AVTEXT Working Group J. Xia
INTERNET-DRAFT R. Even
Intended Status: Standards Track R. Huang
Expires: February 4, 2017 Huawei
L. Deng
China Mobile
August 3, 2016

RTP/RTCP extension for RTP Splicing Notification
draft-ietf-avtext-splicing-notification-09

Abstract

Content splicing is a process that replaces the content of a main multimedia stream with other multimedia content, and delivers the substitutive multimedia content to the receivers for a period of time. The splicer is designed to handle RTP splicing and needs to know when to start and end the splicing.

This memo defines two RTP/RTCP extensions to indicate the splicing related information to the splicer: an RTP header extension that conveys the information in-band and an RTCP packet that conveys the information out-of-band.

Status of this Memo

Table of Contents

1 Introduction

Splicing is a process that replaces some multimedia content with other multimedia content and delivers the substitutive multimedia content to the receivers for a period of time. In some predictable splicing cases, e.g., advertisement insertion, the splicing duration needs to be inside of the specific, pre-designated time slot. Certain timing information about when to start and end the splicing must be first acquired by the splicer in order to start the splicing. This document refers to this information as the Splicing Interval.

[SCTE35] provides a method that encapsulates the Splicing Interval inside the MPEG2-TS layer in cable TV systems. When transported in RTP, an middle box designed as the splicer to decode the RTP packets and search for the Splicing Interval inside the payloads is required. The need for such processing increases the workload of the middle box and limits the number of RTP sessions the middle box can support.

The document defines an RTP header extension [RFC5285bis] used by the main RTP sender to provide the Splicing Interval by including it in the RTP packets.

However, the Splicing Interval conveyed in the RTP header extension might not reach the splicer successfully. Any splicing un-aware middlebox on the path between the RTP sender might strip this RTP header extension.

To increase robustness against such case, the document also defines a new RTCP packet type to carry the same Splicing Interval to the splicer. Since RTCP is also unreliable and may not be so immediate as the in-band way, it's only considered as a complement to the RTP header extension.

1.1 Terminology

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 [RFC2119].

In addition, we define following terminologies:

Main RTP sender:

The sender of RTP packets carrying the main RTP stream.

Splicer:

An intermediary node that inserts substitutive content into a main

RTP stream. The splicer sends substitutive content to the RTP receiver instead of the main content during splicing. It is also responsible for processing RTCP traffic between the RTP sender and the RTP receiver.

Splicing-In Point

A virtual point in the RTP stream, suitable for substitutive content entry, typically in the boundary between two independently decodable frames.

Splicing-Out Point

A virtual point in the RTP stream, suitable for substitutive content exit, typically in the boundary between two independently decodable frames.

Splicing Interval:

The NTP-format timestamps, representing the main RTP sender wallclock time, for the Splicing-In point and Splicing-Out point per [RFC6828] allowing the splicer to know when to start and end the RTP splicing.

Substitutive RTP Sender:

The sender of RTP packets carrying the RTP stream that will replace the content in the main RTP stream.

2 Overview

2.1 Overview of RTP Splicing

RTP Splicing is intended to replace some multimedia content with certain substitutive multimedia content, and then forward it to the receivers for a period of time. This process is authorized by the main RTP sender that offers a specific time window for inserting the substitutive multimedia content in the main content. A typical usage is that IPTV service provider uses its own regional advertising content to replace national advertising content, the time window of which is explicitly indicated by the IPTV service provider.

The splicer is a middlebox handling RTP splicing. It receives main content and substitutive content simultaneously but only chooses to send one of them to the receiver at any point of time. When RTP splicing begins, the splicer sends the substitutive content to the receivers instead of the main content. When RTP splicing ends, the splicer switches back to sending the main content to the receivers.

This implies that the receiver is explicitly configured to receive the traffic via the splicer, and will return any RTCP feedback to it in the presence of the splicer.

The middlebox working as the splicer can be implemented as either an RTP mixer or as an RTP translator. If implemented as an RTP mixer, the splicer will use its own SSRC, sequence number space, and timing model when generating the output stream to receivers, using the CSRC list to indicate whether the original or substitutive content is being delivered. The splicer, on behalf of the content provider, can omit the CSRC list from the RTP packets it generates. This simplifies the design of the receivers, since they don't need to parse the CSRC list, but makes it harder to determine when the splicing is taking place (it requires inspection of the RTP payload data, rather than just the RTP headers). A splicer working as an RTP mixer splits the flow between the sender and receiver into two, and requires separate control loops, for RTCP and congestion control. [RFC6828] offers an example of an RTP mixer approach.

