception Hint Tracks in MP4 file format. Section 4 shows the proposed implementation for the recorded MP4 file playback in VLC media player. Section 5 integrates recording and playback functionalities together into a bidirectional video call system. Section 6 concludes the paper.

II. RTP/RTCP RECEPTION HINT TRACKS

The ISOBMFF uses boxes as a basic syntax element. A box is comprised of a four-character boxtype, the size of the box, and its payload. Boxes may be nested so that the payload contains other boxes. A description of all boxes is given in [4].

ISOBMFF and MP4 tracks are either audio, video, or hint tracks. Hint tracks contain media transmission instructions which can be used, e.g., to record an RTP stream into a file [8]. An RTP stream is composed of RTP packets, each of which consists of a header and a payload. An RTP stream can have an accompanying RTCP stream to convey the relation between RTP and *Network Time Protocol* (*NTP*) timestamps alongside other information [1].

RTP and RTCP Reception Hint Tracks are defined in ISOBMFF [4]. They are modelled after RTP Hint Track. RTP Reception Hint Track samples are composed of a sample header and an equal number of the following 1) details from an RTP packet header; 2) a sample constructor; and 3) an RTP packet payload. RTCP Reception Hint Tracks refer to an RTP Reception Hint Track. Sender reports of the RTCP stream are recorded as RTCP Reception Hint Track samples. RTCP Reception Hint Tracks are used to maintain inter-stream synchronization especially when recording starts mid-stream or clock drift exists [4], [8].

III. VIDEO CALL RECORDING

In the proposed system, Linphone [5] is extended to record 1) incoming audio/video RTP packets into RTP Reception Hint Tracks; 2) RTCP sender reports into RTCP Reception hint tracks; and 3) outgoing audio/video content into media tracks. The incoming packets are assumed to originate from another Linphone running on a separate computer. GSM codec was chosen for audio and H.264 codec for video because they were supported by both Linphone and L-SMASH.

A. Proposed implementation in Linphone

Linphone consists of five libraries. They are depicted in Fig. 1. Only minor changes were made in the libraries marked in light grey and none in Belle-sip.

Windows and *Linux* applications use the GTK [9] for their *Graphical User Interface* (*GUI*). Liblinphone is a library used for configuration and initiation of the phone calls. The proposed changes in Liblinphone allow video recording besides its existing audio recording functionality. Liblinphone is also made to capture the *Session Description Protocol* (*SDP*) [10] message. Belle-sip library is responsible for *Session Initiation Protocol* (*SIP*) [11] and SDP protocol communication with other clients. RTP protocol is implemented by oRTP library.

The libraries marked in dark gray in Fig. 1 contain most of our modifications. Mediastreamer2 library provides functionality for codecs and filter architecture. For both video and audio, it uses a separate filter graph to capture, process, encode, decode, send, and receive media samples that are transformed using oRTP library to RTP packets.

Fig. 2 and Fig. 3 describe audio and video filter graphs of Mediastreamer2 library on Windows, respectively. Each box in these graphs is a filter and each arrow shows how the data moves. A data flow graph for sending is decribed on the left and a data flow graph for receiving on the right. The white boxes are existing filters that did not require modifications whereas gray boxes are new filters added by this work to facilitate different recording operations.

In order to get the RTP packets in a correct format, an RTP receiver filter (*MSRtpRecv*) was modified so that it does not strip the needed RTP header, but a separate filter output was created for the raw RTP packet. In both video and audio graphs, a new RTP recording filter (*RtpRec*) was attached to this output.

The RTP recording is done by first recording incoming RTP packets until a RTP packet with a different RTP timestamp arrives. In a H.264 video RTP stream, a picture is divided into slices which are put to separate RTP packets. RTP packets having the same timestamp are then grouped into the same RTP sample with a sample header indicating the number of samples. One GSM audio sample is contained in one RTP packet.

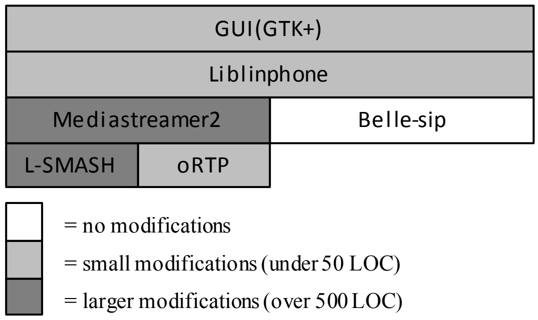


Fig. 1. Linphone libraries.

