

Internet Engineering Task Force  
Internet-Draft  
Intended status: Best Current Practice  
Expires: May 4, 2020

G. Hellstrom  
Omnitor  
November 1, 2019

Real-time text media handling in multi-party conferences  
draft-hellstrom-mmusic-multi-party-rtt-00

Abstract

This memo specifies methods for Real-Time Text (RTT) media handling in multi-party calls. The main solution is to carry Real-Time text by the RTP protocol in a time-sampled mode according to RFC 4103. The main solution for centralized multi-party handling of real-time text is achieved through a media control unit coordinating multiple RTP text streams into one RTP session.

Identification for the streams are provided through the RTCP messages. This mechanism enables the receiving application to present the received real-time text medium in different ways according to user preferences. Some presentation related features are also described explaining suitable variations of transmission and presentation of text.

Call control features are described for the SIP environment. A number of alternative methods for providing the multi-party negotiation, transmission and presentation are discussed and a recommendation for the main one is provided. Two alternative methods using a single RTP stream and source identification inline in the text stream are also described, one of them being provided as a lower functionality fallback method for endpoints with no multi-party awareness for RTT.

Brief information is also provided for multi-party RTT in the WebRTC environment.

EDITOR NOTE: A number of alternatives are specified for discussion. A decision is needed which alternatives are preferred and then how the preferred alternatives shall be emphasized.

Status of This Memo

This Internet-Draft is submitted 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 <https://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 May 4, 2020.

#### Copyright Notice

Copyright (c) 2019 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 (<https://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
1.1. Requirements Language . . . . .	4
2. Centralized conference model . . . . .	4
3. Requirements on multi-party RTT . . . . .	5
4. Coordination of text RTP streams . . . . .	6
4.1. RTP Translator sending one RTT stream per participant . .	6
4.2. RTP Mixer indicating participants in CSRC . . . . .	7
4.3. RTP Mixer indicating participants by a control code in the stream . . . . .	8
4.4. Mesh of RTP endpoints . . . . .	9
4.5. Multiple RTP sessions, one for each participant . . . . .	10
4.6. Mixing for conference-unaware user agents . . . . .	10
5. RTT bridging in WebRTC . . . . .	12
5.1. RTT bridging in WebRTC with one data channel per source .	12
5.2. RTT bridging in WebRTC with one common data channel . . .	12
6. Preferred multi-party RTT transport method . . . . .	13
7. Session control of multi-party RTT sessions . . . . .	14
7.1. Implicit RTT multi-party capability indication . . . . .	15
7.2. RTT multi-party capability declared by SIP media-tags . .	16
7.3. SDP media attribute for RTT multi-party capability indication . . . . .	17

7.4. Preferred capability declaration method. . . . .	18
8. Identification of the source of text . . . . .	18
9. Presentation of multi-party text . . . . .	19
9.1. Associating identities with text streams . . . . .	19
9.2. Presentation details for multi-party aware UAs. . . . .	19
9.2.1. Bubble style presentation . . . . .	20
9.2.2. Other presentation styles . . . . .	21
10. Presentation details for multi-party unaware UAs. . . . .	21
11. Transmission of text from each user . . . . .	21
12. Robustness and indication of possible loss . . . . .	21
13. Performance . . . . .	22
14. Security Considerations . . . . .	22
15. IANA Considerations . . . . .	22
16. Congestion considerations . . . . .	23
17. Acknowledgements . . . . .	23
18. References . . . . .	23
18.1. Normative References . . . . .	23
18.2. Informative References . . . . .	24
Appendix A. Mixing for a conference-unaware UA . . . . .	24
A.1. Short description . . . . .	24
A.2. Functionality goals and drawbacks . . . . .	25
A.3. Definitions . . . . .	25
A.4. Presentation level procedures . . . . .	27
A.4.1. Structure . . . . .	28
A.4.2. Action on reception . . . . .	28
A.5. Display examples . . . . .	31
A.6. Summary of configurable parameters . . . . .	33
A.7. References for this Appendix . . . . .	35
A.8. Acknowledgement . . . . .	36
Author's Address . . . . .	36

## 1. Introduction

Real-time text (RTT) is a medium in real-time conversational sessions. Text entered by participants in a session is transmitted in a time-sampled fashion, so that no specific user action is needed to cause transmission. This gives a direct flow of text in the rate it is created, that is suitable in a real-time conversational setting. The real-time text medium can be combined with other media in multimedia sessions.

Media from a number of multimedia session participants can be combined in a multi-party session. This memo specifies how the real-time text streams are handled in multi-party sessions.

The description is mainly focused on the transport level, but also describes a few session and presentation level aspects.

Transport of real-time text is specified in RFC 4103 [RFC4103] RTP Payload for text conversation. It makes use of RFC 3550 [RFC3550] Real Time Protocol, for transport. Robustness against network transmission problems is normally achieved through redundancy transmission based on the principle from RFC 2198, with one primary and two redundant transmission of each text element. Primary and redundant transmissions are combined in packets and described by a redundancy header. This transport is usually used in the SIP Session Initiation Protocol RFC 3261 [RFC3261] environment.

A very brief overview of functions for real-time text handling in multi-party sessions is described in RFC 4597 [RFC4597] Conferencing Scenarios, sections 4.8 and 4.10. This specification builds on that description and indicates which protocol mechanisms should be used to implement multi-party handling of real-time text.

EDITOR NOTE: A number of alternatives are specified for discussion. A decision is needed which alternatives are preferred and then how the preferred alternatives shall be emphasized.

### 1.1. Requirements Language

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].

## 2. Centralized conference model

In the centralized conference model for SIP, introduced in RFC 4353 [RFC4353] A Framework for Conferencing with the Session Initiation Protocol (SIP), one function co-ordinates the communication with participants in the multi-party session. This function also controls media mixer functions for the media appearing in the session. The central function is common for control of all media, while the media mixers may work differently for each medium.

The central function is called the Focus UA and may be co-located in an advanced terminal including multi-party control functions, or it may be located in a separate location. Many variants exist for setting up sessions including the multipoint control centre. It is not within scope of this description to describe these, but rather the media specific handling in the mixer required to handle multi-party calls with RTT.

The main principle for handling real-time text media in a centralized conference is that one RTP session for real-time text is established including the multipoint media control centre and the participating

endpoints which are going to have real-time text exchange with the others.

The different possible mechanisms for mixing and transporting RTT differs in the way they multiplex the text streams and how they identify the sources of the streams. RFC 7667 [RFC7667] describes a number of possible use cases for RTP. This specification refers to different sections of RFC 7667 for further reading of the situations caused by the different possible design choices.

### 3. Requirements on multi-party RTT

The following requirements are placed on multi-party RTT:

The solution shall be applicable to IMS (3GPP TS 22.173), SIP based VoIP and Next Generation Emergency Services (NENA i3, EENA NG LTD, RFC 6443).

The transmission interval for text must not be longer than 500 milliseconds when there is anything available to send. Ref ITU-T T.140.

If text loss is detected or suspected, a missing text marker shall be inserted in the text stream where the loss is detected or suspected. Ref ITU-T T.140 Amendment 1. ETSI EN 301 549

The display of text from the members of the conversation shall be arranged so that the text from each participant is clearly readable, and its source and the relative timing of entered text is visualized in the display. Mechanisms for looking back in the contents from the current session should be provided. The text should be displayed as soon as it is received. Ref ITU-T T.140

Bridges must be multimedia capable (voice, video, text). Ref NENA i3 STA-010.2.

R7: It MUST be possible to use real-time text in conferences both as a medium of discussion between individual participants (for example, for sidebar discussions in real-time text while listening to the main conference audio) and for central support of the conference with real-time text interpretation of speech. Ref RFC 5194.

It should be possible to protect RTT contents with usual means for privacy and integrity. Ref RFC 6881 section 16

Conferencing procedures are documented in RFC 4579. Ref NENA i3 STA-010.2.

Conferencing applies to any kind of media stream by which users may want to communicate... Ref 3GPP TS 24.147

The framework for SIP conferences is specified in RFC 4353. Ref 3GPP TS 24.147

#### 4. Coordination of text RTP streams

Coordinating and sending text RTP streams in the multi-party session can be done in a number of ways. The most suitable methods are specified here with pros and cons.

A receiving UA SHOULD separate text from the different sources and identify and display them accordingly.

##### 4.1. RTP Translator sending one RTT stream per participant

Within the RTP session, text from each participant is transmitted from the RTP media translator in a separate RTP stream, thus using the same destination address/port combination, but separate RTP SSRC parameters and sequence number series as described in Section 7.1 and 7.2 of RTP RFC 3550 [RFC3550] about the Translator function. The sources of the text in each RTP packet are identified by the SSRC parameters in the RTP packets, containing the SSRC of the initial sources of text.

A receiving UA is supposed to separate text items from the different sources and identify and display them in a suitable way.

This method is described in RFC 7667, section 3.5.1 Relay-transport translator or 3.5.2 Media translator.

The identification of the source is made through the RTCP SDPS CNAME and NAME packets as described in RTP[RFC3550].

##### Pros:

This method has moderate overhead. When loss of packets occur, it is possible to recover text from redundancy at loss of up to the number of redundancy levels carried in the RFC 4103 stream. (normally primary and two redundant levels.

More loss than what can be recovered, can be detected and the marker for text loss can be inserted in the correct stream.

It may be possible in some scenarios to keep the text encrypted through the Translator.

**Cons:**

There may be RTP implementations not supporting the Translator model.

It is even most likely that this configuration is not supported by current media declarations in sdp. RFC 3264 specifies in many places that one media description is supposed to describe just one RTP stream.

**4.2. RTP Mixer indicating participants in CSRC**

An RTP media mixer combines text from all participants except from the receiving endpoint into one RTP stream, thus all using the same destination address/port combination, the same RTP SSRC and, one sequence number series as described in Section 7.1 and 7.3 of RTP RFC 3550 [RFC3550] about the Mixer function. The sources of the text in each RTP packet are identified by the CSRC parameters in the RTP packets, containing the SSRC of the initial sources of text. The order of the CSRC parameters are the same as the order of the redundant and primary data fields in the packet. If all redundancy blocks in a packet are from the same source, then it is allowed to use only one CSRC in the RTP packet. This method is described in RFC 7667, section 3.6.3 Media switching mixer.

The identification of the source is made through the RTCP SDES CNAME and NAME packets as described in RTP[RFC3550].

A receiving UA is supposed to separate text items from the different sources and identify and display them accordingly.

It is likely that the conference server need to have authority to decrypt the payload in the RTP packets in order to be able to recover text from redundant data or insert the missing text marker in the stream, and repack the text in new packets. Further study is needed

**Pros:**

This method has moderate overhead.

When loss of packets occur, it is possible to recover text from redundancy at loss of up to the number of redundancy levels carried in the RFC 4103 stream. (normally primary and two redundant levels.

This method can be implemented with most RTP implementations.

**Cons:**

When more consecutive packet loss than the number of generations of redundant data appears, it is not possible to deduct the source of the totally lost data. Therefore it is not possible to know in which stream to insert the missing text marker. It MAY be acceptable to either indicate a general loss indication, or insert a loss marker in all streams. Calculations of most likely source can however be made from received RTP and RTCP contents so that the loss marker can be inserted in the most likely struck stream.

The conference server need to be allowed to decrypt/encrypt the packet payload.

#### 4.3. RTP Mixer indicating participants by a control code in the stream

Text from all participants except the receiving one is transmitted from the media mixer in the same RTP session and stream, thus all using the same destination address/port combination, the same RTP SSRC and , one sequence number series as described in Section 7.1 and 7.3 of RTP RFC 3550 [RFC3550] about the Mixer function. The sources of the text in each RTP packet are identified by a new defined T.140 control code "c" followed by a unique identification of the source in UTF-8 string format.

The receiver can use the string for presenting the source of text. This method is on the RTP level described in RFC 7667, section 3.6.2 Media mixing mixer.

The inline coding of the source of text is applied in the data stream itself, and an RTP mixer function is used for coordinating the sources of text into one RTP stream.

Information uniquely identifying each user in the multi-party session is placed as the parameter value "n" in the T.140 application protocol function with the function code "c". The identifier shall thus be formatted like this: SOS c n ST, where SOS and ST are coded as specified in ITU-T T.140 [T.140]. The "c" is the letter "c". The n parameter value is a string uniquely identifying the source. This parameter shall be kept short so that it can be repeated in the transmission without concerns for network load.

A receiving UA is supposed to separate text items from the different sources and identify and display them accordingly.

The conference server need to be allowed to decrypt/encrypt the packet payload in order to check the source and repack the text.

Pros:



If loss of packets occur, it is possible to recover text from redundancy at loss of up to the number of redundancy levels carried in the RFC 4103 stream. (normally primary and two redundant levels.

This method can be implemented with most RTP implementations.

Transmitted text can also be used with other transports than RTP

#### Cons:

If more consecutive packet loss than the number of generations of redundant data appears, it is not possible to deduct the source of the totally lost data. Therefore it is not possible to know in which stream to insert the missing text marker. Calculations of most likely source can however be made from recent history, so that it is quite likely that the marker is inserted in the correct stream. Such loss should however be rare, and a general warning that there might have been text loss in the session might be acceptable.

The mixer needs to be able to generate suitable and unique source identifications which are suitable as labels for the sources.

Requires an extension on the ITU-T T.140 standard, best made by the ITU.

The conference server need to be allowed to decrypt/encrypt the packet payload.

The conference server need to be allowed to decrypt/encrypt the packet payload.

#### 4.4. Mesh of RTP endpoints

Text from all participants are transmitted directly to all others in one RTP session, without a central bridge. The sources of the text in each RTP packet are identified by the source network address and the SSRC.

This method is described in RFC 7667, section 3.4 Point to multi-point using mesh.

#### Pros:

When loss of packets occur, it is possible to recover text from redundancy at loss of up to the number of redundancy levels carried in the RFC 4103 stream. (normally primary and two redundant levels.

This method can be implemented with most RTP implementations.

Transmitted text can also be used with other transports than RTP

Cons:

This model is not described in IMS, NENA and EENA specifications, and does therefore not meet the requirements.

#### 4.5. Multiple RTP sessions, one for each participant

Text from all participants are transmitted directly to all others in one RTP session each, without a central bridge. Each session is established with a separate media description in SDP. The sources of the text in each RTP packet are identified by the source network address and the SSRC.

This method is out of scope for further discussion here, because the foreseen applications use centralized model conferencing.

Pros:

When loss of packets occur, it is possible to recover text from redundancy at loss of up to the number of redundancy levels carried in the RFC 4103 stream. (normally primary and two redundant levels.

Complete loss of text can be indicated in the received stream.

This method can be implemented with most RTP implementations.

End-to-end encryption is achievable.

Cons:

This method is not described in IMS, NENA and EENA specifications and does therefore not meet the requirements.

A lot of network resources are spent on setting up separate sessions for each participant.

#### 4.6. Mixing for conference-unaware user agents

Multi-party real-time text contents can be transmitted to conference-unaware user agents if source labeling and formatting of the text is performed by a mixer. This method has the limitations that the layout of the presentation and the format of source identification is purely controlled by the mixer, and that only one source at a time is allowed to present in real-time. Other sources need to be stored temporarily waiting for an appropriate moment to switch the source of transmitted text. The mixer controls the switching of sources and

inserts a source identifier in text format at the beginning of text after switch of source. The logic of the mixer to detect when a switch is appropriate should detect a number of places in text where a switch can be allowed, including new line, end of sentence, end of phrase, a period of inactivity, and a word separator after a long time of active transmission.

This method MAY be used when no support for multi-party awareness is detected in the receiving endpoint. The base for this method is described in RFC 7667, section 3.6.2 Media mixing mixer.

See Appendix A for an informative example of a procedure for presenting RTT to a conference-unaware UA.

Pros:

Can be transmitted to conference-unaware endpoints.

Can be used with other transports than RTP

Cons:

Does not allow full real-time presentation of more than one source at a time. Text from other sources will be delayed, even if automatic detection of suitable moments for switching source for presentation is made by the mixer.

The only realistic presentation format is a style with the text from the different sources presented with a text label indicating source, and the text collected in a chat style presentation but with more frequent turn-taking.

Endpoints often have their own system for adding labels to the RTT presentation. In that case there will be two levels of labels in the presentation, one for the mixer and one for the sources.

If loss of more packets than can be recovered by the redundancy appears, it is not possible to detect which source was struck by the loss. It is also possible that a source switch occurred during the loss, and therefore a false indication of the source of text can be provided to the user after such loss.

Because of all these cons, this method MUST NOT be used as the main method, but only as the last resort for backwards interoperability with conference-unaware endpoints.

The conference server need to be allowed to decrypt/encrypt the packet payload.

## 5. RTT bridging in WebRTC

Within WebRTC, real-time text is specified to be carried in WebRTC data channels as specified in draft-ietf-mmusic-t140-usage-data-channel. A few ways to handle multi-party RTT are mentioned briefly. They are explained and further detailed below.

### 5.1. RTT bridging in WebRTC with one data channel per source

A straightforward way to handle multi-party RTT is for the bridge to open one T.140 data channel per source towards the receiving participants.

The stream-id forms a unique stream identification.

The identification of the source is made through the Label property of the channel, and session information belonging to the source. The UA can compose a readable label for the presentation from this information.

Pros:

This is a straightforward solution.

Cons:

With a high number of participants, the overhead of establishing the high number of data channels required may be high.

### 5.2. RTT bridging in WebRTC with one common data channel

A way to handle multi-party RTT in WebRTC is for the bridge combine text from all sources into one data channel and insert the sources in the stream by a T.140 control code for source.

This method is described in a corresponding section for RTP transmission above.

The identification of the source is made through insertion in the beginning of each text transmission from a source of a control code extension "c" followed by a string representing the source, framed by the control code start and end flags SOS and ST (See ITU-T T.140 [T.140]).

A receiving UA is supposed to separate text items from the different sources and identify and display them in a suitable way.

The UA does not always display the source identification in the received text at the place where it is received, but has the information as a guide for planning the presentation of received text. A label corresponding to the source identification is presented when needed depending on the selected presentation style.

Pros:

This solution has relatively low overhead on session and network level

Cons:

This solution has higher overhead on the media contents level than the WebRTC solution above.

Standardisation of the new control code "c" in ITU-T T.140 is required.

The conference server need to be allowed to decrypt/encrypt the data channel contents.

## 6. Preferred multi-party RTT transport method

EDITOR NOTE: The recommendations here need to be validated, and the proposed further studies performed.

For RTP transport of RTT, two methods for multi-party mixing and transport for conference-aware parties stand out as fulfilling the goals best: "RTP Mixer indicating participants in CSRC" and "RTP Mixer indicating participants by a control code in the stream". The CSRC based method has a slightly better opportunity to use a robust and well defined procedure in the server. The inline stream based method has the slightly better opportunity for ease of interworking with other environments for RTT where the in-line identification also could be used. The inline method can also be applied in the case when an ad-hoc method for conferencing is used, and the source of text only detectable inline. The possibility to use such methods for conferencing and the interoperability opportunities are important, and therefore the method to implement for multi-party RTT with or without conference-aware parties when no other method is explicitly agreed between implementing parties for SIP with RTP is "RTP Mixer indicating participants by a control code in the stream".

Further studies should be made to find out if assessment of the source for lost text can be better done, and if operation without letting the conference server decrypt data can be specified.

For WebRTC, one method is to prefer because of the same interoperability reasons, and because of the lower network resource usage. So, for WebRTC, the method to implement for multi-party RTT with conference-aware parties when no other method is explicitly agreed between implementing parties is: "RTT bridging in WebRTC with one common data channel".

Further studies are needed to check if it can be possible to let the conference server act without decrypting the text.

As a last resort, when the UA is not conference-aware, the method for mixing for multi-party-unaware user agents may be used for both RTP and WebRTC data channel solutions considering that this method provides a reduced impression of the real time characteristics and may delay presentation of text.

## 7. Session control of multi-party RTT sessions

General session control aspects for multi-party sessions are described in RFC 4575 [RFC4575] A Session Initiation Protocol (SIP) Event Package for Conference State, and RFC 4579 [RFC4579] Session Initiation Protocol (SIP) Call Control - Conferencing for User Agents. The nomenclature of these specifications are used here.

The procedures for a conference-aware model for RTT-transmission shall only be applied if a capability exchange for conference-aware real-time text transmission has been completed and a supported method for multi-party real-time text transmission can be identified.

A method for detection of conference-awareness for centralized SIP conferencing in general is specified in RFC 4579 [RFC4579]. The focus sends the "isfocus" feature tag in a SIP Contact header. This causes the conference-aware UA to subscribe to conference notifications from the focus. The focus then sends notifications to the UA about entering and disappearing conference participants and their media capabilities. The information is carried XML-formatted in a 'conference-info' block in the notification according to RFC 4575. The mechanism is described in detail in RFC 4575 [RFC4575].

Before a conference media server starts sending multi-party RTT to a UA, a verification of its ability to handle multi-party RTT must be made. A decision on which mechanism to use for identifying text from the different participants must also be taken, implicitly or explicitly. These verifications and decisions can be done in a number of ways. The most apparent ways are specified here and their pros and cons described. One of the methods is selected to be the one to be used by implementations according to this specification.

### 7.1. Implicit RTT multi-party capability indication

Capability for RTT multi-party handling can be decided to be implicitly indicated by session control items.

The focus may implicitly indicate multi-party RTT capability by including the media child with value "text" in the RFC 4575 conference-info provided in conference notifications.

A UA may implicitly indicate multi-party RTT capability by including the text media in the SDP in the session control transactions with the conference focus after the subscription to the conference has taken place.

The implicit RTT capability indication means for the focus that it can handle multi-party RTT according to the preferred method indicated in the RTT multi-party methods section above.

The implicit RTT capability indication means for the UA that it can handle multi-party RTT according to the preferred method indicated in the RTT multi-party methods section above.

If the focus detects that a UA implicitly declared RTT multi-party capability, it SHALL provide RTT according to the preferred method.

If the focus detects that the UA does not indicate any RTT multi-party capability, then it shall provide RTT multi-party text in the way specified for conference-unaware UA above.

If the UA detects that the focus has implicitly declared RTT multi-party capability, it shall be prepared to present RTT in a multi-party fashion according to the preferred method.

#### Pros:

Acceptance of implicit multi-party capability implies that no standardisation of explicit RTT multi-party capability exchange is required.

#### Cons:

There may be a desire to indicate conference-awareness in general, but not for RTT. Then the method called "Mixing for conference-unaware user agents" should be used as a lower functionality fallback. There is no way to provide that indication by the UA according to the specification of the implicit method above. The solution must be that no conference awareness is indicated by the UA when it has no RTT multi-party capability.

If other methods for multi-party RTT are to be used in the same implementation environment as the preferred ones, then capability exchange needs to be defined for them.

## 7.2. RTT multi-party capability declared by SIP media-tags

Specifications for RTT multi-party capability declarations can be agreed for use as SIP media feature tags, to be exchanged during SIP call control operation according to the mechanisms in RFC 3840 and RFC 3841. Capability for the RTT Multi-party capability is then indicated by the media feature tag "rtt-mixer", with one or more of its possible values in a comma-separated list.

The possible values in the list are:

rtp-translator

rtp-mixer

t140-mixer

rtp-mesh

multi-session

rtp-translator indicates capability for using the RTP-translator based coordination of multi-party text.

rtp-mixer indicates capability for using the RTP-mixer based presentation of multi-party text.

t140-mixer indicates capability for using the T.140 control code source indicators in a mixer.

text-mixer indicates capability for using the fallback method with text formatting for conference-unaware endpoints.

rtp-mesh indicates capability for using the mesh based transmission of multi-party text.

multi-session indicates capability for using separate point-to-point RTP sessions between all participants.

An offer-answer exchange should take place and the common method selected by the answering party shall be used in the session with that UA.



When no common method is declared, then only the fallback method can be used.

If more than one text media line is included in SDP, all must be capable of using the declared RTT multi-party method.

Pros:

Provides a clear decision method.

Can be extended with new mixing methods.

Can guide call routing to a suitable capable focus.

Cons:

Requires standardization and IANA registration.

Cannot be used in the WebRTC environment.

### 7.3. SDP media attribute for RTT multi-party capability indication

An attribute can be specified on media level, to be used in text media SDP declarations for negotiating RTT multi-party capabilities. The attribute can have the name "rtt-mixer", with one or more of its possible values in a comma-separated list.

The possible values in the list are:

rtp-translator

rtp-mixer

t140-mixer

rtp-mesh

multi-session

rtp-translator indicates capability for using the RTP-translator based coordination of multi-party text.

rtp-mixer indicates capability for using the RTP-mixer based presentation of multi-party text.

t140-mixer indicates capability for using the T.140 control code source indicators in a mixer.

text-mixer indicates capability for using the fallback method with text formatting for conference-unaware endpoints.

rtp-mesh indicates capability for using the mesh based transmission of multi-party text.

multi-session indicates capability for using separate point-to-point RTP sessions between all participants.

An offer-answer exchange should take place and the common method selected by the answering party shall be used in the session with that UA.

When no common method is declared, then only the fallback method can be used.

Pros:

Provides a clear decision method.

Can be extended with new mixing methods.

Can be used on specific text media.

Can be used also for SDP-controlled WebRTC sessions with multiple streams in the same data channel.

Cons:

Requires standardization and IANA registration.

Is not well defined for multi-party methods involving more than one media section for text.

Cannot guide SIP routing.

#### 7.4. Preferred capability declaration method.

The preferred capability declaration method is the one with SDP attributes because it is partially usable also for WebRTC.

#### 8. Identification of the source of text

EDITOR NOTE: The text in the following sections need to be adapted after recommendations for the main methods for coordination of RTT has been selected. Details should be provided mainly for the recommended method.

As soon as a new member is added to the RTP session, its characteristics shall be transmitted in RTCP SDES CNAME and NAME reports according to section 6.5 in RFC 3550. The information about the participant MUST also be included in the conference data including the text media member in a notification according to RFC 4575.

The RTCP SDES report, SHOULD contain identification of the source represented by the SSRC/CSRC identifier. This identification MUST contain the CNAME field and MAY contain the NAME field and other defined fields of the SDES report.

A focus UA SHOULD primarily convey SDES information received from the sources of the session members. When such information is not available, the focus UA SHOULD compose SSRC/CSRC, CNAME and NAME information from available information from the SIP session with the participant.

## 9. Presentation of multi-party text

All session participants MUST observe the SSRC/CSRC field of incoming text RTP packets, and make note of what source they came from in order to be able to present text in a way that makes it easy to read text from each participant in a session, and get information about the source of the text.

### 9.1. Associating identities with text streams

A source identity SHOULD be composed from available information sources and displayed together with the text as indicated in ITU-T T.140 Appendix[T.140].

The source identity should primarily be the NAME field from incoming SDES packets. If this information is not available, and the session is a two-party session, then the T.140 source identity SHOULD be composed from the SIP session participant information. For multi-party sessions the source identity may be composed by local information if sufficient information is not available in the session.

Applications may abbreviate the presented source identity to a suitable form for the available display.

### 9.2. Presentation details for multi-party aware UAs.

The multi-party aware UA should after any action for recovery of data from lost packets, separate the incoming streams and present them according to the style that the receiving application supports and

the user has selected. The decisions taken for presentation of the multi-party interchange shall be purely on the receiving side. The sending application must not insert any item in the stream to influence presentation that is not requested by the sending participant.

#### 9.2.1. Bubble style presentation

One often used style is to present real-time text in chunks in readable bubbles identified by labels containing names of sources. Bubbles are placed in one column in the presentation area and are closed and moved upwards in the presentation area after certain items or events, when there is also newer text from another source that would go into a new bubble. The text items that allows bubble closing are any character closing a phrase or sentence followed by a space or a timeout of a suitable time (about 10 seconds).

Real-time active text sent from the local user should be presented in a separate area. When there is a reason to close a bubble from the local user, the bubble should be placed above all real-time active bubbles, so that the time order that real-time text entries were completed is visible.

Scrolling is usually provided for viewing of recent or older text. When scrolling is done to an earlier point in the text, the presentation shall not move the scroll position by new received text. It must be the decision of the local user to return to automatic viewing of latest text actions. It may be useful with an indication that there is new text to read after scrolling to an earlier position has been activated.

The presentation area may become too small to present all text in all real-time active bubbles. Various techniques can be applied to provide a good overview and good reading opportunity even in such situations. The active real-time bubble may have a limited number of lines and if their contents need more lines, then a scrolling opportunity within the real-time active bubble is provided. Another method can be to only show the label and the last line of the active real-time bubble contents, and make it possible to expand or compress the bubble presentation between full view and one line view.

Erasures require special consideration. Erasure within a real-time active bubble is straightforward. But if erasure from one participant affects the last character before a bubble, the whole previous bubble becomes the actual bubble for real-time action by that participant and is placed below all other bubbles in the presentation area. If the border between bubbles was caused by the CRLF characters, only one erasure action is required to erase this

bubble border. When a bubble is closed, it is moved up, above all real-time active bubbles.

#### 9.2.2. Other presentation styles

Other presentation styles than the bubble style may be arranged and appreciated by the users. In a video conference one way may be to have a real-time text area under the video view of each participant. Another view may be to provide one column in a presentation area for each participant and place the text entries in a relative vertical position corresponding to when text entry in them was completed. The labels can then be placed in the column header. The considerations for ending and moving and erasure of entered text discussed above for the bubble style are valid also for these styles.

#### 10. Presentation details for multi-party unaware UAs.

Multi-party unaware UA:s are prepared only for presentation of two sources of text, the local user and a remote user. In order to enable some multi-party communication with such UA, the mixer need to plan the presentation and insert labels and line breaks before labels. Many limitations appear for this presentation mode, and it must be seen as a fallback and a last resort.

See Appendix A for an informative example of a procedure for presenting RTT to a conference-unaware UA.

#### 11. Transmission of text from each user

UAs participating in sessions with real-time text, SHOULD send SDP packets in RTCP giving values to appropriate identification fields.

The CNAME field SHALL be included in SDP packets.

The NAME field should be given a value that is suitable as an identifier of text from the user of the UA.

#### 12. Robustness and indication of possible loss

This section discusses the means for robustness against loss of text that is already specified and their performance in the multi-party situation. means for reducing the risk for loss is discussed, as well as ways to detect in which stream loss has occurred.

TBD

### 13. Performance

This section discusses performance and performance limitations for the different transport solutions, and indicates which means for performance increase versus load limitations can be suitable to apply compared to the point-to-point case.

TBD

### 14. Security Considerations

The security considerations valid for RFC 4103 and RFC 3550 are valid also for the multi-party sessions with text.

### 15. IANA Considerations

EDITOR NOTE: TBD after decision of proposed preferences in the draft.

This document Introduces the TBD /SIP media tag/SDP media level attribute/ rtt-mixer, with a comma-separated parameter list containing the following possible values:

rtsp-translator

rtsp-mixer

t140-mixer

rtsp-mesh

multi-session

rtsp-translator indicates capability for using the RTP-translator based coordination of multi-party text.

rtsp-mixer indicates capability for using the RTP-mixer based presentation of multi-party text.

t140-mixer indicates capability for using the T.140 control code source indicators in a mixer.

text-mixer indicates capability for using the fallback method with text formatting for conference-unaware endpoints.

rtsp-mesh indicates capability for using the mesh based transmission of multi-party text.

multi-session indicates capability for using separate point-to-point RTP sessions between all participants.

#### 16. Congestion considerations

The congestion considerations described in RFC 4103 are valid also for multi-party use of the real-time text RTP transport. A risk for congestion may appear if a number of conference participants are active transmitting text simultaneously, because this multi-party transmission method does not allow multiple sources of text to contribute to the same packet.

In situations of risk for congestion, the Focus UA MAY combine packets from the same source to increase the transmission interval per source up to one second. Local conference policy in the Focus UA may be used to decide which streams shall be selected for such transmission frequency reduction.

#### 17. Acknowledgements

Arnoud van Wijk for contributions to an earlier, expired draft of this memo.

#### 18. References

##### 18.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, DOI 10.17487/RFC3261, June 2002, <<https://www.rfc-editor.org/info/rfc3261>>.
- [RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, DOI 10.17487/RFC3550, July 2003, <<https://www.rfc-editor.org/info/rfc3550>>.
- [RFC4103] Hellstrom, G. and P. Jones, "RTP Payload for Text Conversation", RFC 4103, DOI 10.17487/RFC4103, June 2005, <<https://www.rfc-editor.org/info/rfc4103>>.

- [RFC4575] Rosenberg, J., Schulzrinne, H., and O. Levin, Ed., "A Session Initiation Protocol (SIP) Event Package for Conference State", RFC 4575, DOI 10.17487/RFC4575, August 2006, <<https://www.rfc-editor.org/info/rfc4575>>.
- [RFC4579] Johnston, A. and O. Levin, "Session Initiation Protocol (SIP) Call Control - Conferencing for User Agents", BCP 119, RFC 4579, DOI 10.17487/RFC4579, August 2006, <<https://www.rfc-editor.org/info/rfc4579>>.
- [T.140] "Protocol for multimedia application text conversation", 1998, <<http://www.itu.int/rec/T-REC-T.140/en>>.

## 18.2. Informative References

- [RFC4353] Rosenberg, J., "A Framework for Conferencing with the Session Initiation Protocol (SIP)", RFC 4353, DOI 10.17487/RFC4353, February 2006, <<https://www.rfc-editor.org/info/rfc4353>>.
- [RFC4597] Even, R. and N. Ismail, "Conferencing Scenarios", RFC 4597, DOI 10.17487/RFC4597, August 2006, <<https://www.rfc-editor.org/info/rfc4597>>.
- [RFC7667] Westerlund, M. and S. Wenger, "RTP Topologies", RFC 7667, DOI 10.17487/RFC7667, November 2015, <<https://www.rfc-editor.org/info/rfc7667>>.

## Appendix A. Mixing for a conference-unaware UA

This informational appendix describes media mixer procedures for a multi-party conference server to format real-time text from a number of participants into one single text stream to a participant with a terminal that has no features for multi-party text display. The procedures are intended for implementations using ITU-T T.140 [T.140] for the real-time text coding and presentation.

### A.1. Short description

The media mixer procedures described here are intended to make real-time text from a number of call participants be coordinated into one text stream to a terminal originally intended for two-party calls. A conference server is supposed to apply the procedures.

The procedures may also be applied on a terminal for display of multiple streams of real-time text in one area.



The intention is that text from each participant shall be displayed in suitable sections so that it is easy to read, and text from one active participant at a time is sent and displayed in real-time. The receiving terminal is assumed to have one display area for received text. The display is arranged by this procedure in a text chat style, with a name label in front of each text section where switch of source of the text has taken place.

When more than one participant transmits text at the same time, the text from only one of them is transmitted directly to the receiving terminals. Text from the other participants is stored in buffers in the conference server for transmission at a later time, when a suitable situation for switch of current transmitter can take place.

#### A.2. Functionality goals and drawbacks

The procedures are intended to make best efforts to present a multi-party text conversation on a terminal that has no awareness of multi-party calls. There are some obvious drawbacks, and a terminal designed with multi-party awareness will be able to present multi-party call contents in a more flexible way. Only two parties at a time will be allowed to display added text in real-time, while the other parties' produced text will need to be stored in the multi-party server for a moment awaiting a suitable occasion to be displayed. There are also some cases of erasure that will not be performed on the target text but only indicated in another way. Even with these drawbacks, the procedure provides an opportunity to display text from more than two parties in a smooth and readable way.

This specification does not introduce any new protocol element, and does not rely on anything else than basic two-party terminal functionality with presentation level according to ITU-T T.140 [T.140]. It is a description of a best current practice for mixing and presentation of the real-time text component in multi-party calls with terminals without multi-party awareness.

The procedures are applicable to scenarios, when the conference focus and a User Agent have not gone through any successfully completed negotiation about conference awareness for the real-time text medium neither on the transport level, nor on the presentation level.

#### A.3. Definitions

Active participant: Any user sending text, or being in a pending period.

BOM Byte-Order-Mark, the Unicode character FEFF in UCS-16.

**Buffer:** A buffer intended for unsent text collected per participant.

**Contributing participants:** The participants selected to contribute to the text stream sent to the recipients.

By default all participants except the recipient are contributing participants for transmission to the recipient.

**Current participant:** The participant for whom text currently is transmitted to the recipient in real time.

**Current Recipients:** By default all participants.

**Display Counter:** A counter for the number of displayable characters in a participant's buffer or in the current entry. Used for controlling how far erasure may be performed.

**Erasur replacement** A character to be displayed when an erasure was done, but the text to erase is not reachable on the multi-party display. Default 'X'.

**Message delimiter:** Character(s) forming the end of an imagined message. A configurable set of alternatives, consisting by default of: Line Separator, Paragraph Separator, CR, CRLF, LF.

**Pending period:** A configurable time period of inactivity from a participant, by default set to 7 seconds after each reception of characters from that participant, evaluated as current time minus time stamp of latest entered character.

**Sentence delimiter:** Characters forming end of sentence: A configurable set of alternatives, by default consisting of: dot '.', question mark '?' and exclamation mark '!' followed by a space.

**Label:** A readable unique name for a participant, created by the server from a suitable source related to the participant, e.g. part of the SIP Display name, surrounded by the Label delimiters. The label should have a settable maximum length, with 12 being the default.

**Label delimiters** A configurable set of characters at the edges of the Label, by default being a left bracket [ at the leading edge and a closing bracket ] followed by a space at the trailing edge.

**Line Separator** Unicode UCS-16 2028. Used to request NewLine in Real-Time Text.

Maximum waiting time: The maximum time any participant's text shall be allowed to wait for transmission, by default set to 20 seconds.

Recipient: The terminal receiving the mixed text stream.

SGR Select Graphic Rendition, a control code to specify colours etc.

Switch Reason: A set of reasons to switch Current Participant, consisting of the following

- Waiting time higher for any other participant than the current participant combined with any of the following states:

- A message delimiter was the latest transmitted item

- A sentence delimiter was the latest transmitted item

- A Pending Period has expired and still no text has been transmitted

- The Maximum Waiting time has expired followed by a Word Delimiter or an expired Time Extension.

Waiting time: The time the first character in queue for transmission from a participant has been waiting in a buffer for transmission. The granularity shall be 0.3 Seconds or finer.

Word delimiter: Character forming end of word: space

Time extension: A configurable short extension time allowed after the Maximum waiting time during which a suitable moment for switching Current Participant is awaited, by default set to 7 seconds.

#### A.4. Presentation level procedures

The conference server applies these mixing procedures to text transmitted to all call participants who have not gone through a completed negotiation for conference awareness in real-time text presentation.

All the participants and the conference server use real-time text conversation presentation coding according to ITU-T T.140 [T.140]. A consequence is that real-time text transmissions are UTF-8 coded, with control codes selected from ISO 6429 [ISO 6429].

The description is from the conference server point of view.

#### A.4.1. Structure

The real-time text mixer structure described here is supposed to be placed in the media path so that it is implemented with one mixer per recipient. A mixer contains buffers for temporary storage of text intended for the recipient. Each mixer has one buffer for each contributing participant. A set of status variables is maintained per buffer and is used in the mixer actions. The mixer logic decides for each moment which participant's buffer content is to be sent on to the recipient. By default, the recipient does not contribute text to its own mixer. Text transmitted by a participant is usually displayed locally and will only cause confusion if it appears also in received text.

If there is a reason, own text can be configured to be transmitted also to the participants. That can enable a simplification of the mixer design to have only one common set of buffers instead of a set per recipient. That simplification will however hamper the flow of the conversation severely and is therefore NOT RECOMMENDED.

#### A.4.2. Action on reception

This description of the mixer is valid per recipient.

Text from each contributing participant is checked for a set of characteristics on reception.

Delete BOM: BOM characters are deleted.

Insert in buffer: Resulting text is put into the contributing participant's buffer in the receiving participant's mixer.

Maintain a display counter: For each text character that will take a position on the receiving display, a Display Counter for each participant is increased by one.

There is one T.140 real-time text item that consists of two characters, but is regarded to be a unit and therefore increase the Display Counter with one only. That is CRLF.

Furthermore, the following control codes are regarded units that shall not take any position on the receiving display and shall therefore not increase the Display Counter:

0098 string 009C (SOS-ST strings)

ESC 0061 (INT)

009B Ps 006D (the SGR code, with special handling described below)

BEL (Alert in session)

See the section on control codes below for details.

Combination characters: Also note that it is possible to use combination characters in Unicode. Such combination characters contain more than one character part. They shall only increase the Display Counter with one. The combination characters mainly have components in the series 0300 ? 0361 and 20D0 ? 20E1.

Erase: If the control code for erasure, BS, is received, the following shall be done: If the Display Counter is 0, an Erasure Replacement character, by default being ?X? is inserted in the buffer instead of the erasure, to mark that erasure was intended in earlier transmitted entries. ( this matches traditional habits in real-time text when participants sometimes type XXX to indicate erasure they do not bother to make explicit). If the Display Counter is >0, then the counter is reduced by one, and the erasure control code BS put into the buffer.

Initial action in the session: BOM shall be sent initially to the recipients in the beginning of the session.

Maintaining a waiting time per participant: The time that text has been in the buffer is maintained as the waiting time for each buffer. A granularity of 0.3 seconds is sufficient.

Storing time of reception for each character: Each character that is stored in a buffer shall be assigned with a time stamp indicating its time of reception. A granularity of 0.3 seconds is sufficient. This time stamp is used for calculation of idle time and waiting time in the evaluation of switch reasons.

Initial assignment of the Current Participant: The first contributing participant to send text in the session is assigned to be the Current Participant.

Actions on assignment of a Current Participant: When a participant becomes the Current Participant, the following initial actions shall be performed:

1. Scanning transmissions and timers for a Switch Reason is inactivated.

2. The Current Recipients are set so that all transmissions go to the new set of Current Recipients (See definition).

3. A Line Separator is transmitted if the switch reason was any other than a message delimiter.

4. The Label is transmitted

5. Any stored SGR code is transmitted

6. Scanning transmissions and timers for a Switch Reason is activated.

7. Text in the buffer is transmitted, recalculating and setting the waiting time for each transmitted character based on the time of reception of next character in the buffer. If a switch occurs during transmission from the buffer, the remaining buffer contents is maintained and transmission can continue next time this transmitter becomes the current participant. Any text entered into the buffer for the current participant is after that sent to the recipient until a Switch Reason occurs.

Actions on transmission and during the session: Transmissions are checked for control codes to act on at transmission as described below in the section about handling of control codes and such actions are performed. When the scanning of transmission and timers for a Switch Reason is active, the timers and the transmission to the recipient is analyzed for detection if a Switch Reason has occurred. See the definition of Switch Reasons for details.

Actions when a Switch Reason has occurred: If a Switch Reason has occurred, then the following actions shall be performed:

1. The Display Counter of the Current Participant is set to zero

2. If there is an SGR code stored for the Current Participant, a reset of SGR shall be sent by the sequence SGR 0 [009B 0000 006D].

3. A participant with the longest waiting time is assigned to be the Current Participant, and the procedure for assignment of a Current Participant described above is performed.

Handling of Control codes: The following control codes are specified by ITU-T T.140. Some of them require consideration in the conference server. Note that the codes presented here are expressed in UCS-16, while transmission is made in UTF-8 transform

of these codes. Other sections specify procedures for handling of specific control codes in the conference server.

BEL 0007 Bell, provides for alerting during an active session.

BS 0008 Back Space, erases the last entered character.

NEW LINE 2028 Line separator.

CR LF 000D 000A A supported, but not preferred way of requesting a new line.

INT ESC 0061 Interrupt (used to initiate mode negotiation procedure).

SGR 009B Ps 006D Select graphic rendition. Ps is rendition parameters specified in ISO 6429.

SOS 0098 Start of string, used as a general protocol element introducer, followed by a maximum 256 bytes string.

ST 009C String terminator, end of SOS string.

ESC 001B Escape - used in control strings.

Byte order mark FEFF Zero width, no break space, used for synchronization.

Missing text mark FFFD Replacement character, marks place in stream of possible text loss.

Code for message border, useful, but not mentioned in T.140: New Message 2029 Paragraph separator

Handling of Graphic Rendition SGR: The following procedure shall be followed in order to let the participants control the graphic rendition of their entries without disturbing other participants' graphic rendition. The text stream sent to a recipient shall be monitored for the SGR sequence. The latest conveyed SGR sequence is also stored as a status variable for the recipient. If the SGR 0 code initiated from the current participant is transmitted, the SGR storage shall be cleared.

#### A.5. Display examples

The following pictures are examples of the view on a participant's display.

Conference	Alice
[Bob]:My flight is to Orly. [Eve]:Hi all, can we plan for the seminar.	I will arrive by TGV. Convenient to the main station.
[Bob]:Eve, will you do your presentation on Friday? [Eve]:Yes, Friday at 10. [Bob]: Fine, wo	We need to meet befo

Figure 2 : Alice who has a conference-unaware client is receiving the multi-party real-time text in a single-stream. This figure shows how a coordinated column view MAY be presented on Alice's device.

[Alice] Hi, Alice here.	^
[Bob] Bob as well.	
[Eve] Hi, this is Eve, calling from Paris. I thought you should be here.	
[Alice] I am coming on Thursday, my performance is not until Friday morning.	
[Bob] And I on Wednesday evening.	
[Eve] we can have dinner and then take a walk	
[Eve-typing] But I need to be back to the hotel by 11 because I need	- - v
of course, I underst	

Figure 3 shows a conference view with real-time text preview. Bob's text is buffering until a Current switch reason.



## A.6. Summary of configurable parameters

A number of configurable parameters are described in this specification. This table provides a summary of the parameters on presentation level. A service provider implementing a multi-party service may want to set specific values on these parameters to adapt the characteristics of the service. It is possible to control them per recipient, if desired.

Parameter: Current Recipients

Purpose: Control if participant shall get their own text.

Possible values: Exclude or Include Current Participant

Default value: Exclude

Comment: Own transmissions are usually displayed sufficiently locally

Parameter: Erasure replacement

Purpose: Character to show erasure, when erasure cannot be done

Possible values: Character

Default value: X

Comment: May need to have other value for other than Latin script.

Parameter: Message delimiter

Purpose: Detection of suitable place in text for switching Current Participant

Possible values: List of Unicode editing codes

Default value: Line Separator, Paragraph Separator, CR, CRLF, LF

Comment: Other than Latin based scripts may have other conventions

Parameter: Pending period

Purpose: Inactivity timer for detection of time to Switch Current Participant

Possible values: Time in seconds

Default value: 7

Comment: Longer times may cause inefficient transmission. Shorter time may cause unwanted switching cutting lines of thought inconveniently

Parameter: Sentence delimiter

Purpose: Characters forming end of sentence

Possible values: List of delimiters.