A splicer implemented as an RTP translator [RFC3550] will forward the RTP packets from the original and substitutive senders with their SSRCs intact, but will need to rewrite RTCP sender report packets to account for the splicing. In this case, the congestion control loops run between original sender and receiver, and between the substitutive sender and receiver. The splicer needs to ensure that the RTCP feedback message from the receiver are passed to the right sender to let the congestion control work.

2.2 Overview of Splicing Interval

To handle splicing on the RTP layer at the reserved time slots set by the main RTP sender, the splicer must first know the Splicing Interval from the main RTP sender before it can start splicing.

When a new splicing is forthcoming, the main RTP sender needs to send the Splicing Interval to the splicer. The Splicing Interval SHOULD be sent by RTP header extension or RTCP extension message more than once to mitigate the possible packet loss. To enable the splicer to get the substitutive content before the splicing starts, the main RTP sender MUST send the Splicing Interval far ahead. For example, the main RTP sender can estimate when to send the Splicing Interval based on the round-trip time (RTT) following the mechanisms in section 6.4.1 of [RFC3550] when the splicer sends RTCP RR to the main sender.

The substitutive sender also needs to learn the Splicing Interval from the main RTP sender in advance, and thus estimates when to transfer the substitutive content to the splicer. The Splicing Interval could be transmitted from the main RTP sender to the

substitutive content using some out-of-band mechanisms, for example, a proprietary mechanism to exchange the Splicing Interval, or the substitutive sender is implemented together with the main RTP sender inside a single device. To ensure the Splicing Interval is valid for both the main RTP sender and the substitutive RTP sender, the two senders MUST share a common reference clock so that the splicer can achieve accurate splicing. The requirements for the common reference clock (e.g. resolution, skew) depend on the codec used by the media content.

In this document, the main RTP sender uses a pair of NTP-format timestamps, to indicate when to start and end the splicing to the splicer: the timestamp of the first substitutive RTP packet at the splicing in point, and the timestamp of the first main RTP packet at the splicing out point.

When the substitutive RTP sender gets the Splicing Interval, it must prepare the substitutive stream. The main and the substitutive content providers MUST ensure that the RTP timestamp of the first substitutive RTP packet that would be presented to the receivers corresponds to the same time instant as the former NTP-format timestamp in the Splicing Interval. To enable the splicer to know the first substitutive RTP packet it needs to send, the substitutive RTP sender MUST send the substitutive RTP packet ahead of the Splicing In point, allowing the splicer to find out the timestamp of this first RTP packet in the substitutive RTP stream, e.g., using a prior RTCP SR (Sender Report) message.

When the splicing will end, the main content provider and the substitutive content provider MUST ensure the RTP timestamp of the first main RTP packet that would be presented on the receivers corresponds to the same time instant as the latter NTP-format timestamp in the Splicing Interval.

3 Conveying Splicing Interval in RTP/RTCP extensions

This memo defines two backwards compatible RTP extensions to convey the Splicing Interval to the splicer: an RTP header extension and an RTCP splicing notification message.

3.1 RTP Header Extension

The RTP header extension mechanism defined in [RFC5285bis] can be adapted to carry the Splicing Interval consisting of a pair of NTP-format timestamps.

This RTP header extension carries the 7 octets splicing-out NTP-format timestamp (lower 24-bit part of the Seconds of a NTP-format timestamp and the 32 bits of the Fraction of a NTP-format timestamp as defined in [RFC5905]), followed by the 8 octets splicing-in NTP-format timestamp (64-bit NTP-format timestamp as defined in [RFC5905]). The top 8 bits of the splicing-out NTP timestamp are inferred from the top 8 bits of the splicing-in NTP timestamp, under the assumption that the splicing-out time is after the splicing-in time, and the splicing interval is less than 2^25 seconds. Therefore, if the value of 7 octets splicing-out NTP-format timestamp is smaller than the value of 7 lower octets splicing-in NTP-format, it implies a wrap of the 56-bit splicing-out NTP-format timestamp which means the top 8-bit value of the 64-bit splicing-out is equal to the top 8-bit value of splicing-in NTP Timestamp plus 0x01. Otherwise, the top 8 bits of splicing-out NTP timestamp is equal to the top 8 bits of splicing-in NTP Timestamp.

