

AVTEXT Working Group
Internet-Draft
Intended status: Informational
Expires: August 30, 2011

J. Xia
Huawei
February 26, 2011

Content Splicing for RTP Sessions
draft-xia-avtext-splicing-for-rtp-00

Abstract

This memo outlines RTP splicing. Splicing is a process that replaces the content of the main multimedia stream with other multimedia content, and delivers the substitutive multimedia content to receiver for a period of time. This memo discusses the requirements of splicing process on RTP layer, thus provides guideline for how a RTP Translator works to satisfy these requirements.

Status of this Memo

This Internet-Draft is submitted to IETF 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 August 30, 2011.

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. Terminology 3
- 3. RTP Splicing Discussion and Requirements 4
- 4. Using RTP Translator for RTP Splicing 6
- 5. Processing Splicing on RTP Translator 7
- 6. Implementation considerations 8
- 7. Security Considerations 9
- 8. IANA Considerations 9
- 9. Acknowledgments 9
- 10. Change Log 9
 - 10.1. draft-xia-avtext-splicing-for-rtp-00 9
- 11. Normative References 10
- Author's Address 11

1. Introduction

This document outlines how splicing can be used for RTP sessions. Splicing is a process that replaces the content of the main multimedia stream with other multimedia content, and delivers the substitutive multimedia content to receiver for a period of time. The substitutive content can be provided for example via another RTP stream or local media file storage.

One representative use case for splicing is targeted advertisements insertion, which allows operators to replace a national advertising slot with its own regional advertising content prior to providing to receiver. So far [SCTE30] and [SCTE35] have standardized MPEG2-TS splicing running over cable, but to date there is no guidelines for how to use the RTP Translator functionality defined in [RFC3550] to implement an RTP Splicer.

In this document, we describe the basic requirements of RTP splicing, then analyze how an RTP Translator can be used to implement RTP splicing to satisfy these requirements from the aspect of feasibility, implementation complexity and backward compatibility.

2. Terminology

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

Current RTP Stream

The RTP stream that the RTP receiver is currently receiving. The content of current RTP stream can be either main content or substitutive content.

Main RTP Stream

The RTP stream that the Splicer is receiving. The content of main RTP stream can be replaced by substitutive content for a period of time.

Main Content

The multimedia content that are conveyed in main RTP stream. main content will be replaced by the substitutive content during splicing period.

Substitutive Content

The multimedia content that replaces the content of main RTP stream during splicing period. The substitutive content can for example be contained in an RTP stream from a media source or fetched from local media file storage.

Insertion RTP Stream

A RTP stream that may provide substitutive content. If the substitutive content is provided from insertion RTP stream, the insertion RTP Stream must be terminated on Splicer.

Splicer

An intermediary node that inserts substitutive content into main RTP stream. Splicer sends substitutive content to RTP receiver as the payload of the current RTP stream.

3. RTP Splicing Discussion and Requirements

In this document, we assume an intermediary network element, which is referred to as Splicer, to play the key role to handle RTP splicing. When RTP splicing begins, Splicer replaces the main content in the main RTP stream with substitutive content and conveys the substituted RTP stream to RTP receiver for a period of time, then resumes the main content when RTP splicing finishes. The RTP splicing may happen more than once if substitutive content will be inserted in multiple time segments during the lifetime of the main RTP session.

To realize a seamless splicing on RTP receiver, Splicer must not cause any perceptible media clipping at the splicing joint. Therefore, main media source must reserve specific time slot in the main RTP stream for splicing and notify Splicer this specific time slot information. The substitutive content time length SHOULD be the same as the reserved time slot length. The details about how the specific time slot information is conveyed to Splicer are outside the scope of this memo.

Besides the media clipping avoidance, there are also a set of concrete requirements that must be satisfied on Splicer:

REQ-1:

Splicer MUST operate in either unicast or multicast session environment.

REQ-2:

Splicer MUST guarantee content substitution process is invisible to RTP receiver to prevent the RTP receiver from easily identifying the substitutive content.

The RTP packets whose payloads are replaced by substitutive content are required to keep their RTP header information to be consistent with those of main RTP packets whose payloads are unaltered:

The value of SSRC field in RTP header MUST be same as the value of corresponding field in main content RTP header, while the value of payload type field SHOULD be same as the value of corresponding field in main content RTP header. Note that in some cases, it may be necessary to transcode the substitutive content to ensure the payload type is the same, and that this may be prohibitively expensive, so it might be acceptable to leave the payload untranscoded.

The value of sequence number field in RTP header MUST be contiguous regardless of whether the substitutive content is inserted or not. It should be noted that the number of packets in the substitutive sequence number range may be different from the number of packets in the overridden main sequence number range due to the different characteristics or entropy coding.