Default value: . or ? or ! followed by a space

Comment: Used for deciding on a position in the text to switch Current Participant according to configured logic.

Parameter: Label length

Purpose: Length of label put in front of or above entry.

Possible values: Number of characters

Default value: 12

Comment: Includes any surrounding characters

Parameter: Label delimiters

Purpose: Set of characters at the edges of the label

Possible values: Two strings. One in the beginning, one after.

Default value: [] followed by a space

Comment: It may be valid to include a Line Separator instead of the space

Parameter: Maximum waiting time

Purpose: The maximum time any participant's text shall be allowed to wait for transmission

Possible values: Seconds

Default value: 20

Comment After this time a Switch will be forced within the Time Extension

Parameter: Word delimiter

Purpose: Delimiter for words

Possible values: List of characters

Default value: Space

Comment: Used for detection of suitable switch position if Maximum Waiting time has passed.

Parameter: Time extension

Purpose: Time for maximum further waiting for a Switch Reason

Possible values: Time in seconds

Default value: 7

Comment: After this time a Switch is forced.

#### A.7. References for this Appendix

[T.140] ITU-T T.140 Application protocol, text conversation (including amendment 1.)

[RFC 4103] IETF RFC 4103 RTP Payload for text conversation

[RTP] IETF RFC 3550 RTP: A Transport Protocol for Real-Time Applications.

[RFC 4579] IETF RFC 4579 SIP Call Control ? Conferencing for user agents.

[ISO 6429] ISO 6429 Control functions for coded character sets.

[UTF-8] IETF RFC 3629 UTF-8, a transformation format of ISO 10646

[Unicode] The Unicode Consortium, "The Unicode Standard ? Version 4.0?"

[ISO 10?646-1] ISO 10?646 Universal multiple-octet coded character set (UCS)

[UCS-16] See ISO 10?646-1

## A.8. Acknowledgement

This appendix was developed with funding in part from the National Institute on Disability and Rehabilitation Research, U.S. Department of Education, RERC on Telecommunications Access, grant # H133E090001. However, the contents do not necessarily represent the policy of the Department of Education, and you should not assume endorsement by the Federal Government.

## Author's Address

Gunnar Hellstrom  
Omnitor  
Esplanaden 30  
Vendelso SE-136 70  
SE

Phone: +46 708 204 288  
Email: [gunnar.hellstrom@omnitor.se](mailto:gunnar.hellstrom@omnitor.se)  
URI: [www.omnitor.se](http://www.omnitor.se)

IETF RMCAT Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: May 6, 2021

Z. Sarker  
Ericsson AB  
C. Perkins  
University of Glasgow  
V. Singh  
callstats.io  
M. Ramalho  
November 2, 2020

RTP Control Protocol (RTCP) Feedback for Congestion Control  
draft-ietf-avtcore-cc-feedback-message-09

Abstract

An effective RTP congestion control algorithm requires more fine-grained feedback on packet loss, timing, and ECN marks than is provided by the standard RTP Control Protocol (RTCP) Sender Report (SR) and Receiver Report (RR) packets. This document describes an RTCP feedback message intended to enable congestion control for interactive real-time traffic using RTP. The feedback message is designed for use with a sender-based congestion control algorithm, in which the receiver of an RTP flow sends RTCP feedback packets to the sender containing the information the sender needs to perform congestion control.

Status of This Memo

This Internet-Draft is submitted 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 <https://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 May 6, 2021.

Copyright Notice

Copyright (c) 2020 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 (<https://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 . . . . .	2
2. Terminology . . . . .	3
3. RTCP Feedback for Congestion Control . . . . .	3
3.1. RTCP Congestion Control Feedback Report . . . . .	4
4. Feedback Frequency and Overhead . . . . .	7
5. Response to Loss of Feedback Packets . . . . .	8
6. SDP Signalling . . . . .	8
7. Relation to RFC 6679 . . . . .	9
8. Design Rationale . . . . .	10
9. Acknowledgements . . . . .	11
10. IANA Considerations . . . . .	11
11. Security Considerations . . . . .	12
12. References . . . . .	13
12.1. Normative References . . . . .	13
12.2. Informative References . . . . .	14
Authors' Addresses . . . . .	15

## 1. Introduction

For interactive real-time traffic, such as video conferencing flows, the typical protocol choice is the Real-time Transport Protocol (RTP) [RFC3550] running over the User Datagram Protocol (UDP). RTP does not provide any guarantee of Quality of Service (QoS), reliability, or timely delivery, and expects the underlying transport protocol to do so. UDP alone certainly does not meet that expectation. However, the RTP Control Protocol (RTCP) [RFC3550] provides a mechanism by which the receiver of an RTP flow can periodically send transport and media quality metrics to the sender of that RTP flow. This information can be used by the sender to perform congestion control. In the absence of standardized messages for this purpose, designers of congestion control algorithms have developed proprietary RTCP messages that convey only those parameters needed for their respective designs. As a direct result, the different congestion control designs are not interoperable. To enable algorithm evolution as well as interoperability across designs (e.g., different rate

adaptation algorithms), it is highly desirable to have a generic congestion control feedback format.

To help achieve interoperability for unicast RTP congestion control, this memo proposes a common RTCP feedback packet format that can be used by NADA [RFC8698], SReAM [RFC8298], Google Congestion Control [I-D.ietf-rmcat-gcc] and Shared Bottleneck Detection [RFC8382], and hopefully also by future RTP congestion control algorithms.

## 2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

In addition the terminology defined in [RFC3550], [RFC4585], and [RFC5506] applies.

## 3. RTCP Feedback for Congestion Control

Based on an analysis of NADA [RFC8698], SReAM [RFC8298], Google Congestion Control [I-D.ietf-rmcat-gcc] and Shared Bottleneck Detection [RFC8382], the following per-RTP packet congestion control feedback information has been determined to be necessary:

- o RTP sequence number: The receiver of an RTP flow needs to feed the sequence numbers of the received RTP packets back to the sender, so the sender can determine which packets were received and which were lost. Packet loss is used as an indication of congestion by many congestion control algorithms.
- o Packet Arrival Time: The receiver of an RTP flow needs to feed the arrival time of each RTP packet back to the sender. Packet delay and/or delay variation (jitter) is used as a congestion signal by some congestion control algorithms.
- o Packet Explicit Congestion Notification (ECN) Marking: If ECN [RFC3168], [RFC6679] is used, it is necessary to feed back the 2-bit ECN mark in received RTP packets, indicating for each RTP packet whether it is marked not-ECT, ECT(0), ECT(1), or ECN-CE. If the path used by the RTP traffic is ECN capable the sender can use Congestion Experienced (ECN-CE) marking information as a congestion control signal.

Every RTP flow is identified by its Synchronization Source (SSRC) identifier. Accordingly, the RTCP feedback format needs to group its reports by SSRC, sending one report block per received SSRC.

As a practical matter, we note that host operating system (OS) process interruptions can occur at inopportune times. Accordingly, recording RTP packet send times at the sender, and the corresponding RTP packet arrival times at the receiver, needs to be done with deliberate care. This is because the time duration of host OS interruptions can be significant relative to the precision desired in the one-way delay estimates. Specifically, the send time needs to be recorded at the last opportunity prior to transmitting the RTP packet at the sender, and the arrival time at the receiver needs to be recorded at the earliest available opportunity.

### 3.1. RTCP Congestion Control Feedback Report

Congestion control feedback can be sent as part of a regular scheduled RTCP report, or in an RTP/AVPF early feedback packet. If sent as early feedback, congestion control feedback MAY be sent in a non-compound RTCP packet [RFC5506] if the RTP/AVPF profile [RFC4585] or the RTP/SAVPF profile [RFC5124] is used.

Irrespective of how it is transported, the congestion control feedback is sent as a Transport Layer Feedback Message (RTCP packet type 205). The format of this RTCP packet is shown in Figure 1:



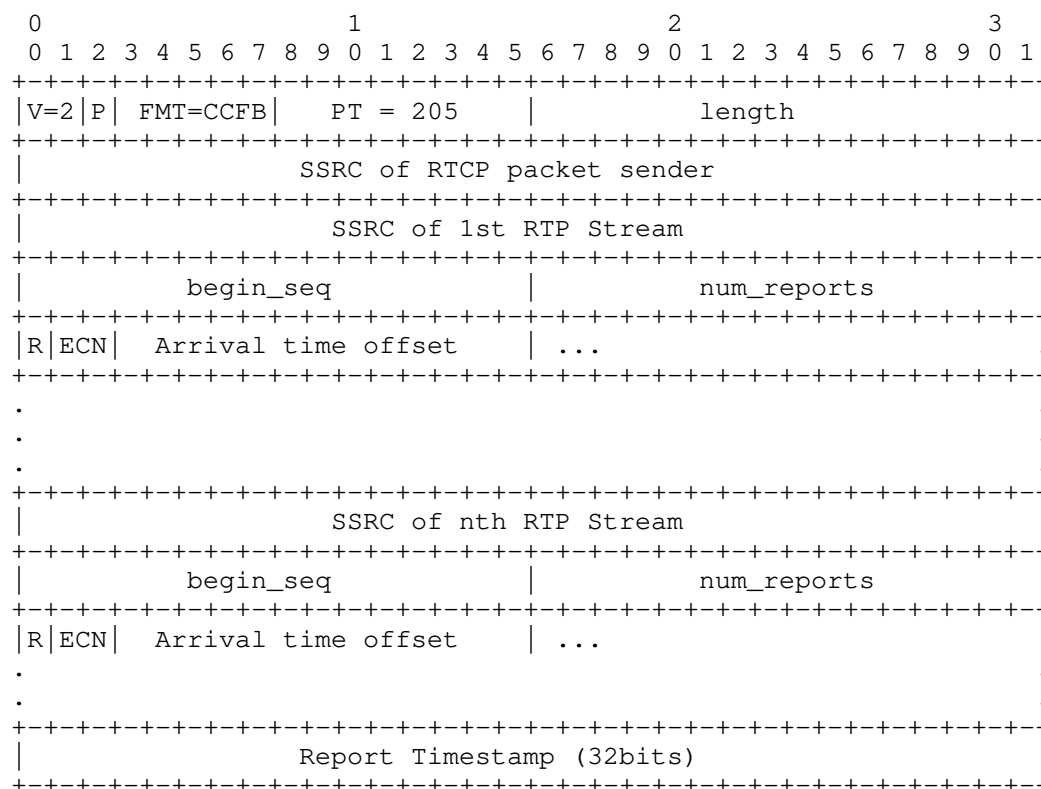


Figure 1: RTCP Congestion Control Feedback Packet Format

The first eight octets comprise a standard RTCP header, with PT=205 and FMT=CCFB indicating that this is a congestion control feedback packet, and with the SSRC set to that of the sender of the RTCP packet. (NOTE TO RFC EDITOR: please replace CCFB here and in the above diagram with the IANA assigned RTCP feedback packet type, and remove this note)

Section 6.1 of [RFC4585] requires the RTCP header to be followed by the SSRC of the RTP flow being reported upon. Accordingly, the RTCP header is followed by a report block for each SSRC from which RTP packets have been received, followed by a Report Timestamp.

Each report block begins with the SSRC of the received RTP Stream on which it is reporting. Following this, the report block contains a 16-bit packet metric block for each RTP packet with sequence number in the range begin\_seq to begin\_seq+num\_reports inclusive (calculated using arithmetic modulo 65536 to account for possible sequence number wrap-around). If the number of 16-bit packet metric blocks included

in the report block is not a multiple of two, then 16 bits of zero padding MUST be added after the last packet metric block, to align the end of the packet metric blocks with the next 32 bit boundary. The value of num\_reports MAY be zero, indicating that there are no packet metric blocks included for that SSRC. Each report block MUST NOT include more than 16384 packet metric blocks (i.e., it MUST NOT report on more than one quarter of the sequence number space in a single report).

The contents of each 16-bit packet metric block comprises the R, ECN, and ATO fields as follows:

- o Received (R, 1 bit): is a boolean to indicate if the packet was received. 0 represents that the packet was not yet received and the subsequent 15-bits (ECN and ATO) in this 16-bit packet metric block are also set to 0 and MUST be ignored. 1 represents that the packet was received and the subsequent bits in the block need to be parsed.
- o ECN (2 bits): is the echoed ECN mark of the packet. These are set to 00 if not received, or if ECN is not used.
- o Arrival time offset (ATO, 13 bits): is the arrival time of the RTP packet at the receiver, as an offset before the time represented by the Report Timestamp (RTS) field of this RTCP congestion control feedback report. The ATO field is in units of 1/1024 seconds (this unit is chosen to give exact offsets from the RTS field) so, for example, an ATO value of 512 indicates that the corresponding RTP packet arrived exactly half a second before the time instant represented by the RTS field. If the measured value is greater than 8189/1024 seconds (the value that would be coded as 0x1FFD), the value 0x1FFE MUST be reported to indicate an over-range measurement. If the measurement is unavailable, or if the arrival time of the RTP packet is after the time represented by the RTS field, then an ATO value of 0x1FFF MUST be reported for the packet.

The RTCP congestion control feedback report packet concludes with the Report Timestamp field (RTS, 32 bits). This denotes the time instant on which this packet is reporting, and is the instant from which the arrival time offset values are calculated. The value of RTS field is derived from the same clock used to generate the NTP timestamp field in RTCP Sender Report (SR) packets. It is formatted as the middle 32 bits of an NTP format timestamp, as described in Section 4 of [RFC3550].

RTCP congestion control feedback packets SHOULD include a report block for every active SSRC. The sequence number ranges reported on

in consecutive reports for a given SSRC will generally be contiguous, but overlapping reports MAY be sent (and need to be sent in cases where RTP packet reordering occurs across the boundary between consecutive reports). If an RTP packet was reported as received in one report, that packet MUST also be reported as received in any overlapping reports sent later that cover its sequence number range. If reports covering overlapping sequence number ranges are sent, information in later reports updates that sent in previous reports for RTP packets included in both reports.

RTCP congestion control feedback packets can be large if they are sent infrequently relative to the number of RTP data packets. If an RTCP congestion control feedback packet is too large to fit within the path MTU, its sender SHOULD split it into multiple feedback packets. The RTCP reporting interval SHOULD be chosen such that feedback packets are sent often enough that they are small enough to fit within the path MTU ([I-D.ietf-rmcat-rtp-cc-feedback] discusses how to choose the reporting interval; specifications for RTP congestion control algorithms can also provide guidance).

If duplicate copies of a particular RTP packet are received, then the arrival time of the first copy to arrive MUST be reported. If any of the copies of the duplicated packet are ECN-CE marked, then an ECN-CE mark MUST be reported that for packet; otherwise the ECN mark of the first copy to arrive is reported.

If no packets are received from an SSRC in a reporting interval, a report block MAY be sent with `begin_seq` set to the highest sequence number previously received from that SSRC and `num_reports` set to zero (or, the report can simply be omitted). The corresponding SR/RR packet will have a non-increased extended highest sequence number received field that will inform the sender that no packets have been received, but it can ease processing to have that information available in the congestion control feedback reports too.

A report block indicating that certain RTP packets were lost is not to be interpreted as a request to retransmit the lost packets. The receiver of such a report might choose to retransmit such packets, provided a retransmission payload format has been negotiated, but there is no requirement that it do so.

#### 4. Feedback Frequency and Overhead

There is a trade-off between speed and accuracy of reporting, and the overhead of the reports. [I-D.ietf-rmcat-rtp-cc-feedback] discusses this trade-off, suggests desirable RTCP feedback rates, and provides guidance on how to configure the RTCP bandwidth fraction, etc., to make appropriate use of the reporting block described in this memo.

Specifications for RTP congestion control algorithms can also provide guidance.

It is generally understood that congestion control algorithms work better with more frequent feedback. However, RTCP bandwidth and transmission rules put some upper limits on how frequently the RTCP feedback messages can be sent from an RTP receiver to the RTP sender. In many cases, sending feedback once per frame is an upper bound before the reporting overhead becomes excessive, although this will depend on the media rate and more frequent feedback might be needed with high-rate media flows [I-D.ietf-rmcat-rtp-cc-feedback]. Analysis [feedback-requirements] has also shown that some candidate congestion control algorithms can operate with less frequent feedback, using a feedback interval range of 50-200ms. Applications need to negotiate an appropriate congestion control feedback interval at session setup time, based on the choice of congestion control algorithm, the expected media bit rate, and the acceptable feedback overhead.

## 5. Response to Loss of Feedback Packets

Like all RTCP packets, RTCP congestion control feedback packets might be lost. All RTP congestion control algorithms MUST specify how they respond to the loss of feedback packets.

RTCP packets do not contain a sequence number, so loss of feedback packets has to be inferred based on the time since the last feedback packet. If only a single congestion control feedback packet is lost, an appropriate response is to assume that the level of congestion has remained roughly the same as the previous report. However, if multiple consecutive congestion control feedback packets are lost, then the media sender SHOULD rapidly reduce its sending rate as this likely indicates a path failure. The RTP circuit breaker [RFC8083] provides further guidance.

## 6. SDP Signalling

A new "ack" feedback parameter, "ccfb", is defined for use with the "a=rtcp-fb:" SDP extension to indicate the use of the RTP Congestion Control feedback packet format defined in Section 3. The ABNF definition of this SDP parameter extension is:

```
rtcp-fb-ack-param = <See Section 4.2 of [RFC4585]>
rtcp-fb-ack-param =/ ccfb-par
ccfb-par           = SP "ccfb"
```

The payload type used with "ccfb" feedback MUST be the wildcard type ("\*"). This implies that the congestion control feedback is sent for

all payload types in use in the session, including any FEC and retransmission payload types. An example of the resulting SDP attribute is:

```
a=rtcp-fb:* ack ccfb
```

The offer/answer rules for these SDP feedback parameters are specified in Section 4.2 of the RTP/AVPF profile [RFC4585].

An SDP offer might indicate support for both the congestion control feedback mechanism specified in this memo and one or more alternative congestion control feedback mechanisms that offer substantially the same semantics. In this case, the answering party **SHOULD** include only one of the offered congestion control feedback mechanisms in its answer. If a re-invite offering the same set of congestion control feedback mechanisms is received, the generated answer **SHOULD** choose the same congestion control feedback mechanism as in the original answer where possible.

When the SDP BUNDLE extension [I-D.ietf-mmusic-sdp-bundle-negotiation] is used for multiplexing, the "a=rtcp-fb:" attribute has multiplexing category IDENTICAL-PER-PT [I-D.ietf-mmusic-sdp-mux-attributes].

## 7. Relation to RFC 6679

Use of Explicit Congestion Notification (ECN) with RTP is described in [RFC6679]. That specifies how to negotiate the use of ECN with RTP, and defines an RTCP ECN Feedback Packet to carry ECN feedback reports. It uses an SDP "a=ecn-capable-rtp:" attribute to negotiate use of ECN, and the "a=rtcp-fb:" attributes with the "nack" parameter "ecn" to negotiate the use of RTCP ECN Feedback Packets.

The RTCP ECN Feedback Packet is not useful when ECN is used with the RTP Congestion Control Feedback Packet defined in this memo since it provides duplicate information. When congestion control feedback is to be used with RTP and ECN, the SDP offer generated **MUST** include an "a=ecn-capable-rtp:" attribute to negotiate ECN support, along with an "a=rtcp-fb:" attribute with the "ack" parameter "ccfb" to indicate that the RTP Congestion Control Feedback Packet can be used. The "a=rtcp-fb:" attribute **MAY** also include the "nack" parameter "ecn", to indicate that the RTCP ECN Feedback Packet is also supported. If an SDP offer signals support for both RTP Congestion Control Feedback Packets and the RTCP ECN Feedback Packet, the answering party **SHOULD** signal support for one, but not both, formats in its SDP answer to avoid sending duplicate feedback.

When using ECN with RTP, the guidelines in Section 7.2 of [RFC6679] MUST be followed to initiate the use of ECN in an RTP session. The guidelines in Section 7.3 of [RFC6679] MUST also be followed about ongoing use of ECN within an RTP session, with the exception that feedback is sent using the RTCP Congestion Control Feedback Packets described in this memo rather than using RTP ECN Feedback Packets. Similarly, the guidance in Section 7.4 of [RFC6679] around detecting failures MUST be followed, with the exception that the necessary information is retrieved from the RTCP Congestion Control Feedback Packets rather than from RTP ECN Feedback Packets.

## 8. Design Rationale

The primary function of RTCP SR/RR packets is to report statistics on the reception of RTP packets. The reception report blocks sent in these packets contain information about observed jitter, fractional packet loss, and cumulative packet loss. It was intended that this information could be used to support congestion control algorithms, but experience has shown that it is not sufficient for that purpose. An efficient congestion control algorithm requires more fine-grained information on per-packet reception quality than is provided by SR/RR packets to react effectively. The feedback format defined in this memo provides such fine-grained feedback.

Several other RTCP extensions also provide more detailed feedback than SR/RR packets:

**TMMBR:** The Codec Control Messages for the RTP/AVPF profile [RFC5104] include a Temporary Maximum Media Bit Rate (TMMBR) message. This is used to convey a temporary maximum bit rate limitation from a receiver of RTP packets to their sender. Even though it was not designed to replace congestion control, TMMBR has been used as a means to do receiver based congestion control where the session bandwidth is high enough to send frequent TMMBR messages, especially when used with non-compound RTCP packets [RFC5506]. This approach requires the receiver of the RTP packets to monitor their reception, determine the level of congestion, and recommend a maximum bit rate suitable for current available bandwidth on the path; it also assumes that the RTP sender can/will respect that bit rate. This is the opposite of the sender-based congestion control approach suggested in this memo, so TMMBR cannot be used to convey the information needed for a sender-based congestion control. TMMBR could, however, be viewed a complementary mechanism that can inform the sender of the receiver's current view of acceptable maximum bit rate. Mechanisms that convey the receiver's estimate of the maximum available bit-rate provide similar feedback.

RTCP Extended Reports (XR): Numerous RTCP extended report (XR) blocks have been defined to report details of packet loss, arrival times [RFC3611], delay [RFC6843], and ECN marking [RFC6679]. It is possible to combine several such XR blocks into a compound RTCP packet, to report the detailed loss, arrival time, and ECN marking information needed for effective sender-based congestion control. However, the result has high overhead both in terms of bandwidth and complexity, due to the need to stack multiple reports.

Transport-wide Congestion Control: The format defined in this memo provides individual feedback on each SSRC. An alternative is to add a header extension to each RTP packet, containing a single, transport-wide, packet sequence number, then have the receiver send RTCP reports giving feedback on these additional sequence numbers [I-D.holmer-rmcat-transport-wide-cc-extensions]. Such an approach adds the per-packet overhead of the header extension (8 octets per packet in the referenced format), but reduces the size of the feedback packets, and can simplify the rate calculation at the sender if it maintains a single rate limit that applies to all RTP packets sent irrespective of their SSRC. Equally, the use of transport-wide feedback makes it more difficult to adapt the sending rate, or respond to lost packets, based on the reception and/or loss patterns observed on a per-SSRC basis (for example, to perform differential rate control and repair for audio and video flows, based on knowledge of what packets from each flow were lost). Transport-wide feedback is also a less natural fit with the wider RTP framework, which makes extensive use of per-SSRC sequence numbers and feedback.

Considering these issues, we believe it appropriate to design a new RTCP feedback mechanism to convey information for sender-based congestion control algorithms. The new congestion control feedback RTCP packet described in Section 3 provides such a mechanism.

## 9. Acknowledgements

This document is based on the outcome of a design team discussion in the RTP Media Congestion Avoidance Techniques (RMCAT) working group. The authors would like to thank David Hayes, Stefan Holmer, Randell Jesup, Ingemar Johansson, Jonathan Lennox, Sergio Mena, Nils Ohlmeier, Magnus Westerlund, and Xiaoqing Zhu for their valuable feedback.

## 10. IANA Considerations

The IANA is requested to register one new RTP/AVPF Transport-Layer Feedback Message in the table for FMT values for RTPFB Payload Types [RFC4585] as defined in Section 3.1:

Name: CCFB  
Long name: RTP Congestion Control Feedback  
Value: (to be assigned by IANA)  
Reference: (RFC number of this document, when published)

The IANA is also requested to register one new SDP "rtcp-fb" attribute "ack" parameter, "ccfb", in the SDP ("ack" and "nack" Attribute Values) registry:

Value name: ccfb  
Long name: Congestion Control Feedback  
Usable with: ack  
Mux: IDENTICAL-PER-PT  
Reference: (RFC number of this document, when published)

## 11. Security Considerations

The security considerations of the RTP specification [RFC3550], the applicable RTP profile (e.g., [RFC3551], [RFC3711], or [RFC4585]), and the RTP congestion control algorithm that is in use (e.g., [RFC8698], [RFC8298], [I-D.ietf-rmcat-gcc], or [RFC8382]) apply.

A receiver that intentionally generates inaccurate RTCP congestion control feedback reports might be able to trick the sender into sending at a greater rate than the path can support, thereby causing congestion on the path. This will negatively impact the quality of experience of that receiver, and potentially cause denial of service to other traffic sharing the path and excessive resource usage at the media sender. Since RTP is an unreliable transport, a sender can intentionally drop a packet, leaving a gap in the RTP sequence number space without causing serious harm, to check that the receiver is correctly reporting losses (this needs to be done with care and some awareness of the media data being sent, to limit impact on the user experience).

An on-path attacker that can modify RTCP congestion control feedback packets can change the reports to trick the sender into sending at either an excessively high or excessively low rate, leading to denial of service. The secure RTCP profile [RFC3711] can be used to authenticate RTCP packets to protect against this attack.

An off-path attacker that can spoof RTCP congestion control feedback packets can similarly trick a sender into sending at an incorrect rate, leading to denial of service. This attack is difficult, since the attacker needs to guess the SSRC and sequence number in addition to the destination transport address. As with on-path attacks, the secure RTCP profile [RFC3711] can be used to authenticate RTCP packets to protect against this attack.



## 12. References

### 12.1. Normative References

- [I-D.ietf-mmusic-sdp-bundle-negotiation]  
Holmberg, C., Alvestrand, H., and C. Jennings,  
"Negotiating Media Multiplexing Using the Session  
Description Protocol (SDP)", draft-ietf-mmusic-sdp-bundle-  
negotiation-54 (work in progress), December 2018.
- [I-D.ietf-mmusic-sdp-mux-attributes]  
Nandakumar, S., "A Framework for SDP Attributes when  
Multiplexing", draft-ietf-mmusic-sdp-mux-attributes-19  
(work in progress), August 2020.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate  
Requirement Levels", BCP 14, RFC 2119,  
DOI 10.17487/RFC2119, March 1997,  
<<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3168] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition  
of Explicit Congestion Notification (ECN) to IP",  
RFC 3168, DOI 10.17487/RFC3168, September 2001,  
<<https://www.rfc-editor.org/info/rfc3168>>.
- [RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V.  
Jacobson, "RTP: A Transport Protocol for Real-Time  
Applications", STD 64, RFC 3550, DOI 10.17487/RFC3550,  
July 2003, <<https://www.rfc-editor.org/info/rfc3550>>.
- [RFC3551] Schulzrinne, H. and S. Casner, "RTP Profile for Audio and  
Video Conferences with Minimal Control", STD 65, RFC 3551,  
DOI 10.17487/RFC3551, July 2003,  
<<https://www.rfc-editor.org/info/rfc3551>>.
- [RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K.  
Norrman, "The Secure Real-time Transport Protocol (SRTP)",  
RFC 3711, DOI 10.17487/RFC3711, March 2004,  
<<https://www.rfc-editor.org/info/rfc3711>>.
- [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,  
DOI 10.17487/RFC4585, July 2006,  
<<https://www.rfc-editor.org/info/rfc4585>>.

- [RFC5124] Ott, J. and E. Carrara, "Extended Secure RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/SAVPF)", RFC 5124, DOI 10.17487/RFC5124, February 2008, <<https://www.rfc-editor.org/info/rfc5124>>.
- [RFC5506] Johansson, I. and M. Westerlund, "Support for Reduced-Size Real-Time Transport Control Protocol (RTCP): Opportunities and Consequences", RFC 5506, DOI 10.17487/RFC5506, April 2009, <<https://www.rfc-editor.org/info/rfc5506>>.
- [RFC6679] Westerlund, M., Johansson, I., Perkins, C., O'Hanlon, P., and K. Carlberg, "Explicit Congestion Notification (ECN) for RTP over UDP", RFC 6679, DOI 10.17487/RFC6679, August 2012, <<https://www.rfc-editor.org/info/rfc6679>>.
- [RFC8083] Perkins, C. and V. Singh, "Multimedia Congestion Control: Circuit Breakers for Unicast RTP Sessions", RFC 8083, DOI 10.17487/RFC8083, March 2017, <<https://www.rfc-editor.org/info/rfc8083>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

## 12.2. Informative References

- [feedback-requirements]  
"RMCAT Feedback Requirements",  
<[://www.ietf.org/proceedings/95/slides/slides-95-rmcat-1.pdf](https://www.ietf.org/proceedings/95/slides/slides-95-rmcat-1.pdf)>.
- [I-D.alvestrand-rmcat-remb]  
Alvestrand, H., "RTCP message for Receiver Estimated Maximum Bitrate", draft-alvestrand-rmcat-remb-03 (work in progress), October 2013.
- [I-D.holmer-rmcat-transport-wide-cc-extensions]  
Holmer, S., Flodman, M., and E. Sprang, "RTP Extensions for Transport-wide Congestion Control", draft-holmer-rmcat-transport-wide-cc-extensions-01 (work in progress), October 2015.
- [I-D.ietf-rmcat-gcc]  
Holmer, S., Lundin, H., Carlucci, G., Cicco, L., and S. Mascolo, "A Google Congestion Control Algorithm for Real-Time Communication", draft-ietf-rmcat-gcc-02 (work in progress), July 2016.

- [I-D.ietf-rmcat-rtp-cc-feedback]  
Perkins, C., "RTP Control Protocol (RTCP) Feedback for Congestion Control in Interactive Multimedia Conferences", draft-ietf-rmcat-rtp-cc-feedback-05 (work in progress), November 2019.
- [RFC3611] Friedman, T., Ed., Caceres, R., Ed., and A. Clark, Ed., "RTP Control Protocol Extended Reports (RTCP XR)", RFC 3611, DOI 10.17487/RFC3611, November 2003, <<https://www.rfc-editor.org/info/rfc3611>>.
- [RFC5104] Wenger, S., Chandra, U., Westerlund, M., and B. Burman, "Codec Control Messages in the RTP Audio-Visual Profile with Feedback (AVPF)", RFC 5104, DOI 10.17487/RFC5104, February 2008, <<https://www.rfc-editor.org/info/rfc5104>>.
- [RFC6843] Clark, A., Gross, K., and Q. Wu, "RTP Control Protocol (RTCP) Extended Report (XR) Block for Delay Metric Reporting", RFC 6843, DOI 10.17487/RFC6843, January 2013, <<https://www.rfc-editor.org/info/rfc6843>>.
- [RFC8298] Johansson, I. and Z. Sarker, "Self-Clocked Rate Adaptation for Multimedia", RFC 8298, DOI 10.17487/RFC8298, December 2017, <<https://www.rfc-editor.org/info/rfc8298>>.
- [RFC8382] Hayes, D., Ed., Ferlin, S., Welzl, M., and K. Hiorth, "Shared Bottleneck Detection for Coupled Congestion Control for RTP Media", RFC 8382, DOI 10.17487/RFC8382, June 2018, <<https://www.rfc-editor.org/info/rfc8382>>.
- [RFC8698] Zhu, X., Pan, R., Ramalho, M., and S. Mena, "Network-Assisted Dynamic Adaptation (NADA): A Unified Congestion Control Scheme for Real-Time Media", RFC 8698, DOI 10.17487/RFC8698, February 2020, <<https://www.rfc-editor.org/info/rfc8698>>.

#### Authors' Addresses

Zaheduzzaman Sarker  
Ericsson AB  
Torshamnsgatan 21  
Stockholm 164 40  
Sweden

Phone: +46107173743  
Email: [zaheduzzaman.sarker@ericsson.com](mailto:zaheduzzaman.sarker@ericsson.com)

Colin Perkins  
University of Glasgow  
School of Computing Science  
Glasgow G12 8QQ  
United Kingdom

Email: [csp@csp Perkins.org](mailto:csp@csp Perkins.org)

Varun Singh  
CALLSTATS I/O Oy  
Annankatu 31-33 C 42  
Helsinki 00100  
Finland

Email: [varun.singh@iki.fi](mailto:varun.singh@iki.fi)  
URI: <http://www.callstats.io/>

Michael A. Ramalho  
6310 Watercrest Way Unit 203  
Lakewood Ranch, FL 34202-5122  
USA

Phone: +1 732 832 9723  
Email: [mar42@cornell.edu](mailto:mar42@cornell.edu)  
URI: <http://ramalho.webhop.info/>

AVTCORE WG  
Internet-Draft  
Updates: 3550, 3551 (if approved)  
Intended status: Standards Track  
Expires: June 20, 2016

M. Westerlund  
Ericsson  
C. Perkins  
University of Glasgow  
J. Lennox  
Vidyo  
December 18, 2015

Sending Multiple Types of Media in a Single RTP Session  
draft-ietf-avtccore-multi-media-rtp-session-13

Abstract

This document specifies how an RTP session can contain RTP Streams with media from multiple media types such as audio, video, and text. This has been restricted by the RTP Specification, and thus this document updates RFC 3550 and RFC 3551 to enable this behaviour for applications that satisfy the applicability for using multiple media types in a single RTP session.

Status of This Memo

This Internet-Draft is submitted 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 June 20, 2016.

Copyright Notice

Copyright (c) 2015 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 . . . . .	2
2. Terminology . . . . .	3
3. Background and Motivation . . . . .	3
4. Applicability . . . . .	4
5. Using Multiple Media Types in a Single RTP Session . . . . .	6
5.1. Allowing Multiple Media Types in an RTP Session . . . . .	6
5.2. Demultiplexing media types within an RTP session . . . . .	7
5.3. Per-SSRC Media Type Restrictions . . . . .	8
5.4. RTCP Considerations . . . . .	8
6. Extension Considerations . . . . .	9
6.1. RTP Retransmission Payload Format . . . . .	9
6.2. RTP Payload Format for Generic FEC . . . . .	10
6.3. RTP Payload Format for Redundant Audio . . . . .	11
7. Signalling . . . . .	12
8. Security Considerations . . . . .	12
9. IANA Considerations . . . . .	13
10. Acknowledgements . . . . .	13
11. References . . . . .	13
11.1. Normative References . . . . .	13
11.2. Informative References . . . . .	14
Authors' Addresses . . . . .	15

## 1. Introduction

The Real-time Transport Protocol [RFC3550] was designed to use separate RTP sessions to transport different types of media. This implies that different transport layer flows are used for different RTP streams. For example, a video conferencing application might send audio and video traffic RTP flows on separate UDP ports. With increased use of network address/port translation, firewalls, and other middleboxes it is, however, becoming difficult to establish multiple transport layer flows between endpoints. Hence, there is pressure to reduce the number of concurrent transport flows used by RTP applications.

This memo updates [RFC3550] and [RFC3551] to allow multiple media types to be sent in a single RTP session in certain cases, thereby reducing the number of transport layer flows that are needed. It makes no changes to RTP behaviour when using multiple RTP streams containing media of the same type (e.g., multiple audio streams or multiple video streams) in a single RTP session. However

[I-D.ietf-avtcore-rtp-multi-stream] provides important clarifications to RTP behaviour in that case.

This memo is structured as follows. Section 2 defines terminology. Section 3 further describes the background to, and motivation for, this memo and Section 4 describes the scenarios where this memo is applicable. Section 5 discusses issues arising from the base RTP and RTCP specification when using multiple types of media in a single RTP session, while Section 6 considers the impact of RTP extensions. We discuss signalling in Section 7. Finally, security considerations are discussed in Section 8.

## 2. Terminology

The terms Encoded Stream, Endpoint, Media Source, RTP Session, and RTP Stream are used as defined in [RFC7656]. We also define the following terms:

**Media Type:** The general type of media data used by a real-time application. The media type corresponds to the value used in the <media> field of an SDP m= line. The media types defined at the time of this writing are "audio", "video", "text", "image", "application", and "message". [RFC4566] [RFC6466]

**Quality of Service (QoS):** Network mechanisms that are intended to ensure that the packets within a flow or with a specific marking are transported with certain properties.

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

## 3. Background and Motivation

RTP was designed to support multimedia sessions, containing multiple types of media sent simultaneously, by using multiple transport layer flows. The existence of network address translators, firewalls, and other middleboxes complicates this, however, since a mechanism is needed to ensure that all the transport layer flows needed by the application can be established. This has three consequences:

1. increased delay to establish a complete session, since each of the transport layer flows needs to be negotiated and established;
2. increased state and resource consumption in the middleboxes that can lead to unexpected behaviour when middlebox resource limits are reached; and

3. increased risk that a subset of the transport layer flows will fail to be established, thus preventing the application from communicating.

Using fewer transport layer flows can hence be seen to reduce the risk of communication failure, and can lead to improved reliability and performance.

One of the benefits of using multiple transport layer flows is that it makes it easy to use network layer quality of service (QoS) mechanisms to give differentiated performance for different flows. However, we note that many RTP-using application don't use network QoS features, and don't expect or desire any separation in network treatment of their media packets, independent of whether they are audio, video or text. When an application has no such desire, it doesn't need to provide a transport flow structure that simplifies flow based QoS.

Given the above issues, it might seem appropriate for RTP-based applications to send all their RTP streams bundled into one RTP session, running over a single transport layer flow. However, this is prohibited by the RTP specification, because the design of RTP makes certain assumptions that can be incompatible with sending multiple media types in a single RTP session. Specifically, the RTP control protocol (RTCP) timing rules assume that all RTP media flows in a single RTP session have broadly similar RTCP reporting and feedback requirements, which can be problematic when different types of media are multiplexed together. Various RTP extensions also make assumptions about SSRC use and RTCP reporting that are incompatible with sending different media types in a single RTP session.

This memo updates [RFC3550] and [RFC3551] to allow RTP sessions to contain more than one media type in certain circumstances, and gives guidance on when it is safe to send multiple media types in a single RTP session.

#### 4. Applicability

This specification has limited applicability, and anyone intending to use it needs to ensure that their application and use case meets the following criteria:

Equal treatment of media: The use of a single RTP session normally results in similar network treatment for all types of media used within the session. Applications that require significantly different network quality of service (QoS) or RTCP configuration for different RTP streams are better suited by sending those RTP streams in separate RTP session, using separate transport layer



flows for each, since that gives greater flexibility. Further guidance on how to provide differential treatment for some media is given in [I-D.ietf-avtcore-multiplex-guidelines] and [RFC7657].

**Compatible RTCP Behaviour:** The RTCP timing rules enforce a single RTCP reporting interval for all participants in an RTP session. Flows with very different media sending rate or RTCP feedback requirements cannot be multiplexed together, since this leads to either excessive or insufficient RTCP for some flows, depending on how the RTCP session bandwidth, and hence reporting interval, is configured. For example, it is likely infeasible to find a single RTCP configuration that simultaneously suits both a low-rate audio flow with no feedback, and a high-quality video flow with sophisticated RTCP-based feedback. Thus, combining these into a single RTP session is difficult and/or inadvisable.

**Signalled Support:** The extensions defined in this memo are not compatible with unmodified [RFC3550]-compatible endpoints. Their use requires signalling and mutual agreement by all participants within an RTP session. This requirement can be a problem for signalling solutions that can't negotiate with all participants. For declarative signalling solutions, mandating that the session is using multiple media types in one RTP session can be a way of attempting to ensure that all participants in the RTP session follow the requirement. However, for signalling solutions that lack methods for enforcing that a receiver supports a specific feature, this can still cause issues.

**Consistent support for multiparty RTP sessions:** If it is desired to send multiple types of media in a multiparty RTP session, then all participants in that session need to support sending multiple type of media in a single RTP session. It is not possible, in the general case, to implement a gateway that can interconnect an endpoint using multiple types of media sent using separate RTP sessions, with one or more endpoints that send multiple types of media in a single RTP session.

One reason for this is that the same SSRC value can safely be used for different streams in multiple RTP sessions, but when collapsed to a single RTP session there is an SSRC collision. This would not be an issue, since SSRC collision detection will resolve the conflict, except that some RTP payload formats and extensions use matching SSRCS to identify related flows, and break when a single RTP session is used.

A middlebox that remaps SSRC values when combining multiple RTP sessions into one also needs to be aware of all possible RTCP packet types that might be used, so that it can remap the SSRC

values in those packets. This is impossible to do without restricting the set of RTCP packet types that can be used to those that are known by the middlebox. Such a middlebox might also have difficulty due to differences in configured RTCP bandwidth and other parameters between the RTP sessions.

Finally, the use of a middlebox that translates SSRC values can negatively impact the possibility for loop detection, as SSRC/CSRC can't be used to detect the loops; instead some other RTP stream or media source identity name space that is common across all interconnect parts is needed.

Ability to operate with limited payload type space: An RTP session has only a single 7-bit payload type space for all its payload type numbers. Some applications might find this space limiting when using different media types and RTP payload formats within a single RTP session.

Avoids incompatible Extensions: Some RTP and RTCP extensions rely on the existence of multiple RTP sessions and relate RTP streams between sessions. Others report on particular media types, and cannot be used with other media types. Applications that send multiple types of media into a single RTP session need to avoid such extensions.

## 5. Using Multiple Media Types in a Single RTP Session

This section defines what needs to be done or avoided to make an RTP session with multiple media types function without issues.

### 5.1. Allowing Multiple Media Types in an RTP Session

Section 5.2 of "RTP: A Transport Protocol for Real-Time Applications" [RFC3550] states:

For example, in a teleconference composed of audio and video media encoded separately, each medium SHOULD be carried in a separate RTP session with its own destination transport address.

Separate audio and video streams SHOULD NOT be carried in a single RTP session and demultiplexed based on the payload type or SSRC fields.

This specification changes both of these sentences. The first sentence is changed to:

For example, in a teleconference composed of audio and video media encoded separately, each medium SHOULD be carried in a separate

RTP session with its own destination transport address, unless specification [RFCXXXX] is followed and the application meets the applicability constraints.

The second sentence is changed to:

Separate audio and video media sources SHOULD NOT be carried in a single RTP session, unless the guidelines specified in [RFCXXXX] are followed.

Second paragraph of Section 6 in RTP Profile for Audio and Video Conferences with Minimal Control [RFC3551] says:

The payload types currently defined in this profile are assigned to exactly one of three categories or media types: audio only, video only and those combining audio and video. The media types are marked in Tables 4 and 5 as "A", "V" and "AV", respectively. Payload types of different media types SHALL NOT be interleaved or multiplexed within a single RTP session, but multiple RTP sessions MAY be used in parallel to send multiple media types. An RTP source MAY change payload types within the same media type during a session. See the section "Multiplexing RTP Sessions" of RFC 3550 for additional explanation.

This specification's purpose is to override that existing SHALL NOT under certain conditions. Thus this sentence also has to be changed to allow for multiple media type's payload types in the same session. The sentence containing "SHALL NOT" in the above paragraph is changed to:

Payload types of different media types SHALL NOT be interleaved or multiplexed within a single RTP session unless [RFCXXXX] is used, and the application conforms to the applicability constraints. Multiple RTP sessions MAY be used in parallel to send multiple media types.

RFC-Editor Note: Please replace RFCXXXX with the RFC number of this specification when assigned.

## 5.2. Demultiplexing media types within an RTP session

When receiving packets from a transport layer flow, an endpoint will first separate the RTP and RTCP packets from the non-RTP packets, and pass them to the RTP/RTCP protocol handler. The RTP and RTCP packets are then demultiplexed based on their SSRC into the different RTP streams. For each RTP stream, incoming RTCP packets are processed, and the RTP payload type is used to select the appropriate media

decoder. This process remains the same irrespective of whether multiple media types are sent in a single RTP session or not.

As explained below, it is important to note that the RTP payload type is never used to distinguish RTP streams. The RTP packets are demultiplexed into RTP streams based on their SSRC, then the RTP payload type is used to select the correct media decoding pathway for each RTP stream.

### 5.3. Per-SSRC Media Type Restrictions

An SSRC in an RTP session can change between media formats of the same type, subject to certain restrictions [RFC7160], but MUST NOT change media type during its lifetime. For example, an SSRC can change between different audio formats, but cannot start sending audio then change to sending video. The lifetime of an SSRC ends when an RTCP BYE packet for that SSRC is sent, or when it ceases transmission for long enough that it times out for the other participants in the session.

The main motivation is that a given SSRC has its own RTP timestamp and sequence number spaces. The same way that you can't send two encoded streams of audio with the same SSRC, you can't send one encoded audio and one encoded video stream with the same SSRC. Each encoded stream when made into an RTP stream needs to have the sole control over the sequence number and timestamp space. If not, one would not be able to detect packet loss for that particular encoded stream. Nor can one easily determine which clock rate a particular SSRCs timestamp will increase with. For additional arguments why RTP payload type based multiplexing of multiple media sources doesn't work, see [I-D.ietf-avtcore-multiplex-guidelines].

Within an RTP session where multiple media types have been configured for use, an SSRC can only send one type of media during its lifetime (i.e., it can switch between different audio codecs, since those are both the same type of media, but cannot switch between audio and video). Different SSRCs MUST be used for the different media sources, the same way multiple media sources of the same media type already have to do. The payload type will inform a receiver which media type the SSRC is being used for. Thus the payload type MUST be unique across all of the payload configurations independent of media type that is used in the RTP session.

### 5.4. RTCP Considerations

When sending multiple types of media that have different rates in a single RTP session, endpoints MUST follow the guidelines for handling RTCP described in Section 7 of [I-D.ietf-avtcore-rtp-multi-stream].

## 6. Extension Considerations

This section outlines known issues and incompatibilities with RTP and RTCP extensions when multiple media types are used in a single RTP sessions. Future extensions to RTP and RTCP need to consider, and document, any potential incompatibility.

### 6.1. RTP Retransmission Payload Format

The RTP Retransmission Payload Format [RFC4588] can operate in either SSRC-multiplexed mode or session-multiplex mode.