This RTP header extension can be encoded using either the one-byte or two-byte header defined in [RFC5285bis]. Figure 1 and 2 show the splicing interval header extension with each of the two header formats.

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+E
   |  ID   | L=14  | OUT NTP timestamp format - Seconds (bit 8-31) |x
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+t
   |      OUT NTP timestamp format - Fraction (bit 0-31)          |e
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+n
   |       IN  NTP timestamp format - Seconds (bit 0-31)          |s
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+i
   |       IN  NTP timestamp format - Fraction (bit 0-31)         |o
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+n
```

Figure 1: Splicing Interval
Using the One-Byte Header Format


```
    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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+E
|   ID          |    L=15       | OUT NTP timestamp  - Seconds  |x
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+t
|Out Secds(cont)|      OUT NTP timestamp format - Fraction      |e
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+n
|Out Fract(cont)|      IN  NTP timestamp format - Seconds       |s
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+i
| In Secds(cont)|     IN  NTP timestamp format - Fraction       |o
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+n
| In Fract(cont)| 0 (pad)       |              ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: Splicing Interval
Using the Two-Byte Header Format

Since the inclusion of an RTP header extension will reduce the efficiency of RTP header compression, it is RECOMMENDED that the main sender inserts the RTP header extensions into only a number of RTP packets, instead of all the RTP packets, prior to the splicing in.

After the splicer obtains the RTP header extension and derives the Splicing Interval, it generates its own stream and is not allowed to include the RTP header extension in outgoing packets to reduce header overhead.

3.2 RTCP Splicing Notification Message

In addition to the RTP header extension, the main RTP sender includes the Splicing Interval in an RTCP splicing notification message. Whether or not the timestamps are included in the RTP header extension, the main RTP sender MUST send the RTCP splicing notification message. This provide robustness in the case where a middlebox strips RTP header extensions. The main RTP sender MUST make sure the splicing information contained in the RTCP splicing notification message consistent with the information included in the RTP header extensions.

The RTCP splicing notification message is a new RTCP packet type. It has a fixed header followed by a pair of NTP-format timestamps:

```
  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
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |V=2|P|reserved |    PT=TBA     |             length            |
```

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            SSRC                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             IN NTP Timestamp (most significant word)          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             IN NTP Timestamp (least significant word)         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             OUT NTP Timestamp (most significant word)         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             OUT NTP Timestamp (least significant word)        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: RTCP Splicing Notification Message

The RSI packet includes the following fields:

Length: 16 bits

As defined in [RFC3550], the length of the RTCP packet in 32-bit words minus one, including the header and any padding.

SSRC: 32 bits

The SSRC of the Main RTP Sender.

Timestamp: 64 bits

Indicates the wallclock time when this splicing starts and ends. The full-resolution NTP-format timestamp is used, which is a 64-bit, unsigned, fixed-point number with the integer part in the first 32 bits and the fractional part in the last 32 bits. This format is same as the NTP timestamp field in the RTCP Sender Report (Section 6.4.1 of [RFC3550]).

The RTCP splicing notification message can be included in the RTCP compound packet together with RTCP SR generated at the main RTP sender, and hence follows the compound RTCP rules defined in Section 6.1 in [RFC3550].

If the use of non-compound RTCP [RFC5506] was previously negotiated between the sender and the splicer, the RTCP splicing notification message may be sent as non-compound RTCP packets. In some cases that the mapping from RTP timestamp to NTP timestamp changes, e.g., clock drift happening before the splicing event, it may be required to send

RTCP SR or even updated Splicing Interval information timely to update the timestamp mapping for accurate splicing.

Since the RTCP splicing notification message is intentionally sent by the main RTP sender to the splicer, the splicer is not allowed to forward this message to the receivers so as to avoid their useless processing and additional RTCP bandwidth consumption in the downstream.