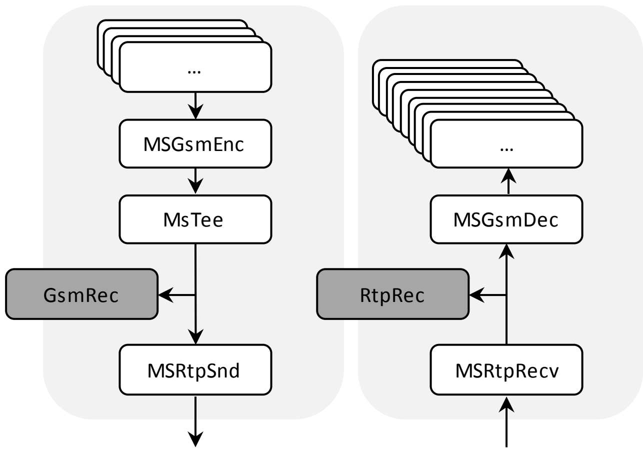


Fig. 2. Mediastreamer2 library: audio filter graph.

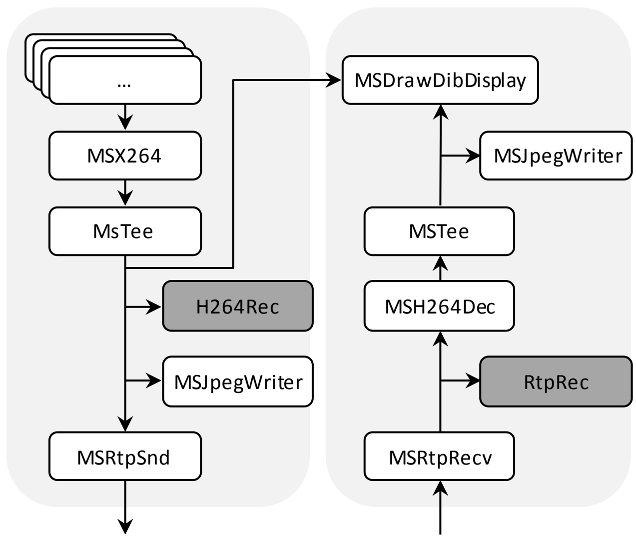


Fig. 3. Mediastreamer2 library: video filter graph.

The RTP Reception Hint Tracks involves a constructor (RTPsampleconstructor in [4]) for each packet. RTP Reception Hint Tracks are not synchronized during recording but synchronization is done during playback using the recorded RTCP Reception Hint Tracks.

A video call being sent to the other end is recorded as two media tracks, one track for audio and the other for video. Media track recording is implemented by new audio (Fig. 2) and video (Fig. 3) recording filters that capture the stream before it is sent.

B. *Adaptation of L-SMASH*

Linphone lacks the functionality to encapsulate media data into an MP4 file. Therefore, L-SMASH was adopted for multiplexing. L-SMASH is an MP4 writer library written in C. It supports a wide variety of codecs. L-SMASH can create multiple tracks inside an MP4 file and samples are appended individually to a particular track. The MP4 header data is written after the recording is finished.

This work extends L-SMASH with support for RTP/RTCP Reception Hint Tracks. The modified version of L-SMASH library can also be used to record RTP/RTCP Reception Hint Tracks in other RTP applications.

IV. Video Call Playback

VLC media player reads the MP4 file containing the RTP Reception Hint Tracks. It demultiplexes RTP Reception Hint Tracks to elemental streams and synchronizes them using corresponding RTCP Reception Hint Tracks.

A. *Proposed implementation in VLC*

In the proposed VLC extension, the playback of RTP Reception Hint Track is implemented as suggested in [4] and [8]. Fig. 4 shows the main modules of VLC file processing. An MP4 file is first read by an *access* module which in this case is the filesystem. Then, the *stream* module takes care of the byte order. In this case, a *demux* module represents an MP4 demultiplexer after which a *decoder* module decodes the stream. An *output* module includes the video or audio filter modules and the modules for displaying the graphical elements.

Changes to the VLC player were limited to the demux module where the playback of RTP Reception Hint Track is implemented. The implementation strips away the sample structure of RTP Reception Hint Tracks. It also converts the payload to a media sample which is passed on to the decoder as an elemental stream.

The first step to add support for RTP Reception Hint Track is initializing data needed for playback of a track. An SDP message is also parsed to find out whether the RTP Reception Hint Track contains audio or video. Next, an elemental stream is created. This can be done by parsing the rest of the SDP message to get the coding format of the stream. Finally, each sample header, RTP packet, and constructor are stripped from sample after which the RTP packet payload is sent forward as a media sample. In case of H.264 video, all RTP payloads of one sample are combined to a single frame.