In SSRC-multiplexed mode, retransmitted RTP packets are sent in the same RTP session as the original packets, but use a different SSRC with the same RTCP SDES CNAME. If each endpoint sends only a single original RTP stream and a single retransmission RTP stream in the session, this is sufficient. If an endpoint sends multiple original and retransmission RTP streams, as would occur when sending multiple media types in a single RTP session, then each original RTP stream and the retransmission RTP stream have to be associated using heuristics. By having retransmission requests outstanding for only one SSRC not yet mapped, a receiver can determine the binding between original and retransmission RTP stream. Another alternative is the use of different RTP payload types, allowing the signalled "apt" (associated payload type) parameter of the RTP retransmission payload format to be used to associate retransmitted and original packets.

Session-multiplexed mode sends the retransmission RTP stream in a separate RTP session to the original RTP stream, but using the same SSRC for each, with association being done by matching SSRCs between the two sessions. This is unaffected by the use of multiple media types in a single RTP session, since each media type will be sent using a different SSRC in the original RTP session, and the same SSRCs can be used in the retransmission session, allowing the streams to be associated. This can be signalled using SDP with the BUNDLE [I-D.ietf-mmusic-sdp-bundle-negotiation] and FID grouping [RFC5888] extensions. These SDP extensions require each "m=" line to only be included in a single FID group, but the RTP retransmission payload format uses FID groups to indicate the m= lines that form an original and retransmission pair. Accordingly, when using the BUNDLE extension to allow multiple media types to be sent in a single RTP session, each original media source (m= line) that is retransmitted needs a corresponding m= line in the retransmission RTP session. In case there are multiple media lines for retransmission, these media lines will form an independent BUNDLE group from the BUNDLE group with the source streams.

An example SDP fragment showing the grouping structures is provided in Figure 1. This example is not legal SDP and only the most important attributes have been left in place. Note that this SDP is not an initial BUNDLE offer. As can be seen there are two bundle groups, one for the source RTP session and one for the retransmissions. Then each of the media sources are grouped with its retransmission flow using FID, resulting in three more groupings.

```

a=group:BUNDLE foo bar fiz
a=group:BUNDLE zoo kelp glo
a=group:FID foo zoo
a=group:FID bar kelp
a=group:FID fiz glo
m=audio 10000 RTP/AVP 0
a=mid:foo
a=rtpmap:0 PCMU/8000
m=video 10000 RTP/AVP 31
a=mid:bar
a=rtpmap:31 H261/90000
m=video 10000 RTP/AVP 31
a=mid:fiz
a=rtpmap:31 H261/90000
m=audio 40000 RTP/AVPF 99
a=rtpmap:99 rtx/90000
a=fmtp:99 apt=0;rtx-time=3000
a=mid:zoo
m=video 40000 RTP/AVPF 100
a=rtpmap:100 rtx/90000
a=fmtp:100 apt=31;rtx-time=3000
a=mid:kelp
m=video 40000 RTP/AVPF 100
a=rtpmap:100 rtx/90000
a=fmtp:100 apt=31;rtx-time=3000
a=mid:glo

```

Figure 1: SDP example of Session Multiplexed RTP Retransmission

## 6.2. RTP Payload Format for Generic FEC

The RTP Payload Format for Generic Forward Error Correction (FEC) [RFC5109] (and its predecessor [RFC2733]) can either send the FEC stream as a separate RTP stream, or it can send the FEC combined with the original RTP stream as a redundant encoding [RFC2198].

When sending FEC as a separate stream, the RTP Payload Format for generic FEC requires that FEC stream to be sent in a separate RTP session to the original stream, using the same SSRC, with the FEC stream being associated by matching the SSRC between sessions. The

RTP session used for the original streams can include multiple RTP streams, and those RTP streams can use multiple media types. The repair session only needs one RTP Payload type to indicate FEC data, irrespective of the number of FEC streams sent, since the SSRC is used to associate the FEC streams with the original streams. Hence, it is RECOMMENDED that the FEC stream use the "application/ulpfec" media type for [RFC5109], and the "application/parityfec" media type for [RFC2733]. It is legal, but NOT RECOMMENDED, to send FEC streams using media specific payload format names (e.g., using both the "audio/ulpfec" and "video/ulpfec" payload formats for a single RTP session containing both audio and video flows), since this unnecessarily uses up RTP payload type values, and adds no value for demultiplexing since there might be multiple streams of the same media type).

The combination of an original RTP session using multiple media types with an associated generic FEC session can be signalled using SDP with the BUNDLE extension [I-D.ietf-mmusic-sdp-bundle-negotiation]. In this case, the RTP session carrying the FEC streams will be its own BUNDLE group. The m= line for each original stream and the m= line for the corresponding FEC stream are grouped using the SDP grouping framework using either the FEC-FR [RFC5956] grouping or, for backwards compatibility, the FEC [RFC4756] grouping. This is similar to the situation that arises for RTP retransmission with session multiplexing discussed in Section 6.1.

The Source-Specific Media Attributes [RFC5576] specification defines an SDP extension (the "FEC" semantic of the "ssrc-group" attribute) to signal FEC relationships between multiple RTP streams within a single RTP session. This cannot be used with generic FEC, since the FEC repair packets need to have the same SSRC value as the source packets being protected. There was work on an Unequal Layer Protection (ULP) extension to allow it be use FEC RTP streams within the same RTP Session as the source stream [I-D.lennox-payload-ulp-ssrc-mux].

When the FEC is sent as a redundant encoding, the considerations in Section 6.3 apply.

### 6.3. RTP Payload Format for Redundant Audio

The RTP Payload Format for Redundant Audio [RFC2198] can be used to protect audio streams. It can also be used along with the generic FEC payload format to send original and repair data in the same RTP packets. Both are compatible with RTP sessions containing multiple media types.

This payload format requires each different redundant encoding use a different RTP payload type number. When used with generic FEC in sessions that contain multiple media types, this requires each media type to use a different payload type for the FEC stream. For example, if audio and text are sent in a single RTP session with generic ULP FEC sent as a redundant encoding for each, then payload types need to be assigned for FEC using the audio/ulpfec and text/ulpfec payload formats. If multiple original payload types are used in the session, different redundant payload types need to be allocated for each one. This has potential to rapidly exhaust the available RTP payload type numbers.

## 7. Signalling

Establishing a single RTP session using multiple media types requires signalling. This signalling has to:

1. ensure that any participant in the RTP session is aware that this is an RTP session with multiple media types;
2. ensure that the payload types in use in the RTP session are using unique values, with no overlap between the media types;
3. ensure RTP session level parameters, for example the RTCP RR and RS bandwidth modifiers, the RTP/AVPF trr-int parameter, transport protocol, RTCP extensions in use, and any security parameters, are consistent across the session; and
4. ensure that RTP and RTCP functions that can be bound to a particular media type are reused where possible, rather than configuring multiple code-points for the same thing.

When using SDP signalling, the BUNDLE extension [I-D.ietf-mmusic-sdp-bundle-negotiation] is used to signal RTP sessions containing multiple media types.

## 8. Security Considerations

RTP provides a range of strong security mechanisms that can be used to secure sessions [RFC7201], [RFC7202]. The majority of these are independent of the type of media sent in the RTP session; however it is important to check that the security mechanism chosen is compatible with all types of media sent within the session.

Sending multiple media types in a single RTP session will generally require that all use the same security mechanism, whereas media sent using different RTP sessions can be secured in different ways. When different media types have different security requirements, it might



be necessary to send them using separate RTP sessions to meet those different requirements. This can have significant costs in terms of resource usage, session set-up time, etc.

## 9. IANA Considerations

This memo makes no request of IANA.

## 10. Acknowledgements

The authors would like to thank Christer Holmberg, Gunnar Hellstroem, Charles Eckel, Tolga Asveren, Warren Kumari, and Meral Shirazipour for their feedback on the document.

## 11. References

### 11.1. Normative References

- [I-D.ietf-avtcore-rtp-multi-stream]  
Lennox, J., Westerlund, M., Wu, Q., and C. Perkins,  
"Sending Multiple RTP Streams in a Single RTP Session",  
draft-ietf-avtcore-rtp-multi-stream-11 (work in progress),  
December 2015.
- [I-D.ietf-mmusic-sdp-bundle-negotiation]  
Holmberg, C., Alvestrand, H., and C. Jennings,  
"Negotiating Media Multiplexing Using the Session  
Description Protocol (SDP)", draft-ietf-mmusic-sdp-bundle-  
negotiation-23 (work in progress), July 2015.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate  
Requirement Levels", BCP 14, RFC 2119,  
DOI 10.17487/RFC2119, March 1997,  
<<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V.  
Jacobson, "RTP: A Transport Protocol for Real-Time  
Applications", STD 64, RFC 3550, DOI 10.17487/RFC3550,  
July 2003, <<http://www.rfc-editor.org/info/rfc3550>>.
- [RFC3551] Schulzrinne, H. and S. Casner, "RTP Profile for Audio and  
Video Conferences with Minimal Control", STD 65, RFC 3551,  
DOI 10.17487/RFC3551, July 2003,  
<<http://www.rfc-editor.org/info/rfc3551>>.

## 11.2. Informative References

- [I-D.ietf-avtcore-multiplex-guidelines]  
Westerlund, M., Perkins, C., and H. Alvestrand,  
"Guidelines for using the Multiplexing Features of RTP to  
Support Multiple Media Streams", draft-ietf-avtcore-  
multiplex-guidelines-03 (work in progress), October 2014.
- [I-D.lennox-payload-ulp-ssrc-mux]  
Lennox, J., "Supporting Source-Multiplexing of the Real-  
Time Transport Protocol (RTP) Payload for Generic Forward  
Error Correction", draft-lennox-payload-ulp-ssrc-mux-00  
(work in progress), February 2013.
- [RFC2198] Perkins, C., Kouvelas, I., Hodson, O., Hardman, V.,  
Handley, M., Bolot, J., Vega-Garcia, A., and S. Fosse-  
Parisis, "RTP Payload for Redundant Audio Data", RFC 2198,  
DOI 10.17487/RFC2198, September 1997,  
<<http://www.rfc-editor.org/info/rfc2198>>.
- [RFC2733] Rosenberg, J. and H. Schulzrinne, "An RTP Payload Format  
for Generic Forward Error Correction", RFC 2733,  
DOI 10.17487/RFC2733, December 1999,  
<<http://www.rfc-editor.org/info/rfc2733>>.
- [RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session  
Description Protocol", RFC 4566, DOI 10.17487/RFC4566,  
July 2006, <<http://www.rfc-editor.org/info/rfc4566>>.
- [RFC4588] Rey, J., Leon, D., Miyazaki, A., Varsa, V., and R.  
Hakenberg, "RTP Retransmission Payload Format", RFC 4588,  
DOI 10.17487/RFC4588, July 2006,  
<<http://www.rfc-editor.org/info/rfc4588>>.
- [RFC4756] Li, A., "Forward Error Correction Grouping Semantics in  
Session Description Protocol", RFC 4756,  
DOI 10.17487/RFC4756, November 2006,  
<<http://www.rfc-editor.org/info/rfc4756>>.
- [RFC5109] Li, A., Ed., "RTP Payload Format for Generic Forward Error  
Correction", RFC 5109, DOI 10.17487/RFC5109, December  
2007, <<http://www.rfc-editor.org/info/rfc5109>>.
- [RFC5576] Lennox, J., Ott, J., and T. Schierl, "Source-Specific  
Media Attributes in the Session Description Protocol  
(SDP)", RFC 5576, DOI 10.17487/RFC5576, June 2009,  
<<http://www.rfc-editor.org/info/rfc5576>>.

- [RFC5888] Camarillo, G. and H. Schulzrinne, "The Session Description Protocol (SDP) Grouping Framework", RFC 5888, DOI 10.17487/RFC5888, June 2010, <<http://www.rfc-editor.org/info/rfc5888>>.
- [RFC5956] Begen, A., "Forward Error Correction Grouping Semantics in the Session Description Protocol", RFC 5956, DOI 10.17487/RFC5956, September 2010, <<http://www.rfc-editor.org/info/rfc5956>>.
- [RFC6466] Salgueiro, G., "IANA Registration of the 'image' Media Type for the Session Description Protocol (SDP)", RFC 6466, DOI 10.17487/RFC6466, December 2011, <<http://www.rfc-editor.org/info/rfc6466>>.
- [RFC7160] Petit-Huguenin, M. and G. Zorn, Ed., "Support for Multiple Clock Rates in an RTP Session", RFC 7160, DOI 10.17487/RFC7160, April 2014, <<http://www.rfc-editor.org/info/rfc7160>>.
- [RFC7201] Westerlund, M. and C. Perkins, "Options for Securing RTP Sessions", RFC 7201, DOI 10.17487/RFC7201, April 2014, <<http://www.rfc-editor.org/info/rfc7201>>.
- [RFC7202] Perkins, C. and M. Westerlund, "Securing the RTP Framework: Why RTP Does Not Mandate a Single Media Security Solution", RFC 7202, DOI 10.17487/RFC7202, April 2014, <<http://www.rfc-editor.org/info/rfc7202>>.
- [RFC7656] Lennox, J., Gross, K., Nandakumar, S., Salgueiro, G., and B. Burman, Ed., "A Taxonomy of Semantics and Mechanisms for Real-Time Transport Protocol (RTP) Sources", RFC 7656, DOI 10.17487/RFC7656, November 2015, <<http://www.rfc-editor.org/info/rfc7656>>.
- [RFC7657] Black, D., Ed. and P. Jones, "Differentiated Services (Diffserv) and Real-Time Communication", RFC 7657, DOI 10.17487/RFC7657, November 2015, <<http://www.rfc-editor.org/info/rfc7657>>.

Authors' Addresses

Magnus Westerlund  
Ericsson  
Farogatan 6  
SE-164 80 Kista  
Sweden

Phone: +46 10 714 82 87  
Email: [magnus.westerlund@ericsson.com](mailto:magnus.westerlund@ericsson.com)

Colin Perkins  
University of Glasgow  
School of Computing Science  
Glasgow G12 8QQ  
United Kingdom

Email: [csp@csp Perkins.org](mailto:csp@csp Perkins.org)

Jonathan Lennox  
Vidyo, Inc.  
433 Hackensack Avenue  
Seventh Floor  
Hackensack, NJ 07601  
US

Email: [jonathan@vidyo.com](mailto:jonathan@vidyo.com)

Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: June 17, 2019

M. Westerlund  
B. Burman  
Ericsson  
C. Perkins  
University of Glasgow  
H. Alvestrand  
Google  
R. Even  
Huawei  
December 14, 2018

Guidelines for using the Multiplexing Features of RTP to Support  
Multiple Media Streams  
draft-ietf-avtcore-multiplex-guidelines-08

Abstract

The Real-time Transport Protocol (RTP) is a flexible protocol that can be used in a wide range of applications, networks, and system topologies. That flexibility makes for wide applicability, but can complicate the application design process. One particular design question that has received much attention is how to support multiple media streams in RTP. This memo discusses the available options and design trade-offs, and provides guidelines on how to use the multiplexing features of RTP to support multiple media streams.

Status of This Memo

This Internet-Draft is submitted 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 <https://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 June 17, 2019.

## Copyright Notice

Copyright (c) 2018 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 (<https://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. Definitions . . . . .	4
2.1. Terminology . . . . .	4
2.2. Subjects Out of Scope . . . . .	5
3. RTP Multiplexing Overview . . . . .	5
3.1. Reasons for Multiplexing and Grouping RTP Streams . . . . .	5
3.2. RTP Multiplexing Points . . . . .	6
3.2.1. RTP Session . . . . .	7
3.2.2. Synchronisation Source (SSRC) . . . . .	8
3.2.3. Contributing Source (CSRC) . . . . .	10
3.2.4. RTP Payload Type . . . . .	10
3.3. Issues Related to RTP Topologies . . . . .	11
3.4. Issues Related to RTP and RTCP Protocol . . . . .	12
3.4.1. The RTP Specification . . . . .	13
3.4.2. Multiple SSRCs in a Session . . . . .	15
3.4.3. Binding Related Sources . . . . .	15
3.4.4. Forward Error Correction . . . . .	17
4. Considerations for RTP Multiplexing . . . . .	17
4.1. Interworking Considerations . . . . .	17
4.1.1. Application Interworking . . . . .	17
4.1.2. RTP Translator Interworking . . . . .	18
4.1.3. Gateway Interworking . . . . .	18
4.1.4. Multiple SSRC Legacy Considerations . . . . .	19
4.2. Network Considerations . . . . .	20
4.2.1. Quality of Service . . . . .	20
4.2.2. NAT and Firewall Traversal . . . . .	21
4.2.3. Multicast . . . . .	22
4.3. Security and Key Management Considerations . . . . .	24
4.3.1. Security Context Scope . . . . .	24
4.3.2. Key Management for Multi-party sessions . . . . .	25
4.3.3. Complexity Implications . . . . .	25

5.	RTP Multiplexing Design Choices . . . . .	26
5.1.	Multiple Media Types in one Session . . . . .	26
5.2.	Multiple SSRCs of the Same Media Type . . . . .	27
5.3.	Multiple Sessions for one Media type . . . . .	28
5.4.	Single SSRC per Endpoint . . . . .	29
5.5.	Summary . . . . .	31
6.	Guidelines . . . . .	31
7.	IANA Considerations . . . . .	32
8.	Security Considerations . . . . .	33
9.	Contributors . . . . .	33
10.	References . . . . .	33
10.1.	Normative References . . . . .	33
10.2.	Informative References . . . . .	33
Appendix A.	Dismissing Payload Type Multiplexing . . . . .	37
Appendix B.	Signalling Considerations . . . . .	39
B.1.	Session Oriented Properties . . . . .	40
B.2.	SDP Prevents Multiple Media Types . . . . .	40
B.3.	Signalling RTP stream Usage . . . . .	41
Authors' Addresses	. . . . .	41

## 1. Introduction

The Real-time Transport Protocol (RTP) [RFC3550] is a commonly used protocol for real-time media transport. It is a protocol that provides great flexibility and can support a large set of different applications. RTP was from the beginning designed for multiple participants in a communication session. It supports many topology paradigms and usages, as defined in [RFC7667]. RTP has several multiplexing points designed for different purposes. These enable support of multiple RTP streams and switching between different encoding or packetization of the media. By using multiple RTP sessions, sets of RTP streams can be structured for efficient processing or identification. Thus, the question for any RTP application designer is how to best use the RTP session, the RTP stream identifier (SSRC), and the RTP payload type to meet the application's needs.

There have been increased interest in more advanced usage of RTP. For example, multiple RTP streams can be used when a single endpoint has multiple media sources (like multiple cameras or microphones) that need to be sent simultaneously. Consequently, questions are raised regarding the most appropriate RTP usage. The limitations in some implementations, RTP/RTCP extensions, and signalling has also been exposed. The authors also hope that clarification on the usefulness of some functionalities in RTP will result in more complete implementations in the future.

The purpose of this document is to provide clear information about the possibilities of RTP when it comes to multiplexing. The RTP application designer needs to understand the implications that come from a particular usage of the RTP multiplexing points. The document will recommend against some usages as being unsuitable, in general or for particular purposes.

The document starts with some definitions and then goes into the existing RTP functionalities around multiplexing. Both the desired behaviour and the implications of a particular behaviour depend on which topologies are used, which requires some consideration. This is followed by a discussion of some choices in multiplexing behaviour and their impacts. Some designs of RTP usage are discussed. Finally, some guidelines and examples are provided.

## 2. Definitions

### 2.1. Terminology

The definitions in Section 3 of [RFC3550] are referenced normatively.

The taxonomy defined in [RFC7656] is referenced normatively.

The following terms and abbreviations are used in this document:

**Multiparty:** A communication situation including multiple endpoints. In this document, it will be used to refer to situations where more than two endpoints communicate.

**Multiplexing:** The operation of taking multiple entities as input, aggregating them onto some common resource while keeping the individual entities addressable such that they can later be fully and unambiguously separated (de-multiplexed) again.

**RTP Receiver:** An Endpoint or Middlebox receiving RTP streams and RTCP messages. It uses at least one SSRC to send RTCP messages. An RTP Receiver may also be an RTP sender.

**RTP Sender:** An Endpoint sending one or more RTP streams, but also sending RTCP messages.

**RTP Session Group:** One or more RTP sessions that are used together to perform some function. Examples are multiple RTP sessions used to carry different layers of a layered encoding. In an RTP Session Group, CNAMEs are assumed to be valid across all RTP sessions, and designate synchronisation contexts that can cross RTP sessions; i.e. SSRCs that map to a common CNAME can be assumed



to have RTCP SR timing information derived from a common clock such that they can be synchronised for playout.

Signalling: The process of configuring endpoints to participate in one or more RTP sessions.

Note: The above definitions of RTP Receiver and RTP sender are intended to be consistent with the usage in [RFC3550].

## 2.2. Subjects Out of Scope

This document is focused on issues that affect RTP. Thus, issues that involve signalling protocols, such as whether SIP, Jingle or some other protocol is in use for session configuration, the particular syntaxes used to define RTP session properties, or the constraints imposed by particular choices in the signalling protocols, are mentioned only as examples in order to describe the RTP issues more precisely.

This document assumes the applications will use RTCP. While there are applications that don't send RTCP, they do not conform to the RTP specification, and thus can be regarded as reusing the RTP packet format but not implementing the RTP protocol.

## 3. RTP Multiplexing Overview

### 3.1. Reasons for Multiplexing and Grouping RTP Streams

There are several reasons why an endpoint might choose to send multiple media streams. In the below discussion, please keep in mind that the reasons for having multiple RTP streams vary and include but are not limited to the following:

- o Multiple media sources
- o Multiple RTP streams might be needed to represent one media source (for instance when using layered encodings)
- o A retransmission stream might repeat some parts of the content of another RTP stream
- o An FEC stream might provide material that can be used to repair another RTP stream
- o Alternative encodings, for instance using different codecs for the same audio stream

- o Alternative formats, for instance multiple resolutions of the same video stream

For each of these reasons, it is necessary to decide if each additional RTP stream is sent within the same RTP session as the other RTP streams, or if it is necessary to use additional RTP sessions to group the RTP streams. The choice suitable for one reason, might not be the choice suitable for another reason. The clearest understanding is associated with multiplexing multiple media sources of the same media type. However, all reasons warrant discussion and clarification on how to deal with them. As the discussion below will show, in reality we cannot choose a single one of SSRC or RTP session multiplexing solutions. To utilise RTP well and as efficiently as possible, both are needed. The real issue is finding the right guidance on when to create additional RTP sessions and when additional RTP streams in the same RTP session is the right choice.

### 3.2. RTP Multiplexing Points

This section describes the multiplexing points present in the RTP protocol that can be used to distinguish RTP streams and groups of RTP streams. Figure 1 outlines the process of demultiplexing incoming RTP streams:

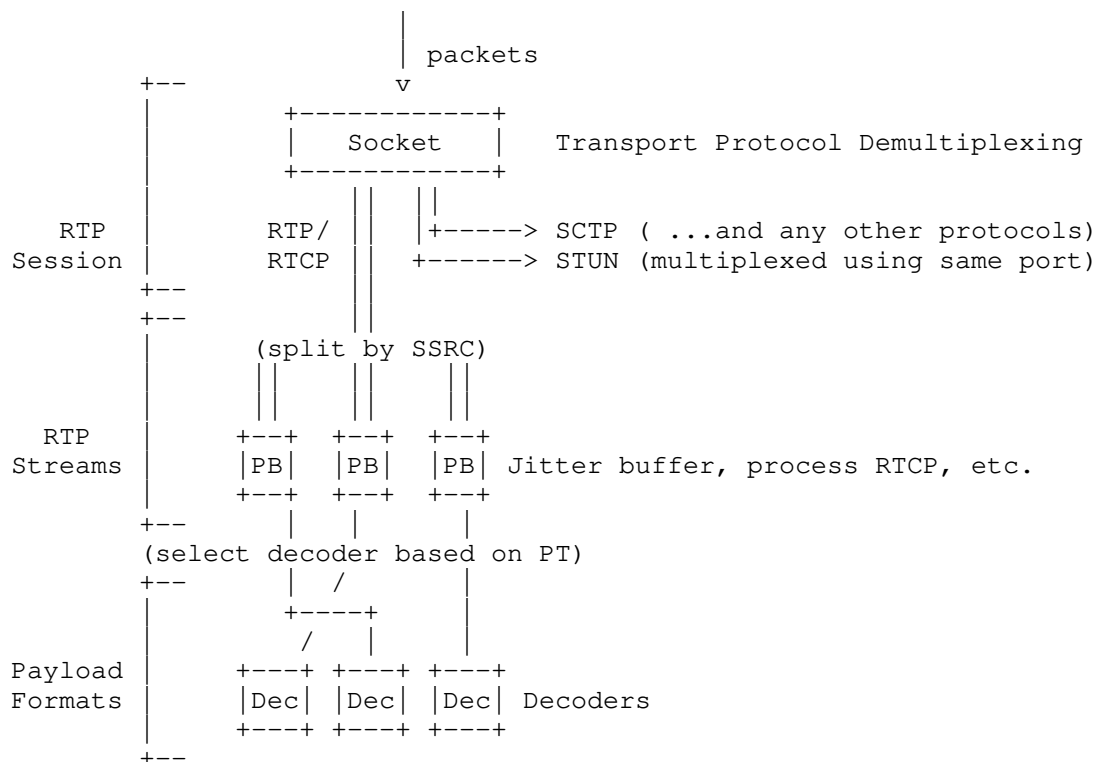


Figure 1: RTP Demultiplexing Process

### 3.2.1. RTP Session

An RTP Session is the highest semantic layer in the RTP protocol, and represents an association between a group of communicating endpoints. RTP does not contain a session identifier, yet RTP sessions must be possible to separate both across different endpoints and within a single endpoint.

For RTP session separation across endpoints, the set of participants that form an RTP session is defined as those that share a single synchronisation source space [RFC3550]. That is, if a group of participants are each aware of the synchronisation source identifiers belonging to the other participants, then those participants are in a single RTP session. A participant can become aware of a synchronisation source identifier by receiving an RTP packet containing it in the SSRC field or CSRC list, by receiving an RTCP packet mentioning it in an SSRC field, or through signalling (e.g., the Session Description Protocol (SDP) [RFC4566] "a=ssrc:" attribute

[RFC5576]). Thus, the scope of an RTP session is determined by the participants' network interconnection topology, in combination with RTP and RTCP forwarding strategies deployed by the endpoints and any middleboxes, and by the signalling.

For RTP session separation within a single endpoint, RTP relies on the underlying transport layer, and on the signalling to identify RTP sessions in a manner that is meaningful to the application. A single endpoint can have one or more transport flows for the same RTP session, and a single RTP session can span multiple transport layer flows. The signalling layer might give RTP sessions an explicit identifier, or the identification might be implicit based on the addresses and ports used. Accordingly, a single RTP session can have multiple associated identifiers, explicit and implicit, belonging to different contexts. For example, when running RTP on top of UDP/IP, an endpoint can identify and delimit an RTP session from other RTP sessions by receiving the multiple UDP flows used as identified based on their UDP source and destination IP addresses and UDP port numbers. Another example is SDP media descriptions (the "m=" line and the following associated lines) signals the transport flow and RTP session configuration for the endpoints part of the RTP session. SDP grouping framework [RFC5888] allows labeling of the media descriptions, for example used so that RTP Session Groups can be created. With Negotiating Media Multiplexing Using the Session Description Protocol (SDP) [I-D.ietf-mmusic-sdp-bundle-negotiation], multiple media descriptions where each represents the RTP streams sent or received for a media source are part of a common RTP session.

The RTP protocol makes no normative statements about the relationship between different RTP sessions, however the applications that use more than one RTP session will have some higher layer understanding of the relationship between the sessions they create.

### 3.2.2. Synchronisation Source (SSRC)

A synchronisation source (SSRC) identifies an source of an RTP stream or an RTP receiver when sending RTCP. Every endpoint has at least one SSRC identifier, even if it does not send RTP packets. RTP endpoints that are only RTP receivers still send RTCP and use their SSRC identifiers in the RTCP packets they send. An endpoint can have multiple SSRC identifiers if it sends multiple RTP streams. Endpoints that are both RTP sender and RTP receiver use the same SSRC in both roles.

The SSRC is a 32-bit identifier. It is present in every RTP and RTCP packet header, and in the payload of some RTCP packet types. It can also be present in SDP signalling. Unless pre-signalled, e.g. using the SDP "a=ssrc:" attribute [RFC5576], the SSRC is chosen at random.

It is not dependent on the network address of the endpoint, and is intended to be unique within an RTP session. SSRC collisions can occur, and are handled as specified in [RFC3550] and [RFC5576], resulting in the SSRC of the colliding RTP streams or receivers changing. An endpoint that changes its network transport address during a session have to choose a new SSRC identifier to avoid being interpreted as looped source, unless the transport layer mechanism, e.g. ICE [RFC8445], handles such changes.

SSRC identifiers that belong to the same synchronisation context (i.e., that represent RTP streams that can be synchronised using information in RTCP SR packets) use identical CNAME chunks in corresponding RTCP SDES packets. SDP signalling can also be used to provide explicit SSRC grouping [RFC5576].

In some cases, the same SSRC identifier value is used to relate streams in two different RTP sessions, such as in RTP retransmission [RFC4588]. This is to be avoided since there is no guarantee that SSRC values are unique across RTP sessions. For the RTP retransmission [RFC4588] case it is recommended to use explicit binding of the source RTP stream and the redundancy stream, e.g. using the RepairedRtpStreamId RTCP SDES item [I-D.ietf-avtext-rid].

Note that RTP sequence number and RTP timestamp are scoped by the SSRC and thus specific per RTP stream.

Different types of entities use a SSRC to identify themselves, as follows:

A real media source: Uses the SSRC to identify a "physical" media source.

A conceptual media source: Uses the SSRC to identify the result of applying some filtering function in a network node, for example a filtering function in an RTP mixer that provides the most active speaker based on some criteria, or a mix representing a set of other sources.

An RTP receiver: Uses the SSRC to identify itself as the source of its RTCP reports.

Note that an endpoint that generates more than one media type, e.g. a conference participant sending both audio and video, need not (and should not) use the same SSRC value across RTP sessions. RTCP compound packets containing the CNAME SDES item is the designated method to bind an SSRC to a CNAME, effectively cross-correlating SSRCs within and between RTP Sessions as coming from the same endpoint. The main property attributed to SSRCs associated with the

same CNAME is that they are from a particular synchronisation context and can be synchronised at playback.

An RTP receiver receiving a previously unseen SSRC value will interpret it as a new source. It might in fact be a previously existing source that had to change SSRC number due to an SSRC conflict. However, the originator of the previous SSRC ought to have ended the conflicting source by sending an RTCP BYE for it prior to starting to send with the new SSRC, so the new SSRC is anyway effectively a new source.

### 3.2.3. Contributing Source (CSRC)

The Contributing Source (CSRC) is not a separate identifier. Rather an SSRC identifier is listed as a CSRC in the RTP header of a packet generated by an RTP mixer, if the corresponding SSRC was in the header of one of the packets that contributed to the mix.

It is not possible, in general, to extract media represented by an individual CSRC since it is typically the result of a media mixing (merge) operation by an RTP mixer on the individual media streams corresponding to the CSRC identifiers. The exception is the case when only a single CSRC is indicated as this represent forwarding of an RTP stream, possibly modified. The RTP header extension for Mixer-to-Client Audio Level Indication [RFC6465] expands on the receiver's information about a packet with a CSRC list. Due to these restrictions, CSRC will not be considered a fully qualified multiplexing point and will be disregarded in the rest of this document.

### 3.2.4. RTP Payload Type

Each RTP stream utilises one or more RTP payload formats. An RTP payload format describes how the output of a particular media codec is framed and encoded into RTP packets. The payload format used is identified by the payload type (PT) field in the RTP packet header. The combination of SSRC and PT therefore identifies a specific RTP stream encoding format. The format definition can be taken from [RFC3551] for statically allocated payload types, but ought to be explicitly defined in signalling, such as SDP, both for static and dynamic payload types. The term "format" here includes whatever can be described by out-of-band signalling means. In SDP, the term "format" includes media type, RTP timestamp sampling rate, codec, codec configuration, payload format configurations, and various robustness mechanisms such as redundant encodings [RFC2198].

The RTP payload type is scoped by the sending endpoint within an RTP session. PT has the same meaning across all RTP streams in an RTP

session. All SSRCs sent from a single endpoint share the same payload type definitions. The RTP payload type is designed such that only a single payload type is valid at any time instant in the RTP stream's timestamp time line, effectively time-multiplexing different payload types if any change occurs. The payload type used can change on a per-packet basis for an SSRC, for example a speech codec making use of generic comfort noise [RFC3389]. If there is a true need to send multiple payload types for the same SSRC that are valid for the same instant, then redundant encodings [RFC2198] can be used. Several additional constraints than the ones mentioned above need to be met to enable this use, one of which is that the combined payload sizes of the different payload types ought not exceed the transport MTU. If it is acceptable to send multiple formats of the same media source as separate RTP streams (with separate SSRC), simulcast [I-D.ietf-mmusic-sdp-simulcast] can be used.

Other aspects of RTP payload format use are described in How to Write an RTP Payload Format [RFC8088].

The payload type is not a multiplexing point at the RTP layer (see Appendix A for a detailed discussion of why using the payload type as an RTP multiplexing point does not work). The RTP payload type is, however, used to determine how to consume and decode an RTP stream. The RTP payload type number is sometimes used to associate an RTP stream with the signalling; this is not recommended since a specific payload type value can be used in multiple bundled "m=" sections [I-D.ietf-mmusic-sdp-bundle-negotiation]. This association is only possible if unique RTP payload type numbers are used in each context.

### 3.3. Issues Related to RTP Topologies

The impact of how RTP multiplexing is performed will in general vary with how the RTP session participants are interconnected, described by RTP Topology [RFC7667].

Even the most basic use case, denoted Topo-Point-to-Point in [RFC7667], raises a number of considerations that are discussed in detail in following sections. They range over such aspects as:

- o Does my communication peer support RTP as defined with multiple SSRCs per RTP session?
- o Do I need network differentiation in form of QoS?
- o Can the application more easily process and handle the media streams if they are in different RTP sessions?

- o Do I need to use additional RTP streams for RTP retransmission or FEC?

For some point to multi-point topologies (e.g. Topo-ASM and Topo-SSM in [RFC7667]), multicast is used to interconnect the session participants. Special considerations (documented in Section 4.2.3) are then needed as multicast is a one-to-many distribution system.

Sometimes an RTP communication can end up in a situation when the communicating peers are not compatible for various reasons:

- o No common media codec for a media type thus requiring transcoding.
- o Different support for multiple RTP streams and RTP sessions.
- o Usage of different media transport protocols, i.e., RTP or other.
- o Usage of different transport protocols, e.g., UDP, DCCP, or TCP.
- o Different security solutions, e.g., IPsec, TLS, DTLS, or SRTP with different keying mechanisms.

In many situations this is resolved by the inclusion of a translator between the two peers, as described by Topo-PtP-Translator in [RFC7667]. The translator's main purpose is to make the peers look compatible to each other. There can also be other reasons than compatibility to insert a translator in the form of a middlebox or gateway, for example a need to monitor the RTP streams. If the stream transport characteristics are changed by the translator, appropriate media handling can require thorough understanding of the application logic, specifically any congestion control or media adaptation.

The point to point topology can contain one to many RTP sessions with one to many media sources per session, each having one or more RTP streams per media source.

### 3.4. Issues Related to RTP and RTCP Protocol

Using multiple RTP streams is a well-supported feature of RTP. However, for most implementers or people writing RTP/RTCP applications or extensions attempting to apply multiple streams, it can be unclear when it is most appropriate to add an additional RTP stream in an existing RTP session and when it is better to use multiple RTP sessions. This section discusses the various considerations needed.



### 3.4.1. The RTP Specification

RFC 3550 contains some recommendations and a bullet list with 5 arguments for different aspects of RTP multiplexing. Let's review Section 5.2 of [RFC3550], reproduced below:

"For efficient protocol processing, the number of multiplexing points should be minimised, as described in the integrated layer processing design principle [ALF]. In RTP, multiplexing is provided by the destination transport address (network address and port number) which is different for each RTP session. For example, in a teleconference composed of audio and video media encoded separately, each medium SHOULD be carried in a separate RTP session with its own destination transport address.

Separate audio and video streams SHOULD NOT be carried in a single RTP session and demultiplexed based on the payload type or SSRC fields. Interleaving packets with different RTP media types but using the same SSRC would introduce several problems:

1. If, say, two audio streams shared the same RTP session and the same SSRC value, and one were to change encodings and thus acquire a different RTP payload type, there would be no general way of identifying which stream had changed encodings.
2. An SSRC is defined to identify a single timing and sequence number space. Interleaving multiple payload types would require different timing spaces if the media clock rates differ and would require different sequence number spaces to tell which payload type suffered packet loss.
3. The RTCP sender and receiver reports (see Section 6.4) can only describe one timing and sequence number space per SSRC and do not carry a payload type field.
4. An RTP mixer would not be able to combine interleaved streams of incompatible media into one stream.
5. Carrying multiple media in one RTP session precludes: the use of different network paths or network resource allocations if appropriate; reception of a subset of the media if desired, for example just audio if video would exceed the available bandwidth; and receiver implementations that use separate processes for the different media, whereas using separate RTP sessions permits either single- or multiple-process implementations.

Using a different SSRC for each medium but sending them in the same RTP session would avoid the first three problems but not the last two.

On the other hand, multiplexing multiple related sources of the same medium in one RTP session using different SSRC values is the norm for multicast sessions. The problems listed above don't apply: an RTP mixer can combine multiple audio sources, for example, and the same treatment is applicable for all of them. It might also be appropriate to multiplex streams of the same medium using different SSRC values in other scenarios where the last two problems do not apply."

Let's consider one argument at a time. The first argument is for using different SSRC for each individual RTP stream, which is fundamental to RTP operation.

The second argument is advocating against demultiplexing RTP streams within a session based on their RTP payload type numbers, which still stands as can be seen by the extensive list of issues found in Appendix A.

The third argument is yet another argument against payload type multiplexing.

The fourth argument is against multiplexing RTP packets that require different handling into the same session. As we saw in the discussion of RTP mixers, the RTP mixer must embed application logic to handle streams anyway; the separation of streams according to stream type is just another piece of application logic, which might or might not be appropriate for a particular application. One type of application that can mix different media sources "blindly" is the audio-only "telephone" bridge; most other types of applications need application-specific logic to perform the mix correctly.

The fifth argument discusses network aspects that we will discuss more below in Section 4.2. It also goes into aspects of implementation, like Split Component Terminal (see Section 3.10 of [RFC7667]) endpoints where different processes or inter-connected devices handle different aspects of the whole multi-media session.

A summary of RFC 3550's view on multiplexing is to use unique SSRCs for anything that is its own media/packet stream, and to use different RTP sessions for media streams that don't share a media type. This document supports the first point; it is very valid. The latter needs further discussion, as imposing a single solution on all usages of RTP is inappropriate. Multiple Media Types in an RTP Session specification [I-D.ietf-avtcore-multi-media-rtp-session]

provides a detailed analysis of the potential issues in having multiple media types in the same RTP session. This document provides a wider scope for an RTP session and considers multiple media types in one RTP session as a possible choice for the RTP application designer.

#### 3.4.2. Multiple SSRCs in a Session

Using multiple SSRCs at one endpoint in an RTP session requires resolving some unclear aspects of the RTP specification. These could potentially lead to some interoperability issues as well as some potential significant inefficiencies, as further discussed in "RTP Considerations for Endpoints Sending Multiple Media Streams" [RFC8108]. An RTP application designer should consider these issues and the possible application impact from lack of appropriate RTP handling or optimization in the peer endpoints.

Using multiple RTP sessions can potentially mitigate application issues caused by multiple SSRCs in an RTP session.

#### 3.4.3. Binding Related Sources

A common problem in a number of various RTP extensions has been how to bind related RTP streams together. This issue is common to both using additional SSRCs and multiple RTP sessions.

The solutions can be divided into a few groups:

- o RTP/RTCP based
- o Signalling based (SDP)
- o Grouping related RTP sessions
- o Grouping SSRCs within an RTP session

Most solutions are explicit, but some implicit methods have also been applied to the problem.

The SDP-based signalling solutions are:

SDP Media Description Grouping: The SDP Grouping Framework [RFC5888] uses various semantics to group any number of media descriptions. These has previously been considered primarily as grouping RTP sessions, [I-D.ietf-mmusic-sdp-bundle-negotiation] groups multiple media descriptions as a single RTP session.

SDP SSRC grouping: Source-Specific Media Attributes in SDP [RFC5576]

includes a solution for grouping SSRCs the same way as the Grouping framework groups Media Descriptions.

This supports a lot of use cases. All these solutions have shortcomings in cases where the session's dynamic properties are such that it is difficult or resource consuming to keep the list of related SSRCs up to date.

An RTP/RTCP-based solution is to use the RTCP SDES CNAME to bind the RTP streams to an endpoint or synchronization context. For applications with a single RTP stream per type (Media, Source or Redundancy) this is sufficient independent if one or more RTP sessions are used. However, some applications choose not to use it because of perceived complexity or a desire not to implement RTCP and instead use the same SSRC value to bind related RTP streams across multiple RTP sessions. RTP Retransmission [RFC4588] in multiple RTP session mode and Generic FEC [RFC5109] both use this method. This method may work but might have some downsides in RTP sessions with many participating SSRCs. When an SSRC collision occurs, this will force one to change SSRC in all RTP sessions and thus resynchronize all of them instead of only the single media stream having the collision. Therefore, it is not recommended to use identical SSRC values to relate RTP streams.

Another solution to bind SSRCs is an implicit method used by RTP Retransmission [RFC4588] when doing retransmissions in the same RTP session as the source RTP stream. The receiver missing a packet issues an RTP retransmission request, and then awaits a new SSRC carrying the RTP retransmission payload and where that SSRC is from the same CNAME. This limits a requester to having only one outstanding request on any new source SSRCs per endpoint.

RTP Payload Format Restrictions [I-D.ietf-mmusic-rid] provides an RTP/RTCP based mechanism to unambiguously identify the RTP streams within an RTP session and restrict the streams' payload format parameters in a codec-agnostic way beyond what is provided with the regular Payload Types. The mapping is done by specifying an "a=rid" value in the SDP offer/answer signalling and having the corresponding "rtp-stream-id" value as an SDES item and an RTP header extension. The RID solution also includes a solution for binding redundancy RTP streams to their original source RTP streams, given that those use RID identifiers.

It can be noted that Section 8.3 of the RTP Specification [RFC3550] recommends using a single SSRC space across all RTP sessions for layered coding. Based on the experience so far however, we recommend to use a solution doing explicit binding between the RTP streams so what the used SSRC values are do not matter. That way solutions

using multiple RTP streams in a single RTP session and multiple RTP sessions uses the same solution.

#### 3.4.4. Forward Error Correction

There exist a number of Forward Error Correction (FEC) based schemes for how to reduce the packet loss of the original streams. Most of the FEC schemes will protect a single source flow. The protection is achieved by transmitting a certain amount of redundant information that is encoded such that it can repair one or more packet losses over the set of packets the redundant information protects. This sequence of redundant information also needs to be transmitted as its own media stream, or in some cases, instead of the original media stream. Thus, many of these schemes create a need for binding related flows as discussed above. Looking at the history of these schemes, there are schemes using multiple SSRCs and schemes using multiple RTP sessions, and some schemes that support both modes of operation.

Using multiple RTP sessions supports the case where some set of receivers might not be able to utilise the FEC information. By placing it in a separate RTP session and if separating RTP sessions on transport level, FEC can easily be ignored already on transport level.

In usages involving multicast, having the FEC information on its own multicast group allows for similar flexibility. This is especially useful when receivers see very heterogeneous packet loss rates. Those receivers that are not seeing packet loss don't need to join the multicast group with the FEC data, and so avoid the overhead of receiving unnecessary FEC packets, for example.

### 4. Considerations for RTP Multiplexing

#### 4.1. Interworking Considerations

There are several different kinds of interworking, and this section discusses two; interworking between different applications including the implications of potentially different RTP multiplexing point choices and limitations that have to be considered when working with some legacy applications.

##### 4.1.1. Application Interworking

It is not uncommon that applications or services of similar but not identical usage, especially the ones intended for interactive communication, encounter a situation where one want to interconnect two or more of these applications.

In these cases, one ends up in a situation where one might use a gateway to interconnect applications. This gateway must then either change the multiplexing structure or adhere to the respective limitations in each application.

There are two fundamental approaches to building a gateway: using an RTP Translator interworking (RTP bridging), where the gateway acts as an RTP Translator, with the two applications being members of the same RTP session; or Gateway Interworking with RTP termination, where there are independent RTP sessions running from each interconnected application to the gateway.

#### 4.1.2. RTP Translator Interworking

From an RTP perspective, the RTP Translator approach could work if all the applications are using the same codecs with the same payload types, have made the same multiplexing choices, and have the same capabilities in number of simultaneous RTP streams combined with the same set of RTP/RTCP extensions being supported. Unfortunately, this might not always be true.

When a gateway is implemented via an RTP Translator, an important consideration is if the two applications being interconnected need to use the same approach to multiplexing. If one side is using RTP session multiplexing and the other is using SSRC multiplexing with bundle, it is possible for the RTP translator to map the RTP streams between both sides if the order of SDP "m=" lines between both sides are the same. There are also challenges with SSRC collision handling since there may be a collision on the SSRC multiplexing side but the RTP session multiplexing side will not be aware of any collision unless SSRC translation is applied on the RTP translator. Furthermore, if one of the applications is capable of working in several modes (such as being able to use additional RTP streams in one RTP session or multiple RTP sessions at will), and the other one is not, successful interconnection depends on locking the more flexible application into the operating mode where interconnection can be successful, even if no participants are using the less flexible application when the RTP sessions are being created.

#### 4.1.3. Gateway Interworking

When one terminates RTP sessions at the gateway, there are certain tasks that the gateway has to carry out:

- o Generating appropriate RTCP reports for all RTP streams (possibly based on incoming RTCP reports), originating from SSRCs controlled by the gateway.

- o Handling SSRC collision resolution in each application's RTP sessions.
- o Signalling, choosing and policing appropriate bit-rates for each session.