REQ-3:

Splicer SHOULD ensure that the main media source can learn the real performance of the path between media source and downstream receiver for adaptation purpose.

REQ-4:

Splicer MUST be backward compatible with basic characteristics of [RFC3550], e.g., SSRC identifier collision resolution and loop detection.

REQ-5:

Splicer MUST have the capability to correctly handle any packet loss events regardless of whether the lost content is main content or substitutive content.

REQ-6:

If substitutive content comes from an insertion RTP stream, Splicer MUST terminate this stream, in such case, Splicer should generate RTCP reports upstream towards the insertion media source using its own SSRC.

4. Using RTP Translator for RTP Splicing

An RTP Translator that acts as media transcoder can replace the payloads of the main RTP packets with the substitutive content, and can assign new sequence numbers to the substituted packets with main RTP packets SSRC identifier intact. Note that the new sequence numbers of substituted RTP packets must seamlessly follow the sequence numbers of the previous main RTP packets. When splicing ends, Translator must switch back to the main RTP stream and output it to RTP receiver until next splicing occurs.

With respect to RTCP flow, Translator acting as a media transcoder will need to interpose itself into the RTCP flow as specified in section 7.2 of [RFC3550]. However, the substitutive content might have different characteristics compared to the main content it replaces. As a result, the translated RTCP Receiver Reports received by the main media source might be somewhat misleading for adaptation purposes, since they relate to other content that the main media source cannot control during the splicing period. Fortunately Translator has the capability to act as quality monitor and generate RTCP reports upstream towards main media source with its own SSRC thus reflecting the real characteristics of the main RTP stream and the reception quality on the Translator. These RTCP reports, as well as the translated RTCP reports sent from the downstream receiver, can provide main media source the general performance of the different segments of the path between main media source and RTP receiver. If the substitutive content is fetched from the insertion RTP stream, Translator acting as RTP receiver should generate RTCP receiver reports upstream towards the insertion media source to reflect the reception quality of the insertion RTP stream on the Translator.

When RTP receiver detects any packet loss during splicing, it may generate RTCP NACK message for packet loss recovery [RFC4585]. If Translator receives any RTCP NACK message from RTP receiver, Translator first needs to determine the sequence number range of lost packets requested by RTP receiver. Since parts of lost packets may contain substitutive content which does not relate to the main RTP stream, the translated RTCP NACK message might be somewhat misleading for packet loss recovery. Thus, when Translator is intercepting RTCP NACK packets, it should only pass those RTCP NACK packets that relate

to the relevant content upstream.

5. Processing Splicing on RTP Translator

Once Translator has learnt when to process splicing from main RTP source, it must get ready for the coming splicing in advance, e.g., fetches the substitutive content either from local media file storage or via insertion RTP stream. If the substitutive content comes from the insertion RTP stream, Translator must act as a RTP receiver and terminate this insertion RTP stream.

First the Translator should guarantee the media encoding of substitutive content to be same as the media encoding of main content. If the substitutive content uses other media encoding, Translator should perform media transcoding procedure for substitutive content.

When splicing begins, Translator replaces the main content with the substitutive content as if the substitutive content were sent from the main media source.

When splicing ends, Translator must switch back to main RTP stream, but since the number of packets in the substitutive sequence number may be different from the number of packets in the overridden main sequence number range, Translator still needs to modify the sequence numbers of subsequent main RTP packets prior to outputting them to RTP receiver even if no media transcoding occurs on main RTP stream. For subsequent current RTP packets, its start sequence number should be determined as being one after the end sequence number of previous substitutive RTP packets.

This is perhaps best explained by a pseudo code example:

```
when (splicing begin) {  
  
    the sequence number of the ith substitutive packet = the sequence  
    number of last main packet + i, until the splicing end;  
  
}  
  
when (splicing end) {  
  
    the sequence number of the following kth current packet = the  
    sequence number of last substitutive packet + k, until the next  
    splicing begin;  
  
}
```

With regard to RTCP packets, Translator needs to rewrite RTCP Report. The "sender's byte count" field and the "sender's packet count" field in RTCP Sender Report should be changed. Translator then determines the proportion, P , of the number of packets Translator receives from main media source (numRev) to the number of packets Translator sends to RTP receiver (numSend) during a regular RTCP Reporting interval such that $P = \text{numRev} / \text{numSend}$. It should be noted that the value of P may be not fixed specially at the splicing joint, where the RTCP Reports may reflect the characteristics of the combination of main RTP packets and adjacent substitutive RTP packets. Once the proportion, P , is counted, the values of packet loss fields and the "extended last sequence number" field in RTCP Sender Report are scaled by dividing each of them by P , and reverse rewriting is made to the values of the corresponding fields in RTCP Receiver Report, i.e., multiplying each them by P . Meanwhile, Translator must generate RTCP Receiver Report upstream towards main media source using its own SSRC, reflecting the reception quality of the main RTP stream on the Translator. If the substitutive content comes from the insertion RTP stream, Translator acting as receiver must generate RTCP Receiver Report upstream towards insertion media source using its own SSRC.