4 Reducing Splicing Latency

When splicing starts or ends, the splicer outputs the multimedia content from another sender to the receivers. Given that the receivers must first acquire certain information ([RFC6285] refers to this information as Reference Information) to start processing the multimedia data, either the main RTP sender or the substitutive sender SHOULD provide the Reference Information together with its multimedia content to reduce the delay caused by acquiring the Reference Information. The methods by which the Reference Information is distributed to the receivers is out of scope of this memo.

Another latency element is synchronization caused delay. The receivers must receive enough synchronization metadata prior to synchronizing the separate components of the multimedia streams when splicing starts or ends. Either the main RTP sender or the substitutive sender SHOULD send the synchronization metadata early enough so that the receivers can play out the multimedia in a synchronized fashion. The main RTP sender or the substitutive sender can estimate when to send the synchronization metadata based on, for example, the round-trip time (RTT) following the mechanisms in section 6.4.1 of [RFC3550] when the splicer sends RTCP RR to the main sender or the substitutive sender. The main RTP sender and the substitutive sender can also be coordinated by some proprietary out-of-band mechanisms to decide when and whom to send the metadata. If both send the information, the splicer SHOULD pick one based on the current situation, e.g., choosing main RTP sender when synchronizing the main media content while choosing the information from the substitutive sender when synchronizing the spliced content. The mechanisms defined in [RFC6051] are RECOMMENDED to be adopted to reduce the possible synchronization delay.

5 Failure Cases

This section examines the implications of losing RTCP splicing notification message and the other failure case, e.g., the RTP header extension is stripped on the path.

Given that there may be a splicing un-aware middlebox on the path between the main RTP sender and the splicer, the main and the substitutive RTP senders can use one heuristic to verify whether or not the Splicing Interval reaches the splicer.

The splicer can be implemented to have its own SSRC, and send RTCP reception reports to the senders of the main and substitutive RTP streams. This allows the senders to detect problems on the path to the splicer. Alternatively, it is possible to implement the splicer such that it has no SSRC, and does not send RTCP reports; this prevents the senders from being able to monitor the quality to the path to the splicer.

If the splicer has an SSRC and sends its own RTCP reports, it can choose not to pass RTCP reports it receives from the receivers to the senders. This will stop the senders from being able to monitor the quality of the paths from the splicer to the receivers.

A splicer that has an SSRC can choose to pass RTCP reception reports from the receivers back to the senders, after modifications to account for the splicing. This will allow the senders the monitor the quality of the paths from the splicer to the receivers. A splicer that does not have its own SSRC has to forward and translation RTCP reports from the receiver, otherwise the senders will not see any receivers in the RTP session.

If the splicer is implemented as a mixer, it will have its own SSRC and will send its own RTCP reports, and will forward translated RTCP reports from the receivers.

Upon the detection of a failure, the splicer can communicate with the main sender and the substitutive sender in some out of band signaling ways to fall back to the payload specific mechanisms it supports, e.g., MPEG-TS splicing solution defined in [SCTE35], or just abandon the splicing.

6 Session Description Protocol (SDP) Signaling

This document defines the URI for declaring this header extension in an extmap attribute to be "urn:ietf:params:rtp-hdrext:splicing-interval".

This document extends the standard semantics defined in SDP Grouping Framework [RFC5888] with a new semantic: SPLICE to represent the relationship between the main RTP stream and the substitutive RTP stream. Only 2 m-lines are allowed in the SPLICE group. The main RTP stream is the one with the extended extmap attribute, and the other one is substitutive stream. A single m-line MUST NOT be included in different SPLICE groups at the same time. The main RTP sender provides the information about both main and substitutive sources.

The extended SDP attribute specified in this document is applicable for offer/answer content [RFC3264] and do not affect any rules when negotiating offer and answer. When used with multiple m-lines, substitutive RTP MUST be applied only to the RTP packets whose SDP m-line is in the same group with the substitutive stream using SPLICE and has the extended splicing extmap attribute. This semantic is also applicable for BUNDLE cases.

The following examples show how SDP signaling could be used for splicing in different cases.