For applications that uses any security mechanism, e.g., in the form of SRTP, the gateway needs to be able to decrypt incoming packets and re-encrypt them in the other application's security context. This is necessary even if all that's needed is a simple remapping of SSRC numbers. If this is done, the gateway also needs to be a member of the security contexts of both sides, of course.

Other tasks a gateway might need to apply include transcoding (for incompatible codec types), media-level adaptations that cannot be solved through media negotiation (such as rescaling for incompatible video size requirements), suppression of content that is known not to be handled in the destination application, or the addition or removal of redundancy coding or scalability layers to fit the needs of the destination domain.

From the above, we can see that the gateway needs to have an intimate knowledge of the application requirements; a gateway is by its nature application specific, not a commodity product.

This fact reveals the potential for these gateways to block application evolution by blocking RTP and RTCP extensions that the applications have been extended with but that are unknown to the gateway.

If one uses security functions, like SRTP, and as can be seen from above, they incur both additional risk due to the requirement to have the gateway in the security association between the endpoints (unless the gateway is on the transport level), and additional complexities in form of the decrypt-encrypt cycles needed for each forwarded packet. SRTP, due to its keying structure, also requires that each RTP session needs different master keys, as use of the same key in two RTP sessions can for some ciphers result in two-time pads that completely breaks the confidentiality of the packets.

#### 4.1.4. Multiple SSRC Legacy Considerations

Historically, the most common RTP use cases have been point to point Voice over IP (VoIP) or streaming applications, commonly with no more than one media source per endpoint and media type (typically audio or video). Even in conferencing applications, especially voice-only, the conference focus or bridge has provided a single stream with a mix of the other participants to each participant. It is also common

to have individual RTP sessions between each endpoint and the RTP mixer, meaning that the mixer functions as an RTP-terminating gateway.

When establishing RTP sessions that can contain endpoints that aren't updated to handle multiple streams following these recommendations, a particular application can have issues with multiple SSRCs within a single session. These issues include:

1. Need to handle more than one stream simultaneously rather than replacing an already existing stream with a new one.
2. Be capable of decoding multiple streams simultaneously.
3. Be capable of rendering multiple streams simultaneously.

This indicates that gateways attempting to interconnect to this class of devices has to make sure that only one RTP stream of each type gets delivered to the endpoint if it's expecting only one, and that the multiplexing format is what the device expects. It is highly unlikely that RTP translator-based interworking can be made to function successfully in such a context.

#### 4.2. Network Considerations

The RTP multiplexing choice has impact on network level mechanisms that need to be considered by the implementer.

##### 4.2.1. Quality of Service

When it comes to Quality of Service mechanisms, they are either flow based or packet marking based. RSVP [RFC2205] is an example of a flow based mechanism, while Diff-Serv [RFC2474] is an example of a packet marking based one. For a packet marking based scheme, the method of multiplexing will not affect the possibility to use QoS.

However, for a flow based scheme there is a clear difference between the multiplexing methods. Additional SSRC will result in all RTP streams being part of the same 5-tuple (protocol, source address, destination address, source port, destination port) which is the most common selector for flow based QoS.

It must also be noted that packet marking based QoS mechanisms can have limitations. A general observation is that different Differentiated Services Code Points (DSCP) can be assigned to different packets within a flow as well as within an RTP stream. However, care must be taken when considering which forwarding behaviours that are applied on path due to these DSCPs. In some



cases the forwarding behaviour can result in packet reordering. For more discussion of this see [RFC7657].

The method for assigning marking to packets can impact what number of RTP sessions to choose. If this marking is done using a network ingress function, it can have issues discriminating the different RTP streams. The network API on the endpoint also needs to be capable of setting the marking on a per-packet basis to reach the full functionality.

#### 4.2.2. NAT and Firewall Traversal

In today's network there exist a large number of middleboxes. The ones that normally have most impact on RTP are Network Address Translators (NAT) and Firewalls (FW).

Below we analyse and comment on the impact of requiring more underlying transport flows in the presence of NATs and Firewalls:

**End-Point Port Consumption:** A given IP address only has 65536 available local ports per transport protocol for all consumers of ports that exist on the machine. This is normally never an issue for an end-user machine. It can become an issue for servers that handle large number of simultaneous streams. However, if the application uses ICE to authenticate STUN requests, a server can serve multiple endpoints from the same local port, and use the whole 5-tuple (source and destination address, source and destination port, protocol) as identifier of flows after having securely bound them to the remote endpoint address using the STUN request. In theory the minimum number of media server ports needed are the maximum number of simultaneous RTP Sessions a single endpoint can use. In practice, implementation will probably benefit from using more server ports to simplify implementation or avoid performance bottlenecks.

**NAT State:** If an endpoint sits behind a NAT, each flow it generates to an external address will result in a state that has to be kept in the NAT. That state is a limited resource. In home or Small Office/Home Office (SOHO) NATs, memory or processing are usually the most limited resources. For large scale NATs serving many internal endpoints, available external ports are likely the scarce resource. Port limitations is primarily a problem for larger centralised NATs where endpoint independent mapping requires each flow to use one port for the external IP address. This affects the maximum number of internal users per external IP address. However, it is worth pointing out that a real-time video conference session with audio and video is likely using less than

10 UDP flows, compared to certain web applications that can use 100+ TCP flows to various servers from a single browser instance.

**NAT Traversal Extra Delay:** Performing the NAT/FW traversal takes a certain amount of time for each flow. It also takes time in a phase of communication between accepting to communicate and the media path being established which is fairly critical. The best case scenario for how much extra time it takes after finding the first valid candidate pair following the specified ICE procedures are:  $1.5 \cdot \text{RTT} + T_a \cdot (\text{Additional\_Flows} - 1)$ , where  $T_a$  is the pacing timer. That assumes a message in one direction, and then an immediate triggered check back. The reason it isn't more, is that ICE first finds one candidate pair that works prior to attempting to establish multiple flows. Thus, there is no extra time until one has found a working candidate pair. Based on that working pair the needed extra time is to in parallel establish the, in most cases 2-3, additional flows. However, packet loss causes extra delays, at least 100 ms, which is the minimal retransmission timer for ICE.

**NAT Traversal Failure Rate:** Due to the need to establish more than a single flow through the NAT, there is some risk that establishing the first flow succeeds but that one or more of the additional flows fail. The risk that this happens is hard to quantify, but ought to be fairly low as one flow from the same interfaces has just been successfully established. Thus only rare events such as NAT resource overload, or selecting particular port numbers that are filtered etc., ought to be reasons for failure.

**Deep Packet Inspection and Multiple Streams:** Firewalls differ in how deeply they inspect packets. There exist some potential that deeply inspecting firewalls will have similar legacy issues with multiple SSRCs as some stack implementations.

Using additional RTP streams in the same RTP session and transport flow does not introduce any additional NAT traversal complexities per RTP stream. This can be compared with normally one or two additional transport flows per RTP session when using multiple RTP sessions. Additional lower layer transport flows will be needed, unless an explicit de-multiplexing layer is added between RTP and the transport protocol. At time of writing no such mechanism was defined.

#### 4.2.3. Multicast

Multicast groups provides a powerful tool for a number of real-time applications, especially the ones that desire broadcast-like behaviours with one endpoint transmitting to a large number of receivers, like in IPTV. There are also the RTP/RTCP extension to

better support Source Specific Multicast (SSM) [RFC5760]. Another application is the Many to Many communication, which RTP [RFC3550] was originally built to support, but the multicast semantics do result in a certain number of limitations.

One limitation is that for any group, sender side adaptation to the actual receiver properties causes degradation for all participants to what is supported by the receiver with the worst conditions among the group participants. For broadcast type of applications this is not acceptable. Instead, various receiver-based solutions are employed to ensure that the receivers achieve best possible performance. By using scalable encoding and placing each scalability layer in a different multicast group, the receiver can control the amount of traffic it receives. To have each scalability layer on a different multicast group, one RTP session per multicast group is used.

In addition, the transport flow considerations in multicast are a bit different from unicast; NATs with port translation are not useful in the multicast environment, meaning that the entire port range of each multicast address is available for distinguishing between RTP sessions.

Thus, when using broadcast applications it appears easiest and most straightforward to use multiple RTP sessions for sending different media flows used for adapting to network conditions. It is also common that streams that improve transport robustness are sent in their own multicast group to allow for interworking with legacy or to support different levels of protection.

For many to many applications there are different needs. Here, the most appropriate choice will depend on how the actual application is realized. With sender side congestion control there might not exist any benefit with using multiple RTP sessions.

The properties of a broadcast application using RTP multicast:

1. Uses a group of RTP sessions, not one. Each endpoint will need to be a member of a number of RTP sessions in order to perform well.
2. Within each RTP session, the number of RTP receivers is likely to be much larger than the number of RTP senders.
3. The applications need signalling functions to identify the relationships between RTP sessions.

4. The applications need signalling or RTP/RTCP functions to identify the relationships between SSRCs in different RTP sessions when needs beyond CNAME exist.

Both broadcast and many to many multicast applications do share a signalling requirement; all of the participants will need to have the same RTP and payload type configuration. Otherwise, A could for example be using payload type 97 as the video codec H.264 while B thinks it is MPEG-2. It is to be noted that SDP offer/answer [RFC3264] is not appropriate for ensuring this property in broadcast/multicast context. The signalling aspects of broadcast/multicast are not explored further in this memo.

Security solutions for this type of group communications are also challenging. First, the key-management and the security protocol need to support group communication. Second, source authentication requires special solutions. For more discussion on this please review Options for Securing RTP Sessions [RFC7201].

#### 4.3. Security and Key Management Considerations

When dealing with point-to-point, 2-member RTP sessions only, there are few security issues that are relevant to the choice of having one RTP session or multiple RTP sessions. However, there are a few aspects of multiparty sessions that might warrant consideration. For general information of possible methods of securing RTP, please review RTP Security Options [RFC7201].

##### 4.3.1. Security Context Scope

When using SRTP [RFC3711] the security context scope is important and can be a necessary differentiation in some applications. As SRTP's crypto suites are (so far) built around symmetric keys, the receiver will need to have the same key as the sender. This results in that no one in a multi-party session can be certain that a received packet really was sent by the claimed sender and not by another party having access to the key. At least unless TESLA source authentication [RFC4383], which adds delay to achieve source authentication. In most cases symmetric ciphers provide sufficient security properties, but there are a few cases where this does create issues.

The first case is when someone leaves a multi-party session and one wants to ensure that the party that left can no longer access the RTP streams. This requires that everyone re-keys without disclosing the keys to the excluded party.

A second case is when using security as an enforcing mechanism for differentiation. Take for example a scalable layer or a high quality

simulcast version that only premium users are allowed to access. The mechanism preventing a receiver from getting the high quality stream can be based on the stream being encrypted with a key that user can't access without paying premium, having the key-management limit access to the key.

SRTP [RFC3711] has no special functions for dealing with different sets of master keys for different SSRCs. The key-management functions have different capabilities to establish different sets of keys, normally on a per endpoint basis. For example, DTLS-SRTP [RFC5764] and Security Descriptions [RFC4568] establish different keys for outgoing and incoming traffic from an endpoint. This key usage has to be written into the cryptographic context, possibly associated with different SSRCs.

#### 4.3.2. Key Management for Multi-party sessions

Performing key-management for multi-party sessions can be a challenge. This section considers some of the issues.

Multi-party sessions, such as transport translator based sessions and multicast sessions, can neither use Security Description [RFC4568] nor DTLS-SRTP [RFC5764] without an extension as each endpoint provides its set of keys. In centralised conferences, the signalling counterpart is a conference server and the media plane unicast counterpart (to which DTLS messages would be sent) is the transport translator. Thus, an extension like Encrypted Key Transport [I-D.ietf-perc-srtp-ekt-diet] or a MIKEY [RFC3830] based solution that allows for keying all session participants with the same master key is needed.

#### 4.3.3. Complexity Implications

The usage of security functions can surface complexity implications from the choice of multiplexing and topology. This becomes especially evident in RTP topologies having any type of middlebox that processes or modifies RTP/RTCP packets. Where there is very small overhead for an RTP translator or mixer to rewrite an SSRC value in the RTP packet of an unencrypted session, the cost is higher when using cryptographic security functions. For example, if using SRTP [RFC3711], the actual security context and exact crypto key are determined by the SSRC field value. If one changes SSRC, the encryption and authentication must use another key. Thus, changing the SSRC value implies a decryption using the old SSRC and its security context, followed by an encryption using the new one.

## 5. RTP Multiplexing Design Choices

This section discusses how some RTP multiplexing design choices can be used in applications to achieve certain goals, and a summary of the implications of such choices. For each design there is discussion of benefits and downsides.

### 5.1. Multiple Media Types in one Session

This design uses a single RTP session for multiple different media types, like audio and video, and possibly also transport robustness mechanisms like FEC or Retransmission. An endpoint can have zero, one or more media sources per media type, resulting in a number of RTP streams of various media types and both source and redundancy type.

The Pros:

1. Single RTP session which implies:
  - \* Minimal NAT/FW state.
  - \* Minimal NAT/FW Traversal Cost.
  - \* Fate-sharing for all media flows.
2. Can handle dynamic allocations of RTP streams well on an RTP level. Depends on the application's needs for explicit indication of the stream usage and how timely that can be signalled.
3. Minimal overhead for security association establishment.

The Cons:

- a. Less suitable for interworking with other applications that uses individual RTP sessions per media type or multiple sessions for a single media type, due to the potential need of SSRC translation.
- b. Negotiation of bandwidth for the different media types is currently only possible using RID [I-D.ietf-mmusic-rid] in SDP.
- c. Not suitable for Split Component Terminal (see Section 3.10 of [RFC7667]).
- d. Flow-based QoS cannot provide separate treatment of RTP streams compared to others in the single RTP session.

- e. If there is significant asymmetry between the RTP streams' RTCP reporting needs, there are some challenges in configuration and usage to avoid wasting RTCP reporting on the RTP stream that does not need that frequent reporting.
- f. Not suitable for applications where some receivers like to receive only a subset of the RTP streams, especially if multicast or transport translator is being used.
- g. Additional concern with legacy implementations that do not support the RTP specification fully when it comes to handling multiple SSRC per endpoint, as also multiple simultaneous media types need to be handled.
- h. If the applications need finer control over which session participants that are included in different sets of security associations, most key-management will have difficulties establishing such a session.

## 5.2. Multiple SSRCs of the Same Media Type

In this design, each RTP session serves only a single media type. The RTP session can contain multiple RTP streams, either from a single endpoint or from multiple endpoints. This commonly creates a low number of RTP sessions, typically only one for audio and one for video, with a corresponding need for two listening ports when using RTP/RTCP multiplexing.

The Pros:

1. Works well with Split Component Terminal (see Section 3.10 of [RFC7667]) where the split is per media type.
2. Enables Flow-based QoS with different prioritisation between media types.
3. For applications with dynamic usage of RTP streams, i.e. frequently added and removed, having much of the state associated with the RTP session rather than per individual SSRC can avoid the need for in-session signalling of meta-information about each SSRC.
4. Low overhead for security association establishment.

The Cons:

- a. Slightly higher number of RTP sessions needed compared to Multiple Media Types in one Session Section 5.1. This implies:

- \* More NAT/FW state
- \* Increased NAT/FW Traversal Cost in both processing and delay.
- b. Some potential for concern with legacy implementations that don't support the RTP specification fully when it comes to handling multiple SSRC per endpoint.
- c. Not possible to control security association for sets of RTP streams within the same media type with today's key- management mechanisms, unless these are split into different RTP sessions.

For RTP applications where all RTP streams of the same media type share same usage, this structure provides efficiency gains in amount of network state used and provides more fate sharing with other media flows of the same type. At the same time, it is still maintaining almost all functionalities when it comes to negotiation in the signalling of the properties for the individual media type, and also enables flow based QoS prioritisation between media types. It handles multi-party session well, independently of multicast or centralised transport distribution, as additional sources can dynamically enter and leave the session.

### 5.3. Multiple Sessions for one Media type

This design goes one step further than above (Section 5.2) by using multiple RTP sessions also for a single media type. The main reason for going in this direction is that the RTP application needs separation of the RTP streams due to their usage. Some typical reasons for going to this design are scalability over multicast, simulcast, need for extended QoS prioritisation of RTP streams due to their usage in the application, or the need for fine-grained signalling using today's tools.

The Pros:

1. More suitable for multicast usage where receivers can individually select which RTP sessions they want to participate in, assuming each RTP session has its own multicast group.
2. The application can indicate its usage of the RTP streams on RTP session level, in case multiple different usages exist.
3. Less need for SSRC specific explicit signalling for each media stream and thus reduced need for explicit and timely signalling.
4. Enables detailed QoS prioritisation for flow-based mechanisms.



5. Works well with Split Component Terminal (see Section 3.10 of [RFC7667]).
6. The scope for who is included in a security association can be structured around the different RTP sessions, thus enabling such functionality with existing key-management.

The Cons:

- a. Increases the amount of RTP sessions compared to Multiple SSRCs of the Same Media Type.
- b. Increased amount of session configuration state.
- c. For RTP streams that are part of scalability, simulcast or transport robustness, a method to bind sources across multiple RTP sessions is needed.
- d. Some potential for concern with legacy implementations that does not support the RTP specification fully when it comes to handling multiple SSRC per endpoint.
- e. Higher overhead for security association establishment due to the increased number of RTP sessions.
- f. If the applications need finer control than on RTP session level over which participants that are included in different sets of security associations, most of today's key-management will have difficulties establishing such a session.

For more complex RTP applications that have several different usages for RTP streams of the same media type, or uses scalability or simulcast, this solution can enable those functions at the cost of increased overhead associated with the additional sessions. This type of structure is suitable for more advanced applications as well as multicast-based applications requiring differentiation to different participants.

#### 5.4. Single SSRC per Endpoint

In this design each endpoint in a point-to-point session has only a single SSRC, thus the RTP session contains only two SSRCs, one local and one remote. This session can be used both unidirectional, i.e. only a single RTP stream or bi-directional, i.e. both endpoints have one RTP stream each. If the application needs additional media flows between the endpoints, they will have to establish additional RTP sessions.

The Pros:

1. This design has great legacy interoperability potential as it will not tax any RTP stack implementations.
2. The signalling has good possibilities to negotiate and describe the exact formats and bit-rates for each RTP stream, especially using today's tools in SDP.
3. It is possible to control security association per RTP stream with current key-management, since each RTP stream is directly related to an RTP session, and the most used keying mechanisms operates on a per-session basis.

The Cons:

- a. The number of RTP sessions grows directly in proportion with the number of RTP streams, which has the implications:
  - \* Linear growth of the amount of NAT/FW state with number of RTP streams.
  - \* Increased delay and resource consumption from NAT/FW traversal.
  - \* Likely larger signalling message and signalling processing requirement due to the amount of session related information.
  - \* Higher potential for a single RTP stream to fail during transport between the endpoints.
- b. When the number of RTP sessions grows, the amount of explicit state for relating RTP streams also grows, depending on how the application needs to relate RTP streams.
- c. The port consumption might become a problem for centralised services, where the central node's port or 5-tuple filter consumption grows rapidly with the number of sessions.
- d. For applications where the RTP stream usage is highly dynamic, i.e. entering and leaving, the amount of signalling can grow high. Issues can also arise from the timely establishment of additional RTP sessions.
- e. If, against the recommendation, the same SSRC value is reused in multiple RTP sessions rather than being randomly chosen, interworking with applications that use a different multiplexing structure will require SSRC translation.

RTP applications that need to interwork with legacy RTP applications can potentially benefit from this structure. However, a large number of media descriptions in SDP can also run into issues with existing implementations. For any application needing a larger number of media flows, the overhead can become very significant. This structure is also not suitable for multi-party sessions, as any given RTP stream from each participant, although having same usage in the application, needs its own RTP session. In addition, the dynamic behaviour that can arise in multi-party applications can tax the signalling system and make timely media establishment more difficult.

### 5.5. Summary

There are some clear similarities between these designs. Both the "Single SSRC per Endpoint" and the "Multiple Media Types in one Session" are cases that require full explicit signalling of the media stream relations. However, they operate on two different levels where the first primarily enables session level binding, and the second needs SSRC level binding. From another perspective, the two solutions are the two extreme points when it comes to number of RTP sessions needed.

The two other designs "Multiple SSRCs of the Same Media Type" and "Multiple Sessions for one Media Type" are two examples that primarily allows for some implicit mapping of the role or usage of the RTP streams based on which RTP session they appear in. It thus potentially allows for less signalling and in particular reduces the need for real-time signalling in dynamic sessions. They also represent points in between the first two designs when it comes to amount of RTP sessions established, i.e. representing an attempt to balance the amount of RTP sessions with the functionality the communication session provides both on network level and on signalling level.

## 6. Guidelines

This section contains a number of multi-stream guidelines for implementers or specification writers.

Do not require use of the same SSRC value across RTP sessions:

As discussed in Section 3.4.3 there exist drawbacks in using the same SSRC in multiple RTP sessions as a mechanism to bind related RTP streams together. It is instead recommended to use a mechanism to explicitly signal the relation, either in RTP/RTCP or in the signalling mechanism used to establish the RTP session(s).

Use additional RTP streams for additional media sources: In the cases where an RTP endpoint needs to transmit additional RTP

streams of the same media type in the application, with the same processing requirements at the network and RTP layers, it is suggested to send them in the same RTP session. For example a telepresence room where there are three cameras, and each camera captures 2 persons sitting at the table, sending each camera as its own RTP stream within a single RTP session is suggested.

Use additional RTP sessions for streams with different requirements:

When RTP streams have different processing requirements from the network or the RTP layer at the endpoints, it is suggested that the different types of streams are put in different RTP sessions. This includes the case where different participants want different subsets of the set of RTP streams.

When using multiple RTP Sessions, use grouping: When using Multiple RTP session solutions, it is suggested to explicitly group the involved RTP sessions when needed using a signalling mechanism, for example The Session Description Protocol (SDP) Grouping Framework [RFC5888], using some appropriate grouping semantics.

RTP/RTCP Extensions Support Multiple RTP Streams as well as Multiple RTP sessions:

When defining an RTP or RTCP extension, the creator needs to consider if this extension is applicable to use with additional SSRCs and multiple RTP sessions. Any extension intended to be generic must support both. Extensions that are not as generally applicable will have to consider if interoperability is better served by defining a single solution or providing both options.

Transport Support Extensions: When defining new RTP/RTCP extensions intended for transport support, like the retransmission or FEC mechanisms, they must include support for both multiple RTP streams in the same RTP sessions and multiple RTP sessions, such that application developers can choose freely from the set of mechanisms without concerning themselves with which of the multiplexing choices a particular solution supports.

## 7. IANA Considerations

This document makes no request of IANA.

Note to RFC Editor: this section can be removed on publication as an RFC.

## 8. Security Considerations

The security considerations of the RTP specification [RFC3550] and any applicable RTP profile [RFC3551], [RFC4585], [RFC3711], the extensions for sending multiple media types in a single RTP session [I-D.ietf-avtcore-multi-media-rtp-session], RID [I-D.ietf-mmusic-rid], BUNDLE [I-D.ietf-mmusic-sdp-bundle-negotiation], [RFC5760], [RFC5761], apply if selected and thus needs to be considered in the evaluation.

There is discussion of the security implications of choosing multiple SSRC vs multiple RTP sessions in Section 4.3.

## 9. Contributors

Hui Zheng (Marvin) from Huawei contributed to WG draft versions -04 and -05 of the document.

## 10. References

### 10.1. Normative References

- [RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, DOI 10.17487/RFC3550, July 2003, <<https://www.rfc-editor.org/info/rfc3550>>.
- [RFC7656] Lennox, J., Gross, K., Nandakumar, S., Salgueiro, G., and B. Burman, Ed., "A Taxonomy of Semantics and Mechanisms for Real-Time Transport Protocol (RTP) Sources", RFC 7656, DOI 10.17487/RFC7656, November 2015, <<https://www.rfc-editor.org/info/rfc7656>>.

### 10.2. Informative References

- [ALF] Clark, D. and D. Tennenhouse, "Architectural Considerations for a New Generation of Protocols", SIGCOMM Symposium on Communications Architectures and Protocols (Philadelphia, Pennsylvania), pp. 200--208, IEEE Computer Communications Review, Vol. 20(4), September 1990.
- [I-D.ietf-avtcore-multi-media-rtp-session] Westerlund, M., Perkins, C., and J. Lennox, "Sending Multiple Types of Media in a Single RTP Session", draft-ietf-avtcore-multi-media-rtp-session-13 (work in progress), December 2015.

- [I-D.ietf-avtext-rid]  
Roach, A., Nandakumar, S., and P. Thatcher, "RTP Stream Identifier Source Description (SDS)", draft-ietf-avtext-rid-09 (work in progress), October 2016.
- [I-D.ietf-mmusic-rid]  
Roach, A., "RTP Payload Format Restrictions", draft-ietf-mmusic-rid-15 (work in progress), May 2018.
- [I-D.ietf-mmusic-sdp-bundle-negotiation]  
Holmberg, C., Alvestrand, H., and C. Jennings, "Negotiating Media Multiplexing Using the Session Description Protocol (SDP)", draft-ietf-mmusic-sdp-bundle-negotiation-53 (work in progress), September 2018.
- [I-D.ietf-mmusic-sdp-simulcast]  
Burman, B., Westerlund, M., Nandakumar, S., and M. Zanaty, "Using Simulcast in SDP and RTP Sessions", draft-ietf-mmusic-sdp-simulcast-13 (work in progress), June 2018.
- [I-D.ietf-perc-srtp-ekt-diet]  
Jennings, C., Mattsson, J., McGrew, D., Wing, D., and F. Andreassen, "Encrypted Key Transport for DTLS and Secure RTP", draft-ietf-perc-srtp-ekt-diet-09 (work in progress), October 2018.
- [RFC2198] Perkins, C., Kouvelas, I., Hodson, O., Hardman, V., Handley, M., Bolot, J., Vega-Garcia, A., and S. Fosse-Parisis, "RTP Payload for Redundant Audio Data", RFC 2198, DOI 10.17487/RFC2198, September 1997, <<https://www.rfc-editor.org/info/rfc2198>>.
- [RFC2205] Braden, R., Ed., Zhang, L., Berson, S., Herzog, S., and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, DOI 10.17487/RFC2205, September 1997, <<https://www.rfc-editor.org/info/rfc2205>>.
- [RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, DOI 10.17487/RFC2474, December 1998, <<https://www.rfc-editor.org/info/rfc2474>>.
- [RFC2974] Handley, M., Perkins, C., and E. Whelan, "Session Announcement Protocol", RFC 2974, DOI 10.17487/RFC2974, October 2000, <<https://www.rfc-editor.org/info/rfc2974>>.

- [RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, DOI 10.17487/RFC3261, June 2002, <<https://www.rfc-editor.org/info/rfc3261>>.
- [RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, DOI 10.17487/RFC3264, June 2002, <<https://www.rfc-editor.org/info/rfc3264>>.
- [RFC3389] Zopf, R., "Real-time Transport Protocol (RTP) Payload for Comfort Noise (CN)", RFC 3389, DOI 10.17487/RFC3389, September 2002, <<https://www.rfc-editor.org/info/rfc3389>>.
- [RFC3551] Schulzrinne, H. and S. Casner, "RTP Profile for Audio and Video Conferences with Minimal Control", STD 65, RFC 3551, DOI 10.17487/RFC3551, July 2003, <<https://www.rfc-editor.org/info/rfc3551>>.
- [RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, DOI 10.17487/RFC3711, March 2004, <<https://www.rfc-editor.org/info/rfc3711>>.
- [RFC3830] Arkko, J., Carrara, E., Lindholm, F., Naslund, M., and K. Norrman, "MIKEY: Multimedia Internet KEYing", RFC 3830, DOI 10.17487/RFC3830, August 2004, <<https://www.rfc-editor.org/info/rfc3830>>.
- [RFC4103] Hellstrom, G. and P. Jones, "RTP Payload for Text Conversation", RFC 4103, DOI 10.17487/RFC4103, June 2005, <<https://www.rfc-editor.org/info/rfc4103>>.
- [RFC4383] Baugher, M. and E. Carrara, "The Use of Timed Efficient Stream Loss-Tolerant Authentication (TESLA) in the Secure Real-time Transport Protocol (SRTP)", RFC 4383, DOI 10.17487/RFC4383, February 2006, <<https://www.rfc-editor.org/info/rfc4383>>.
- [RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, DOI 10.17487/RFC4566, July 2006, <<https://www.rfc-editor.org/info/rfc4566>>.
- [RFC4568] Andreasen, F., Baugher, M., and D. Wing, "Session Description Protocol (SDP) Security Descriptions for Media Streams", RFC 4568, DOI 10.17487/RFC4568, July 2006, <<https://www.rfc-editor.org/info/rfc4568>>.

- [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, DOI 10.17487/RFC4585, July 2006, <<https://www.rfc-editor.org/info/rfc4585>>.
- [RFC4588] Rey, J., Leon, D., Miyazaki, A., Varsa, V., and R. Hakenberg, "RTP Retransmission Payload Format", RFC 4588, DOI 10.17487/RFC4588, July 2006, <<https://www.rfc-editor.org/info/rfc4588>>.
- [RFC5104] Wenger, S., Chandra, U., Westerlund, M., and B. Burman, "Codec Control Messages in the RTP Audio-Visual Profile with Feedback (AVPF)", RFC 5104, DOI 10.17487/RFC5104, February 2008, <<https://www.rfc-editor.org/info/rfc5104>>.
- [RFC5109] Li, A., Ed., "RTP Payload Format for Generic Forward Error Correction", RFC 5109, DOI 10.17487/RFC5109, December 2007, <<https://www.rfc-editor.org/info/rfc5109>>.
- [RFC5576] Lennox, J., Ott, J., and T. Schierl, "Source-Specific Media Attributes in the Session Description Protocol (SDP)", RFC 5576, DOI 10.17487/RFC5576, June 2009, <<https://www.rfc-editor.org/info/rfc5576>>.
- [RFC5760] Ott, J., Chesterfield, J., and E. Schooler, "RTP Control Protocol (RTCP) Extensions for Single-Source Multicast Sessions with Unicast Feedback", RFC 5760, DOI 10.17487/RFC5760, February 2010, <<https://www.rfc-editor.org/info/rfc5760>>.
- [RFC5761] Perkins, C. and M. Westerlund, "Multiplexing RTP Data and Control Packets on a Single Port", RFC 5761, DOI 10.17487/RFC5761, April 2010, <<https://www.rfc-editor.org/info/rfc5761>>.
- [RFC5764] McGrew, D. and E. Rescorla, "Datagram Transport Layer Security (DTLS) Extension to Establish Keys for the Secure Real-time Transport Protocol (SRTP)", RFC 5764, DOI 10.17487/RFC5764, May 2010, <<https://www.rfc-editor.org/info/rfc5764>>.
- [RFC5888] Camarillo, G. and H. Schulzrinne, "The Session Description Protocol (SDP) Grouping Framework", RFC 5888, DOI 10.17487/RFC5888, June 2010, <<https://www.rfc-editor.org/info/rfc5888>>.



- [RFC6465] Ivov, E., Ed., Marocco, E., Ed., and J. Lennox, "A Real-time Transport Protocol (RTP) Header Extension for Mixer-to-Client Audio Level Indication", RFC 6465, DOI 10.17487/RFC6465, December 2011, <<https://www.rfc-editor.org/info/rfc6465>>.
- [RFC7201] Westerlund, M. and C. Perkins, "Options for Securing RTP Sessions", RFC 7201, DOI 10.17487/RFC7201, April 2014, <<https://www.rfc-editor.org/info/rfc7201>>.
- [RFC7657] Black, D., Ed. and P. Jones, "Differentiated Services (Diffserv) and Real-Time Communication", RFC 7657, DOI 10.17487/RFC7657, November 2015, <<https://www.rfc-editor.org/info/rfc7657>>.
- [RFC7667] Westerlund, M. and S. Wenger, "RTP Topologies", RFC 7667, DOI 10.17487/RFC7667, November 2015, <<https://www.rfc-editor.org/info/rfc7667>>.
- [RFC7826] Schulzrinne, H., Rao, A., Lanphier, R., Westerlund, M., and M. Stiemerling, Ed., "Real-Time Streaming Protocol Version 2.0", RFC 7826, DOI 10.17487/RFC7826, December 2016, <<https://www.rfc-editor.org/info/rfc7826>>.
- [RFC8088] Westerlund, M., "How to Write an RTP Payload Format", RFC 8088, DOI 10.17487/RFC8088, May 2017, <<https://www.rfc-editor.org/info/rfc8088>>.
- [RFC8108] Lennox, J., Westerlund, M., Wu, Q., and C. Perkins, "Sending Multiple RTP Streams in a Single RTP Session", RFC 8108, DOI 10.17487/RFC8108, March 2017, <<https://www.rfc-editor.org/info/rfc8108>>.
- [RFC8445] Keranen, A., Holmberg, C., and J. Rosenberg, "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal", RFC 8445, DOI 10.17487/RFC8445, July 2018, <<https://www.rfc-editor.org/info/rfc8445>>.

#### Appendix A. Dismissing Payload Type Multiplexing

This section documents a number of reasons why using the payload type as a multiplexing point is unsuitable for most things related to multiple RTP streams. If one attempts to use Payload type multiplexing beyond its defined usage, that has well known negative effects on RTP. To use payload type as the single discriminator for multiple streams implies that all the different RTP streams are being

sent with the same SSRC, thus using the same timestamp and sequence number space. This has many effects:

1. Putting restraint on RTP timestamp rate for the multiplexed media. For example, RTP streams that use different RTP timestamp rates cannot be combined, as the timestamp values need to be consistent across all multiplexed media frames. Thus streams are forced to use the same RTP timestamp rate. When this is not possible, payload type multiplexing cannot be used.
2. Many RTP payload formats can fragment a media object over multiple RTP packets, like parts of a video frame. These payload formats need to determine the order of the fragments to correctly decode them. Thus, it is important to ensure that all fragments related to a frame or a similar media object are transmitted in sequence and without interruptions within the object. This can relatively simple be solved on the sender side by ensuring that the fragments of each RTP stream are sent in sequence.
3. Some media formats require uninterrupted sequence number space between media parts. These are media formats where any missing RTP sequence number will result in decoding failure or invoking a repair mechanism within a single media context. The text/T140 payload format [RFC4103] is an example of such a format. These formats will need a sequence numbering abstraction function between RTP and the individual RTP stream before being used with payload type multiplexing.
4. Sending multiple streams in the same sequence number space makes it impossible to determine which payload type, which stream a packet loss relates to, and thus to which stream to potentially apply packet loss concealment or other stream-specific loss mitigation mechanisms.
5. If RTP Retransmission [RFC4588] is used and there is a loss, it is possible to ask for the missing packet(s) by SSRC and sequence number, not by payload type. If only some of the payload type multiplexed streams are of interest, there is no way of telling which missing packet(s) belong to the interesting stream(s) and all lost packets need be requested, wasting bandwidth.
6. The current RTCP feedback mechanisms are built around providing feedback on RTP streams based on stream ID (SSRC), packet (sequence numbers) and time interval (RTP Timestamps). There is almost never a field to indicate which payload type is reported,

so sending feedback for a specific RTP payload type is difficult without extending existing RTCP reporting.

7. The current RTCP media control messages [RFC5104] specification is oriented around controlling particular media flows, i.e. requests are done addressing a particular SSRC. Such mechanisms would need to be redefined to support payload type multiplexing.
8. The number of payload types are inherently limited. Accordingly, using payload type multiplexing limits the number of streams that can be multiplexed and does not scale. This limitation is exacerbated if one uses solutions like RTP and RTCP multiplexing [RFC5761] where a number of payload types are blocked due to the overlap between RTP and RTCP.
9. At times, there is a need to group multiplexed streams and this is currently possible for RTP sessions and for SSRC, but there is no defined way to group payload types.
10. It is currently not possible to signal bandwidth requirements per RTP stream when using payload type multiplexing.
11. Most existing SDP media level attributes cannot be applied on a per payload type level and would require re-definition in that context.
12. A legacy endpoint that does not understand the indication that different RTP payload types are different RTP streams might be slightly confused by the large amount of possibly overlapping or identically defined RTP payload types.

## Appendix B. Signalling Considerations

Signalling is not an architectural consideration for RTP itself, so this discussion has been moved to an appendix. However, it is hugely important for anyone building complete applications, so it is deserving of discussion.

The issues raised here need to be addressed in the WGs that deal with signalling; they cannot be addressed by tweaking, extending or profiling RTP.

There exist various signalling solutions for establishing RTP sessions. Many are SDP [RFC4566] based, however SDP functionality is also dependent on the signalling protocols carrying the SDP. RTSP [RFC7826] and SAP [RFC2974] both use SDP in a declarative fashion, while SIP [RFC3261] uses SDP with the additional definition of Offer/Answer [RFC3264]. The impact on signalling and especially SDP needs

to be considered as it can greatly affect how to deploy a certain multiplexing point choice.

#### B.1. Session Oriented Properties

One aspect of the existing signalling is that it is focused around RTP sessions, or at least in the case of SDP the media description. There are a number of things that are signalled on media description level but those are not necessarily strictly bound to an RTP session and could be of interest to signal specifically for a particular RTP stream (SSRC) within the session. The following properties have been identified as being potentially useful to signal not only on RTP session level:

- o Bitrate/Bandwidth exist today only at aggregate or as a common "any RTP stream" limit, unless either codec-specific bandwidth limiting or RTCP signalling using TMMBR is used.
- o Which SSRC that will use which RTP payload types (this will be visible from the first media packet, but is sometimes useful to know before packet arrival).

Some of these issues are clearly SDP's problem rather than RTP limitations. However, if the aim is to deploy an solution using additional SSRCs that contains several sets of RTP streams with different properties (encoding/packetization parameter, bit-rate, etc.), putting each set in a different RTP session would directly enable negotiation of the parameters for each set. If insisting on additional SSRC only, a number of signalling extensions are needed to clarify that there are multiple sets of RTP streams with different properties and that they need in fact be kept different, since a single set will not satisfy the application's requirements.

For some parameters, such as RTP payload type, resolution and framerate, a SSRC-linked mechanism has been proposed in [I-D.ietf-mmusic-rid]

#### B.2. SDP Prevents Multiple Media Types

SDP chose to use the m= line both to delineate an RTP session and to specify the top level of the MIME media type; audio, video, text, image, application. This media type is used as the top-level media type for identifying the actual payload format and is bound to a particular payload type using the rtpmap attribute. This binding has to be loosened in order to use SDP to describe RTP sessions containing multiple MIME top level types.

[I-D.ietf-mmusic-sdp-bundle-negotiation] describes how to let multiple SDP media descriptions use a single underlying transport in SDP, which allows to define one RTP session with media types having different MIME top level types.

### B.3. Signalling RTP stream Usage

RTP streams being transported in RTP has some particular usage in an RTP application. This usage of the RTP stream is in many applications so far implicitly signalled. For example, an application might choose to take all incoming audio RTP streams, mix them and play them out. However, in more advanced applications that use multiple RTP streams there will be more than a single usage or purpose among the set of RTP streams being sent or received. RTP applications will need to signal this usage somehow. The signalling used will have to identify the RTP streams affected by their RTP-level identifiers, which means that they have to be identified either by their session or by their SSRC + session.

In some applications, the receiver cannot utilise the RTP stream at all before it has received the signalling message describing the RTP stream and its usage. In other applications, there exists a default handling that is appropriate.

If all RTP streams in an RTP session are to be treated in the same way, identifying the session is enough. If SSRCS in a session are to be treated differently, signalling needs to identify both the session and the SSRC.

If this signalling affects how any RTP central node, like an RTP mixer or translator that selects, mixes or processes streams, treats the streams, the node will also need to receive the same signalling to know how to treat RTP streams with different usage in the right fashion.

### Authors' Addresses

Magnus Westerlund  
Ericsson  
Torshamsgatan 23  
SE-164 80 Kista  
Sweden

Phone: +46 10 714 82 87  
Email: magnus.westerlund@ericsson.com

Bo Burman  
Ericsson  
Gronlandsgatan 31  
SE-164 60 Kista  
Sweden

Phone: +46 10 714 13 11  
Email: bo.burman@ericsson.com

Colin Perkins  
University of Glasgow  
School of Computing Science  
Glasgow G12 8QQ  
United Kingdom

Email: csp@cspcrkins.org

Harald Tveit Alvestrand  
Google  
Kungsbron 2  
Stockholm 11122  
Sweden

Email: harald@alvestrand.no

Roni Even  
Huawei

Email: roni.even@huawei.com

AVTCORE WG  
Internet-Draft  
Intended status: Standards Track  
Expires: September 3, 2016

J. Lennox  
Vidyo  
M. Westerlund  
Ericsson  
Q. Wu  
Huawei  
C. Perkins  
University of Glasgow  
March 2, 2016

Sending Multiple RTP Streams in a Single RTP Session: Grouping RTCP  
Reception Statistics and Other Feedback  
draft-ietf-avtccore-rtp-multi-stream-optimisation-12

Abstract

RTP allows multiple RTP streams to be sent in a single session, but requires each Synchronisation Source (SSRC) to send RTCP reception quality reports for every other SSRC visible in the session. This causes the number of RTCP reception reports to grow with the number of SSRCs, rather than the number of endpoints. In many cases most of these RTCP reception reports are unnecessary, since all SSRCs of an endpoint are normally co-located and see the same reception quality. This memo defines a Reporting Group extension to RTCP to reduce the reporting overhead in such scenarios.

Status of This Memo

This Internet-Draft is submitted 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 September 3, 2016.

## Copyright Notice

Copyright (c) 2016 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 . . . . .	2
2. Terminology . . . . .	3
3. RTCP Reporting Groups . . . . .	3
3.1. Semantics and Behaviour of RTCP Reporting Groups . . . . .	4
3.2. Identifying Members of an RTCP Reporting Group . . . . .	5
3.2.1. Definition and Use of the RTCP RGRP SDDES Item . . . . .	5
3.2.2. Definition and Use of the RTCP RGRS Packet . . . . .	6
3.3. Interactions with the RTP/AVPF Feedback Profile . . . . .	8
3.4. Interactions with RTCP Extended Report (XR) Packets . . . . .	9
3.5. Middlebox Considerations . . . . .	9
3.6. SDP Signalling for Reporting Groups . . . . .	10
4. Properties of RTCP Reporting Groups . . . . .	12
4.1. Bandwidth Benefits of RTCP Reporting Groups . . . . .	12
4.2. Compatibility of RTCP Reporting Groups . . . . .	13
5. Security Considerations . . . . .	13
6. IANA Considerations . . . . .	15
7. References . . . . .	16
7.1. Normative References . . . . .	16
7.2. Informative References . . . . .	16
Authors' Addresses . . . . .	18

## 1. Introduction

The Real-time Transport Protocol (RTP) [RFC3550] is a protocol for group communication, supporting multiparty multimedia sessions. A single RTP session can support multiple participants sending at once, and can also support participants sending multiple simultaneous RTP streams. Examples of the latter might include a participant with multiple cameras who chooses to send multiple views of a scene, or a participant that sends audio and video flows multiplexed in a single RTP session. Rules for handling RTP sessions containing multiple RTP



streams are described in [RFC3550] with some clarifications in [I-D.ietf-avtcore-rtp-multi-stream].

An RTP endpoint will have one or more synchronisation sources (SSRCs). It will have at least one RTP Stream, and thus SSRC, for each media source it sends, and might use multiple SSRCs per media source when using media scalability features [RFC6190], forward error correction, RTP retransmission [RFC4588], or similar mechanisms. An endpoint that is not sending any RTP stream, will have at least one SSRC to use for reporting and any feedback messages. Each SSRC has to send RTCP sender reports corresponding to the RTP packets it sends, and receiver reports for traffic it receives. That is, every SSRC will send RTCP packets to report on every other SSRC. This rule is simple, but can be quite inefficient for endpoints that send large numbers of RTP streams in a single RTP session. Consider a session comprising ten participants, each sending three media sources, each with their own RTP stream. There will be 30 SSRCs in such an RTP session, and each of those 30 SSRCs will send an RTCP Sender Report/Receiver Report packet (containing several report blocks) per reporting interval as each SSRC reports on all the others. However, the three SSRCs comprising each participant are commonly co-located such that they see identical reception quality. If there was a way to indicate that several SSRCs are co-located, and see the same reception quality, then two-thirds of those RTCP reports could be suppressed. This would allow the remaining RTCP reports to be sent more often, while keeping within the same RTCP bandwidth fraction.

This memo defines such an RTCP extension, RTCP Reporting Groups. This extension is used to indicate the SSRCs that originate from the same endpoint, and therefore have identical reception quality, hence allowing the endpoints to suppress unnecessary RTCP reception quality reports.

## 2. 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 [RFC2119].

## 3. RTCP Reporting Groups

An RTCP Reporting Group is a set of synchronization sources (SSRCs) that are co-located at a single endpoint (which could be an end host or a middlebox) in an RTP session. Since they are co-located, every SSRC in the RTCP reporting group will have an identical view of the network conditions, and see the same lost packets, jitter, etc. This allows a single representative to send RTCP reception quality reports

on behalf of the rest of the reporting group, reducing the number of RTCP packets that need to be sent without loss of information.

### 3.1. Semantics and Behaviour of RTCP Reporting Groups

A group of co-located SSRCs that see identical network conditions can form an RTCP reporting group. If reporting groups are in use, an RTP endpoint with multiple SSRCs MAY put those SSRCs into a reporting group if their view of the network is identical; i.e., if they report on traffic received at the same interface of an RTP endpoint. SSRCs with different views of the network MUST NOT be put into the same reporting group.

An endpoint that has combined its SSRCs into an RTCP reporting group will choose one (or a subset) of those SSRCs to act as "reporting source(s)" for that RTCP reporting group. A reporting source will send RTCP SR/RR reception quality reports on behalf of the other members of the RTCP reporting group. A reporting source MUST suppress the RTCP SR/RR reports that relate to other members of the reporting group, and only report on remote SSRCs. The other members (non reporting sources) of the RTCP reporting group will suppress their RTCP reception quality reports, and instead send an RTCP RGRS packet (see Section 3.2.2) to indicate that they are part of an RTCP reporting group and give the SSRCs of the reporting sources.

If there are large numbers of remote SSRCs in the RTP session, then the reception quality reports generated by the reporting source might grow too large to fit into a single compound RTCP packet, forcing the reporting source to use a round-robin policy to determine what remote SSRCs it includes in each compound RTCP packet, and so reducing the frequency of reports on each SSRC. To avoid this, in sessions with large numbers of remote SSRCs, an RTCP reporting group MAY use more than one reporting source. If several SSRCs are acting as reporting sources for an RTCP reporting group, then each reporting source MUST have non-overlapping sets of remote SSRCs it reports on.