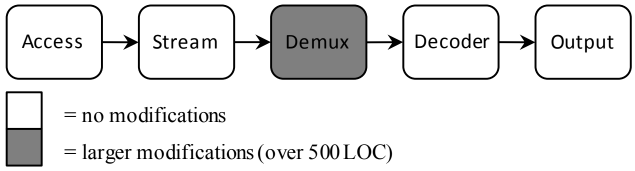
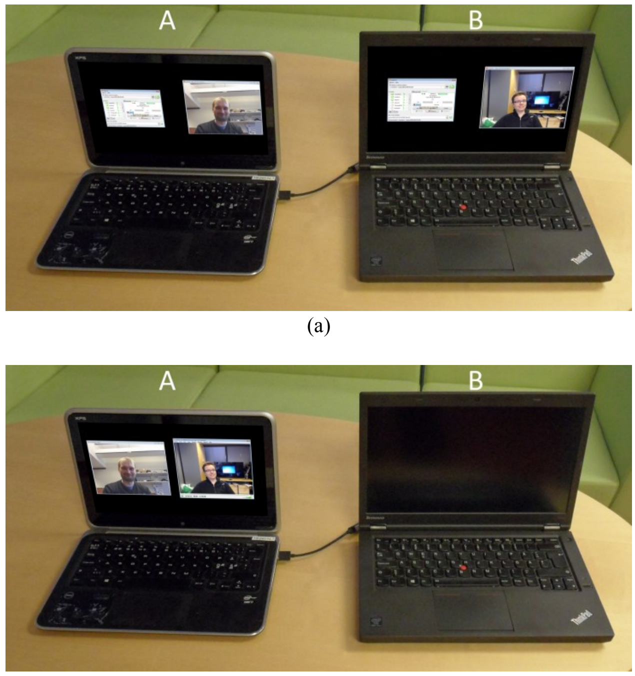


Fig. 4. The general architecture of VLC media player.



(b)

Fig. 5. A two-way video call in the proposed system. (a) Recording. (b) Playback.

There is a random offset in the first samples of a track due to the arbitrary starting point of the recording. The offset is synchronized using the NTP timestamps of RTCP sender report to make sure both tracks start simultaneously.

V. Overall System

Fig. 5 presents two snapshots of the proposed video call system with two participants (*A* and *B*). In Fig. 5(a), the participant *A* is recording a two-way Linphone video call. Linphone creates an MP4 file using the L-SMASH library. The MP4 file includes RTP/RTCP Reception Hint Tracks for incoming GSM audio/H.264 video and media tracks for outgoing GSM audio/H.264 video. The tracks are played back by VLC media player as shown Fig. 5(b) where only the playback functionality of the instance *A* is depicted. The respective block diagram of this proof-of-concept architecture is shown in Fig. 6.

The proposed system was benchmarked with three video call tests lasting 1 min, 10 min, and 60 min. In these tests, the computer *A* was equipped with Intel i7-3537U (2 cores, 2.0 GHz) processor and 8 GB of memory whereas computer *B* had Intel i7-4800MQ (4 cores, 2.7 GHz) processor and 16 GB of memory. The recorded video format was 800 × 600. The