6.1 Declarative SDP

```
v=0
o=xia 1122334455 1122334466 IN IP4 splicing.example.com
s=RTP Splicing Example
```

```
t=0 0
a=group:SPLICE 1 2
m=video 30000 RTP/AVP 100
i=Main RTP Stream
c=IN IP4 233.252.0.1/127
a=rtpmap:100 MP2T/90000
a=extmap:1 urn:ietf:params:rtp-hdrext:splicing-interval
a=mid:1
m=video 30002 RTP/AVP 100
i=Substitutive RTP Stream
c=IN IP4 233.252.0.2/127
a=sendonly
a=rtpmap:100 MP2T/90000
a=mid:2
```

Figure 3: Example SDP for a single-channel splicing scenario

The splicer receiving the SDP message above receives one MPEG2-TS stream (payload 100) from the main RTP sender (with multicast destination address of 233.252.0.1) on port 30000, and/or receives another MPEG2-TS stream from the substitutive RTP sender (with multicast destination address of 233.252.0.2) on port 30002. But at a particular point in time, the splicer only selects one stream and

outputs the content from the chosen stream to the downstream receivers.

6.2 Offer/Answer without BUNDLE

SDP Offer - from main RTP sender

```
v=0
o=xia 1122334455 1122334466 IN IP4 splicing.example.com
s=RTP Splicing Example
t=0 0
a=group:SPLICE 1 2
m=video 30000 RTP/AVP 31 100
i=Main RTP Stream
c=IN IP4 splicing.example.com
a=rtpmap:31 H261/90000
a=rtpmap:100 MP2T/90000
a=extmap:1 urn:ietf:params:rtp-hdrext:splicing-interval
```

```
a=sendonly
a=mid:1
m=video 40000 RTP/AVP 31 100
i=Substitutive RTP Stream
c=IN IP4 substitutive.example.com
a=rtpmap:31 H261/90000
a=rtpmap:100 MP2T/90000
a=sendonly
a=mid:2
```

SDP Answer - from splicer

```
v=0
o=xia 1122334455 1122334466 IN IP4 splicer.example.com
s=RTP Splicing Example
t=0 0
a=group:SPLICE 1 2
m=video 30000 RTP/AVP 100
i=Main RTP Stream
c=IN IP4 splicer.example.com
a=rtpmap:100 MP2T/90000
a=extmap:1 urn:ietf:params:rtp-hdrext:splicing-interval
a=recvonly
a=mid:1
m=video 40000 RTP/AVP 100
i=Substitutive RTP Stream
c=IN IP4 splicer.example.com
a=rtpmap:100 MP2T/90000
a=recvonly
a=mid:2
```


6.3 Offer/Answer with BUNDLE: All Media are spliced

In this example, the bundled audio and video media have their own substitutive media for splicing:

1. An Offer, in which the offerer assigns a unique address and a substitutive media to each bundled "m="line for splicing within the BUNDLE group.

2. An answer, in which the answerer selects its own BUNDLE address, and leave the substitutive media untouched.

SDP Offer - from main RTP sender

```
v=0
o=alice 1122334455 1122334466 IN IP4 splicing.example.com
s=RTP Splicing Example
c=IN IP4 splicing.example.com
t=0 0
a=group:SPLICE foo 1
a=group:SPLICE bar 2
a=group:BUNDLE foo bar
m=audio 10000 RTP/AVP 0 8 97
a=mid:foo
b=AS:200
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:97 iLBC/8000
a=extmap:1 urn:ietf:params:rtp-hdrext:splicing-interval
a=sendonly
m=video 10002 RTP/AVP 31 32
a=mid:bar
b=AS:1000
a=rtpmap:31 H261/90000
a=rtpmap:32 MPV/90000
a=extmap:2 urn:ietf:params:rtp-hdrext:splicing-interval
a=sendonly
m=audio 20000 RTP/AVP 0 8 97
i=Substitutive audio RTP Stream
c=IN IP4 substitive.example.com
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:97 iLBC/8000
a=sendonly
a=mid:1
m=video 20002 RTP/AVP 31 32
i=Substitutive video RTP Stream
```



```
c=IN IP4 substitive.example.com
a=rtpmap:31 H261/90000
a=rtpmap:32 MPV/90000
a=mid:2
```

```
a=sendonly
```