An endpoint MUST NOT create an RTCP reporting group that comprises only a single local SSRC (i.e., an RTCP reporting group where the reporting source is the only member of the group), unless it is anticipated that the group might have additional SSRCs added to it in the future.

If a reporting source leaves the RTP session (i.e., if it sends a RTCP BYE packet, or leaves the session without sending BYE under the rules of [RFC3550] section 6.3.7), the remaining members of the RTCP reporting group MUST either (a) have another reporting source, if one exists, report on the remote SSRCs the leaving SSRC reported on, (b) choose a new reporting source, or (c) disband the RTCP reporting

group and begin sending reception quality reports following [RFC3550] and [I-D.ietf-avtcore-rtp-multi-stream].

The RTCP timing rules assign different bandwidth fractions to senders and receivers. This lets senders transmit RTCP reception quality reports more often than receivers. If a reporting source in an RTCP reporting group is a receiver, but one or more non-reporting SSRCs in the RTCP reporting group are senders, then the endpoint MAY treat the reporting source as a sender for the purpose of RTCP bandwidth allocation, increasing its RTCP bandwidth allocation, provided it also treats one of the senders as if it were a receiver and makes the corresponding reduction in RTCP bandwidth for that SSRC. However, the application needs to consider the impact on the frequency of transmitting of the synchronization information included in RTCP Sender Reports.

### 3.2. Identifying Members of an RTCP Reporting Group

When RTCP Reporting Groups are in use, the other SSRCs in the RTP session need to be able to identify which SSRCs are members of an RTCP reporting group. Two RTCP extensions are defined to support this: the RTCP RGRP SDES item is used by the reporting source(s) to identify an RTCP reporting group, and the RTCP RGRS packet is used by other members of an RTCP reporting group to identify the reporting source(s).

#### 3.2.1. Definition and Use of the RTCP RGRP SDES Item

This document defines a new RTCP SDES item to identify an RTCP reporting group. The motivation for giving a reporting group an identify is to ensure that the RTCP reporting group and its member SSRCs can be correctly associated when there are multiple reporting sources, and to ensure that a reporting SSRC can be associated with the correct reporting group if an SSRC collision occurs.

This document defines the RTCP Source Description (SDES) RGRP item. The RTCP SDES RGRP item MUST be sent by the reporting sources in a reporting group, and MUST NOT be sent by other members of the reporting group or by SSRCs that are not members of any RTCP reporting group. Specifically, every reporting source in an RTCP reporting group MUST include an RTCP SDES packet containing an RGRP item in every compound RTCP packet in which it sends an RR or SR packet (i.e., in every RTCP packet it sends, unless Reduced-Size RTCP [RFC5506] is in use).

Syntactically, the format of the RTCP SDES RGRP item is identical to that of the RTCP SDES CNAME item [RFC7022], except that the SDES item type field MUST have value RGRP=(TBA) instead of CNAME=1. The value

of the RTCP SDES RGRP item MUST be chosen with the same concerns about global uniqueness and the same privacy considerations as the RTCP SDES CNAME. The value of the RTCP SDES RGRP item MUST be stable throughout the lifetime of the reporting group, even if some or all of the reporting sources change their SSRC due to collisions, or if the set of reporting sources changes.

Note to RFC Editor: please replace (TBA) in the above paragraph with the RTCP SDES item type number assigned to the RGRP item, then delete this note.

An RTP mixer or translator that forwards RTCP SR or RR packets from members of a reporting group MUST forward the corresponding RTCP SDES RGRP items as well, even if it otherwise strips SDES items other than the CNAME item.

### 3.2.2. Definition and Use of the RTCP RGRS Packet

A new RTCP packet type is defined to allow the members of an RTCP reporting group to identify the reporting sources for that group. This allows participants in an RTP session to distinguish an SSRC that is sending empty RTCP reception reports because it is a member of an RTCP reporting group, from an SSRC that is sending empty RTCP reception reports because it is not receiving any traffic. It also explicitly identifies the reporting sources, allowing other members of the RTP session to know which SSRCs are acting as the reporting sources for an RTCP reporting group, and allowing them to detect if RTCP packets from any of the reporting sources are being lost.

The format of the RTCP RGRS packet is defined below. It comprises the fixed RTCP header that indicates the packet type and length, the SSRC of the packet sender, and a list of reporting sources for the RTCP reporting group of which the packet sender is a member.

```

      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|      SC      | PT=RGRS(TBA) |      length      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|
|      SSRC of packet sender
|
+===+===+===+===+===+===+===+===+===+===+===+===+===+===+===+
:      List of SSRC(s) for the Reporting Source(s)      :
:                                                         :
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

The fields in the RTCP RGRS packet have the following definition:

version (V): 2 bits unsigned integer. This field identifies the RTP version. The current RTP version is 2.

padding (P): 1 bit. If set, the padding bit indicates that the RTCP packet contains additional padding octets at the end that are not part of the control information but are included in the length field. See [RFC3550].

Source Count (SC): 5 bits unsigned integer. Indicates the number of reporting source SSRCs that are included in this RTCP packet. As the RTCP RGRS packet MUST NOT be sent by reporting sources, all the SSRCs in the list of reporting sources will be different from the SSRC of the packet sender. Every RTCP RGRS packet MUST contain at least one reporting source SSRC.

Payload type (PT): 8 bits unsigned integer. The RTCP packet type number that identifies the packet as being an RTCP RGRS packet. The RGRS RTCP packet has the value [TBA].

Note to RFC Editor: please replace [TBA] here, and in the packet format diagram above, with the RTCP packet type that IANA assigns to the RTCP RGRS packet.

Length: 16 bits unsigned integer. The length of this packet in 32-bit words minus one, including the header and any padding. This is in line with the definition of the length field used in RTCP sender and receiver reports [RFC3550]. Since all RTCP RGRS packets include at least one reporting source SSRC, the length will always be 2 or greater.

SSRC of packet sender: 32 bits. The SSRC of the sender of this packet.

List of SSRCs for the Reporting Source(s): A variable length size (as indicated by SC header field) of the 32 bit SSRC values of the reporting sources for the RTCP Reporting Group of which the packet sender is a member.

Every source that belongs to an RTCP reporting group but is not a reporting source MUST include an RTCP RGRS packet in every compound RTCP packet in which it sends an RR or SR packet (i.e., in every RTCP packet it sends, unless Reduced-Size RTCP [RFC5506] is in use). Each RTCP RGRS packet MUST contain the SSRC identifier of at least one reporting source. If there are more reporting sources in an RTCP reporting group than can fit into an RTCP RGRS packet, the members of that reporting group MUST send the SSRCs of the reporting sources in a round-robin fashion in consecutive RTCP RGRS packets, such that all

the SSRCs of the reporting sources are included over the course of several RTCP reporting intervals.

An RTP mixer or translator that forwards RTCP SR or RR packets from members of a reporting group **MUST** also forward the corresponding RGRS RTCP packets. If the RTP mixer or translator rewrites SSRC values of the packets it forwards, it **MUST** make the corresponding changes to the RTCP RGRS packets.

### 3.3. Interactions with the RTP/AVPF Feedback Profile

Use of the RTP/AVPF Feedback Profile [RFC4585] allows SSRCs to send rapid RTCP feedback requests and codec control messages. If use of the RTP/AVPF profile has been negotiated in an RTP session, members of an RTCP reporting group can send rapid RTCP feedback and codec control messages following [RFC4585] and [RFC5104], as updated by Section 5.4 of [I-D.ietf-avtcore-rtp-multi-stream], and by the following considerations.

The members of an RTCP reporting group will all see identical network conditions. Accordingly, one might therefore think that it doesn't matter which SSRC in the reporting group sends the RTP/AVPF feedback or codec control messages. There might be, however, cases where the sender of the feedback/codec control message has semantic importance, or when only a subset of the members of an RTCP reporting group might want to send RTP/AVPF feedback or a codec control message in response to a particular event. For example, an RTP video sender might choose to treat packet loss feedback received from SSRCs known to be audio receivers with less urgency than feedback that it receives from video receivers when deciding what packets to retransmit, and a multimedia receiver using reporting groups might want to choose the outgoing SSRC for feedback packets to reflect this.

Each member of an RTCP reporting group **SHOULD** therefore send RTP/AVPF feedback/codec control messages independently of the other members of the reporting group, to respect the semantic meaning of the message sender. The suppression rules of [RFC4585] will ensure that only a single copy of each feedback packet is (typically) generated, even if several members of a reporting group send the same feedback. When an endpoint knows that several members of its RTCP reporting group will be sending identical feedback, and that the sender of the feedback is not semantically important, then that endpoint **MAY** choose to send all its feedback from the reporting source and deterministically suppress feedback packets generated by the other sources in the reporting group.

It is important to note that the RTP/AVPF timing rules operate on a per-SSRC basis. Using a single reporting source to send all feedback

for a reporting group will hence limit the amount of feedback that can be sent to that which can be sent by one SSRC. If this limit is a problem, then the reporting group can allow each of its members to send its own feedback, using its own SSRC.

If the RTP/AVPF feedback messages or codec control requests are sent as compound RTCP packets, then those compound RTCP packets **MUST** include either an RTCP RGRS packet or an RTCP SDES RGRP item, depending on whether they are sent by the reporting source or a non-reporting source in the RTCP reporting group respectively. The contents of non-compound RTCP feedback or codec control messages are not affected by the use of RTCP reporting groups.

#### 3.4. Interactions with RTCP Extended Report (XR) Packets

When using RTCP Extended Reports (XR) [RFC3611] with RTCP reporting groups, it is **RECOMMENDED** that the reporting source is used to send the RTCP XR packets. If multiple reporting sources are in use, the reporting source that sends the SR/RR packets that relate to a particular remote SSRC **SHOULD** send the RTCP XR reports about that SSRC. This is motivated as one commonly combine the RTCP XR metrics with the regular report block to more fully understand the situation. Receiving these blocks in different compound packets reduces their value as the measuring intervals are not synchronized in those cases.

Some RTCP XR report blocks are specific to particular types of media, and might be relevant to only some members of a reporting group. For example, it would make no sense for an SSRC that is receiving video to send a VoIP metric RTCP XR report block. Such media specific RTCP XR report blocks **MUST** be sent by the SSRC to which they are relevant, and **MUST NOT** be included in the common report sent by the reporting source. This might mean that some SSRCs send RTCP XR packets in compound RTCP packets that contain an empty RTCP SR/RR packet, and that the time period covered by the RTCP XR packet is different to that covered by the RTCP SR/RR packet. If it is important that the RTCP XR packet and RTCP SR/RR packet cover the same time period, then that source **SHOULD** be removed from the RTCP reporting group, and send standard RTCP packets instead.

#### 3.5. Middlebox Considerations

Many different types of middlebox are used with RTP. RTCP reporting groups are potentially relevant to those types of RTP middlebox that have their own SSRCs and generate RTCP reports for the traffic they receive. RTP middleboxes that do not have their own SSRC, and that don't send RTCP reports on the traffic they receive, cannot use the RTCP reporting groups extension, since they generate no RTCP reports to group.

An RTP middlebox that has several SSRCs of its own can use the RTCP reporting groups extension to group the RTCP reports it generates. This can occur, for example, if a middlebox is acting as an RTP mixer for both audio and video flows that are multiplexed onto a single RTP session, where the middlebox has one SSRC for the audio mixer and one for the video mixer part, and when the middlebox wants to avoid cross reporting between audio and video.

A middlebox cannot use the RTCP reporting groups extension to group RTCP packets from the SSRCs that it is forwarding. It can, however, group the RTCP packets from the SSRCs it is forwarding into compound RTCP packets following the rules in Section 6.1 of [RFC3550] and Section 5.3 of [I-D.ietf-avtcore-rtp-multi-stream]. If the middlebox is using RTCP reporting groups for its own SSRCs, it MAY include RTCP packets from the SSRCs that it is forwarding as part of the compound RTCP packets its reporting source generates.

A middlebox that forwards RTCP SR or RR packets sent by members of a reporting group MUST forward the corresponding RTCP SDES RGRP items, as described in Section 3.2.1. A middlebox that forwards RTCP SR or RR packets sent by member of a reporting group MUST also forward the corresponding RTCP RGRS packets, as described in Section 3.2.2. Failure to forward these packets can cause compatibility problems, as described in Section 4.2.

If a middlebox rewrites SSRC values in the RTP and RTCP packets that it is forwarding, then it MUST make the corresponding changes in RTCP SDES packets containing RGRP items and in RTCP RGRS packets, to allow them to be associated with the rewritten SSRCs.

### 3.6. SDP Signalling for Reporting Groups

This document defines the "a=rtcp-rgrp" Session Description Protocol (SDP) [RFC4566] attribute to indicate if the session participant is capable of supporting RTCP Reporting Groups for applications that use SDP for configuration of RTP sessions. It is a property attribute, and hence takes no value. The multiplexing category [I-D.ietf-mmusic-sdp-mux-attributes] is IDENTICAL, as the functionality applies on RTP session level. A participant that proposes the use of RTCP Reporting Groups SHALL itself support the reception of RTCP Reporting Groups. The formal definition of this attribute is:



Name: rtcp-rgrp  
Value:  
Usage Level: session, media  
Charset Dependent: no  
Example:  
    a=rtcp-rgrp

When using SDP Offer/Answer [RFC3264], the following procedures are to be used:

- o Generating the initial SDP offer: If the offerer supports the RTCP reporting group extensions, and is willing to accept RTCP packets containing those extensions, then it MUST include an "a=rtcp-rgrp" attribute in the initial offer. If the offerer does not support RTCP reporting groups extensions, or is not willing to accept RTCP packets containing those extensions, then it MUST NOT include the "a=rtcp-rgrp" attribute in the offer.
- o Generating the SDP answer: If the SDP offer contains an "a=rtcp-rgrp" attribute, and if the answerer supports RTCP reporting groups and is willing to receive RTCP packets using the RTCP reporting groups extensions, then the answerer MAY include an "a=rtcp-rgrp" attribute in the answer and MAY send RTCP packets containing the RTCP reporting groups extensions. If the offer does not contain an "a=rtcp-rgrp" attribute, or if the offer does contain such an attribute but the answerer does not wish to accept RTCP packets using the RTCP reporting groups extensions, then the answerer MUST NOT include an "a=rtcp-rgrp" attribute.
- o Offerer Processing of the SDP Answer: If the SDP answer contains an "a=rtcp-rgrp" attribute, and the corresponding offer also contained an "a=rtcp-rgrp" attribute, then the offerer MUST be prepared to accept and process RTCP packets that contain the reporting groups extension, and MAY send RTCP packets that contain the reporting groups extension. If the SDP answer contains an "a=rtcp-rgrp" attribute, but the corresponding offer did not contain the "a=rtcp-rgrp" attribute, then the offerer MUST reject the call. If the SDP answer does not contain an "a=rtcp-rgrp" attribute, then the offerer MUST NOT send packets containing the RTCP reporting groups extensions, and does not need to process packet containing the RTCP reporting groups extensions.

In declarative usage of SDP, such as the Real Time Streaming Protocol (RTSP) [RFC2326] and the Session Announcement Protocol (SAP) [RFC2974], the presence of the attribute indicates that the session participant MAY use RTCP Reporting Groups in its RTCP transmissions. An implementation that doesn't explicitly support RTCP Reporting Groups MAY join a RTP session as long as it has been verified that

the implementation doesn't suffer from the problems discussed in Section 4.2.

#### 4. Properties of RTCP Reporting Groups

This section provides additional information on what the resulting properties are with the design specified in Section 3. The content of this section is non-normative.

##### 4.1. Bandwidth Benefits of RTCP Reporting Groups

To understand the benefits of RTCP reporting groups, consider a scenario in which the two endpoints in a session each have a hundred sources, of which eight each are sending within any given reporting interval.

For ease of analysis, we can make the simplifying approximation that the duration of the RTCP reporting interval is equal to the total size of the RTCP packets sent during an RTCP interval, divided by the RTCP bandwidth. (This will be approximately true in scenarios where the bandwidth is not so high that the minimum RTCP interval is reached.) For further simplification, we can assume RTCP senders are following the recommendations regarding Compound RTCP Packets in [I-D.ietf-avtcore-rtp-multi-stream]; thus, the per-packet transport-layer overhead will be small relative to the RTCP data. Thus, only the actual RTCP data itself need be considered.

In a report interval in this scenario, there will, as a baseline, be 200 SDES packets, 184 RR packets, and 16 SR packets. This amounts to approximately 6.5 kB of RTCP per report interval, assuming 16-byte CNAMEs and no other SDES information.

Using the original [RFC3550] everyone-reports-on-every-sender feedback rules, each of the 184 receivers will send 16 report blocks, and each of the 16 senders will send 15. This amounts to approximately 76 kB of report block traffic per interval; 92% of RTCP traffic consists of report blocks.

If reporting groups are used, however, there is only 0.4 kB of reports per interval, with no loss of useful information. Additionally, there will be (assuming 16-byte RGRPs, and a single reporting source per reporting group) an additional 2.4 kB per cycle of RGRP SDES items and RGRS packets. Put another way, the unmodified [RFC3550] reporting interval is approximately 9 times longer than if reporting groups are in use.

#### 4.2. Compatibility of RTCP Reporting Groups

The RTCP traffic generated by receivers using RTCP Reporting Groups might appear, to observers unaware of these semantics, to be generated by receivers who are experiencing a network disconnection, as the non-reporting sources appear not to be receiving a given sender at all.

This could be a potentially critical problem for such a sender using RTCP for congestion control, as such a sender might think that it is sending so much traffic that it is causing complete congestion collapse.

However, such an interpretation of the session statistics would require a fairly sophisticated RTCP analysis. Any receiver of RTCP statistics which is just interested in information about itself needs to be prepared that any given reception report might not contain information about a specific media source, because reception reports in large conferences can be round-robin.

Thus, it is unclear to what extent such backward compatibility issues would actually cause trouble in practice.

#### 5. Security Considerations

The security considerations of [RFC3550] and [I-D.ietf-avtcore-rtp-multi-stream] apply. If the RTP/AVPF profile is in use, then the security considerations of [RFC4585] (and [RFC5104], if used) also apply. If RTCP XR is used, the security consideration of [RFC3611] and any XR report blocks used also apply.

The RTCP SDES RGRP item is vulnerable to malicious modifications unless integrity protection is used. A modification of this item's length field cause the parsing of the RTCP packet in which it is contained to fail. Depending on the implementation, parsing of the full compound RTCP packet can also fail causing the whole packet to be discarded. A modification to the value of this SDES item would make the receiver of the report think that the sender of the report was a member of a different RTCP reporting group. This will potentially create an inconsistency, when the RGRS reports the source as being in the same reporting group as another source with another reporting group identifier. What impact on a receiver implementation such inconsistencies would have are difficult to fully predict. One case is when congestion control or other adaptation mechanisms are used, an inconsistent report can result in a media sender to reduce its bit-rate. However, a direct modification of the receiver report or a feedback message itself would be a more efficient attack, and equally costly to perform.

The new RGRS RTCP Packet type is very simple. The common RTCP packet type header shares the security risks with previous RTCP packet types. Errors or modification of the length field can cause the full compound packet to fail header validation (see Appendix A.2 in [RFC3550]) resulting in the whole compound RTCP packet being discarded. Modification of the SC or P fields would cause inconsistency when processing the RTCP packet, likely resulting it being classified as invalid. A modification of the PT field would cause the packet being interpreted under some other packet type's rules. In such case the result might be more or less predictable but packet type specific. Modification of the SSRC of packet sender would attribute this packet to another sender. Resulting in a receiver believing the reporting group applies also for this SSRC, if it exists. If it doesn't exist, unless also corresponding modifications are done on a SR/RR packet and a SDES packet the RTCP packet SHOULD be discarded. If consistent changes are done, that could be part of a resource exhaustion attack on a receiver implementation. Modification of the "List of SSRCs for the Reporting Source(s)" would change the SSRC the receiver expect to report on behalf of this SSRC. If that SSRC exist, that could potentially change the report group used for this SSRC. A change to another reporting group belonging to another endpoint is likely detectable as there would be a mismatch between the SSRC of the packet sender's endpoint information, transport addresses, SDES CNAME etc and the corresponding information from the reporting group indicated.

In general the reporting group is providing limited impacts attacks. The most significant result from an deliberate attack would be to cause the information to be discarded or be inconsistent, including discard of all RTCP packets that are modified. This causes a lack of information at any receiver entity, possibly disregarding the endpoints participation in the session.

To protect against this type of attacks from external non trusted entities, integrity and source authentication SHOULD be applied. This can be done, for example, by using SRTP [RFC3711] with appropriate key-management, other options exist as discussed in RTP Security Options [RFC7201].

The Report Group Identifier has a potential privacy impacting properties. If this would be generated by an implementation in such a way that is long term stable or predictable, it could be used for tracking a particular end-point. Therefore it is RECOMMENDED that it be generated as a short-term persistent RGRP, following the rules for short-term persistent CNAMEs in [RFC7022]. The rest of the information revealed, i.e. the SSRCs, the size of reporting group and the number of reporting sources in a reporting group is of less sensitive nature, considering that the SSRCs and the communication

would anyway be revealed without this extension. By encrypting the report group extensions the SSRC values would be preserved confidential, but can still be revealed if SRTP [RFC3711] is used. The size of the reporting groups and number of reporting sources are likely determinable from analysis of the packet pattern and sizes. However, this information appears to have limited value.

## 6. IANA Considerations

(Note to the RFC-Editor: in the following, please replace "TBA" with the IANA-assigned value, and "XXXX" with the number of this document, then delete this note)

The IANA is requested to register one new RTCP SDES item in the "RTCP SDES Item Types" registry, as follows:

Value	Abbrev	Name	Reference
TBA	RGRP	Reporting Group Identifier	[RFCXXXX]

The definition of the RTCP SDES RGRP item is given in Section 3.2.1 of this memo.

The IANA is also requested to register one new RTCP packet type in the "RTCP Control Packet Types (PT)" Registry as follows:

Value	Abbrev	Name	Reference
TBA	RGRS	Reporting Group Reporting Sources	[RFCXXXX]

The definition of the RTCP RGRS packet type is given in Section 3.2.2 of this memo.

The IANA is also requested to register one new SDP attribute:

SDP Attribute ("att-field"):

Attribute name:	rtcp-rgrp
Long form:	RTCP Reporting Groups
Type of name:	att-field
Type of attribute:	Media or session level
Subject to charset:	No
Purpose:	Negotiate or configure the use of the RTCP Reporting Group Extension.
Reference:	[RFCXXXX]
Values:	None

The definition of the "a=rtcp-rgrp" SDP attribute is given in Section 3.6 of this memo.

## 7. References

### 7.1. Normative References

- [I-D.ietf-avtcore-rtp-multi-stream]  
Lennox, J., Westerlund, M., Wu, Q., and C. Perkins,  
"Sending Multiple RTP Streams in a Single RTP Session",  
draft-ietf-avtcore-rtp-multi-stream-11 (work in progress),  
December 2015.
- [I-D.ietf-mmusic-sdp-mux-attributes]  
Nandakumar, S., "A Framework for SDP Attributes when  
Multiplexing", draft-ietf-mmusic-sdp-mux-attributes-12  
(work in progress), January 2016.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate  
Requirement Levels", BCP 14, RFC 2119,  
DOI 10.17487/RFC2119, March 1997,  
<<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model  
with Session Description Protocol (SDP)", RFC 3264,  
DOI 10.17487/RFC3264, June 2002,  
<<http://www.rfc-editor.org/info/rfc3264>>.
- [RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V.  
Jacobson, "RTP: A Transport Protocol for Real-Time  
Applications", STD 64, RFC 3550, DOI 10.17487/RFC3550,  
July 2003, <<http://www.rfc-editor.org/info/rfc3550>>.
- [RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session  
Description Protocol", RFC 4566, DOI 10.17487/RFC4566,  
July 2006, <<http://www.rfc-editor.org/info/rfc4566>>.
- [RFC7022] Begen, A., Perkins, C., Wing, D., and E. Rescorla,  
"Guidelines for Choosing RTP Control Protocol (RTCP)  
Canonical Names (CNAMEs)", RFC 7022, DOI 10.17487/RFC7022,  
September 2013, <<http://www.rfc-editor.org/info/rfc7022>>.

### 7.2. Informative References

- [RFC2326] Schulzrinne, H., Rao, A., and R. Lanphier, "Real Time  
Streaming Protocol (RTSP)", RFC 2326,  
DOI 10.17487/RFC2326, April 1998,  
<<http://www.rfc-editor.org/info/rfc2326>>.

- [RFC2974] Handley, M., Perkins, C., and E. Whelan, "Session Announcement Protocol", RFC 2974, DOI 10.17487/RFC2974, October 2000, <<http://www.rfc-editor.org/info/rfc2974>>.
- [RFC3611] Friedman, T., Ed., Caceres, R., Ed., and A. Clark, Ed., "RTP Control Protocol Extended Reports (RTCP XR)", RFC 3611, DOI 10.17487/RFC3611, November 2003, <<http://www.rfc-editor.org/info/rfc3611>>.
- [RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, DOI 10.17487/RFC3711, March 2004, <<http://www.rfc-editor.org/info/rfc3711>>.
- [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, DOI 10.17487/RFC4585, July 2006, <<http://www.rfc-editor.org/info/rfc4585>>.
- [RFC4588] Rey, J., Leon, D., Miyazaki, A., Varsa, V., and R. Hakenberg, "RTP Retransmission Payload Format", RFC 4588, DOI 10.17487/RFC4588, July 2006, <<http://www.rfc-editor.org/info/rfc4588>>.
- [RFC5104] Wenger, S., Chandra, U., Westerlund, M., and B. Burman, "Codec Control Messages in the RTP Audio-Visual Profile with Feedback (AVPF)", RFC 5104, DOI 10.17487/RFC5104, February 2008, <<http://www.rfc-editor.org/info/rfc5104>>.
- [RFC5506] Johansson, I. and M. Westerlund, "Support for Reduced-Size Real-Time Transport Control Protocol (RTCP): Opportunities and Consequences", RFC 5506, DOI 10.17487/RFC5506, April 2009, <<http://www.rfc-editor.org/info/rfc5506>>.
- [RFC6190] Wenger, S., Wang, Y., Schierl, T., and A. Eleftheriadis, "RTP Payload Format for Scalable Video Coding", RFC 6190, DOI 10.17487/RFC6190, May 2011, <<http://www.rfc-editor.org/info/rfc6190>>.
- [RFC7201] Westerlund, M. and C. Perkins, "Options for Securing RTP Sessions", RFC 7201, DOI 10.17487/RFC7201, April 2014, <<http://www.rfc-editor.org/info/rfc7201>>.

## Authors' Addresses

Jonathan Lennox  
Vidyo, Inc.  
433 Hackensack Avenue  
Seventh Floor  
Hackensack, NJ 07601  
US

Email: [jonathan@vidyo.com](mailto:jonathan@vidyo.com)

Magnus Westerlund  
Ericsson  
Farogatan 2  
SE-164 80 Kista  
Sweden

Phone: +46 10 714 82 87  
Email: [magnus.westerlund@ericsson.com](mailto:magnus.westerlund@ericsson.com)

Qin Wu  
Huawei  
101 Software Avenue, Yuhua District  
Nanjing, Jiangsu 210012  
China

Email: [bill.wu@huawei.com](mailto:bill.wu@huawei.com)

Colin Perkins  
University of Glasgow  
School of Computing Science  
Glasgow G12 8QQ  
United Kingdom

Email: [csp@csp Perkins.org](mailto:csp@csp Perkins.org)



Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: September 29, 2019

M. Zanaty  
E. Berger  
S. Nandakumar  
Cisco Systems  
March 28, 2019

Frame Marking RTP Header Extension  
draft-ietf-avtext-framemarking-09

Abstract

This document describes a Frame Marking RTP header extension used to convey information about video frames that is critical for error recovery and packet forwarding in RTP middleboxes or network nodes. It is most useful when media is encrypted, and essential when the middlebox or node has no access to the media decryption keys. It is also useful for codec-agnostic processing of encrypted or unencrypted media, while it also supports extensions for codec-specific information.

Status of This Memo

This Internet-Draft is submitted 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 <https://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 September 29, 2019.

Copyright Notice

Copyright (c) 2019 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 (<https://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 . . . . .	2
2. Key Words for Normative Requirements . . . . .	4
3. Frame Marking RTP Header Extension . . . . .	4
3.1. Short Extension for Non-Scalable Streams . . . . .	4
3.2. Long Extension for Scalable Streams . . . . .	5
3.2.1. Layer ID Mappings for Scalable Streams . . . . .	7
3.2.1.1. H265 LID Mapping . . . . .	7
3.2.1.2. H264-SVC LID Mapping . . . . .	7
3.2.1.3. H264 (AVC) LID Mapping . . . . .	8
3.2.1.4. VP8 LID Mapping . . . . .	8
3.2.1.5. Future Codec LID Mapping . . . . .	8
3.3. Signaling Information . . . . .	8
3.4. Usage Considerations . . . . .	9
3.4.1. Relation to Layer Refresh Request (LRR) . . . . .	9
3.4.2. Scalability Structures . . . . .	9
4. Security Considerations . . . . .	10
5. Acknowledgements . . . . .	10
6. IANA Considerations . . . . .	10
7. References . . . . .	10
7.1. Normative References . . . . .	10
7.2. Informative References . . . . .	11
Authors' Addresses . . . . .	12

## 1. Introduction

Many widely deployed RTP [RFC3550] topologies [RFC7667] used in modern voice and video conferencing systems include a centralized component that acts as an RTP switch. It receives voice and video streams from each participant, which may be encrypted using SRTP [RFC3711], or extensions that provide participants with private media [I-D.ietf-perc-private-media-framework] via end-to-end encryption where the switch has no access to media decryption keys. The goal is to provide a set of streams back to the participants which enable them to render the right media content. In a simple video configuration, for example, the goal will be that each participant sees and hears just the active speaker. In that case, the goal of the switch is to receive the voice and video streams from each participant, determine the active speaker based on energy in the voice packets, possibly using the client-to-mixer audio level RTP header extension [RFC6464], and select the corresponding video stream for transmission to participants; see Figure 1.

In this document, an "RTP switch" is used as a common short term for the terms "switching RTP mixer", "source projecting middlebox", "source forwarding unit/middlebox" and "video switching MCU" as discussed in [RFC7667].

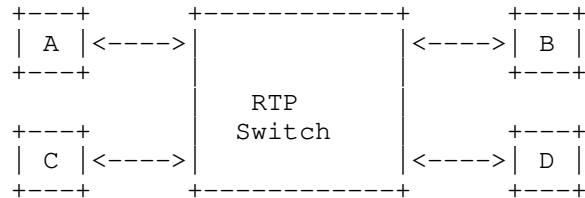


Figure 1: RTP switch

In order to properly support switching of video streams, the RTP switch typically needs some critical information about video frames in order to start and stop forwarding streams.

- o Because of inter-frame dependencies, it should ideally switch video streams at a point where the first frame from the new speaker can be decoded by recipients without prior frames, e.g switch on an intra-frame.
- o In many cases, the switch may need to drop frames in order to realize congestion control techniques, and needs to know which frames can be dropped with minimal impact to video quality.
- o Furthermore, it is highly desirable to do this in a payload format-agnostic way which is not specific to each different video codec. Most modern video codecs share common concepts around frame types and other critical information to make this codec-agnostic handling possible.
- o It is also desirable to be able to do this for SRTP without requiring the video switch to decrypt the packets. SRTP will encrypt the RTP payload format contents and consequently this data is not usable for the switching function without decryption, which may not even be possible in the case of end-to-end encryption of private media [I-D.ietf-perc-private-media-framework].

By providing meta-information about the RTP streams outside the encrypted media payload, an RTP switch can do codec-agnostic selective forwarding without decrypting the payload. This document specifies the necessary meta-information in an RTP header extension.

## 2. Key Words for Normative Requirements

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 [RFC2119].

## 3. Frame Marking RTP Header Extension

This specification uses RTP header extensions as defined in [RFC8285]. A subset of meta-information from the video stream is provided as an RTP header extension to allow an RTP switch to do generic selective forwarding of video streams encoded with potentially different video codecs.

The Frame Marking RTP header extension is encoded using the one-byte header or two-byte header as described in [RFC8285]. The one-byte header format is used for examples in this memo. The two-byte header format is used when other two-byte header extensions are present in the same RTP packet, since mixing one-byte and two-byte extensions is not possible in the same RTP packet.

This extension is only specified for Source (not Redundancy) RTP Streams [RFC7656] that carry video payloads. It is not specified for audio payloads, nor is it specified for Redundancy RTP Streams. The (separate) specifications for Redundancy RTP Streams often include provisions for recovering any header extensions that were part of the original source packet. Such provisions SHALL be followed to recover the Frame Marking RTP header extension of the original source packet. Source packet frame markings may be useful when generating Redundancy RTP Streams; for example, the I and D bits can be used to generate extra or no redundancy, respectively, and redundancy schemes with source blocks can align source block boundaries with Independent frame boundaries as marked by the I bit.

A frame, in the context of this specification, is the set of RTP packets with the same RTP timestamp from a specific RTP synchronization source (SSRC).

### 3.1. Short Extension for Non-Scalable Streams

The following RTP header extension is RECOMMENDED for non-scalable streams. It MAY also be used for scalable streams if the sender has limited or no information about stream scalability. The ID is assigned per [RFC8285], and the length is encoded as L=0 which indicates 1 octet of data.

```

      0                               1
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|  ID=?  |  L=0  |S|E|I|D|0 0 0 0|
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+

```

The following information are extracted from the media payload and sent in the Frame Marking RTP header extension.

- o S: Start of Frame (1 bit) - MUST be 1 in the first packet in a frame; otherwise MUST be 0.
- o E: End of Frame (1 bit) - MUST be 1 in the last packet in a frame; otherwise MUST be 0. SHOULD match the RTP header marker bit in payload formats with such semantics for marking end of frame.
- o I: Independent Frame (1 bit) - MUST be 1 for frames that can be decoded independent of temporally prior frames, e.g. intra-frame, VPX keyframe, H.264 IDR [RFC6184], H.265 IDR/CRA/BLA/RAP [RFC7798]; otherwise MUST be 0.
- o D: Discardable Frame (1 bit) - MUST be 1 for frames the sender knows can be discarded, and still provide a decodable media stream; otherwise MUST be 0.
- o The remaining (4 bits) - are reserved for future use for non-scalable streams; they MUST be set to 0 upon transmission and ignored upon reception.

### 3.2. Long Extension for Scalable Streams

The following RTP header extension is RECOMMENDED for scalable streams. It MAY also be used for non-scalable streams, in which case TID, LID and TL0PICIDX MUST be 0 or omitted. The ID is assigned per [RFC8285], and the length is encoded as L=2 which indicates 3 octets of data when nothing is omitted, or L=1 for 2 octets when TL0PICIDX is omitted, or L=0 for 1 octet when both LID and TL0PICIDX are omitted.

```

      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
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| ID=? | L=2 | S|E|I|D|B| TID | LID | TL0PICIDX |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
                        or
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| ID=? | L=1 | S|E|I|D|B| TID | LID | (TL0PICIDX omitted)
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
                        or
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| ID=? | L=0 | S|E|I|D|B| TID | (LID and TL0PICIDX omitted)
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

The following information are extracted from the media payload and sent in the Frame Marking RTP header extension.

- o S: Start of Frame (1 bit) - MUST be 1 in the first packet in a frame within a layer; otherwise MUST be 0.
- o E: End of Frame (1 bit) - MUST be 1 in the last packet in a frame within a layer; otherwise MUST be 0. Note that the RTP header marker bit MAY be used to infer the last packet of the highest enhancement layer, in payload formats with such semantics.
- o I: Independent Frame (1 bit) - MUST be 1 for frames that can be decoded independent of temporally prior frames, e.g. intra-frame, VPX keyframe, H.264 IDR [RFC6184], H.265 IDR/CRA/BLA/RAP [RFC7798]; otherwise MUST be 0. Note that this bit only signals temporal independence, so it can be 1 in spatial or quality enhancement layers that depend on temporally co-located layers but not temporally prior frames.
- o D: Discardable Frame (1 bit) - MUST be 1 for frames the sender knows can be discarded, and still provide a decodable media stream; otherwise MUST be 0.
- o B: Base Layer Sync (1 bit) - MUST be 1 if the sender knows this frame only depends on the base temporal layer; otherwise MUST be 0. If no scalability is used, this MUST be 0.
- o TID: Temporal ID (3 bits) - The base temporal layer starts with 0, and increases with 1 for each higher temporal layer/sub-layer. If no scalability is used, this MUST be 0.
- o LID: Layer ID (8 bits) - Identifies the spatial and quality layer encoded, starting with 0 and increasing with higher fidelity. If no scalability is used, this MUST be 0 or omitted to reduce length. When omitted, TL0PICIDX MUST also be omitted.
- o TL0PICIDX: Temporal Layer 0 Picture Index (8 bits) - Running index of base temporal layer 0 frames when TID is 0. When TID is not 0, this indicates a dependency on the given index. If no scalability is used, or the running index is unknown, this MUST be omitted to

reduce length. Note that 0 is a valid running index value for TL0PICIDX.

The layer information contained in TID and LID convey useful aspects of the layer structure that can be utilized in selective forwarding. Without further information about the layer structure, these identifiers can only be used for relative priority of layers. They convey a layer hierarchy with TID=0 and LID=0 identifying the base layer. Higher values of TID identify higher temporal layers with higher frame rates. Higher values of LID identify higher spatial and/or quality layers with higher resolutions and/or bitrates.

With further information, for example, possible future RTCP SDES items that convey full layer structure information, it may be possible to map these TIDs and LIDs to specific frame rates, resolutions and bitrates. Such additional layer information may be useful for forwarding decisions in the RTP switch, but is beyond the scope of this memo. The relative layer information is still useful for many selective forwarding decisions even without such additional layer information.

### 3.2.1. Layer ID Mappings for Scalable Streams

### 3.2.1.1. H265 LID Mapping

The following shows the H265 [RFC7798] LayerID (6 bits) and TID (3 bits) from the NAL unit header mapped to the generic LID and TID fields.

The I bit MUST be 1 when the NAL unit type is 16-23 (inclusive), otherwise it MUST be 0.

The S and E bits MUST match the corresponding bits in PACI:PHES:TSCI payload structures.

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	2	3	4	5	6	7	8	9	0	1																		
ID=2										L=2										S E I D B										TID 0 0										LayerID										TL0PICIDX									

### 3.2.1.2. H264-SVC LID Mapping

The following shows H264-SVC [RFC6190] Layer encoding information (3 bits for spatial/dependency layer, 4 bits for quality layer and 3 bits for temporal layer) mapped to the generic LID and TID fields.

The S, E, I and D bits MUST match the corresponding bits in PACSI payload structures.

```

      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
+-----+-----+-----+-----+-----+-----+-----+-----+
| ID=2 | L=2 | S|E|I|D|B| TID | 0 | DID | QID | TL0PICIDX |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

#### 3.2.1.3. H264 (AVC) LID Mapping

The following shows the header extension for H264 (AVC) [RFC6184] that contains only temporal layer information.

```

      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
+-----+-----+-----+-----+-----+-----+-----+-----+
| ID=2 | L=2 | S|E|I|D|B| TID | 0|0|0|0|0|0|0|0| TL0PICIDX |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

#### 3.2.1.4. VP8 LID Mapping

The following shows the header extension for VP8 [RFC7741] that contains only temporal layer information.

```

      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
+-----+-----+-----+-----+-----+-----+-----+-----+
| ID=2 | L=2 | S|E|I|D|B| TID | 0|0|0|0|0|0|0|0| TL0PICIDX |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

#### 3.2.1.5. Future Codec LID Mapping

The RTP payload format specification for future video codecs SHOULD include a section describing the LID mapping and TID mapping for the codec. For example, the LID/TID mapping for the VP9 codec is described in the VP9 RTP Payload Format [I-D.ietf-payload-vp9].

### 3.3. Signaling Information

The URI for declaring this header extension in an extmap attribute is "urn:ietf:params:rtp-hdext:framemarking". It does not contain any extension attributes.

An example attribute line in SDP:

```
a=extmap:3 urn:ietf:params:rtp-hdext:framemarking
```



### 3.4. Usage Considerations

The header extension values MUST represent what is already in the RTP payload.

When an RTP switch needs to discard a received video frame due to congestion control considerations, it is RECOMMENDED that it preferably drop frames marked with the D (Discardable) bit set, or the highest values of TID and LID, which indicate the highest temporal and spatial/quality enhancement layers, since those typically have fewer dependencies on them than lower layers.

When an RTP switch wants to forward a new video stream to a receiver, it is RECOMMENDED to select the new video stream from the first switching point with the I (Independent) bit set in all spatial layers and forward the same. An RTP switch can request a media source to generate a switching point by sending Full Intra Request (RTCP FIR) as defined in [RFC5104], for example.

#### 3.4.1. Relation to Layer Refresh Request (LRR)

Receivers can use the Layer Refresh Request (LRR) [I-D.ietf-avtext-lrr] RTCP feedback message to upgrade to a higher layer in scalable encodings. The TID/LID values and formats used in LRR messages MUST correspond to the same values and formats specified in Section 3.2.

Because frame marking can only be used with temporally-nested streams, temporal-layer LRR refreshes are unnecessary for frame-marked streams. Other refreshes can be detected based on the I bit being set for the specific spatial layers.

#### 3.4.2. Scalability Structures

The LID and TID information is most useful for fixed scalability structures, such as nested hierarchical temporal layering structures, where each temporal layer only references lower temporal layers or the base temporal layer. The LID and TID information is less useful, or even not useful at all, for complex, irregular scalability structures that do not conform to common, fixed patterns of inter-layer dependencies and referencing structures. Therefore it is RECOMMENDED to use LID and TID information for RTP switch forwarding decisions only in the case of temporally nested scalability structures, and it is NOT RECOMMENDED for other (more complex or irregular) scalability structures.

#### 4. Security Considerations

In the Secure Real-Time Transport Protocol (SRTP) [RFC3711], RTP header extensions are authenticated but usually not encrypted. When header extensions are used some of the payload type information are exposed and visible to middle boxes. The encrypted media data is not exposed, so this is not seen as a high risk exposure.

#### 5. Acknowledgements

Many thanks to Bernard Aboba, Jonathan Lennox, and Stephan Wenger for their inputs.

#### 6. IANA Considerations

This document defines a new extension URI to the RTP Compact HeaderExtensions sub-registry of the Real-Time Transport Protocol (RTP) Parameters registry, according to the following data:

Extension URI: urn:ietf:params:rtp-hdext:framemarkinginfo  
Description: Frame marking information for video streams  
Contact: mzanaty@cisco.com  
Reference: RFC XXXX

Note to RFC Editor: please replace RFC XXXX with the number of this RFC.

#### 7. References

##### 7.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC6184] Wang, Y., Even, R., Kristensen, T., and R. Jesup, "RTP Payload Format for H.264 Video", RFC 6184, DOI 10.17487/RFC6184, May 2011, <<https://www.rfc-editor.org/info/rfc6184>>.
- [RFC6190] Wenger, S., Wang, Y., Schierl, T., and A. Eleftheriadis, "RTP Payload Format for Scalable Video Coding", RFC 6190, DOI 10.17487/RFC6190, May 2011, <<https://www.rfc-editor.org/info/rfc6190>>.

- [RFC7741] Westin, P., Lundin, H., Glover, M., Uberti, J., and F. Galligan, "RTP Payload Format for VP8 Video", RFC 7741, DOI 10.17487/RFC7741, March 2016, <<https://www.rfc-editor.org/info/rfc7741>>.
- [RFC7798] Wang, Y., Sanchez, Y., Schierl, T., Wenger, S., and M. Hannuksela, "RTP Payload Format for High Efficiency Video Coding (HEVC)", RFC 7798, DOI 10.17487/RFC7798, March 2016, <<https://www.rfc-editor.org/info/rfc7798>>.
- [RFC8285] Singer, D., Desineni, H., and R. Even, Ed., "A General Mechanism for RTP Header Extensions", RFC 8285, DOI 10.17487/RFC8285, October 2017, <<https://www.rfc-editor.org/info/rfc8285>>.

## 7.2. Informative References

- [I-D.ietf-avtext-lrr] Lennox, J., Hong, D., Uberti, J., Holmer, S., and M. Flodman, "The Layer Refresh Request (LRR) RTCP Feedback Message", draft-ietf-avtext-lrr-07 (work in progress), July 2017.
- [I-D.ietf-payload-vp9] Uberti, J., Holmer, S., Flodman, M., Lennox, J., and D. Hong, "RTP Payload Format for VP9 Video", draft-ietf-payload-vp9-06 (work in progress), July 2018.
- [I-D.ietf-perc-private-media-framework] Jones, P., Benham, D., and C. Groves, "A Solution Framework for Private Media in Privacy Enhanced RTP Conferencing", draft-ietf-perc-private-media-framework-09 (work in progress), February 2019.
- [RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, DOI 10.17487/RFC3550, July 2003, <<https://www.rfc-editor.org/info/rfc3550>>.
- [RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, DOI 10.17487/RFC3711, March 2004, <<https://www.rfc-editor.org/info/rfc3711>>.
- [RFC5104] Wenger, S., Chandra, U., Westerlund, M., and B. Burman, "Codec Control Messages in the RTP Audio-Visual Profile with Feedback (AVPF)", RFC 5104, DOI 10.17487/RFC5104, February 2008, <<https://www.rfc-editor.org/info/rfc5104>>.

- [RFC6464] Lennox, J., Ed., Ivov, E., and E. Marocco, "A Real-time Transport Protocol (RTP) Header Extension for Client-to-Mixer Audio Level Indication", RFC 6464, DOI 10.17487/RFC6464, December 2011, <<https://www.rfc-editor.org/info/rfc6464>>.
- [RFC7656] Lennox, J., Gross, K., Nandakumar, S., Salgueiro, G., and B. Burman, Ed., "A Taxonomy of Semantics and Mechanisms for Real-Time Transport Protocol (RTP) Sources", RFC 7656, DOI 10.17487/RFC7656, November 2015, <<https://www.rfc-editor.org/info/rfc7656>>.
- [RFC7667] Westerlund, M. and S. Wenger, "RTP Topologies", RFC 7667, DOI 10.17487/RFC7667, November 2015, <<https://www.rfc-editor.org/info/rfc7667>>.

## Authors' Addresses

Mo Zanaty  
Cisco Systems  
170 West Tasman Drive  
San Jose, CA 95134  
US

Email: [mzanaty@cisco.com](mailto:mzanaty@cisco.com)

Espen Berger  
Cisco Systems

Phone: +47 98228179  
Email: [espeberg@cisco.com](mailto:espeberg@cisco.com)

Suhas Nandakumar  
Cisco Systems  
170 West Tasman Drive  
San Jose, CA 95134  
US

Email: [snandaku@cisco.com](mailto:snandaku@cisco.com)

Network Working Group  
Internet-Draft  
Intended status: Experimental  
Expires: 15 May 2022

M. Zanaty  
E. Berger  
S. Nandakumar  
Cisco Systems  
November 2021

Frame Marking RTP Header Extension  
draft-ietf-avtext-framemarking-13

## Abstract

This document describes a Frame Marking RTP header extension used to convey information about video frames that is critical for error recovery and packet forwarding in RTP middleboxes or network nodes. It is most useful when media is encrypted, and essential when the middlebox or node has no access to the media decryption keys. It is also useful for codec-agnostic processing of encrypted or unencrypted media, while it also supports extensions for codec-specific information.

## Status of This Memo

This Internet-Draft is submitted 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 <https://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 5 May 2022.

## Copyright Notice

Copyright (c) 2021 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 (<https://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 . . . . .	2
2. Key Words for Normative Requirements . . . . .	4
3. Frame Marking RTP Header Extension . . . . .	4
3.1. Long Extension for Scalable Streams . . . . .	5
3.2. Short Extension for Non-Scalable Streams . . . . .	7
3.3. Layer ID Mappings for Scalable Streams . . . . .	7
3.3.1. VP9 LID Mapping . . . . .	7
3.3.2. H265 LID Mapping . . . . .	8
3.3.3. H264-SVC LID Mapping . . . . .	9
3.3.4. H264 (AVC) LID Mapping . . . . .	9
3.3.5. VP8 LID Mapping . . . . .	10
3.3.6. Future Codec LID Mapping . . . . .	11
3.4. Signaling Information . . . . .	11
3.5. Usage Considerations . . . . .	11
3.5.1. Relation to Layer Refresh Request (LRR) . . . . .	12
3.5.2. Scalability Structures . . . . .	12
4. Security Considerations . . . . .	12
5. Acknowledgements . . . . .	12
6. IANA Considerations . . . . .	12
7. References . . . . .	13
7.1. Normative References . . . . .	13
7.2. Informative References . . . . .	13
Authors' Addresses . . . . .	14

## 1. Introduction

Many widely deployed RTP [RFC3550] topologies [RFC7667] used in modern voice and video conferencing systems include a centralized component that acts as an RTP switch. It receives voice and video streams from each participant, which may be encrypted using SRTP [RFC3711], or extensions that provide participants with private media [RFC8871] via end-to-end encryption where the switch has no access to media decryption keys. The goal is to provide a set of streams back to the participants which enable them to render the right media content. In a simple video configuration, for example, the goal will be that each participant sees and hears just the active speaker. In that case, the goal of the switch is to receive the voice and video streams from each participant, determine the active speaker based on energy in the voice packets, possibly using the client-to-mixer audio level RTP header extension [RFC6464], and select the corresponding video stream for transmission to participants; see Figure 1.

In this document, an "RTP switch" is used as a common short term for the terms "switching RTP mixer", "source projecting middlebox", "source forwarding unit/middlebox" and "video switching MCU" as discussed in [RFC7667].

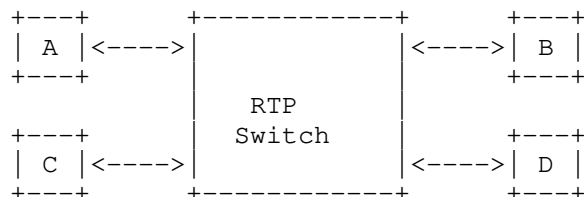


Figure 1: RTP switch

In order to properly support switching of video streams, the RTP switch typically needs some critical information about video frames in order to start and stop forwarding streams.

- \* Because of inter-frame dependencies, it should ideally switch video streams at a point where the first frame from the new speaker can be decoded by recipients without prior frames, e.g. switch on an intra-frame.
- \* In many cases, the switch may need to drop frames in order to realize congestion control techniques, and needs to know which frames can be dropped with minimal impact to video quality.
- \* For scalable streams with dependent layers, the switch may need to selectively forward specific layers to specific recipients due to recipient bandwidth or decoder limits.
- \* Furthermore, it is highly desirable to do this in a payload format-agnostic way which is not specific to each different video codec. Most modern video codecs share common concepts around frame types and other critical information to make this codec-agnostic handling possible.
- \* It is also desirable to be able to do this for SRTP without requiring the video switch to decrypt the packets. SRTP will encrypt the RTP payload format contents and consequently this data is not usable for the switching function without decryption, which may not even be possible in the case of end-to-end encryption of private media [RFC8871].

By providing meta-information about the RTP streams outside the encrypted media payload, an RTP switch can do codec-agnostic selective forwarding without decrypting the payload. This document specifies the necessary meta-information in an RTP header extension.

## 2. Key Words for Normative Requirements

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

## 3. Frame Marking RTP Header Extension

This specification uses RTP header extensions as defined in [RFC8285]. A subset of meta-information from the video stream is provided as an RTP header extension to allow an RTP switch to do generic selective forwarding of video streams encoded with potentially different video codecs.

The Frame Marking RTP header extension is encoded using the one-byte header or two-byte header as described in [RFC8285]. The one-byte header format is used for examples in this memo. The two-byte header format is used when other two-byte header extensions are present in the same RTP packet, since mixing one-byte and two-byte extensions is not possible in the same RTP packet.

This extension is only specified for Source (not Redundancy) RTP Streams [RFC7656] that carry video payloads. It is not specified for audio payloads, nor is it specified for Redundancy RTP Streams. The (separate) specifications for Redundancy RTP Streams often include provisions for recovering any header extensions that were part of the original source packet. Such provisions SHALL be followed to recover the Frame Marking RTP header extension of the original source packet. Source packet frame markings may be useful when generating Redundancy RTP Streams; for example, the I and D bits can be used to generate extra or no redundancy, respectively, and redundancy schemes with source blocks can align source block boundaries with Independent frame boundaries as marked by the I bit.

A frame, in the context of this specification, is the set of RTP packets with the same RTP timestamp from a specific RTP synchronization source (SSRC). A frame within a layer is the set of RTP packets with the same RTP timestamp, SSRC, Temporal ID (TID), and Layer ID (LID).



### 3.1. Long Extension for Scalable Streams

The following RTP header extension is RECOMMENDED for scalable streams. It MAY also be used for non-scalable streams, in which case TID, LID and TLOPICIDX MUST be 0 or omitted. The ID is assigned per [RFC8285], and the length is encoded as L=2 which indicates 3 octets of data when nothing is omitted, or L=1 for 2 octets when TLOPICIDX is omitted, or L=0 for 1 octet when both LID and TLOPICIDX are omitted.

```

      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
+-----+-----+-----+-----+-----+-----+-----+-----+
| ID=? | L=2 | S|E|I|D|B| TID | LID | TLOPICIDX |
+-----+-----+-----+-----+-----+-----+-----+-----+
      or
+-----+-----+-----+-----+-----+-----+-----+-----+
| ID=? | L=1 | S|E|I|D|B| TID | LID | (TLOPICIDX omitted)
+-----+-----+-----+-----+-----+-----+-----+-----+
      or
+-----+-----+-----+-----+-----+-----+-----+-----+
| ID=? | L=0 | S|E|I|D|B| TID | (LID and TLOPICIDX omitted)
+-----+-----+-----+-----+-----+-----+-----+-----+

```

The following information are extracted from the media payload and sent in the Frame Marking RTP header extension.

- \* S: Start of Frame (1 bit) - MUST be 1 in the first packet in a frame within a layer; otherwise MUST be 0.
- \* E: End of Frame (1 bit) - MUST be 1 in the last packet in a frame within a layer; otherwise MUST be 0. Note that the RTP header marker bit MAY be used to infer the last packet of the highest enhancement layer, in payload formats with such semantics.
- \* I: Independent Frame (1 bit) - MUST be 1 for a frame within a layer that can be decoded independent of temporally prior frames, e.g. intra-frame, VPX keyframe, H.264 IDR [RFC6184], H.265 IDR/CRA/BLA/RAP [RFC7798]; otherwise MUST be 0. Note that this bit only signals temporal independence, so it can be 1 in spatial or quality enhancement layers that depend on temporally co-located layers but not temporally prior frames.
- \* D: Discardable Frame (1 bit) - MUST be 1 for a frame within a layer the sender knows can be discarded, and still provide a decodable media stream; otherwise MUST be 0.
- \* B: Base Layer Sync (1 bit) - When TID is not 0, this MUST be 1 if the sender knows this frame within a layer only depends on the base temporal layer; otherwise MUST be 0. When TID is 0 or if no scalability is used, this MUST be 0.

- \* TID: Temporal ID (3 bits) - Identifies the temporal layer/sub-layer encoded, starting with 0 for the base layer, and increasing with higher temporal fidelity. If no scalability is used, this MUST be 0. It is implicitly 0 in the short extension format.
- \* LID: Layer ID (8 bits) - Identifies the spatial and quality layer encoded, starting with 0 for the base layer, and increasing with higher fidelity. If no scalability is used, this MUST be 0 or omitted to reduce length. When omitted, TLOPICIDX MUST also be omitted. It is implicitly 0 in the short extension format or when omitted in the long extension format.
- \* TLOPICIDX: Temporal Layer 0 Picture Index (8 bits) - When TID is 0 and LID is 0, this is a cyclic counter labeling base layer frames. When TID is not 0 or LID is not 0, this indicates a dependency on the given index, such that this frame within this layer depends on the frame with this label in the layer with TID 0 and LID 0. If no scalability is used, or the cyclic counter is unknown, this MUST be omitted to reduce length. Note that 0 is a valid index value for TLOPICIDX.

The layer information contained in TID and LID convey useful aspects of the layer structure that can be utilized in selective forwarding.

Without further information about the layer structure, these TID/LID identifiers can only be used for relative priority of layers and implicit dependencies between layers. They convey a layer hierarchy with TID=0 and LID=0 identifying the base layer. Higher values of TID identify higher temporal layers with higher frame rates. Higher values of LID identify higher spatial and/or quality layers with higher resolutions and/or bitrates. Implicit dependencies between layers assume that a layer with a given TID/LID MAY depend on layer(s) with the same or lower TID/LID, but MUST NOT depend on layer(s) with higher TID/LID.

With further information, for example, possible future RTCP SDES items that convey full layer structure information, it may be possible to map these TIDs and LIDs to specific absolute frame rates, resolutions and bitrates, as well as explicit dependencies between layers. Such additional layer information may be useful for forwarding decisions in the RTP switch, but is beyond the scope of this memo. The relative layer information is still useful for many selective forwarding decisions even without such additional layer information.



The S bit MUST match the B bit in the VP9 payload descriptor.

The E bit MUST match the E bit in the VP9 payload descriptor.

The I bit MUST match the inverse of the P bit in the VP9 payload descriptor.

The D bit MUST be 1 if the refresh\_frame\_flags in the VP9 payload uncompressed header are all 0, otherwise it MUST be 0.

The B bit MUST be 0 if TID is 0; otherwise, if TID is not 0, it MUST match the U bit in the VP9 payload descriptor. Note: When using temporally nested scalability structures as recommended in Section 3.5.2, the B bit and VP9 U bit will always be 1 if TID is not 0, since it is always possible to switch up to a higher temporal layer in such nested structures.

TID and TLOPICIDX MUST match the correspondingly named fields in the VP9 payload descriptor.

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	2	3	4	5	6	7	8	9	0	1																												
ID=?										L=2										S E I D B										TID										0 0 0 0 0										SID										TLOPICIDX									

### 3.3.2. H265 LID Mapping

The following shows the H265 [RFC7798] LayerID (6 bits) and TID (3 bits) from the NAL unit header mapped to the generic LID and TID fields.

The S and E bits MUST match the correspondingly named bits in PACI:PHES:TSCI payload structures.

The I bit MUST be 1 when the NAL unit type is 16-23 (inclusive) or 32-34 (inclusive), or an aggregation packet or fragmentation unit encapsulating any of these types, otherwise it MUST be 0. These ranges cover intra (IRAP) frames as well as critical parameter sets (VPS, SPS, PPS).

The D bit MUST be 1 when the NAL unit type is 0, 2, 4, 6, 8, 10, 12, 14, or 38, or an aggregation packet or fragmentation unit encapsulating only these types, otherwise it MUST be 0. These ranges cover non-reference frames as well as filler data.

The B bit can not be determined reliably from simple inspection of payload headers, and therefore is determined by implementation-specific means. For example, internal codec interfaces may provide information to set this reliably.

```

      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
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| ID=? | L=2 | S|E|I|D|B| TID |0|0| LayerID | TLOPICIDX |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

### 3.3.3. H264-SVC LID Mapping

The following shows H264-SVC [RFC6190] Layer encoding information (3 bits for spatial/dependency layer, 4 bits for quality layer and 3 bits for temporal layer) mapped to the generic LID and TID fields.

The S, E, I and D bits MUST match the correspondingly named bits in PACSI payload structures.

The I bit MUST be 1 when the NAL unit type is 5, 7, 8, 13, or 15, or an aggregation packet or fragmentation unit encapsulating any of these types, otherwise it MUST be 0. These ranges cover intra (IDR) frames as well as critical parameter sets (SPS/PPS variants).

The D bit MUST be 1 when the NAL unit header NRI field is 0, or an aggregation packet or fragmentation unit encapsulating only NAL units with NRI=0, otherwise it MUST be 0. The NRI=0 condition signals non-reference frames.

The B bit can not be determined reliably from simple inspection of payload headers, and therefore is determined by implementation-specific means. For example, internal codec interfaces may provide information to set this reliably.

```

      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
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| ID=? | L=2 | S|E|I|D|B| TID |0| DID | QID | TLOPICIDX |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

### 3.3.4. H264 (AVC) LID Mapping

The following shows the header extension for H264 (AVC) [RFC6184] that contains only temporal layer information.

The S bit MUST be 1 when the timestamp in the RTP header differs from the timestamp in the prior RTP sequence number from the same SSRC, otherwise it MUST be 0.

The E bit MUST match the M bit in the RTP header.

The I bit MUST be 1 when the NAL unit type is 5, 7, or 8, or an aggregation packet or fragmentation unit encapsulating any of these types, otherwise it MUST be 0. These ranges cover intra (IDR) frames as well as critical parameter sets (SPS/PPS).

The D bit MUST be 1 when the NAL unit header NRI field is 0, or an aggregation packet or fragmentation unit encapsulating only NAL units with NRI=0, otherwise it MUST be 0. The NRI=0 condition signals non-reference frames.

The B bit can not be determined reliably from simple inspection of payload headers, and therefore is determined by implementation-specific means. For example, internal codec interfaces may provide information to set this reliably.

```

      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
+-----+-----+-----+-----+-----+-----+-----+-----+
| ID=? | L=2 | S|E|I|D|B| TID | 0|0|0|0|0|0|0|0|0|0| TL0PICIDX |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

### 3.3.5. VP8 LID Mapping

The following shows the header extension for VP8 [RFC7741] that contains only temporal layer information.

The S bit MUST match the correspondingly named bit in the VP8 payload descriptor when PID=0, otherwise it MUST be 0.

The E bit MUST match the M bit in the RTP header.

The I bit MUST match the inverse of the P bit in the VP8 payload header.

The D bit MUST match the N bit in the VP8 payload descriptor.

The B bit MUST match the Y bit in the VP8 payload descriptor. Note: When using temporally nested scalability structures as recommended in Section 3.5.2, the B bit and VP8 Y bit will always be 1 if TID is not 0, since it is always possible to switch up to a higher temporal layer in such nested structures.

TID and TLOPICIDX MUST match the correspondingly named fields in the VP8 payload descriptor.

```

      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
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| ID=? | L=2 | S|E|I|D|B| TID | 0|0|0|0|0|0|0|0|0|0| TLOPICIDX |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

### 3.3.6. Future Codec LID Mapping

The RTP payload format specification for future video codecs SHOULD include a section describing the LID mapping and TID mapping for the codec.

### 3.4. Signaling Information

The URI for declaring this header extension in an extmap attribute is "urn:ietf:params:rtp-hdext:framemarking". It does not contain any extension attributes.

An example attribute line in SDP:

```
a=extmap:3 urn:ietf:params:rtp-hdext:framemarking
```

### 3.5. Usage Considerations

The header extension values MUST represent what is already in the RTP payload.

When an RTP switch needs to discard a received video frame due to congestion control considerations, it is RECOMMENDED that it preferably drop frames marked with the D (Discardable) bit set, or the highest values of TID and LID, which indicate the highest temporal and spatial/quality enhancement layers, since those typically have fewer dependencies on them than lower layers.

When an RTP switch wants to forward a new video stream to a receiver, it is RECOMMENDED to select the new video stream from the first switching point with the I (Independent) bit set in all spatial layers and forward the same. An RTP switch can request a media source to generate a switching point by sending Full Intra Request (RTCP FIR) as defined in [RFC5104], for example.

### 3.5.1. Relation to Layer Refresh Request (LRR)

Receivers can use the Layer Refresh Request (LRR) [I-D.ietf-avtext-lrr] RTCP feedback message to upgrade to a higher layer in scalable encodings. The TID/LID values and formats used in LRR messages MUST correspond to the same values and formats specified in Section 3.1.

Because frame marking can only be used with temporally-nested streams, temporal-layer LRR refreshes are unnecessary for frame-marked streams. Other refreshes can be detected based on the I bit being set for the specific spatial layers.

### 3.5.2. Scalability Structures

The LID and TID information is most useful for fixed scalability structures, such as nested hierarchical temporal layering structures, where each temporal layer only references lower temporal layers or the base temporal layer. The LID and TID information is less useful, or even not useful at all, for complex, irregular scalability structures that do not conform to common, fixed patterns of inter-layer dependencies and referencing structures. Therefore it is RECOMMENDED to use LID and TID information for RTP switch forwarding decisions only in the case of temporally nested scalability structures, and it is NOT RECOMMENDED for other (more complex or irregular) scalability structures.

## 4. Security Considerations

In the Secure Real-Time Transport Protocol (SRTP) [RFC3711], RTP header extensions are authenticated but usually not encrypted. When header extensions are used some of the payload type information are exposed and visible to middle boxes. The encrypted media data is not exposed, so this is not seen as a high risk exposure.

## 5. Acknowledgements

Many thanks to Bernard Aboba, Jonathan Lennox, Stephan Wenger, Dale Worley, and Magnus Westerlund for their inputs.

## 6. IANA Considerations

This document defines a new extension URI to the RTP Compact HeaderExtensions sub-registry of the Real-Time Transport Protocol (RTP) Parameters registry, according to the following data:



Extension URI: urn:ietf:params:rtp-hdrext:frame-marking-info  
Description: Frame marking information for video streams Contact:  
mzanaty@cisco.com Reference: RFC XXXX

Note to RFC Editor: please replace RFC XXXX with the number of this RFC.

## 7. References

### 7.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC8285] Singer, D., Desineni, H., and R. Even, Ed., "A General Mechanism for RTP Header Extensions", RFC 8285, DOI 10.17487/RFC8285, October 2017, <<https://www.rfc-editor.org/info/rfc8285>>.
- [RFC6184] Wang, Y.-K., Even, R., Kristensen, T., and R. Jesup, "RTP Payload Format for H.264 Video", RFC 6184, DOI 10.17487/RFC6184, May 2011, <<https://www.rfc-editor.org/info/rfc6184>>.
- [RFC6190] Wenger, S., Wang, Y.-K., Schierl, T., and A. Eleftheriadis, "RTP Payload Format for Scalable Video Coding", RFC 6190, DOI 10.17487/RFC6190, May 2011, <<https://www.rfc-editor.org/info/rfc6190>>.
- [RFC7741] Westin, P., Lundin, H., Glover, M., Uberti, J., and F. Galligan, "RTP Payload Format for VP8 Video", RFC 7741, DOI 10.17487/RFC7741, March 2016, <<https://www.rfc-editor.org/info/rfc7741>>.
- [RFC7798] Wang, Y.-K., Sanchez, Y., Schierl, T., Wenger, S., and M. Hannuksela, "RTP Payload Format for High Efficiency Video Coding (HEVC)", RFC 7798, DOI 10.17487/RFC7798, March 2016, <<https://www.rfc-editor.org/info/rfc7798>>.

### 7.2. Informative References

- [RFC7656] Lennox, J., Gross, K., Nandakumar, S., Salgueiro, G., and B. Burman, Ed., "A Taxonomy of Semantics and Mechanisms for Real-Time Transport Protocol (RTP) Sources", RFC 7656, DOI 10.17487/RFC7656, November 2015, <<https://www.rfc-editor.org/info/rfc7656>>.

- [RFC7667] Westerlund, M. and S. Wenger, "RTP Topologies", RFC 7667, DOI 10.17487/RFC7667, November 2015, <<https://www.rfc-editor.org/info/rfc7667>>.
- [RFC6464] Lennox, J., Ed., Iovov, E., and E. Marocco, "A Real-time Transport Protocol (RTP) Header Extension for Client-to-Mixer Audio Level Indication", RFC 6464, DOI 10.17487/RFC6464, December 2011, <<https://www.rfc-editor.org/info/rfc6464>>.
- [RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, DOI 10.17487/RFC3550, July 2003, <<https://www.rfc-editor.org/info/rfc3550>>.
- [RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, DOI 10.17487/RFC3711, March 2004, <<https://www.rfc-editor.org/info/rfc3711>>.
- [RFC5104] Wenger, S., Chandra, U., Westerlund, M., and B. Burman, "Codec Control Messages in the RTP Audio-Visual Profile with Feedback (AVPF)", RFC 5104, DOI 10.17487/RFC5104, February 2008, <<https://www.rfc-editor.org/info/rfc5104>>.
- [RFC8871] Jones, P., Benham, D., and C. Groves, "A Solution Framework for Private Media in Privacy-Enhanced RTP Conferencing (PERC)", RFC 8871, DOI 10.17487/RFC8871, January 2021, <<https://www.rfc-editor.org/info/rfc8871>>.
- [I-D.ietf-avtext-lrr]  
Lennox, J., Hong, D., Uberti, J., Holmer, S., and M. Flodman, "The Layer Refresh Request (LRR) RTCP Feedback Message", Work in Progress, Internet-Draft, draft-ietf-avtext-lrr-07, 2 July 2017, <<https://www.ietf.org/archive/id/draft-ietf-avtext-lrr-07.txt>>.
- [I-D.ietf-payload-vp9]  
Uberti, J., Holmer, S., Flodman, M., Hong, D., and J. Lennox, "RTP Payload Format for VP9 Video", Work in Progress, Internet-Draft, draft-ietf-payload-vp9-16, 10 June 2021, <<https://www.ietf.org/internet-drafts/draft-ietf-payload-vp9-16.txt>>.

Authors' Addresses

Mo Zanaty  
Cisco Systems  
170 West Tasman Drive  
San Jose, CA 95134  
United States of America

Email: mzanaty@cisco.com

Espen Berger  
Cisco Systems

Email: espeberg@cisco.com

Suhas Nandakumar  
Cisco Systems  
170 West Tasman Drive  
San Jose, CA 95134  
United States of America

Email: snandaku@cisco.com

Payload Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: December 31, 2017

J. Lennox  
D. Hong  
Vidyo  
J. Uberti  
S. Holmer  
M. Flodman  
Google  
June 29, 2017

The Layer Refresh Request (LRR) RTCP Feedback Message  
draft-ietf-avtext-lrr-07

Abstract

This memo describes the RTCP Payload-Specific Feedback Message "Layer Refresh Request" (LRR), which can be used to request a state refresh of one or more substreams of a layered media stream. It also defines its use with several RTP payloads for scalable media formats.

Status of This Memo

This Internet-Draft is submitted 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 December 31, 2017.

Copyright Notice

Copyright (c) 2017 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 . . . . .	2
2. Conventions, Definitions and Acronyms . . . . .	2
2.1. Terminology . . . . .	3
3. Layer Refresh Request . . . . .	5
3.1. Message Format . . . . .	6
3.2. Semantics . . . . .	7
4. Usage with specific codecs . . . . .	8
4.1. H264 SVC . . . . .	8
4.2. VP8 . . . . .	9
4.3. H265 . . . . .	10
5. Usage with different scalability transmission mechanisms . .	11
6. SDP Definitions . . . . .	11
7. Security Considerations . . . . .	12
8. IANA Considerations . . . . .	12
9. References . . . . .	12
9.1. Normative References . . . . .	12
9.2. Informative References . . . . .	13
Authors' Addresses . . . . .	14

## 1. Introduction

This memo describes an RTCP [RFC3550] Payload-Specific Feedback Message [RFC4585] "Layer Refresh Request" (LRR). It is designed to allow a receiver of a layered media stream to request that one or more of its substreams be refreshed, such that it can then be decoded by an endpoint which previously was not receiving those layers, without requiring that the entire stream be refreshed (as it would be if the receiver sent a Full Intra Request (FIR); [RFC5104] see also [RFC8082]).

The feedback message is applicable both to temporally and spatially scaled streams, and to both single-stream and multi-stream scalability modes.

## 2. Conventions, Definitions and Acronyms

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 [RFC2119].

## 2.1. Terminology

A "Layer Refresh Point" is a point in a scalable stream after which a decoder, which previously had been able to decode only some (possibly none) of the available layers of stream, is able to decode a greater number of the layers.

For spatial (or quality) layers, in normal encoding, a subpicture can depend both on earlier pictures of that spatial layer and also on lower-layer pictures of the current picture. A layer refresh, however, typically requires that a spatial layer picture be encoded in a way that references only the lower-layer subpictures of the current picture, not any earlier pictures of that spatial layer. Additionally, the encoder must promise that no earlier pictures of that spatial layer will be used as reference in the future.

However, even in a layer refresh, layers other than the ones being refreshed may still maintain dependency on earlier content of the stream. This is the difference between a layer refresh and a Full Intra Request [RFC5104]. This minimizes the coding overhead of refresh to only those parts of the stream that actually need to be refreshed at any given time.

An illustration of spatial layer refresh of an enhancement layer is shown below. <-- indicates a coding dependency.

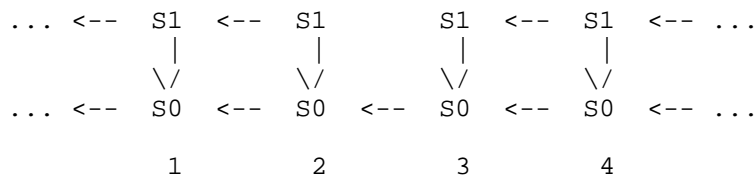


Figure 1

In Figure 1, frame 3 is a layer refresh point for spatial layer S1; a decoder which had previously only been decoding spatial layer S0 would be able to decode layer S1 starting at frame 3.

An illustration of spatial layer refresh of a base layer is shown below. <-- indicates a coding dependency.

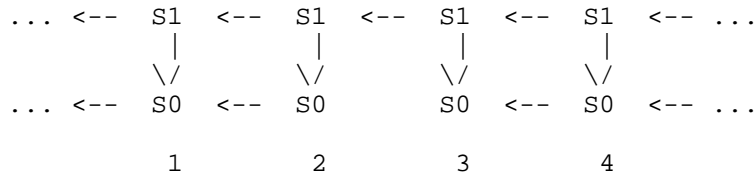


Figure 2

In Figure 2, frame 3 is a layer refresh point for spatial layer S0; a decoder which had previously not been decoding the stream at all could decode layer S0 starting at frame 3.

For temporal layers, while normal encoding allows frames to depend on earlier frames of the same temporal layer, layer refresh requires that the layer be "temporally nested", i.e. use as reference only earlier frames of a lower temporal layer, not any earlier frames of this temporal layer, and also promise that no future frames of this temporal layer will reference frames of this temporal layer before the refresh point. In many cases, the temporal structure of the stream will mean that all frames are temporally nested, in which case decoders will have no need to send LRR messages for the stream.

An illustration of temporal layer refresh is shown below. <-- indicates a coding dependency.

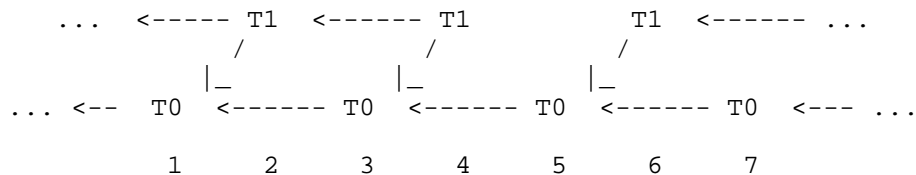


Figure 3

In Figure 3, frame 6 is a layer refresh point for temporal layer T1; a decoder which had previously only been decoding temporal layer T0 would be able to decode layer T1 starting at frame 6.

An illustration of an inherently temporally nested stream is shown below. <-- indicates a coding dependency.

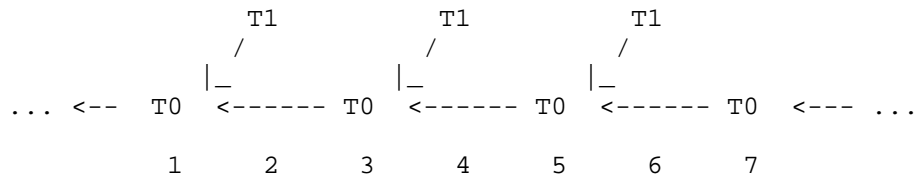


Figure 4

In Figure 4, the stream is temporally nested in its ordinary structure; a decoder receiving layer T0 can begin decoding layer T1 at any point.

A "Layer Index" is a numeric label for a specific spatial and temporal layer of a scalable stream. It consists of the pair of a "temporal ID" identifying the temporal layer, and a "layer ID" identifying the spatial or quality layer. The details of how layers of a scalable stream are labeled are codec-specific. Details for several codecs are defined in Section 4.

### 3. Layer Refresh Request

A layer refresh frame can be requested by sending a Layer Refresh Request (LRR), which is an RTP Control Protocol (RTCP) [RFC3550] payload-specific feedback message [RFC4585] asking the encoder to encode a frame which makes it possible to upgrade to a higher layer. The LRR contains one or two tuples, indicating the temporal and spatial layer the decoder wants to upgrade to, and (optionally) the currently highest temporal and spatial layer the decoder can decode.

The specific format of the tuples, and the mechanism by which a receiver recognizes a refresh frame, is codec-dependent. Usage for several codecs is discussed in Section 4.

LRR follows the model of the Full Intra Request (FIR) [RFC5104] (Section 3.5.1) for its retransmission, reliability, and use in multipoint conferences.

The LRR message is identified by RTCP packet type value PT=PSFB and FMT=TBD. The FCI field MUST contain one or more LRR entries. Each entry applies to a different media sender, identified by its SSRC.

[NOTE TO RFC Editor: Please replace "TBD" with the IANA-assigned payload-specific feedback number.]



### 3.1. Message Format

The Feedback Control Information (FCI) for the Layer Refresh Request consists of one or more FCI entries, the content of which is depicted in Figure 5. The length of the LRR feedback message MUST be set to  $2+3*N$  32-bit words, where N is the number of FCI entries.

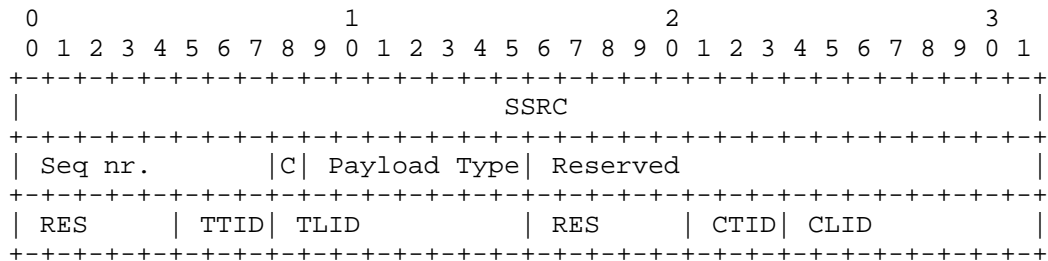


Figure 5

**SSRC (32 bits)** The SSRC value of the media sender that is requested to send a layer refresh point.

**Seq nr. (8 bits)** Command sequence number. The sequence number space is unique for each pairing of the SSRC of command source and the SSRC of the command target. The sequence number SHALL be increased by 1 for each new command (modulo 256, so the value after 255 is 0). A repetition SHALL NOT increase the sequence number. The initial value is arbitrary.

**C (1 bit)** A flag bit indicating whether the "Current Temporal Layer ID (CTID)" and "Current Layer ID (CLID)" fields are present in the FCI. If this bit is 0, the sender of the LRR message is requesting refresh of all layers up to and including the target layer.

**Payload Type (7 bits)** The RTP payload type for which the LRR is being requested. This gives the context in which the target layer index is to be interpreted.

**Reserved (RES) (three separate fields, 16 bits / 5 bits / 5 bits)**  
All bits SHALL be set to 0 by the sender and SHALL be ignored on reception.

**Target Temporal Layer ID (TTID) (3 bits)** The temporal ID of the target layer for which the receiver wishes a refresh point.

Target Layer ID (TLID) (8 bits) The layer ID of the target spatial or quality layer for which the receiver wishes a refresh point. Its format is dependent on the payload type field.

Current Temporal Layer ID (CTID) (3 bits) If C is 1, the ID of the current temporal layer being decoded by the receiver. This message is not requesting refresh of layers at or below this layer. If C is 0, this field SHALL be set to 0 by the sender and SHALL be ignored on reception.

Current Layer ID (CLID) (8 bits) If C is 1, the layer ID of the current spatial or quality layer being decoded by the receiver. This message is not requesting refresh of layers at or below this layer. If C is 0, this field SHALL be set to 0 by the sender and SHALL be ignored on reception.

When C is 1, TTID MUST NOT be less than CTID, and TLID MUST NOT be less than CLID; at least one of TTID or TLID MUST be greater than CTID or CLID respectively. That is to say, the target layer index <TTID, TLID> MUST be a layer upgrade from the current layer index <CTID, CLID>. A sender MAY request an upgrade in both temporal and spatial/quality layers simultaneously.

A receiver receiving an LRR feedback packet which does not satisfy the requirements of the previous paragraph, i.e. one where the C bit is present but TTID is less than CTID or TLID is less than CLID, MUST discard the request.

Note: the syntax of the TTID, TLID, CTID, and CLID fields match, by design, the TID and LID fields in [I-D.ietf-avtext-framemarking].

### 3.2. Semantics

Within the common packet header for feedback messages (as defined in section 6.1 of [RFC4585]), the "SSRC of packet sender" field indicates the source of the request, and the "SSRC of media source" is not used and SHALL be set to 0. The SSRCS of the media senders to which the LRR command applies are in the corresponding FCI entries. A LRR message MAY contain requests to multiple media senders, using one FCI entry per target media sender.

Upon reception of LRR, the encoder MUST send a decoder refresh point (see Section 2.1) as soon as possible.

The sender MUST respect bandwidth limits provided by the application of congestion control, as described in Section 5 of [RFC5104]. As layer refresh points will often be larger than non-refreshing frames,

this may restrict a sender's ability to send a layer refresh point quickly.

LRR MUST NOT be sent as a reaction to picture losses due to packet loss or corruption -- it is RECOMMENDED to use PLI [RFC4585] instead. LRR SHOULD be used only in situations where there is an explicit change in decoders' behavior, for example when a receiver will start decoding a layer which it previously had been discarding.

#### 4. Usage with specific codecs

In order for LRR to be used with a scalable codec, the format of the temporal and layer ID fields (for both the target and current layer indices) needs to be specified for that codec's RTP packetization. New RTP packetization specifications for scalable codecs SHOULD define how this is done. (The VP9 payload [I-D.ietf-payload-vp9], for instance, has done so.) If the payload also specifies how it is used with the Frame Marking RTP Header Extension [I-D.ietf-avtext-framemarking], the syntax MUST be defined in the same manner as the TID and LID fields in that header.

##### 4.1. H264 SVC

H.264 SVC [RFC6190] defines temporal, dependency (spatial), and quality scalability modes.

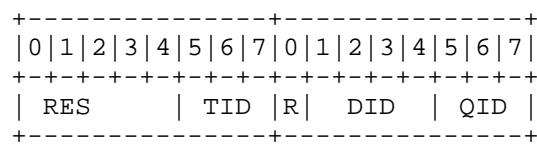


Figure 6

Figure 6 shows the format of the layer index fields for H.264 SVC streams. The "R" and "RES" fields MUST be set to 0 on transmission and ignored on reception. See [RFC6190] Section 1.1.3 for details on the DID, QID, and TID fields.

A dependency or quality layer refresh of a given layer in H.264 SVC can be identified by the "I" bit (idr\_flag) in the extended NAL unit header, present in NAL unit types 14 (prefix NAL unit) and 20 (coded scalable slice). Layer refresh of the base layer can also be identified by its NAL unit type of its coded slices, which is "5" rather than "1". A dependency or quality layer refresh is complete once this bit has been seen on all the appropriate layers (in decoding order) above the current layer index (if any, or beginning from the base layer if not) through the target layer index.

Note that as the "I" bit in a PACSI header is set if the corresponding bit is set in any of the aggregated NAL units it describes; thus, it is not sufficient to identify layer refresh when NAL units of multiple dependency or quality layers are aggregated.

In H.264 SVC, temporal layer refresh information can be determined from various Supplemental Encoding Information (SEI) messages in the bitstream.

Whether an H.264 SVC stream is scalably nested can be determined from the Scalability Information SEI message's temporal\_id\_nesting flag. If this flag is set in a stream's currently applicable Scalability Information SEI, receivers SHOULD NOT send temporal LRR messages for that stream, as every frame is implicitly a temporal layer refresh point. (The Scalability Information SEI message may also be available in the signaling negotiation of H.264 SVC, as the sprop-scalability-info parameter.)

If a stream's temporal\_id\_nesting flag is not set, the Temporal Level Switching Point SEI message identifies temporal layer switching points. A temporal layer refresh is satisfied when this SEI message is present in a frame with the target layer index, if the message's delta\_frame\_num refers to a frame with the requested current layer index. (Alternately, temporal layer refresh can also be satisfied by a complete state refresh, such as an IDR.) Senders which support receiving LRR for non-temporally-nested streams MUST insert Temporal Level Switching Point SEI messages as appropriate.

#### 4.2. VP8

The VP8 RTP payload format [RFC7741] defines temporal scalability modes. It does not support spatial scalability.

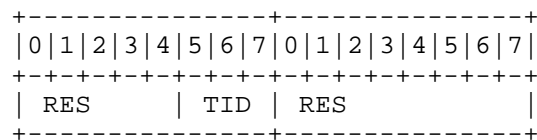


Figure 7

Figure 7 shows the format of the layer index field for VP8 streams. The "RES" fields MUST be set to 0 on transmission and be ignored on reception. See [RFC7741] Section 4.2 for details on the TID field.

A VP8 layer refresh point can be identified by the presence of the "Y" bit in the VP8 payload header. When this bit is set, this and all subsequent frames depend only on the current base temporal layer.

On receipt of an LRR for a VP8 stream, A sender which supports LRR MUST encode the stream so it can set the Y bit in a packet whose temporal layer is at or below the target layer index.

Note that in VP8, not every layer switch point can be identified by the Y bit, since the Y bit implies layer switch of all layers, not just the layer in which it is sent. Thus the use of LRR with VP8 can result in some inefficiency in transmission. However, this is not expected to be a major issue for temporal structures in normal use.

#### 4.3. H265

The initial version of the H.265 payload format [RFC7798] defines temporal scalability, with protocol elements reserved for spatial or other scalability modes (which are expected to be defined in a future version of the specification).

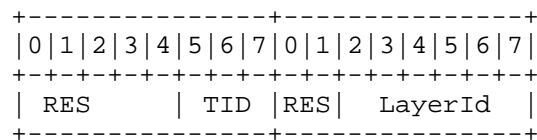


Figure 8

Figure 8 shows the format of the layer index field for H.265 streams. The "RES" fields MUST be set to 0 on transmission and ignored on reception. See [RFC7798] Section 1.1.4 for details on the LayerId and TID fields.

H.265 streams signal whether they are temporally nested, using the `vps_temporal_id_nesting_flag` in the Video Parameter Set (VPS), and the `sps_temporal_id_nesting_flag` in the Sequence Parameter Set (SPS). If this flag is set in a stream's currently applicable VPS or SPS, receivers SHOULD NOT send temporal LRR messages for that stream, as every frame is implicitly a temporal layer refresh point.

If a stream's `sps_temporal_id_nesting_flag` is not set, the NAL unit types 2 to 5 inclusively identify temporal layer switching points. A layer refresh to any higher target temporal layer is satisfied when a NAL unit type of 4 or 5 with TID equal to 1 more than current TID is seen. Alternatively, layer refresh to a target temporal layer can be incrementally satisfied with NAL unit type of 2 or 3. In this case, given current TID = T0 and target TID = TN, layer refresh to TN is satisfied when NAL unit type of 2 or 3 is seen for TID = T1, then TID = T2, all the way up to TID = TN. During this incremental process, layer refresh to TN can be completely satisfied as soon as a NAL unit type of 2 or 3 is seen.

Of course, temporal layer refresh can also be satisfied whenever any Intra Random Access Point (IRAP) NAL unit type (with values 16-23, inclusively) is seen. An IRAP picture is similar to an IDR picture in H.264 (NAL unit type of 5 in H.264) where decoding of the picture can start without any older pictures.

In the (future) H.265 payloads that support spatial scalability, a spatial layer refresh of a specific layer can be identified by NAL units with the requested layer ID and NAL unit types between 16 and 21 inclusive. A dependency or quality layer refresh is complete once NAL units of this type have been seen on all the appropriate layers (in decoding order) above the current layer index (if any, or beginning from the base layer if not) through the target layer index.

## 5. Usage with different scalability transmission mechanisms

Several different mechanisms are defined for how scalable streams can be transmitted in RTP. The RTP Taxonomy [RFC7656] Section 3.7 defines three mechanisms: Single RTP Stream on a Single Media Transport (SRST), Multiple RTP Streams on a Single Media Transport (MRST), and Multiple RTP Streams on Multiple Media Transports (MRMT).

The LRR message is applicable to all these mechanisms. For MRST and MRMT mechanisms, the "media source" field of the LRR FCI is set to the SSRC of the RTP stream containing the layer indicated by the Current Layer Index (if "C" is 1), or the stream containing the base encoded stream (if "C" is 0). For MRMT, it is sent on the RTP session on which this stream is sent. On receipt, the sender MUST refresh all the layers requested in the stream, simultaneously in decode order.

## 6. SDP Definitions

Section 7 of [RFC5104] defines SDP procedures for indicating and negotiating support for codec control messages (CCM) in SDP. This document extends this with a new codec control command, "lrr", which indicates support of the Layer Refresh Request (LRR).

Figure 9 gives a formal Augmented Backus-Naur Form (ABNF) [RFC5234] showing this grammar extension, extending the grammar defined in [RFC5104].