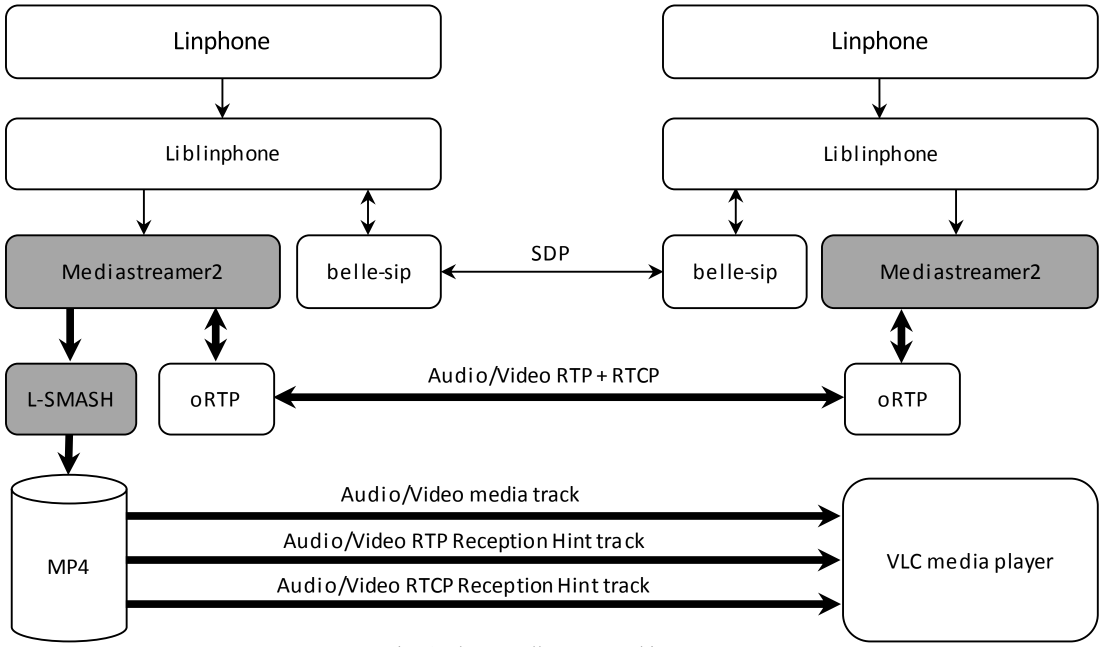


Fig. 6. The overall system architecture.

internal measurement tools of Linphone were used to measure the CPU time of recording RTP Reception Hint Tracks (RtpRec). According to our measurements, the execution of RtpRec accounted for under 1% (0.5-0.9%) of the total Linphone CPU time. That is, the overhead of our proposal is diminutive even if the implementation was not optimized for performance.

The bitrate cost caused by RTP Reception Hint Tracks depends on media encoding parameters and is proportional to the size of individual RTP packets. For a video call, the bitrate increase relative to the bitrate of the RTP packets is under 2%. This comes from sample structures detailed in Section 2.

In addition to GSM codec, the Speex codec was tested and RTP Reception Hint Track recording worked without any modifications to Linphone or L-SMASH, but VLC needed small modifications to recognize a different codec from an SDP message. Hence, new codecs could be added in the proposed system by modifying playback only.

VI. Conclusions

This paper shows an end-to-end video call system that deploys RTP and RTCP Reception Hint Tracks in video call recording and playback. The proposed setup uses Linphone and L-SMASH to record the video call in MP4 format and VLC media player to play the recorded MP4 file. The proposed proof-of-concept system validates the feasibility and benefits of the RTP/RTCP Reception Hint Tracks, so augmenting a video call capture with them is recommended in RTP based traffic, provided that a compatible player is available. VLC extension for RTP Reception Hint Track was merged to VLC 3.0.0 version, which facilitates a broader deployment of our proposal.

In the future, the proposed system could be extended with three new features. Firstly, the RTP Reception Hint Tracks could be used to detect possible packet losses during recording. Secondly, an answer machine for video calls could be implemented by allowing a sender to re-create and send the original RTP streams. Finally, the RTCP Reception Hint Tracks could be used to correct potential clock drift during playback by using the NTP timestamps in RTCP sender reports. The listener could then choose whether to play the file as it was received or as it was sent.

Acknowledgment

This work was supported in part by the Finnish Funding Agency for Innovation (TEKES) and the Academy of Finland (decision no. 301820).

References

[1] *IETF RFC 3550 RTP: A Transport Protocol for Real-Time Applications* [Online]. Available: https://www.ietf.org/rfc/rfc3550.txt

[2] Skype [Online]. Available: http://www.skype.com/

[3] *How can I record my Skype calls*? [Online]. Available: https://support.skype.com/en/faq/FA12395/how-can-i-record-my-skype-calls

[4] ISO Base Media File Format, document ISO/IEC 14496-12 and ISO/IEC 15444-12, ISO/IEC, Jul 2012.

[5] *Linphone open-source voip software* [Online]. Available: http://www.linphone.org/

[6] *L-SMASH* [Online]. Available: https://github.com/l-smash/l-smash

[7] *VLC media player* [Online]. Available: http://www.videolan.org/vlc/index.html

[8] M. M. Hannuksela, S. Döhla, and K. Murray, "The DVB file format," IEEE Signal Process. Mag., vol. 29, no. 2, Mar. 2012, pp. 148-153.

[9] *The GTK+ project* [Online]. Available: http://www.gtk.org/

[10] *IETF RFC 4566 SDP: Session Description Protocol* [Online]. Available: https://www.ietf.org/rfc/rfc4566.txt

[11] *IETF RFC 3261 SIP: Session Initiation Protocol* [Online]. Available: https://www.ietf.org/rfc/rfc3261.txt