SDP Answer - from the splicer

```
v=0
o=bob 2808844564 2808844564 IN IP4 splicer.example.com
s=RTP Splicing Example
c=IN IP4 splicer.example.com
t=0 0
a=group:SPLICE foo 1
a=group:SPLICE bar 2
a=group:BUNDLE foo bar
m=audio 30000 RTP/AVP 0
a=mid:foo
b=AS:200
a=rtpmap:0 PCMU/8000
a=extmap:1 urn:ietf:params:rtp-hdrext:splicing-interval
a=recvonly
m=video 30000 RTP/AVP 32
a=mid:bar
b=AS:1000
a=rtpmap:32 MPV/90000
a=extmap:2 urn:ietf:params:rtp-hdrext:splicing-interval
a=recvonly
m=audio 30002 RTP/AVP 0
i=Substitutive audio RTP Stream
c=IN IP4 splicer.example.com
a=rtpmap:0 PCMU/8000
a=recvonly
a=mid:1
m=video 30004 RTP/AVP 32
i=Substitutive video RTP Stream
c=IN IP4 splicer.example.com
a=rtpmap:32 MPV/90000
a=mid:2
a=recvonly
```

6.4 Offer/Answer with BUNDLE: a Subset of Media are Spliced

In this example, the substitutive media only applies for video when splicing:

1. An Offer, in which the offerer assigns a unique address to each bundled "m="line within the BUNDLE group, and assigns a substitutive

media to the bundled video "m=" line for splicing.

2. An answer, in which the answerer selects its own BUNDLE address, and leave the substitutive media untouched.

SDP Offer - from the main RTP sender:

```
v=0
o=alice 1122334455 1122334466 IN IP4 splicing.example.com
s=RTP Splicing Example
c=IN IP4 splicing.example.com
t=0 0
a=group:SPLICE bar 2
a=group:BUNDLE foo bar
m=audio 10000 RTP/AVP 0 8 97
a=mid:foo
b=AS:200
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:97 iLBC/8000
a=sendonly
m=video 10002 RTP/AVP 31 32
a=mid:bar
b=AS:1000
a=rtpmap:31 H261/90000
a=rtpmap:32 MPV/90000
a=extmap:2 urn:ietf:params:rtp-hdrext:splicing-interval
a=sendonly
m=video 20000 RTP/AVP 31 32
i=Substitutive video RTP Stream
c=IN IP4 substitutive.example.com
a=rtpmap:31 H261/90000
a=rtpmap:32 MPV/90000
a=mid:2
a=sendonly
```

SDP Answer - from the splicer:

```
v=0
o=bob 2808844564 2808844564 IN IP4 splicer.example.com
s=RTP Splicing Example
c=IN IP4 splicer.example.com
t=0 0
a=group:SPLICE bar 2
a=group:BUNDLE foo bar
m=audio 30000 RTP/AVP 0
a=mid:foo
```

```
b=AS:200
```



```
a=rtpmap:0 PCMU/8000
a=recvonly
m=video 30000 RTP/AVP 32
a=mid:bar
b=AS:1000
a=rtpmap:32 MPV/90000
a=extmap:2 urn:ietf:params:rtp-hdrext:splicing-interval
a=recvonly
m=video 30004 RTP/AVP 32
i=Substitutive video RTP Stream
c=IN IP4 splicer.example.com
a=rtpmap:32 MPV/90000
a=mid:2
a=recvonly
```

7 Security Considerations

The security considerations of the RTP specification [RFC3550] and the general mechanism for RTP header extensions [RFC5285bis] apply. The splicer can either be a mixer or a translator, and all the security considerations of these two RTP intermediaries topologies described in [RFC7667] and [RFC7201] are applicable for the splicer.

The splicer replaces some content with other content in RTP packet, thus breaking any RTP-level end-to-end security, such as source authentication and integrity protection. End to end source authentication is not possible with any known existing splicing solution. A new solution can theoretically be developed that enables identification of the participating entities and what each provides, i.e., the different media sources, main and substituting, and the splicer which provides the RTP-level integration of the media payloads in a common timeline and synchronization context.

Since the splicer breaks RTP-level end-to-end security, it needs to be part of the signaling context and the necessary security associations (e.g., SRTP crypto contexts) established for the RTP session participants. When using the Secure Real-Time Transport Protocol (SRTP) [RFC3711], the splicer would have to be provisioned with the same security association as the main RTP sender.