If Translator receives any RTCP NACK message from RTP receiver for packet loss recovery, it first determines if the lost packets contain substitutive content. It is somewhat more complex that the lost packets requested in a single RTCP NACK message contain not only main content but also substitutive content. To address this, Translator must rewrite the translated RTCP NACK message into two separate RTCP NACK messages: one only requests the lost packets that contain main content, and is forwarded using the SSRC of that RTP receiver; and another only requests the lost packets that contain substitutive content, and is sent using the SSRC of the Translator if the substitutive content is delivered via insertion RTP stream, or fetching the lost substitutive content directly from corresponding local media file storage.

6. Implementation considerations

When Translator is used to handle RTP splicing, RTP receiver does not need any RTP/RTCP extension for splicing. As a trade-off, additional overhead could be induced on Translator compared to just translating the main RTP stream, since Translator must coordinate the switch between main content and substitutive content, and generate its own RTCP reports. Even if no new substitutive content will be inserted, the Translator still needs to modify the main RTP packets, and to recalculate the UDP/IP checksum until the main RTP session ends. If Translator serves multiple main RTP streams simultaneously, this may lead to large overhead on the Translator.

7. Security Considerations

The security considerations of the RTP specification [RFC3550], the Extended RTP profile for RTCP-Based Feedback [RFC4585], and the Secure Real-time Transport Protocol [RFC3711] apply. Translator must be trusted by main media source and insertion media source, and must be included in the security context.

8. IANA Considerations

No IANA actions are required.

9. Acknowledgments

The following individuals have reviewed the earlier versions of this specification and provided very valuable comments: Colin Perkins, Magnus Westerlund, Roni Even, Tom Van Caenegem, Joerg Ott, David R Oran, Cullen Jennings, Ali C Begen, and Ning Zong.

10. Change Log

10.1. draft-xia-avtext-splicing-for-rtp-00

The following are the major changes compared to previous version 00:

- o Change primary RTP stream to main RTP stream, add current RTP stream as the streaming received by RTP receiver.
- o Eliminate the ambiguity of inserted content with substitutive content which replaces the main content rather than pause it.
- o Clarify the signaling requirements.
- o Delete the description on Mixer and MCU in section 4, mainly focus on the direction whether a Translator can act as a Splicer.
- o Add section 5 to describe the exact guidance on how an RTP Translator is used to handle splicing.
- o Modify the security considerations section and add acknowledges section.

11. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2250] Hoffman, D., Fernando, G., Goyal, V., and M. Civanlar, "RTP Payload Format for MPEG1/MPEG2 Video", RFC 2250, January 1998.
- [RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.
- [RFC3551] Schulzrinne, H. and S. Casner, "RTP Profile for Audio and Video Conferences with Minimal Control", STD 65, RFC 3551, July 2003.
- [RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, March 2004.
- [RFC4585] Ott, J., Wenger, S., Sato, N., Burmeister, C., and J. Rey, "Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/AVPF)", RFC 4585, July 2006.
- [RFC5117] Westerlund, M. and S. Wenger, "RTP Topologies", RFC 5117, January 2008.
- [RFC5760] Ott, J., Chesterfield, J., and E. Schooler, "RTP Control Protocol (RTCP) Extensions for Single-Source Multicast Sessions with Unicast Feedback", RFC 5760, February 2010.
- [I-D.ietf-avt-ecn-for-rtp]
Westerlund, M., "Explicit Congestion Notification (ECN) for RTP over UDP", draft-ietf-avt-ecn-for-rtp-03 (work in progress), October 2010.
- [SCTE30] Society of Cable Telecommunications Engineers (SCTE), "Digital Program Insertion Splicing API", 2001.
- [SCTE35] Society of Cable Telecommunications Engineers (SCTE), "Digital Program Insertion Cueing Message for Cable", 2004.
- [H.323] ITU-T Recommendation H.323, "Packet-based multimedia communications systems", June 2006.

Author's Address

Jinwei Xia
Huawei
Hui Hong Mansion
Nanjing, Baixia District 210001
China

Phone: +86-025-86622310
Email: xiajinwei@huawei.com