If there is a concern about the confidentiality of the splicing time information, the header extension defined in this document MUST be also protected, for example, header extension encryption [RFC6904] can be used in this case. However, the malicious endpoint may get the splicing time information by other means, e.g., inferring from the communication between the main and substitutive content sources. To avoid the insertion of invalid substitutive content, the splicer MUST have some mechanisms to authenticate the substitutive stream source.

For cases that the splicing time information is changed by a malicious endpoint, the splicing, for example, may fail since it will not be available at the right time for the substitutive media to arrive. Another case is that an attacker may prevent the receivers receiving the content from the main sender by inserting extra splicing time information. To avoid the above cases happening, the authentication of the RTP header extension for splicing time information SHOULD be considered.

When a splicer implemented as a mixer sends the stream to the receivers, CSRC list, which can be used to detect RTP-level forwarding loops as defined in Section 8.2 of [RFC3550], may be removed for simplifying the receivers that can not handle multiple sources in the RTP stream. Hence, loops may occur to cause packets to loop back to upstream of the splicer and may form a serious denial-of-service threat. In such a case, non-RTP means, e.g., signaling among all the participants, MUST be used to detect and resolve loops.

8 IANA Considerations

8.1 RTCP Control Packet Types

Based on the guidelines suggested in [RFC5226], a new RTCP packet format has been registered with the RTCP Control Packet Type (PT) Registry:

Name: SNM

Long name: Splicing Notification Message

Value: TBA

Reference: This document

8.2 RTP Compact Header Extensions

The IANA has also registered a new RTP Compact Header Extension [RFC5285bis], according to the following:

Extension URI: urn:ietf:params:rtp-hdrext:splicing-interval

Description: Splicing Interval

Contact: Jinwei Xia <xiajinwei@huawei.com>

Reference: This document

8.3 SDP Grouping Semantic Extension

This document request IANA to register the new SDP grouping semantic extension called "SPLICE".

Semantics: Splice

Token:SPLICE

Reference: This document

9 Acknowledgement

The authors would like to thank the following individuals who help to review this document and provide very valuable comments: Colin Perkins, Bo Burman, Stephen Botzko, Ben Campbell.

10 References

10.1 Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.

[RFC3264] Rosenberg, J., and H. Schulzrinne, "An Offer/Answer Model with the Session Description Protocol (SDP)", RFC 3264, June 2002.

[RFC5285bis] Even, R., Singer, D. and H. Desineni, "A General Mechanism for RTP Header Extensions", draft-ietf-avtcore-rfc5285-bis-02, May 2016.

[RFC5888] Camarillo, G. and H. Schulzrinne, "The Session Description Protocol (SDP) Grouping Framework", RFC 5888, June 2010.

[RFC5905] Mills, D., Martin, J., Ed., Burbank, J., and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, June 2010.

[RFC6051] Perkins, C. and T. Schierl, "Rapid Synchronisation of RTP Flows", RFC 6051, November 2010.

[RFC7201] Westerlund, M. and C. Perkins, "Options for Securing RTP Sessions", RFC 7201, April 2014.

[RFC7667] Westerlund, M. and S. Wenger, "RTP Topologies", RFC 7667, November 2015.

10.2 Informative References

[RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, March 2004.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

[RFC5506] Johansson, I. and M. Westerlund, "Support for Reduced-Size Real-Time Transport Control Protocol (RTCP): Opportunities and Consequences", RFC 5506, April 2009.

[RFC6285] Ver Steeg, B., Begen, A., Van Caenegem, T., and Z. Vax, "Unicast-Based Rapid Acquisition of Multicast RTP Sessions", RFC 6285, June 2011.

[RFC6904] Lennox, J.,"Encryption of Header Extensions in the Secure Real-Time Transport Protocol (SRTP)", April 2013.

[SCTE35] Society of Cable Telecommunications Engineers (SCTE), "Digital Program Insertion Cueing Message for Cable", 2011.

[RFC6828] Xia, J., "Content Splicing for RTP Sessions", RFC 6828, January 2013.

Authors' Addresses

Jinwei Xia
Huawei

Email: xiajinwei@huawei.com

Roni Even
Huawei

Email: ron.even.tlv@gmail.com

Rachel Huang

Huawei

Email: rachel.huang@huawei.com

Lingli Deng
China Mobile

Email: denglingli@chinamobile.com