```
rtcp-fb-ccm-param =/ SP "lrr" ; Layer Refresh Request
```

Figure 9: Syntax of the "lrr" ccm

The Offer-Answer considerations defined in [RFC5104] Section 7.2 apply.

## 7. Security Considerations

All the security considerations of FIR feedback packets [RFC5104] apply to LRR feedback packets as well. Additionally, media senders receiving LRR feedback packets MUST validate that the payload types and layer indices they are receiving are valid for the stream they are currently sending, and discard the requests if not.

## 8. IANA Considerations

This document defines a new entry to the "Codec Control Messages" subregistry of the "Session Description Protocol (SDP) Parameters" registry, according to the following data:

Value name: lrr

Long name: Layer Refresh Request Command

Usable with: ccm

Mux: IDENTICAL-PER-PT

Reference: RFC XXXX

This document also defines a new entry to the "FMT Values for PSFB Payload Types" subregistry of the "Real-Time Transport Protocol (RTP) Parameters" registry, according to the following data:

Name: LRR

Long Name: Layer Refresh Request Command

Value: TBD

Reference: RFC XXXX

## 9. References

### 9.1. Normative References

[I-D.ietf-avtext-framemarking]

Berger, E., Nandakumar, S., and M. Zanaty, "Frame Marking RTP Header Extension", draft-ietf-avtext-framemarking-04 (work in progress), March 2017.

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, DOI 10.17487/RFC3550, July 2003, <<http://www.rfc-editor.org/info/rfc3550>>.
- [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, DOI 10.17487/RFC4585, July 2006, <<http://www.rfc-editor.org/info/rfc4585>>.
- [RFC5104] Wenger, S., Chandra, U., Westerlund, M., and B. Burman, "Codec Control Messages in the RTP Audio-Visual Profile with Feedback (AVPF)", RFC 5104, DOI 10.17487/RFC5104, February 2008, <<http://www.rfc-editor.org/info/rfc5104>>.
- [RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <<http://www.rfc-editor.org/info/rfc5234>>.
- [RFC6190] Wenger, S., Wang, Y., Schierl, T., and A. Eleftheriadis, "RTP Payload Format for Scalable Video Coding", RFC 6190, DOI 10.17487/RFC6190, May 2011, <<http://www.rfc-editor.org/info/rfc6190>>.
- [RFC7741] Westin, P., Lundin, H., Glover, M., Uberti, J., and F. Galligan, "RTP Payload Format for VP8 Video", RFC 7741, DOI 10.17487/RFC7741, March 2016, <<http://www.rfc-editor.org/info/rfc7741>>.
- [RFC7798] Wang, Y., Sanchez, Y., Schierl, T., Wenger, S., and M. Hannuksela, "RTP Payload Format for High Efficiency Video Coding (HEVC)", RFC 7798, DOI 10.17487/RFC7798, March 2016, <<http://www.rfc-editor.org/info/rfc7798>>.

## 9.2. Informative References

- [I-D.ietf-payload-vp9] Uberti, J., Holmer, S., Flodman, M., Lennox, J., and D. Hong, "RTP Payload Format for VP9 Video", draft-ietf-payload-vp9-03 (work in progress), March 2017.



- [RFC7656] Lennox, J., Gross, K., Nandakumar, S., Salgueiro, G., and B. Burman, Ed., "A Taxonomy of Semantics and Mechanisms for Real-Time Transport Protocol (RTP) Sources", RFC 7656, DOI 10.17487/RFC7656, November 2015, <<http://www.rfc-editor.org/info/rfc7656>>.
- [RFC8082] Wenger, S., Lennox, J., Burman, B., and M. Westerlund, "Using Codec Control Messages in the RTP Audio-Visual Profile with Feedback with Layered Codecs", RFC 8082, DOI 10.17487/RFC8082, March 2017, <<http://www.rfc-editor.org/info/rfc8082>>.

## Authors' Addresses

Jonathan Lennox  
Vidyo, Inc.  
433 Hackensack Avenue  
Seventh Floor  
Hackensack, NJ 07601  
US

Email: [jonathan@vidyo.com](mailto:jonathan@vidyo.com)

Danny Hong  
Vidyo, Inc.  
433 Hackensack Avenue  
Seventh Floor  
Hackensack, NJ 07601  
US

Email: [danny@vidyo.com](mailto:danny@vidyo.com)

Justin Uberti  
Google, Inc.  
747 6th Street South  
Kirkland, WA 98033  
USA

Email: [justin@uberti.name](mailto:justin@uberti.name)

Stefan Holmer  
Google, Inc.  
Kungsbron 2  
Stockholm 111 22  
Sweden

Email: holmer@google.com

Magnus Flodman  
Google, Inc.  
Kungsbron 2  
Stockholm 111 22  
Sweden

Email: mflodman@google.com

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: April 9, 2017

A. Roach  
Mozilla  
S. Nandakumar  
Cisco Systems  
P. Thatcher  
Google  
October 06, 2016

RTP Stream Identifier Source Description (SDES)  
draft-ietf-avtext-rid-09

Abstract

This document defines and registers two new RTCP Stream Identifier Source Description (SDES) items. One, named RtpStreamId, is used for unique identification of RTP streams. The other, RepairedRtpStreamId, can be used to identify which stream a redundancy RTP stream is to be used to repair.

Status of This Memo

This Internet-Draft is submitted 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 April 9, 2017.

Copyright Notice

Copyright (c) 2016 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 . . . . .	2
2. Terminology . . . . .	3
3. Usage of RtpStreamId and RepairedRtpStreamId in RTP and RTCP	3
3.1. RTCP 'RtpStreamId' SDES Extension . . . . .	5
3.2. RTCP 'RepairedRtpStreamId' SDES Extension . . . . .	5
3.3. RTP 'RtpStreamId' and 'RepairedRtpStreamId' Header Extensions . . . . .	5
4. IANA Considerations . . . . .	6
4.1. New RtpStreamId SDES item . . . . .	6
4.2. New RepairRtpStreamId SDES item . . . . .	6
4.3. New RtpStreamId Header Extension URI . . . . .	7
4.4. New RepairRtpStreamId Header Extension URI . . . . .	7
5. Security Considerations . . . . .	7
6. Acknowledgements . . . . .	8
7. References . . . . .	8
7.1. Normative References . . . . .	8
7.2. Informative References . . . . .	9
Authors' Addresses . . . . .	9

## 1. Introduction

RTP sessions frequently consist of multiple streams, each of which is identified at any given time by its SSRC; however, the SSRC associated with a stream is not guaranteed to be stable over its lifetime. Within a session, these streams can be tagged with a number of identifiers, including CNAMEs and MSIDs [I-D.ietf-mmusic-msid]. Unfortunately, none of these have the proper ordinality to refer to an individual stream; all such identifiers can appear in more than one stream at a time. While approaches that use unique Payload Types (PTs) per stream have been used in some applications, this is a semantic overloading of that field, and one for which its size is inadequate: in moderately complex systems that use PT to uniquely identify every potential combination of codec configuration and unique stream, it is possible to simply run out of values.

To address this situation, we define a new RTCP Stream Identifier Source Description (SDES) identifier, RtpStreamId, that uniquely identifies a single RTP stream. A key motivator for defining this identifier is the ability to differentiate among different encodings of a single Source Stream that are sent simultaneously (i.e., simulcast). This need for unique identification extends to dependent

streams (e.g., where layers used by a layered codec are transmitted on separate streams).

At the same time, when redundancy RTP streams are in use, we also need an identifier that connects such streams to the RTP stream for which they are providing redundancy. For this purpose, we define an additional SDES identifier, `RepairedRtpStreamId`. This identifier can appear only in packets associated with a redundancy RTP stream. They carry the same value as the `RtpStreamId` of the RTP stream that the redundant RTP stream is correcting.

## 2. Terminology

In this document, the terms "source stream", "RTP stream", "source RTP stream", "dependent stream", "received RTP stream", and "redundancy RTP stream" are used as defined in [RFC7656].

The following acronyms are also used:

- o CNAME: Canonical End-Point Identifier, defined in [RFC3550]
- o MID: Media Identification, defined in [I-D.ietf-mmusic-sdp-bundle-negotiation]
- o MSID: Media Stream Identifier, defined in [I-D.ietf-mmusic-msid]
- o RTCP: Real-time Transport Control Protocol, defined in [RFC3550]
- o RTP: Real-time Transport Protocol, defined in [RFC3550]
- o SDES: Source Description, defined in [RFC3550]
- o SSRC: Synchronization Source, defined in [RFC3550]

## 3. Usage of `RtpStreamId` and `RepairedRtpStreamId` in RTP and RTCP

The RTP fixed header includes the payload type number and the SSRC values of the RTP stream. RTP defines how you de-multiplex streams within an RTP session; however, in some use cases, applications need further identifiers in order to effectively map the individual RTP Streams to their equivalent payload configurations in the SDP.

This specification defines two new RTCP SDES items [RFC3550]. The first item is '`RtpStreamId`', which is used to carry RTP stream identifiers within RTCP SDES packets. This makes it possible for a receiver to associate received RTP packets (identifying the RTP stream) with a media description having the format constraint specified. The second is '`RepairedRtpStreamId`', which can be used in

redundancy RTP streams to indicate the RTP stream repaired by a redundancy RTP stream.

To be clear: the value carried in a RepairedRtpStreamId will always match the RtpStreamId value from another RTP stream in the same session. For example, if a source RTP stream is identified by RtpStreamId "A", then any redundancy RTP stream that repairs that source RTP stream will contain a RepairedRtpStreamId of "A" (if this mechanism is being used to perform such correlation). These redundant RTP streams may also contain their own unique RtpStreamId.

This specification also uses the RTP header extension for RTCP SDES items [I-D.ietf-avtext-sdes-hdr-ext] to allow carrying RtpStreamId and RepairedRtpStreamId values in RTP packets. This allows correlation at stream startup, or after stream changes where the use of RTCP may not be sufficiently responsive. This speed of response is necessary since, in many cases, the stream cannot be properly processed until it can be identified.

RtpStreamId and RepairedRtpStreamId values are scoped by source identifier (e.g., CNAME) and by media session. When the media is multiplexed using the BUNDLE extension [I-D.ietf-mmusic-sdp-bundle-negotiation], these values are further scoped by their associated MID values. For example: an RtpStreamId of "1" may be present in the stream identified with a CNAME of "1234@example.com", and may also be present in a stream with a CNAME of "5678@example.org", and these would refer to different streams. Similarly, an RtpStreamId of "1" may be present with an MID of "A", and again with a MID of "B", and also refer to two different streams.

Note that the RepairedRtpStreamId mechanism is limited to indicating one repaired stream per redundancy stream. If systems require correlation for schemes in which a redundancy stream contains information used to repair more than one stream, they will have to use a more complex mechanism than the one defined in this specification.

As with all SDES items, RtpStreamId and RepairedRtpStreamId are limited to a total of 255 octets in length. RtpStreamId and RepairedStreamId are constrained to contain only alphanumeric characters. For avoidance of doubt, the only allowed byte values for these IDs are decimal 48 through 57, 65 through 90, and 97 through 122.

### 3.1. RTCP 'RtpStreamId' SDES Extension

```

 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
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|RtpStreamId=TBD|      length      | RtpStreamId                      ...
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

The RtpStreamId payload is ASCII encoded and is not null-terminated.

RFC EDITOR NOTE: Please replace TBD with the assigned SDES identifier value.

### 3.2. RTCP 'RepairedRtpStreamId' SDES Extension

```

 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
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|Repaired...=TBD|      length      | RepairRtpStreamId                ...
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

The RepairedRtpStreamId payload is ASCII encoded and is not null-terminated.

RFC EDITOR NOTE: Please replace TBD with the assigned SDES identifier value.

### 3.3. RTP 'RtpStreamId' and 'RepairedRtpStreamId' Header Extensions

Because recipients of RTP packets will typically need to know which streams they correspond to immediately upon receipt, this specification also defines a means of carrying RtpStreamId and RepairedRtpStreamId identifiers in RTP extension headers, using the technique described in [I-D.ietf-avtext-sdes-hdr-ext].

As described in that document, the header extension element can be encoded using either the one-byte or two-byte header, and the identification-tag payload is ASCII-encoded.

As the identifier is included in an RTP header extension, there should be some consideration given to the packet expansion caused by the identifier. To avoid Maximum Transmission Unit (MTU) issues for the RTP packets, the header extension's size needs to be taken into account when encoding media. Note that the set of header extensions included in the packet needs to be padded to the next 32-bit boundary [RFC5285].

In many cases, a one-byte identifier will be sufficient to distinguish streams in a session; implementations are strongly encouraged to use the shortest identifier that fits their purposes. Implementors are warned, in particular, not to include any information in the identifier that is derived from potentially user-identifying information, such as user ID or IP address. To avoid identification of specific implementations based on their pattern of tag generation, implementations are encouraged to use a simple scheme that starts with the ASCII digit "1", and increments by one for each subsequent identifier.

#### 4. IANA Considerations

##### 4.1. New RtpStreamId SDES item

RFC EDITOR NOTE: Please replace RFCXXXX with the RFC number of this document.

RFC EDITOR NOTE: Please replace TBD with the assigned SDES identifier value.

This document adds the RtpStreamId SDES item to the IANA "RTP SDES item types" registry as follows:

Value:	TBD
Abbrev.:	RtpStreamId
Name:	RTP Stream Identifier
Reference:	RFCXXXX

##### 4.2. New RepairRtpStreamId SDES item

RFC EDITOR NOTE: Please replace RFCXXXX with the RFC number of this document.

RFC EDITOR NOTE: Please replace TBD with the assigned SDES identifier value.

This document adds the RepairedRtpStreamId SDES item to the IANA "RTP SDES item types" registry as follows:

Value:	TBD
Abbrev.:	RepairedRtpStreamId
Name:	Repaired RTP Stream Identifier
Reference:	RFCXXXX



#### 4.3. New RtpStreamId Header Extension URI

RFC EDITOR NOTE: Please replace RFCXXXX with the RFC number of this document.

This document defines a new extension URI in the RTP SDES Compact Header Extensions sub-registry of the RTP Compact Header Extensions registry sub-registry, as follows

Extension URI: urn:ietf:params:rtp-hdext:sdes:rtp-stream-id  
Description: RTP Stream Identifier Contact: adam@nostrum.com  
Reference: RFCXXXX

#### 4.4. New RepairRtpStreamId Header Extension URI

RFC EDITOR NOTE: Please replace RFCXXXX with the RFC number of this document.

This document defines a new extension URI in the RTP SDES Compact Header Extensions sub-registry of the RTP Compact Header Extensions registry sub-registry, as follows

Extension URI: urn:ietf:params:rtp-hdext:sdes:repaired-rtp-stream-id  
Description: RTP Repaired Stream Identifier Contact: adam@nostrum.com  
Reference: RFCXXXX

#### 5. Security Considerations

Although the identifiers defined in this document are limited to be strictly alphanumeric, SDES items have the potential to carry any string. As a consequence, there exists a risk that it might carry privacy-sensitive information. Implementations need to take care when generating identifiers so that they do not contain information that can identify the user or allow for long term tracking of the device. Following the generation recommendations in Section 3.3 will result in non-instance-specific labels, with only minor fingerprinting possibilities in the total number of used RtpStreamIds and RepairedRtpStreamIds.

Even if the SDES items are generated to convey as little information as possible, implementors are strongly encouraged to encrypt SDES items - both in RTCP and RTP header extensions - so as to preserve privacy against third parties.

As the SDES items are used for identification of the RTP streams for different application purposes, it is important that the intended values are received. An attacker, either a third party or malicious RTP middlebox, that removes, or changes the values for these SDES

items, can severely impact the application. The impact can include failure to decode or display the media content of the RTP stream. It can also result in incorrectly attributing media content to identifiers of the media source, such as incorrectly identifying the speaker. To prevent this from occurring due to third party attacks, integrity and source authentication is needed.

Options for Securing RTP Sessions [RFC7201] discusses options for how encryption, integrity and source authentication can be accomplished.

## 6. Acknowledgements

Many thanks for review and input from Cullen Jennings, Magnus Westerlund, Colin Perkins, Jonathan Lennox, and Paul Kyzivat. Magnus Westerlund provided substantially all of the Security Considerations section.

## 7. References

### 7.1. Normative References

- [I-D.ietf-avtext-sdes-hdr-ext]  
Westerlund, M., Burman, B., Even, R., and M. Zanaty, "RTP Header Extension for RTCP Source Description Items", draft-ietf-avtext-sdes-hdr-ext-07 (work in progress), June 2016.
- [I-D.ietf-mmusic-sdp-bundle-negotiation]  
Holmberg, C., Alvestrand, H., and C. Jennings, "Negotiating Media Multiplexing Using the Session Description Protocol (SDP)", draft-ietf-mmusic-sdp-bundle-negotiation-32 (work in progress), August 2016.
- [RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, DOI 10.17487/RFC3550, July 2003, <<http://www.rfc-editor.org/info/rfc3550>>.
- [RFC5285] Singer, D. and H. Desineni, "A General Mechanism for RTP Header Extensions", RFC 5285, DOI 10.17487/RFC5285, July 2008, <<http://www.rfc-editor.org/info/rfc5285>>.
- [RFC7656] Lennox, J., Gross, K., Nandakumar, S., Salgueiro, G., and B. Burman, Ed., "A Taxonomy of Semantics and Mechanisms for Real-Time Transport Protocol (RTP) Sources", RFC 7656, DOI 10.17487/RFC7656, November 2015, <<http://www.rfc-editor.org/info/rfc7656>>.

## 7.2. Informative References

[I-D.ietf-mmusic-msid]

Alvestrand, H., "WebRTC MediaStream Identification in the Session Description Protocol", draft-ietf-mmusic-msid-15 (work in progress), July 2016.

[RFC7201] Westerlund, M. and C. Perkins, "Options for Securing RTP Sessions", RFC 7201, DOI 10.17487/RFC7201, April 2014, <<http://www.rfc-editor.org/info/rfc7201>>.

## Authors' Addresses

Adam Roach  
Mozilla

Email: [adam@nostrum.com](mailto:adam@nostrum.com)

Suhas Nandakumar  
Cisco Systems

Email: [snandaku@cisco.com](mailto:snandaku@cisco.com)

Peter Thatcher  
Google

Email: [pthatcher@google.com](mailto:pthatcher@google.com)

Payload Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: April 11, 2020

S. Lugan  
G. Rouvroy  
A. Descampe  
intoPIX  
T. Richter  
IIS  
A. Willeme  
UCL/ICTEAM  
October 9, 2019

RTP Payload Format for ISO/IEC 21122 (JPEG XS)  
draft-ietf-payload-rtp-jpegxs-02

## Abstract

This document specifies a Real-Time Transport Protocol (RTP) payload format to be used for transporting JPEG XS (ISO/IEC 21122) encoded video. JPEG XS is a low-latency, lightweight image coding system. Compared to an uncompressed video use case, it allows higher resolutions and frame rates, while offering visually lossless quality, reduced power consumption, and end-to-end latency confined to a fraction of a frame.

## Status of This Memo

This Internet-Draft is submitted 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 <https://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 April 11, 2020.

## Copyright Notice

Copyright (c) 2019 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

(<https://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 . . . . .	2
2. Conventions, Definitions, and Abbreviations . . . . .	3
3. Media Format Description . . . . .	4
3.1. Image Data Structures . . . . .	4
3.2. Codestream . . . . .	5
3.3. Video support box and colour specification box . . . . .	5
4. Payload Format . . . . .	5
4.1. Payload Header . . . . .	6
4.2. Payload Data . . . . .	8
4.3. Traffic Shaping and Delivery Timing . . . . .	10
5. Congestion Control Considerations . . . . .	10
6. Payload Format Parameters . . . . .	10
6.1. Media Type Definition . . . . .	10
6.2. Mapping to SDP . . . . .	13
6.2.1. General . . . . .	13
6.2.2. Media type and subtype . . . . .	14
6.2.3. Traffic shaping . . . . .	14
6.2.4. Offer/Answer Considerations . . . . .	14
7. IANA Considerations . . . . .	15
8. Security Considerations . . . . .	15
9. RFC Editor Considerations . . . . .	16
10. References . . . . .	16
10.1. Normative References . . . . .	16
10.2. Informative References . . . . .	18
10.3. URIs . . . . .	18
Authors' Addresses . . . . .	18

## 1. Introduction

This document specifies a payload format for packetization of JPEG XS encoded video signals into the Real-time Transport Protocol (RTP) [RFC3550].

The JPEG XS coding system offers compression and recompression of image sequences with very moderate computational resources while remaining robust under multiple compression and decompression cycles and mixing of content sources, e.g. embedding of subtitles, overlays or logos. Typical target compression ratios ensuring visually

lossless quality are in the range of 2:1 to 10:1, depending on the nature of the source material. The end-to-end latency can be confined to a fraction of a frame, typically between a small number of lines down to below a single line.

## 2. Conventions, Definitions, and Abbreviations

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].

### Application Data Unit (ADU)

The unit of source data provided as payload to the transport layer, and corresponding, in this RTP payload definition, to a single JPEG XS frame.

### Colour specification box

A ISO colour specification box defined in ISO/IEC 21122-3 [ISO21122-3] that includes colour-related metadata required to correctly display JPEG XS frames, such as colour primaries, transfer characteristics and matrix coefficients.

### JPEG XS codestream

A sequence of bytes representing a compressed image formatted according to JPEG XS Part 1 [ISO21122-1], except the End-Of-Codestream (EOC) marker which is omitted in this payload format.

### JPEG XS codestream header

A sequence of bytes at the beginning of each JPEG XS codestream encoded in multiple markers and marker segments that does not carry entropy coded data, but metadata such as the frame dimension and component precision.

### JPEG XS frame

The concatenation of a video support box, as defined in JPEG XS Part 3 [ISO21122-3], a colour specification box, as defined as well in JPEG XS Part 3 [ISO21122-3] and a JPEG XS codestream.

### JPEG XS header segment

The concatenation of a video support box, as defined in JPEG XS Part 3 [ISO21122-3], a colour specification box, as defined as well in JPEG XS Part 3 [ISO21122-3] and a JPEG XS codestream header.

### JPEG XS stream

A sequence of JPEG XS frames

### Marker

A two-byte functional sequence that is part of a JPEG XS codestream starting with a 0xff byte and a subsequent byte defining its function.

Marker segment

A marker along with a 16-bit marker size and payload data following the size.

Slice

The smallest independently decodable unit of a JPEG XS codestream, bearing in mind that it decodes to wavelet coefficients which still require inverse wavelet filtering to give an image.

SOC marker

A marker that consists of the two bytes 0xff10 indicating the start of a JPEG XS codestream.

Video support box

A ISO video support box defined in ISO/IEC 21122-3 [ISO21122-3] that includes metadata required to play back a JPEG XS stream, such as its maximum bitrate, its subsampling structure, its buffer model and its frame rate.

### 3. Media Format Description

#### 3.1. Image Data Structures

JPEG XS is a low-latency lightweight image coding system for coding continuous-tone grayscale or continuous-tone colour digital images.

This coding system provides an efficient representation of image signals through the mathematical tool of wavelet analysis. The wavelet filter process separates each component into multiple bands, where each band consists of multiple coefficients describing the image signal of a given component within a frequency domain specific to the wavelet filter type, i.e. the particular filter corresponding to the band.

Wavelet coefficients are grouped into precincts, where each precinct includes all coefficients over all bands that contribute to a spatial region of the image.

One or multiple precincts are furthermore combined into slices consisting of an integral number of precincts. Precincts do not cross slice boundaries, and wavelet coefficients in precincts that are part of different slices can be decoded independently from each other. Note, however, that the wavelet transformation runs across

slice boundaries. A slice always extends over the full width of the image, but may only cover parts of its height.

Each JPEG XS frame consists of a JPEG XS header segment followed by one or multiple slices completely describing a single frame.

### 3.2. Codestream

The overall codestream format, including the definition of all markers, is further defined in ISO/IEC 21122-1 [ISO21122-1]. It represents sample values of a single frame, bare any interpretation relative to a colour space.

### 3.3. Video support box and colour specification box

While the information defined in the codestream is sufficient to reconstruct the sample values of one video frame, the interpretation of the samples remains undefined by the codestream itself. This interpretation is given by the video support box and the colour specification box which contain significant information to correctly play the JPEG XS stream. The layout and syntax of these boxes, together with their content, are defined in ISO/IEC 21122-3 [ISO21122-3]. The video support box provides information on the maximum bitrate, the frame rate, the subsampling image format, the timecode of the current JPEG XS frame, the profile, level and sublevel used (as defined in ISO/IEC 21122-2 [ISO21122-2]), and optionally on the buffer model and the mastering display metadata. The colour specification box indicates the colour primaries, transfer characteristics, matrix coefficients and video full range flag needed to specify the colour space of the video stream.

## 4. Payload Format

This section specifies the payload format for JPEG XS streams over the Real-time Transport Protocol (RTP) [RFC3550].

In order to be transported over RTP, each JPEG XS stream is transported in a distinct RTP stream, identified by a distinct SSRC.

A JPEG XS stream is divided into Application Data Units (ADUs), each ADU corresponding to a single JPEG XS frame.

An ADU is split into multiple RTP packet payloads. Figure 1 shows an example of how a JPEG XS frame fits into the payload of RTP packets ("Hdr" denotes a RTP packet header). As seen there, each packet contains either part of the JPEG XS header segment or part of a single slice. Both may extend over multiple packets. The payload of every packet shall have the same size (based e.g. on the Maximum



Transfer Unit of the network), except (possibly) the last packet of the JPEG XS header segment or a slice. The boundaries of the JPEG XS header segment and of every slice shall coincide with the boundaries of the payload of a packet, i.e. the first (resp. last) byte of the JPEG XS header segment or a slice shall be the first (resp. last) byte of the payload.

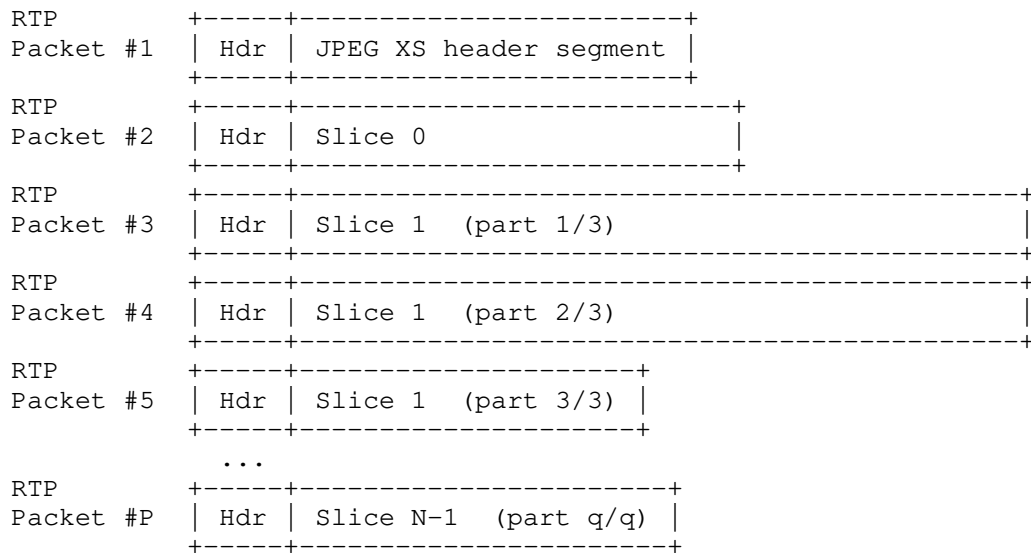


Figure 1: Example of ADU defining a single JPEG XS frame

#### 4.1. Payload Header

Figure 2 illustrates the RTP payload header used in order to transport a JPEG XS stream.

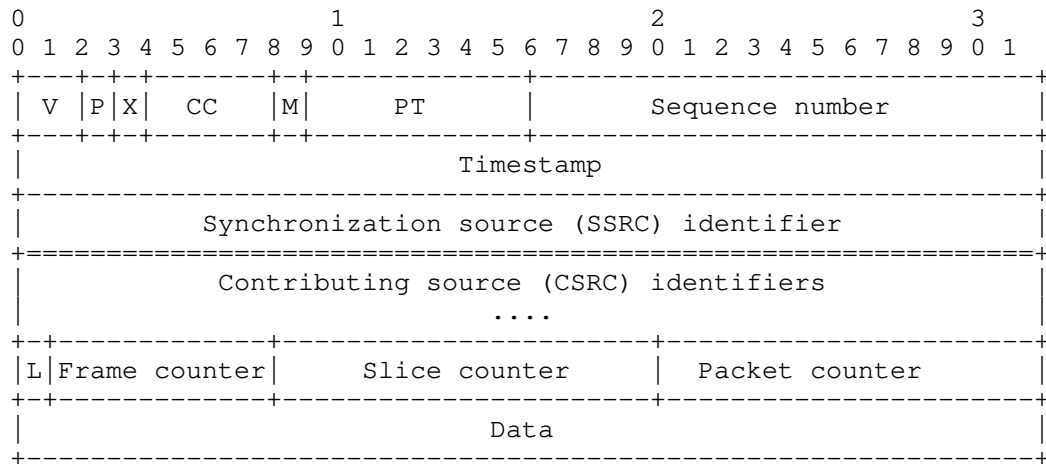


Figure 2: RTP and payload headers

The version (V), padding (P), extension (X), CSRC count (CC), sequence number, synchronization source (SSRC) and contributing source (CSRC) fields follow their respective definitions in RFC 3550 [RFC3550].

The timestamp SHOULD be based on a 90 kHz clock reference.

As per specified in RFC 3550 [RFC3550] and RFC 4175 [RFC4175], the RTP timestamp designates the sampling instant of the first octet of the frame to which the RTP packet belongs. Packets shall not include data from multiple frames, and all packets belonging to the same frame shall have the same timestamp. Several successive RTP packets will consequently have equal timestamps if they belong to the same frame (that is until the marker bit is set to 1, marking the last packet of the frame), and the timestamp is only increased when a new frame begins.

If the sampling instant does not correspond to an integer value of the clock, the value shall be truncated to the next lowest integer, with no ambiguity.

The remaining fields are defined as follows:

Marker (M) [1 bit]:

The M bit is used to indicate the last packet of a frame. This enables a decoder to finish decoding the frame, where it otherwise may need to wait for the next packet to explicitly know that the frame is finished.

**Payload Type (PT) [7 bits]:**

A dynamically allocated payload type field that designates the payload as JPEG XS video.

**Last (L) [1 bit]:**

The L bit is set to indicate the last packet of the JPEG XS header segment or a slice. It enables the decoder to already start decoding a slice without having to wait for the full frame to finish, and thus allows low-latency decoding. As the end of the frame also ends the packet containing the last slice of the frame, the L bit is set whenever the M bit is set.

**Frame counter [7 bits]:**

This field identifies the frame number modulo 128 to which a packet belongs. Frame numbers increment by 1 for each frame transmitted. The frame number, in addition to the time stamp, may help the decoder to manage its input buffer and to bring packets back into their natural order.

**Slice counter [12 bits]:**

This field identifies the slice modulo 4096 to which the packet contributes. If the data belongs to the JPEG XS header segment, this field shall have its maximal value, namely 4095 = 0x0fff. Otherwise, it is the slice index modulo 4096. Slice indices count from 0 at the top of the frame to their maximum number.

**Packet counter [12 bits]:**

This field identifies the packet number modulo 4096 within the JPEG XS header segment or a slice. The packet counter is set to 0 at the start of the JPEG XS header segment and incremented by 1 for every subsequent packet (if any) of this JPEG XS header segment. The packet counter is then reset to 0 at the start of every slice, and incremented by 1 for every packet that contributes to the same slice.

#### 4.2. Payload Data

The payload data of a JPEG XS RTP stream consists of a concatenation of multiple JPEG XS frames.

Each JPEG XS frame is the concatenation of a JPEG XS header segment followed by one or several slices completely defining a single frame. Figure 3 depicts this layout.

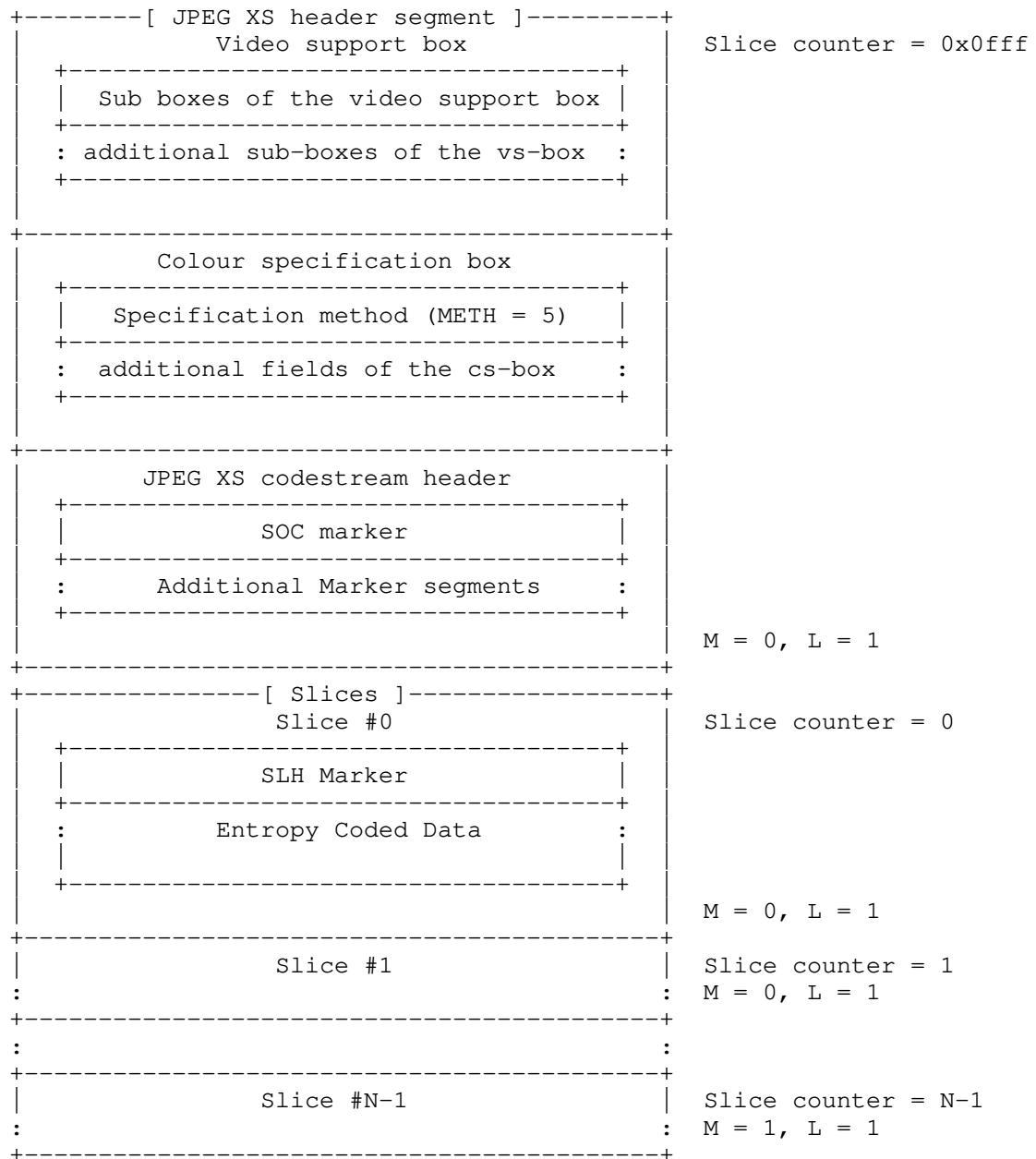


Figure 3: JPEG XS Payload Data

#### 4.3. Traffic Shaping and Delivery Timing

The traffic shaping and delivery timing shall be in accordance with the Network Compatibility Model compliance definitions specified in SMPTE ST 2110-21 [SMPTE-ST2110-21] for either Narrow Linear Senders (Type NL) or Wide Senders (Type W). The session description shall include a format-specific parameter of either TP=2110TPNL or TP=2110TPW to indicate compliance with Type NL or Type W respectively.

NOTE: The Virtual Receiver Buffer Model compliance definitions of ST 2110-21 do not apply.

#### 5. Congestion Control Considerations

Congestion control for RTP SHALL be used in accordance with RFC 3550 [RFC3550], and with any applicable RTP profile: e.g., RFC 3551 [RFC3551]. An additional requirement if best-effort service is being used is users of this payload format MUST monitor packet loss to ensure that the packet loss rate is within acceptable parameters. Circuit Breakers [RFC8083] is an update to RTP [RFC3550] that defines criteria for when one is required to stop sending RTP Packet Streams and applications implementing this standard MUST comply with it. RFC 8085 [RFC8085] provides additional information on the best practices for applying congestion control to UDP streams.

#### 6. Payload Format Parameters

##### 6.1. Media Type Definition

Type name: video

Subtype name: jxsv

Required parameters:

rate: The RTP timestamp clock rate. Applications using this payload format SHOULD use a value of 90000.

Optional parameters:

profile: The JPEG XS profile in use, as defined in ISO/IEC 21122-2 (JPEG XS Part 2) [ISO21122-2].

level: The JPEG XS level in use, as defined in ISO/IEC 21122-2 (JPEG XS Part 2) [ISO21122-2].

sublevel: The JPEG XS sublevel in use, as defined in ISO/IEC 21122-2 (JPEG XS Part 2) [ISO21122-2].

sampling: Signals the colour difference signal sub-sampling structure.

Signals utilizing the non-constant luminance Y'C'B C'R signal format of Recommendation ITU-R BT.601-7, Recommendation ITU-R BT.709-6, Recommendation ITU-R BT.2020-2, or Recommendation ITU-R BT.2100 shall use the appropriate one of the following values for the Media Type Parameter "sampling":

YCbCr-4:4:4 (4:4:4 sampling)  
YCbCr-4:2:2 (4:2:2 sampling)  
YCbCr-4:2:0 (4:2:0 sampling)

Signals utilizing the Constant Luminance Y'C C'BC C'RC signal format of Recommendation ITU-R BT.2020-2 shall use the appropriate one of the following values for the Media Type Parameter "sampling":

CLYCbCr-4:4:4 (4:4:4 sampling)  
CLYCbCr-4:2:2 (4:2:2 sampling)  
CLYCbCr-4:2:0 (4:2:0 sampling)

Signals utilizing the constant intensity I CT CP signal format of Recommendation ITU-R BT.2100 shall use the appropriate one of the following values for the Media Type Parameter "sampling":

ICtCp-4:4:4 (4:4:4 sampling)  
ICtCp-4:2:2 (4:2:2 sampling)  
ICtCp-4:2:0 (4:2:0 sampling)

Signals utilizing the 4:4:4 R' G' B' or RGB signal format (such as that of Recommendation ITU-R BT.601, Recommendation ITU-R BT.709, Recommendation ITU-R BT.2020, Recommendation ITU-R BT.2100, SMPTE ST 2065-1 or ST 2065-3) shall use the following value for the Media Type Parameter sampling.

RGB      RGB or R' G' B' samples

Signals utilizing the 4:4:4 X' Y' Z' signal format (such as defined in SMPTE ST 428-1) shall use the following value for the Media Type Parameter sampling.

XYZ      X' Y' Z' samples

Key signals as defined in SMPTE RP 157 shall use the value key for the Media Type Parameter sampling. The Key signal is represented as a single component.

KEY        samples of the key signal

depth: Determines the number of bits per sample. This is an integer with typical values including 8, 10, 12, and 16.

width: Determines the number of pixels per line. This is an integer between 1 and 32767.

height: Determines the number of lines per frame. This is an integer between 1 and 32767.

exactframerate: Signals the frame rate in frames per second. Integer frame rates shall be signaled as a single decimal number (e.g. "25") whilst non-integer frame rates shall be signaled as a ratio of two integer decimal numbers separated by a "forward-slash" character (e.g. "30000/1001"), utilizing the numerically smallest numerator value possible.

colorimetry: Specifies the system colorimetry used by the image samples. Valid values and their specification are:

BT601-5	ITU Recommendation BT.601-5
BT709-2	ITU Recommendation BT.709-2
SMPTE240M	SMPTE standard 240M
BT601	as specified in Recommendation ITU-R BT.601-7
BT709	as specified in Recommendation ITU-R BT.709-6
BT2020	as specified in Recommendation ITU-R BT.2020-2
BT2100	as specified in Recommendation ITU-R BT.2100
	Table 2 titled "System colorimetry"
ST2065-1	as specified in SMPTE ST 2065-1 Academy Color
	Encoding Specification (ACES)
ST2065-3	as specified for Academy Density Exchange
	Encoding (ADX) in SMPTE ST 2065-3
XYZ	as specified in ISO 11664-1 section titled
	"1931 Observer"

Signals utilizing the Recommendation ITU-R BT.2100 colorimetry should also signal the representational range using the optional parameter RANGE defined below.

interlace: If this OPTIONAL parameter name is present, it indicates that the video is interlaced. If this parameter name is not present, the progressive video format shall be assumed.

TCS: Transfer Characteristic System. This parameter specifies the transfer characteristic system of the image samples. Valid values and their specification are:

- |     |  |
|-----|--|
| SDR | (Standard Dynamic Range) Video streams of standard dynamic range, that utilize the OETF of Recommendation ITU-R BT.709 or Recommendation ITU-R BT.2020. Such streams shall be assumed to target the EOTF specified in ITU-R BT.1886. |
| PQ  | Video streams of high dynamic range video that utilize the Perceptual Quantization system of Recommendation ITU-R BT.2100  |
| HLG | Video streams of high dynamic range video that utilize the Hybrid Log-Gamma system of Recommendation ITU-R BT.2100   |

RANGE: This parameter should be used to signal the encoding range of the sample values within the stream. When paired with ITU Rec BT.2100 colorimetry, this parameter has two allowed values NARROW and FULL, corresponding to the ranges specified in table 9 of ITU Rec BT.2100. In any other context, this parameter has three allowed values: NARROW, FULLPROTECT, and FULL, which correspond to the ranges specified in SMPTE RP 2077. In the absence of this parameter, NARROW shall be the assumed value in either case.

Encoding considerations:

This media type is framed and binary; see Section 4.8 in RFC 6838 [RFC6838].

Security considerations:

Please see the Security Considerations section in RFC XXXX

## 6.2. Mapping to SDP

### 6.2.1. General

A Session Description Protocol (SDP) object shall be created for each RTP stream and it shall be in accordance with the provisions of SMPTE ST 2110-10 [SMPTE-ST2110-10].

The information carried in the media type specification has a specific mapping to fields in the Session Description Protocol (SDP), which is commonly used to describe RTP sessions.



### 6.2.2. Media type and subtype

The media type ("video") goes in SDP "m=" as the media name.

The media subtype ("jxsv") goes in SDP "a=rtpmap" as the encoding name, followed by a slash ("/") and the required parameter "rate" corresponding to the RTP timestamp clock rate (which for the payload format defined in this document MUST be 90000). The optional parameters go in the SDP "a=fmtp" attribute by copying them directly from the MIME media type string as a semicolon-separated list of parameter=value pairs.

A sample SDP mapping for JPEG XS video is as follows:

```
m=video 30000 RTP/AVP 112
a=rtpmap:112 jxsv/90000
a=fmtp:112 sampling=YCbCr-4:2:2; width=1920; height=1080;
          depth=10; colorimetry=BT709; TCS=SDR;
          RANGE=FULL; TP=2110TPNL
```

In this example, a JPEG XS RTP stream is being sent to UDP destination port 30000, with an RTP dynamic payload type of 112 and a media clock rate of 90000 Hz. Note that the "a=fmtp:" line has been wrapped to fit this page, and will be a single long line in the SDP file.

### 6.2.3. Traffic shaping

The SDP object shall include the TP parameter (either 2110TPNL or 2110TPW as specified in Section 4.3) and may include the CMAX parameter as specified in SMPTE ST 2110-21 [SMPTE-ST2110-21].

### 6.2.4. Offer/Answer Considerations

The following considerations apply when using SDP offer/answer procedures [RFC3264] to negotiate the use of the JPEG XS payload in RTP:

- o The "encode" parameter can be used for sendrecv, sendonly, and recvonly streams. Each encode type MUST use a separate payload type number.
- o Any unknown parameter in an offer MUST be ignored by the receiver and MUST NOT be included in the answer.

## 7. IANA Considerations

This memo requests that IANA registers video/jxsv as specified in Section 6.1. The media type is also requested to be added to the IANA registry for "RTP Payload Format MIME types" [1].

## 8. Security Considerations

RTP packets using the payload format defined in this specification are subject to the security considerations discussed in the RTP specification [RFC3550] and in any applicable RTP profile such as RTP/AVP [RFC3551], RTP/AVPF [RFC4585], RTP/SAVP [RFC3711], or RTP/SAVPF [RFC5124]. This implies that confidentiality of the media streams is achieved by encryption.

However, as "Securing the RTP Framework: Why RTP Does Not Mandate a Single Media Security Solution" [RFC7202] discusses, it is not an RTP payload format's responsibility to discuss or mandate what solutions are used to meet the basic security goals like confidentiality, integrity, and source authenticity for RTP in general. This responsibility lies on anyone using RTP in an application. They can find guidance on available security mechanisms and important considerations in "Options for Securing RTP Sessions" [RFC7201]. Applications SHOULD use one or more appropriate strong security mechanisms.

This payload format and the JPEG XS encoding do not exhibit any substantial non-uniformity, either in output or in complexity to perform the decoding operation and thus are unlikely to pose a denial-of-service threat due to the receipt of pathological datagrams.

It is important to note that HD or UHDTV JPEG XS-encoded video can have significant bandwidth requirements (typically more than 1 Gbps for ultra high-definition video, especially if using high framerate). This is sufficient to cause potential for denial-of-service if transmitted onto most currently available Internet paths.

Accordingly, if best-effort service is being used, users of this payload format MUST monitor packet loss to ensure that the packet loss rate is within acceptable parameters. Packet loss is considered acceptable if a TCP flow across the same network path, and experiencing the same network conditions, would achieve an average throughput, measured on a reasonable timescale, that is not less than the RTP flow is achieving. This condition can be satisfied by implementing congestion control mechanisms to adapt the transmission rate (or the number of layers subscribed for a layered multicast

session), or by arranging for a receiver to leave the session if the loss rate is unacceptably high.

This payload format may also be used in networks that provide quality-of-service guarantees. If enhanced service is being used, receivers SHOULD monitor packet loss to ensure that the service that was requested is actually being delivered. If it is not, then they SHOULD assume that they are receiving best-effort service and behave accordingly.

## 9. RFC Editor Considerations

Note to RFC Editor: This section may be removed after carrying out all the instructions of this section.

RFC XXXX is to be replaced by the RFC number this specification receives when published.

## 10. References

### 10.1. Normative References

[ISO21122-1]

International Organization for Standardization (ISO) -  
International Electrotechnical Commission (IEC),  
"Information technology - JPEG XS low-latency lightweight  
image coding system - Part 1: Core coding system", ISO/  
IEC PRF 21122-1, under development,  
<<https://www.iso.org/standard/74535.html>>.

[ISO21122-2]

International Organization for Standardization (ISO) -  
International Electrotechnical Commission (IEC),  
"Information technology - JPEG XS low-latency lightweight  
image coding system - Part 2: Profiles and buffer models",  
ISO/IEC PRF 21122-2, under development,  
<<https://www.iso.org/standard/74536.html>>.

[ISO21122-3]

International Organization for Standardization (ISO) -  
International Electrotechnical Commission (IEC),  
"Information technology - JPEG XS low-latency lightweight  
image coding system - Part 3: Transport and container  
formats", ISO/IEC FDIS 21122-3, under development,  
<<https://www.iso.org/standard/74537.html>>.

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, DOI 10.17487/RFC3264, June 2002, <<https://www.rfc-editor.org/info/rfc3264>>.
- [RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, DOI 10.17487/RFC3550, July 2003, <<https://www.rfc-editor.org/info/rfc3550>>.
- [RFC3551] Schulzrinne, H. and S. Casner, "RTP Profile for Audio and Video Conferences with Minimal Control", STD 65, RFC 3551, DOI 10.17487/RFC3551, July 2003, <<https://www.rfc-editor.org/info/rfc3551>>.
- [RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, DOI 10.17487/RFC3711, March 2004, <<https://www.rfc-editor.org/info/rfc3711>>.
- [RFC6838] Freed, N., Klensin, J., and T. Hansen, "Media Type Specifications and Registration Procedures", BCP 13, RFC 6838, DOI 10.17487/RFC6838, January 2013, <<https://www.rfc-editor.org/info/rfc6838>>.
- [RFC8083] Perkins, C. and V. Singh, "Multimedia Congestion Control: Circuit Breakers for Unicast RTP Sessions", RFC 8083, DOI 10.17487/RFC8083, March 2017, <<https://www.rfc-editor.org/info/rfc8083>>.
- [RFC8085] Eggert, L., Fairhurst, G., and G. Shepherd, "UDP Usage Guidelines", BCP 145, RFC 8085, DOI 10.17487/RFC8085, March 2017, <<https://www.rfc-editor.org/info/rfc8085>>.
- [SMPTE-ST2110-10] Society of Motion Picture and Television Engineers, "SMPTE Standard - Professional Media Over Managed IP Networks: System Timing and Definitions", SMPTE ST 2110-10:2017, 2017, <<https://doi.org/10.5594/SMPTE.ST2110-10.2017>>.

[SMPTE-ST2110-21]

Society of Motion Picture and Television Engineers, "SMPTE Standard – Professional Media Over Managed IP Networks: Traffic Shaping and Delivery Timing for Video", SMPTE ST 2110-21:2017, 2017, <<https://doi.org/10.5594/SMPTE.ST2110-21.2017>>.

## 10.2. Informative References

- [RFC4175] Gharai, L. and C. Perkins, "RTP Payload Format for Uncompressed Video", RFC 4175, DOI 10.17487/RFC4175, September 2005, <<https://www.rfc-editor.org/info/rfc4175>>.
- [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, DOI 10.17487/RFC4585, July 2006, <<https://www.rfc-editor.org/info/rfc4585>>.
- [RFC5124] Ott, J. and E. Carrara, "Extended Secure RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/SAVPF)", RFC 5124, DOI 10.17487/RFC5124, February 2008, <<https://www.rfc-editor.org/info/rfc5124>>.
- [RFC7201] Westerlund, M. and C. Perkins, "Options for Securing RTP Sessions", RFC 7201, DOI 10.17487/RFC7201, April 2014, <<https://www.rfc-editor.org/info/rfc7201>>.
- [RFC7202] Perkins, C. and M. Westerlund, "Securing the RTP Framework: Why RTP Does Not Mandate a Single Media Security Solution", RFC 7202, DOI 10.17487/RFC7202, April 2014, <<https://www.rfc-editor.org/info/rfc7202>>.

## 10.3. URIs

- [1] <http://www.iana.org/assignments/rtp-parameters>

## Authors' Addresses

Sebastien Lugan  
intoPIX S.A.  
Rue Emile Francqui, 9  
1435 Mont-Saint-Guibert  
Belgium

Phone: +32 10 23 84 70  
Email: [D313B41E@dynmail.crt1.net](mailto:D313B41E@dynmail.crt1.net)  
URI: <http://www.intopix.com>

Gael Rouvroy  
intoPIX S.A.  
Rue Emile Francqui, 9  
1435 Mont-Saint-Guibert  
Belgium

Phone: +32 10 23 84 70  
Email: g.rouvroy@intopix.com  
URI: <http://www.intopix.com>

Antonin Descampe  
intoPIX S.A.  
Rue Emile Francqui, 9  
1435 Mont-Saint-Guibert  
Belgium

Phone: +32 10 23 84 70  
Email: a.descampe@intopix.com  
URI: <http://www.intopix.com>

Thomas Richter  
Fraunhofer IIS  
Am Wolfsmantel 33  
91048 Erlangen  
Germany

Phone: +49 9131 776 5126  
Email: thomas.richter@iis.fraunhofer.de  
URI: <https://www.iis.fraunhofer.de/>

Alexandre Willeme  
Universite catholique de Louvain  
Place du Levant, 2 - bte L5.04.04  
1348 Louvain-la-Neuve  
Belgium

Phone: +32 10 47 80 82  
Email: alexandre.willeme@uclouvain.be  
URI: <https://uclouvain.be/en/icteam>

A/V Transport Payloads Workgroup  
Internet-Draft  
Intended status: Standards Track  
Expires: 1 May 2020

J. Sandford  
British Broadcasting Corporation  
29 October 2019

RTP Payload for TTML Timed Text  
draft-ietf-payload-rtp-ttml-05

Abstract

This memo describes a Real-time Transport Protocol (RTP) payload format for TTML, an XML based timed text format for live and file based workflows from W3C. This payload format is specifically targeted at live workflows using TTML.

Status of This Memo

This Internet-Draft is submitted 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 <https://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 1 May 2020.

Copyright Notice

Copyright (c) 2019 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 (<https://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 . . . . .	2
2. Conventions, Definitions, and Abbreviations . . . . .	2
3. Media Format Description . . . . .	3
3.1. Relation to Other Text Payload Types . . . . .	3
3.2. TTML2 . . . . .	3
4. Payload Format . . . . .	3
4.1. RTP Header Usage . . . . .	4
4.2. Payload Data . . . . .	5
5. Payload content restrictions . . . . .	5
6. Payload processing requirements . . . . .	6
6.1. TTML Processor profile . . . . .	7
6.1.1. Feature extension designation . . . . .	7
6.1.2. Processor profile document . . . . .	7
6.1.3. Processor profile signalling . . . . .	8
7. Payload Examples . . . . .	9
8. Fragmentation of TTML Documents . . . . .	11
9. Protection Against Loss of Data . . . . .	11
10. Congestion Control Considerations . . . . .	12
11. Payload Format Parameters . . . . .	12
11.1. Clock Rate . . . . .	12
11.2. Mapping to SDP . . . . .	12
11.2.1. Examples . . . . .	13
11.3. Offer/Answer Considerations . . . . .	13
12. IANA Considerations . . . . .	13
13. Security Considerations . . . . .	13
14. Acknowledgements . . . . .	14
15. Normative References . . . . .	15
16. Informative References . . . . .	16
Appendix A. RFC Editor Considerations . . . . .	17
Author's Address . . . . .	17

## 1. Introduction

TTML (Timed Text Markup Language) [TTML2] is a media type for describing timed text such as closed captions and subtitles in television workflows or broadcasts as XML. This document specifies how TTML should be mapped into an RTP stream in live workflows including, but not restricted to, those described in the television broadcast oriented EBU-TT Part 3 [TECH3370] specification. This document does not define a media type for TTML but makes use of the existing application/ttml+xml media type [TTML-MTPR].

## 2. Conventions, Definitions, and Abbreviations

Unless otherwise stated, the term "document" refers to the TTML document being transmitted in the payload of the RTP packet(s).



The term "word" refers to a data word aligned to a specified number of bits in a computing sense and not to refer to linguistic words that might appear in the transported text.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

### 3. Media Format Description

#### 3.1. Relation to Other Text Payload Types

Prior payload types for text are not suited to the carriage of closed captions in Television Workflows. RFC 4103 for Text Conversation [RFC4103] is intended for low data rate conversation with its own session management and minimal formatting capabilities. RFC 4734 Events for Modem, Fax, and Text Telephony Signals [RFC4734] deals in large parts with the control signalling of facsimile and other systems. RFC 4396 for 3rd Generation Partnership Project (3GPP) Timed Text [RFC4396] describes the carriage of a timed text format with much more restricted formatting capabilities than TTML. The lack of an existing format for TTML or generic XML has necessitated the creation of this payload format.

#### 3.2. TTML2

TTML2 (Timed Text Markup Language, Version 2) [TTML2] is an XML-based markup language for describing textual information with associated timing metadata. One of its primary use cases is the description of subtitles and closed captions. A number of profiles exist that adapt TTML2 for use in specific contexts [TTML-MTPR]. These include both file based and streaming workflows.

### 4. Payload Format

In addition to the required RTP headers, the payload contains a section for the TTML document being transmitted (User Data Words), and a field for the Length of that data. Each RTP payload contains one or part of one TTML document.

A representation of the payload format for TTML is Figure 1.

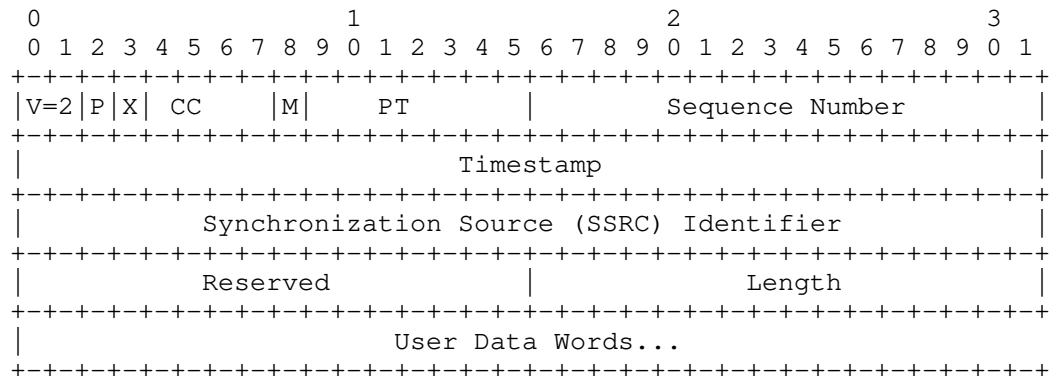


Figure 1: RTP Payload Format for TTML

#### 4.1. RTP Header Usage

RTP packet header fields SHALL be interpreted as per RFC 3550 [RFC3550], with the following specifics:

**Marker Bit (M):** 1 bit The Marker Bit is set to "1" to indicate the last packet of a document. Otherwise set to "0". Note: The first packet might also be the last.

**Timestamp:** 32 bits The RTP Timestamp encodes the epoch of the TTML document in User Data Words. Further detail on its usage may be found in Section 6. The clock frequency used is dependent on the application and is specified in the media type rate parameter as per Section 11.1. Documents spread across multiple packets MUST use the same timestamp but different consecutive Sequence Numbers. Sequential documents MUST NOT use the same timestamp. Because packets do not represent any constant duration, the timestamp cannot be used to directly infer packet loss.

**Reserved:** 16 bits These bits are reserved for future use and MUST be set to 0x0 and ignored at receive.

**Length:** 16 bits The length of User Data Words in bytes.

**User Data Words:** The length of User Data Words MUST match the value specified in the Length field User

Data Words contains the text of the whole document being transmitted or a part of the document being transmitted. Documents using character encodings where characters are not represented by a single byte MUST be serialized in big endian order, a.k.a. network byte order. Where a document will not fit

within the MTU, it may be fragmented across multiple packets. Further detail on fragmentation may be found in Section 8.

#### 4.2. Payload Data

TTML documents define a series of changes to text over time. TTML documents carried in User Data Words are encoded in accordance with one or more of the defined TTML profiles specified in the TTML registry [TTML-MTPR]. These profiles specify the document structure used, systems models, timing, and other considerations. TTML profiles may restrict the complexity of the changes and operational requirements may limit the maximum duration of TTML documents by a deployment configuration. Both of these cases are out of scope of this document.

Documents carried over RTP MUST conform to the following profile in addition to any others used.

#### 5. Payload content restrictions

This section defines constraints on the content of TTML documents carried over RTP.

Multiple TTML subtitle streams MUST NOT be interleaved in a single RTP stream.

The TTML document instance's root "tt" element in the "http://www.w3.org/ns/ttml" namespace MUST include a "timeBase" attribute in the "http://www.w3.org/ns/ttml#parameter" namespace containing the value "media".

This is equivalent to the TTML2 content profile definition document in Figure 2.

```

<?xml version="1.0" encoding="UTF-8"?>
<profile xmlns="http://www.w3.org/ns/ttml#parameter"
  xmlns:ttm="http://www.w3.org/ns/ttml#metadata"
  xmlns:tt="http://www.w3.org/ns/ttml"
  type="content"
  designator="urn:ietf:rfc:XXXX#content"
  combine="mostRestrictive">
  <features xml:base="http://www.w3.org/ns/ttml/feature/">
    <tt:metadata>
      <ttm:desc>
        This document is a minimal TTML2 content profile
        definition document intended to express the
        minimal requirements to apply when carrying TTML
        over RTP.
      </ttm:desc>
    </tt:metadata>
    <feature value="required">#timeBase-media</feature>
    <feature value="prohibited">#timeBase-smpte</feature>
    <feature value="prohibited">#timeBase-clock</feature>
  </features>
</profile>

```

Figure 2: TTML2 Content Profile Definition for Documents Carried Over RTP

## 6. Payload processing requirements

This section defines constraints on the processing of the TTML documents carried over RTP.

If a TTML document is assessed to be invalid then it **MUST** be discarded. This includes empty documents, i.e. those of zero length. When processing a valid document, the following requirements apply.

Each TTML document becomes active at its epoch *E*. *E* **MUST** be set to the RTP Timestamp in the header of the RTP packet carrying the TTML document. Computed TTML media times are offset relative to *E* in accordance with Section I.2 of [TTML2].

When processing a sequence of TTML documents each delivered in the same RTP stream, exactly zero or one document **SHALL** be considered active at each moment in the RTP time line. In the event that a document *D*<sub>(*n*-1)</sub> with *E*<sub>(*n*-1)</sub> is active, and document *D*<sub>(*n*)</sub> is delivered with *E*<sub>(*n*)</sub> where *E*<sub>(*n*-1)</sub> < *E*<sub>(*n*)</sub>, processing of *D*<sub>(*n*-1)</sub> **MUST** be stopped at *E*<sub>(*n*)</sub> and processing of *D*<sub>(*n*)</sub> **MUST** begin.

When all defined content within a document has ended then processing of the document **MAY** be stopped. This can be tested by constructing

the intermediate synchronic document sequence from the document, as defined by [TTML2]. If the last intermediate synchronic document in the sequence is both active and contains no region elements, then all defined content within the document has ended.

As described above, the RTP Timestamp does not specify the exact timing of the media in this payload format. Additionally, documents may be fragmented across multiple packets. This renders the RTCP jitter calculation unusable.

## 6.1. TTML Processor profile

### 6.1.1. Feature extension designation

This specification defines the following TTML feature extension designation:

\* urn:ietf:rfc:XXXX#rtp-relative-media-time

The namespace "urn:ietf:rfc:XXXX" is as defined by [RFC2648].

A TTML content processor supports the "#rtp-relative-media-time" feature extension if it processes media times in accordance with the payload processing requirements specified in this document, i.e. that the epoch E is set to the time equivalent to the RTP Timestamp as detailed above in Section 6.

### 6.1.2. Processor profile document

The required syntax and semantics declared in the minimal TTML2 processor profile in Figure 3 MUST be supported by the receiver, as signified by those "feature" or "extension" elements whose "value" attribute is set to "required".

```

<?xml version="1.0" encoding="UTF-8"?>
<profile xmlns="http://www.w3.org/ns/ttml#parameter"
  xmlns:ttml="http://www.w3.org/ns/ttml#metadata"
  xmlns:tt="http://www.w3.org/ns/ttml"
  type="processor"
  designator="urn:ietf:rfc:XXXX#processor"
  combine="mostRestrictive">
  <features xml:base="http://www.w3.org/ns/ttml/feature/">
    <tt:metadata>
      <ttml:desc>
        This document is a minimal TTML2 processor profile
        definition document intended to express the
        minimal requirements of a TTML processor able to
        process TTML delivered over RTP according to
        RFC XXXX.
      </ttml:desc>
    </tt:metadata>
    <feature value="required">#timeBase-media</feature>
    <feature value="optional">
      #profile-full-version-2
    </feature>
  </features>
  <extensions xml:base="urn:ietf:rfc:XXXX">
    <extension restricts="#timeBase-media" value="required">
      #rtp-relative-media-time
    </extension>
  </extensions>
</profile>

```

Figure 3: TTML2 Processor Profile Definition for Processing Documents Carried Over RTP

Note that this requirement does not imply that the receiver needs to support either TTML1 or TTML2 profile processing, i.e. the TTML2 "#profile-full-version-2" feature or any of its dependent features.

#### 6.1.3. Processor profile signalling

The "codecs" media type parameter MUST specify at least one processor profile. Short codes for TTML profiles are registered at [TTML-MTPR]. The processor profiles specified in "codecs" MUST be compatible with the processor profile specified in this document. Where multiple options exist in "codecs" for possible processor profile combinations (i.e. separated by "|" operator), every permitted option MUST be compatible with the processor profile specified in this document. Where processor profiles other than the one specified in this document are advertised in the "codecs" parameter, the requirements of the processor profile specified in

this document MAY be signalled additionally using the "+" operator with its registered short code.

A processor profile (X) is compatible with the processor profile specified here (P) if X includes all the features and extensions in P, identified by their character content, and the "value" attribute of each is at least as restrictive as the "value" attribute of the feature or extension in P that has the same character content. The term "restrictive" here is as defined in [TTML2] Section 6.

## 7. Payload Examples

Figure 4 is an example of a valid TTML document that may be carried using the payload format described in this document.

```
<?xml version="1.0" encoding="UTF-8"?>
<tt xml:lang="en"
  xmlns="http://www.w3.org/ns/ttml"
  xmlns:ttm="http://www.w3.org/ns/ttml#metadata"
  xmlns:ttp="http://www.w3.org/ns/ttml#parameter"
  xmlns:tts="http://www.w3.org/ns/ttml#styling"
  ttp:timeBase="media"
>
  <head>
    <metadata>
      <ttm:title>Timed Text TTML Example</ttm:title>
      <ttm:copyright>The Authors (c) 2006</ttm:copyright>
    </metadata>
    <styling>
      <!--
        s1 specifies default color, font, and text alignment
      -->
      <style xml:id="s1"
        tts:color="white"
        tts:fontFamily="proportionalSansSerif"
        tts:fontSize="100%"
        tts:textAlign="center"
      />
    </styling>
    <layout>
      <region xml:id="subtitleArea"
        style="s1"
        tts:extent="78% 11%"
        tts:padding="1% 5%"
        tts:backgroundColor="black"
        tts:displayAlign="after"
      />
    </layout>
  </head>
  <body region="subtitleArea">
    <div>
      <p xml:id="subtitle1" dur="5.0s" style="s1">
        How truly delightful!
      </p>
    </div>
  </body>
</tt>
```

Figure 4: Example TTML Document



## 8. Fragmentation of TTML Documents

Many of the use cases for TTML are low bit-rate with RTP packets expected to fit within the MTU. However, some documents may exceed the MTU. In these cases, they may be split between multiple packets. Where fragmentation is used, the following guidelines MUST be followed:

- \* It is RECOMMENDED that documents be fragmented as seldom as possible, i.e., the least possible number of fragments is created out of a document.
- \* Text strings MUST split at character boundaries. This enables decoding of partial documents. As a consequence, document fragmentation requires knowledge of the UTF-8/UTF-16 encoding formats to determine character boundaries.
- \* Document fragments SHOULD be protected against packet losses. More information can be found in Section 9

When a document spans more than one RTP packet, the entire document is obtained by concatenating User Data Words from each consecutive contributing packet in ascending order of Sequence Number.

As described in Section 6, only zero or one TTML document may be active at any point in time. As such, there MUST only be one document transmitted for a given RTP Timestamp. Furthermore, as stated in Section 4.1, the Marker Bit MUST be set for a packet containing the last fragment of a document. A packet following one where the Marker Bit is set contains the first fragment of a new document. The first fragment might also be the last.

## 9. Protection Against Loss of Data

Consideration must be devoted to keeping loss of documents due to packet loss within acceptable limits. What is deemed acceptable limits is dependant on the TTML profile(s) used and use case among other things. As such, specific limits are outside the scope of this document.

Documents MAY be sent without additional protection if end-to-end network conditions allow document loss to be within acceptable limits in all anticipated load conditions. Where such guarantees cannot be provided, implementations MUST use a mechanism to protect against packet loss. Potential mechanisms include FEC [RFC2733], retransmission [RFC4588], duplication [ST2022-7], or an equivalent technique.

## 10. Congestion Control Considerations

Congestion control for RTP SHALL be used in accordance with [RFC3550], and with any applicable RTP profile: e.g., [RFC3551]. Circuit Breakers [RFC8083] is an update to RTP [RFC3550] that defines criteria for when one is required to stop sending RTP Packet Streams. Applications implementing this standard MUST comply with [RFC8083] with particular attention paid to Section 4.4 on Media Usability. [RFC8085] provides additional information on the best practices for applying congestion control to UDP streams.

## 11. Payload Format Parameters

This RTP payload format is identified using the existing application/ttml+xml media type as registered with IANA [IANA] and defined in [TTML-MTPR].

### 11.1. Clock Rate

The default clock rate for TTML over RTP is 1000Hz. The clock rate SHOULD be included in any advertisements of the RTP stream where possible. This parameter has not been added to the media type definition as it is not applicable to TTML usage other than within RTP streams. In other contexts, timing is defined within the TTML document.

When choosing a clock rate, implementers should consider what other media their TTML streams may be used in conjunction with (e.g. video or audio). In these situations, it is RECOMMENDED that streams use the same clock source and Clock Rate as the related media. As TTML streams may be aperiodic, implementers should also consider the frequency range over which they expect packets to be sent and the temporal resolution required.

### 11.2. Mapping to SDP

The mapping of the application/ttml+xml media type and its parameters [TTML-MTPR] SHALL be done according to Section 3 of [RFC4855].

- \* The type name "application" goes in SDP "m=" as the media name.
- \* The media subtype "ttml+xml" goes in SDP "a=rtpmap" as the encoding name,
- \* The clock rate also goes in "a=rtpmap" as the clock rate.

Additional format specific parameters as described in the media type specification SHALL be included in the SDP file in "a=fmtp" as a

semicolon separated list of "parameter=value" pairs as described in [RFC4855]. The "codecs" parameter MUST be included in the "a=fmtp" line of the SDP file. Specific requirements for the "codecs" parameter are included in Section 6.1.3.

#### 11.2.1. Examples

A sample SDP mapping is presented in Figure 5.

```
m=application 30000 RTP/AVP 112
a=rtpmap:112 ttml+xml/90000
a=fmtp:112 charset=utf-8;codecs=imlt
```

Figure 5: Example SDP mapping

In this example, a dynamic payload type 112 is used. The 90 kHz RTP timestamp rate is specified in the "a=rtpmap" line after the subtype. The codecs parameter defined in the "a=fmtp" line indicates that the TTML data conforms to IMSC 1 Text profile.

#### 11.3. Offer/Answer Considerations

All parameters are declarative.

#### 12. IANA Considerations

No IANA action.

#### 13. Security Considerations

RTP packets using the payload format defined in this specification are subject to the security considerations discussed in the RTP specification [RFC3550], and in any applicable RTP profile such as RTP/AVP [RFC3551], RTP/AVPF [RFC4585], RTP/SAVP [RFC3711], or RTP/SAVPF [RFC5124]. However, as "Securing the RTP Protocol Framework: Why RTP Does Not Mandate a Single Media Security Solution" [RFC7202] discusses, it is not an RTP payload format's responsibility to discuss or mandate what solutions are used to meet the basic security goals like confidentiality, integrity, and source authenticity for RTP in general. This responsibility lays on anyone using RTP in an application. They can find guidance on available security mechanisms and important considerations in "Options for Securing RTP Sessions" [RFC7201]. Applications SHOULD use one or more appropriate strong security mechanisms. The rest of this Security Considerations section discusses the security impacting properties of the payload format itself.

To avoid potential buffer overflow attacks, receivers should take care to validate that the User Data Words in the RTP payload are of the appropriate length (using the Length field).

This payload format places no specific restrictions on the size of TTML documents that may be transmitted. As such, malicious implementations could be used to perform denial-of-service (DoS) attacks. RFC 4732 [RFC4732] provides more information on DoS attacks and describes some mitigation strategies. Implementers should take into consideration that the size and frequency of documents transmitted using this format may vary over time. As such, sender implementations should avoid producing streams that exhibit DoS-like behaviour and receivers should avoid false identification of a legitimate stream as malicious.

As with other XML types and as noted in RFC 7303 [RFC7303], XML Media Types, Section 10, repeated expansion of maliciously constructed XML entities can be used to consume large amounts of memory, which may cause XML processors in constrained environments to fail.

In addition, because of the extensibility features for TTML and of XML in general, it is possible that "application/ttml+xml" may describe content that has security implications beyond those described here. However, TTML does not provide for any sort of active or executable content, and if the processor follows only the normative semantics of the published specification, this content will be outside TTML namespaces and may be ignored. Only in the case where the processor recognizes and processes the additional content, or where further processing of that content is dispatched to other processors, would security issues potentially arise. And in that case, they would fall outside the domain of this RTP payload format and the application/ttml+xml registration document.

Although not prohibited, there are no expectations that XML signatures or encryption would normally be employed.

Further information related to privacy and security at a document level can be found in TTML 2 Appendix P [TTML2].

#### 14. Acknowledgements

Thanks to Nigel Megitt, James Gruessing, Robert Wadge, Andrew Bonney, James Weaver, John Fletcher, Frans De jong, and Willem Vermost for their valuable feedback throughout the development of this document. Thanks to the W3C Timed Text Working Group and EBU Timed Text working group for their substantial efforts in developing the timed text formats this payload format is intended to carry.

## 15. Normative References

- [IANA] IANA, "IANA - Media Types - Application", February 2019, <<https://www.iana.org/assignments/media-types/media-types.xhtml#application>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, DOI 10.17487/RFC3550, July 2003, <<https://www.rfc-editor.org/info/rfc3550>>.
- [RFC4103] Hellstrom, G. and P. Jones, "RTP Payload for Text Conversation", RFC 4103, DOI 10.17487/RFC4103, June 2005, <<https://www.rfc-editor.org/info/rfc4103>>.
- [RFC4855] Casner, S., "Media Type Registration of RTP Payload Formats", RFC 4855, DOI 10.17487/RFC4855, February 2007, <<https://www.rfc-editor.org/info/rfc4855>>.
- [RFC7303] Thompson, H. and C. Lilley, "XML Media Types", RFC 7303, DOI 10.17487/RFC7303, July 2014, <<https://www.rfc-editor.org/info/rfc7303>>.
- [RFC8083] Perkins, C. and V. Singh, "Multimedia Congestion Control: Circuit Breakers for Unicast RTP Sessions", RFC 8083, DOI 10.17487/RFC8083, March 2017, <<https://www.rfc-editor.org/info/rfc8083>>.
- [RFC8085] Eggert, L., Fairhurst, G., and G. Shepherd, "UDP Usage Guidelines", BCP 145, RFC 8085, DOI 10.17487/RFC8085, March 2017, <<https://www.rfc-editor.org/info/rfc8085>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [TECH3370] European Broadcasting Union, "TECH 3370 - EBU-TT PART 3: LIVE CONTRIBUTION", May 2017, <<https://tech.ebu.ch/publications/tech3370>>.
- [TTML-MTPR] W3C - Timed Text Working Group, "TTML Media Type

Definition and Profile Registry", January 2017,  
<<https://www.w3.org/TR/ttml-profile-registry/>>.

- [TTML2] W3C - Timed Text Working Group, "Timed Text Markup Language 2 (TTML2)", November 2018,  
<<https://www.w3.org/TR/ttml2/>>.

## 16. Informative References

- [RFC2648] Moats, R., "A URN Namespace for IETF Documents", RFC 2648, DOI 10.17487/RFC2648, August 1999,  
<<https://www.rfc-editor.org/info/rfc2648>>.
- [RFC2733] Rosenberg, J. and H. Schulzrinne, "An RTP Payload Format for Generic Forward Error Correction", RFC 2733, DOI 10.17487/RFC2733, December 1999,  
<<https://www.rfc-editor.org/info/rfc2733>>.
- [RFC3551] Schulzrinne, H. and S. Casner, "RTP Profile for Audio and Video Conferences with Minimal Control", STD 65, RFC 3551, DOI 10.17487/RFC3551, July 2003,  
<<https://www.rfc-editor.org/info/rfc3551>>.
- [RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, DOI 10.17487/RFC3711, March 2004,  
<<https://www.rfc-editor.org/info/rfc3711>>.
- [RFC4396] Rey, J. and Y. Matsui, "RTP Payload Format for 3rd Generation Partnership Project (3GPP) Timed Text", RFC 4396, DOI 10.17487/RFC4396, February 2006,  
<<https://www.rfc-editor.org/info/rfc4396>>.
- [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, DOI 10.17487/RFC4585, July 2006,  
<<https://www.rfc-editor.org/info/rfc4585>>.
- [RFC4588] Rey, J., Leon, D., Miyazaki, A., Varsa, V., and R. Hakenberg, "RTP Retransmission Payload Format", RFC 4588, DOI 10.17487/RFC4588, July 2006,  
<<https://www.rfc-editor.org/info/rfc4588>>.
- [RFC4732] Handley, M., Ed., Rescorla, E., Ed., and IAB, "Internet Denial-of-Service Considerations", RFC 4732, DOI 10.17487/RFC4732, December 2006,  
<<https://www.rfc-editor.org/info/rfc4732>>.

- [RFC4734] Schulzrinne, H. and T. Taylor, "Definition of Events for Modem, Fax, and Text Telephony Signals", RFC 4734, DOI 10.17487/RFC4734, December 2006, <<https://www.rfc-editor.org/info/rfc4734>>.
- [RFC5124] Ott, J. and E. Carrara, "Extended Secure RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/SAVPF)", RFC 5124, DOI 10.17487/RFC5124, February 2008, <<https://www.rfc-editor.org/info/rfc5124>>.
- [RFC7201] Westerlund, M. and C. Perkins, "Options for Securing RTP Sessions", RFC 7201, DOI 10.17487/RFC7201, April 2014, <<https://www.rfc-editor.org/info/rfc7201>>.
- [RFC7202] Perkins, C. and M. Westerlund, "Securing the RTP Framework: Why RTP Does Not Mandate a Single Media Security Solution", RFC 7202, DOI 10.17487/RFC7202, April 2014, <<https://www.rfc-editor.org/info/rfc7202>>.
- [ST2022-7] SMPTE, "ST 2022-7:2019 - Seamless Protection Switching of SMPTE ST 2022 IP Datagrams", November 2019, <<https://ieeexplore.ieee.org/document/8716822>>.

#### Appendix A. RFC Editor Considerations

Note to RFC Editor: This section may be removed after carrying out all the instructions of this section.

The namespace "urn:ietf:rfc:XXXX" is to be replaced with the namespace for this document once it has received an RFC number.

"RFC XXXX" in Figure 3 is to be replaced with the RFC number for this document.

#### Author's Address

James Sandford  
British Broadcasting Corporation  
Dock House, MediaCityUK  
Salford  
United Kingdom

Phone: +44 30304 09549  
Email: [james.sandford@bbc.co.uk](mailto:james.sandford@bbc.co.uk)

payload  
Internet-Draft  
Intended status: Standards Track  
Expires: January 27, 2020

Reisenbauer  
Frequentis  
Brandhuber  
eurofunk  
Hagedorn  
Hagedorn  
Hoehnsch  
T-Systems  
Wenk  
Frequentis  
July 26, 2019

RTP Payload Format for the TETRA Audio Codec  
draft-ietf-payload-tetra-03

Abstract

This document specifies a Real-time Transport Protocol (RTP) payload format for TETRA encoded speech signals. The payload format is designed to be able to interoperate with existing TETRA transport formats on non-IP networks. A media type registration is included, specifying the use of the RTP payload format and the storage format.

Status of This Memo

This Internet-Draft is submitted 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 <https://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 January 27, 2020.

Copyright Notice

Copyright (c) 2019 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



(<https://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 . . . . .	2
2. Conventions Used In This Document . . . . .	3
3. Media Format Background . . . . .	3
4. Payload format . . . . .	4
4.1. RTP Header Usage . . . . .	4
4.2. Payload layout . . . . .	4
4.3. Payload Header . . . . .	5
4.3.1. I bit: Frame Indicator . . . . .	5
4.3.2. F bit: Frame Type . . . . .	6
4.3.3. CTRL: Control bit(5 bits) . . . . .	6
4.3.4. C bit: Failed Crypto operation indication . . . . .	6
4.3.5. FRAME_NR: FN (5 bits) . . . . .	7
4.3.6. R: Audio Signal Relevance (3 bits) . . . . .	7
4.3.7. S: Spare (7 bits) . . . . .	7
4.4. Payload Data . . . . .	8
5. Payload example . . . . .	8
6. Congestion Control Considerations . . . . .	8
7. Payload Format Parameters . . . . .	9
7.1. Media Type Definition . . . . .	9
8. Mapping to SDP . . . . .	10
8.1. Offer/Answer Considerations . . . . .	11
8.2. Declarative SDP Considerations . . . . .	11
9. IANA Considerations . . . . .	12
10. Security Considerations . . . . .	12
11. References . . . . .	12
11.1. Normative References . . . . .	12
11.2. Informative References . . . . .	13
Authors' Addresses . . . . .	14

## 1. Introduction

This document specifies the payload format for packetization of TERrestrial Trunked Radio (TETRA) encoded speech signals [ETSI-TETRA-Codec] into the Real-time Transport Protocol (RTP) [RFC3550]. The payload format supports transmission of multiple frames per payload, robustness against packet loss, and interoperation with existing TETRA transport formats on non-IP networks, as described in Section Section 3.

The payload format itself is specified in Section Section 4.

## 2. Conventions 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 [RFC2119] when they appear in ALL CAPS. These words may also appear in this document in lower case as plain English words, absent their normative meanings.

The following acronyms are used in this document:

- o ETSI: European Telecommunications Standards Institute
- o TETRA: TERrestrial TRunked RADio

The byte order used in this document is network byte order, i.e., the most significant byte first. The bit order is also the most significant bit first. This is presented in all figures as having the most significant bit leftmost on a line and with the lowest number. Some bit fields may wrap over multiple lines in which cases the bits on the first line are more significant than the bits on the next line.

Best current practices for writing an RTP payload format specification were followed [RFC2736] updated with [RFC8088].

## 3. Media Format Background

The TETRA codec is used as vocoder for TETRA systems. The TETRA codec is designed for compressing 30ms of audio speech data into 137 bits. The TETRA codec is designed in such a way that on the air interface two of these 30ms samples are transported together (sub-block 1 and sub-block 2). The codec allows that data of the first 30ms voice frame can be stolen and used for other purposes, e.g. for the exchange of dynamically updated key-material in end-to-end encrypted voice sessions. Codec payload serialisation is specified for TDM lines with 2048 kBit/s within traditional circuit mode based TETRA system. For this purpose two optional formats are defined [ETSI-TETRA-Codec], the first format is called FSTE (First Speech Transport Encoding Format), the other format is called OSTE (Optimized Speech Transport Encoding Format). These two formats differ mainly insofar that the OSTE format transports an additional 5 bit frame number, which provides timing information from the air interface to the receiving side in order to save the need for buffering due to different transports speed on air and in 64 kbit/s circuit switched networks. The RTP payload format is defined such that the value of this frame number can be transported.

#### 4. Payload format

The RTP payload format is designed in such a way that it can carry the information needed to map the audio and control payload from [ETSI-TETRA-ISI]. The RTP format is defined such that both of the independent sub-blocks can be transferred separately or together within one RTP packet. Both of them contain the same information in terms of control bits – the information is propagated redundantly. This redundancy is driven by on one hand to simplify the encoding process in direction from E1 to RTP on the other to provide the option to go for either 30ms or 60ms packet size. The redundant information SHALL be propagated consistently equal – otherwise the behavior of the receiver is unspecified. The payload format is chosen such that the TETRA data bits are octet aligned.

##### 4.1. RTP Header Usage

The format of the RTP header is specified in [RFC3550]. The use of the fields of the RTP header by the TETRA payload format is consistent with that specification.

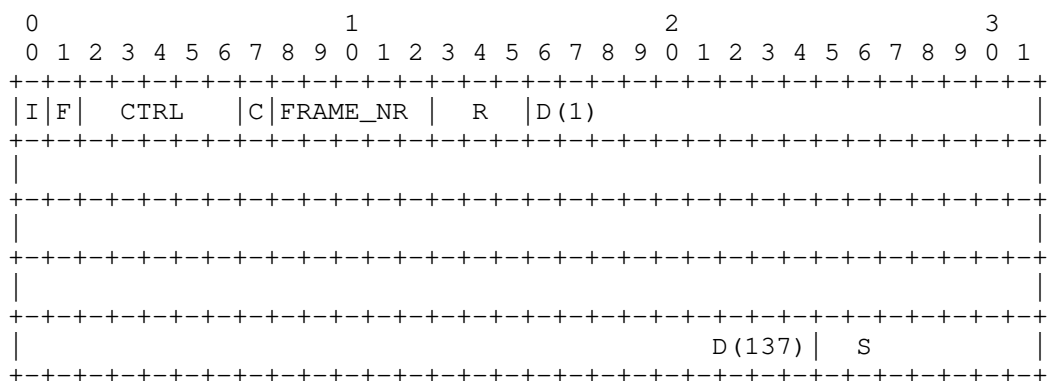
The payload length of TETRA is an integer number of octets; therefore, no padding is necessary.

The timestamp, sequence number, and marker bit (M) of the RTP header are used in accordance with Section 4.1 of [RFC3551].

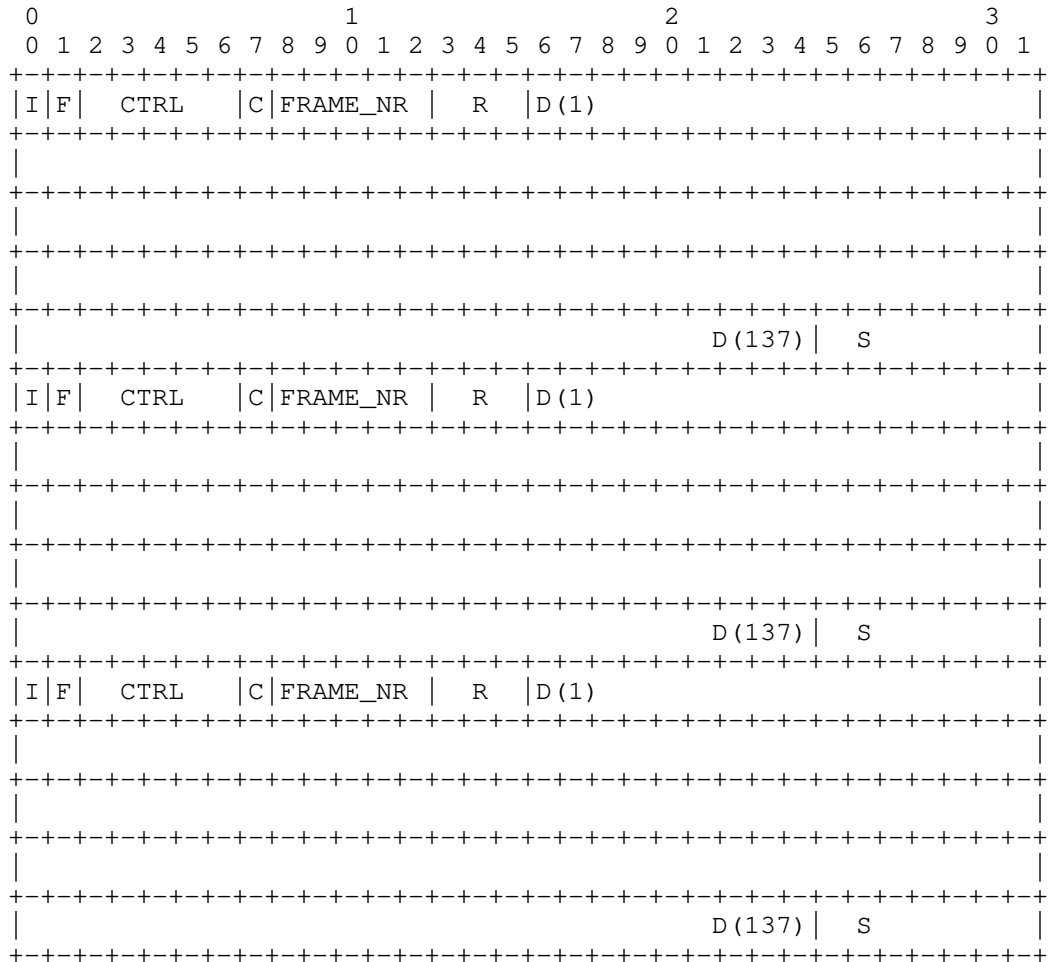
The RTP payload type for Tetra is to be assigned dynamically.

##### 4.2. Payload layout

RTP payload is composed of multiple blocks with TETRA audio data. TETRA Audio data itself contains: – Audio Payload Header – Audio Data (137 Bit) – 7 Spare Bits



RTP payload can be formed by any integer multiple of 30ms audio using following layout (e.g. 90ms audio payload):



#### 4.3. Payload Header

##### 4.3.1. I bit: Frame Indicator

1: The following frame contains a first block of two sub-blocks

0: The following frame contains a separated sub-block. A sub-block marked as such could either be a second sub-block, or an independent block, which does not have a relation with any first block. To distinguish between the one and the other the information of the Control bits has to be evaluated.

## 4.3.2. F bit: Frame Type

Value	Frame contains
0	FSTE encoded data
1	OSTE encoded data

## 4.3.3. CTRL: Control bit (5 bits)

Ctrl 1..3 derived from the information propagated according table 5.7 of [ETSI-TETRA-ISI].

Value	Sub block 1	Sub block 2
000	normal	normal
001	C stolen	normal
010	U stolen	normal
011	C stolen	C stolen
100	C stolen	U stolen
101	U stolen	C stolen
110	U stolen	U stolen
111	O&M ISI block	

Ctrl 4..5 derived from the information propagated according table 5.7 of [ETSI-TETRA-ISI].

Value	Sub block 1	Sub block 2
00	no bad frame indicator	no bad frame indicator
01	no bad frame indicator	bad frame indicator(s)
10	bad frame indicator(s)	no bad frame indicator
11	bad frame indicator(s)	bad frame indicator(s)

NOTE: The interpretation of C4 and C5 is outside the scope of the present document

## 4.3.4. C bit: Failed Crypto operation indication

This bit may be set to "1" if a decryption (encrypted audio along the circuit switched mobile network, decryption at the RTP sender forwarding this audio) operation could not be performed successfully for the specific half-block. Consequently, the encryption status of

the half-block audio data is unknown. Implementation of an RTP receiver has to take into account "C bit" when forwarding such TETRA audio data (either to a decoder directly or via TETRA infrastructure to a TETRA mobile unit), the contained audio might be scrambled - depending if the audio originally was generated as a plain-override half-block or as an encrypted half-block.

#### 4.3.5. FRAME\_NR: FN (5 bits)

The frame number bits contain an uplink frame number as defined in table 5.3 of [ETSI-TETRA-ISI]. If no frame number is available the FRAME\_NR value SHALL be set to 00000.

#### 4.3.6. R: Audio Signal Relevance (3 bits)

The Audio Signal Relevance bits contain information about the Relevance of the voice packet contained here.

R 1

0: no audio signal relevance propagated (R2 and R3 do not contain any valid information)

1: audio signal relevance propagated in R2 and R3

R 2..3 According to table 1 of [BDBOS-BIP20]

value	relevance
00	no audio signal relevance (level ? -72 dBm0)
01	low audio signal relevance (-52dBm0 ? level > -72dBm0)
10	medium audio signal relevance (-32dBm0 ? level > -52dBm0)
11	high audio signal relevance (0dBm0 ? level > -32dBm0)

NOTE: Receiver SHOULD consider stolen or erroneous blocks as not available for audio decoding as indicated by control bits independent of audio signal relevance bits.

#### 4.3.7. S: Spare (7 bits)

The S bits bits are reserved for future use and set to "0" currently.

#### 4.4. Payload Data

The payload itself contains TETRA ACELP coded speech information encoded according to table 4 of [ETSI-TETRA-Codec].

#### 5. Payload example

The following example shows how a first and a second consecutive 30 ms frame is combined into a single 60ms RTP packet. Note: This example shows the usage of OSTE mapping.

```

      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
+-----+-----+-----+-----+-----+-----+-----+-----+
|1|1|  CTRL  |C|0|0|0|0|0|0|0|0|D(1)|
+-----+-----+-----+-----+-----+-----+-----+-----+
|
+-----+-----+-----+-----+-----+-----+-----+-----+
|
+-----+-----+-----+-----+-----+-----+-----+-----+
|
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     D(137) | S
+-----+-----+-----+-----+-----+-----+-----+-----+
|0|1|  CTRL  |C|0|0|0|0|0|0|0|0|D(1)|
+-----+-----+-----+-----+-----+-----+-----+-----+
|
+-----+-----+-----+-----+-----+-----+-----+-----+
|
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     D(137) | S
+-----+-----+-----+-----+-----+-----+-----+-----+

```

Both halves of information contain exact the same CTRL bits

#### 6. Congestion Control Considerations

Tetra uses a fixed bitrate which cannot be adjusted at all.

Since UDP does not provide congestion control, applications that use RTP over UDP SHOULD implement their own congestion control above the UDP layer RFC8085 [RFC8085] and MAY also implement a transport circuit breaker RFC8083 [RFC8083]. Work in the RMCAT working group [RMCAT] describes the interactions and conceptual interfaces necessary between the application components that relate to congestion control, including the RTP layer, the higher-level media

codec control layer, and the lower-level transport interface, as well as components dedicated to congestion control functions.

Congestion control for RTP SHALL be used in accordance with RFC 3550 [RFC3550], and with any applicable RTP profile; e.g., RFC 3551 [RFC3551]. An additional requirement if best-effort service is being used is: users of this payload format MUST monitor packet loss to ensure that the packet loss rate is within acceptable parameters.

## 7. Payload Format Parameters

This RTP payload format is identified using one media subtype (audio/TETRA) which is registered in accordance with RFC 4855 [RFC4855] and per media type registration template from RFC 6838 [RFC6838].

### 7.1. Media Type Definition

The media type for the TETRA codec is expected to be allocated from the IETF tree once this draft turns into an RFC. This media type registration covers both real-time transfer via RTP and non-real-time transfers via stored files.

Type name:

audio

Subtype name:

TETRA

Required parameters:

none

Optional parameters:

These parameters apply to RTP transfer only.

- \* maxptime: The maximum amount of media which can be encapsulated in a payload packet, expressed as time in milliseconds. The time is calculated as the sum of the time that the media present in the packet represents. The time SHOULD be an integer multiple of the frame size. If this parameter is not present, the sender MAY encapsulate any number of speech frames into one RTP packet.

- \* ptime: see RFC 4566 [RFC4566].

Encoding considerations:

This media type is framed and binary according Section 4.8 of RFC 6838 [RFC6838].

Security considerations:

See Section Section 10 of RFC XXXX. [RFC Editor: Upon publication as an RFC, please replace "XXXX" with the number assigned to this document and remove this note.]

Interoperability considerations: N/A



## Published specification:

RFC XXXX [RFC Editor: Upon publication as an RFC, please replace "XXXX" with the number assigned to this document and remove this note.]

## Applications that use this media type:

This media type is used in applications needing transport or storage of encoded voice. Some examples include; Voice over IP, streaming media, voice messaging, and voice recording on recording systems.

## Additional Information:

- Deprecated alias names for this type: N/A
- Magic number(s): N/A
- File extension(s): N/A
- Macintosh file type code(s): N/A

## Person &amp; email address to contact for further information:

Andreas Reisenbauer <mailto:andreas.reisenbauer@frequentis.com>

IETF Payload Working Group <mailto:payload@ietf.org>

## Intended usage:

COMMON

## Restrictions on usage:

This media subtype depends on RTP framing and hence is only defined for transfer via RTP RFC 3550 [RFC3550]. Transport within other framing protocols is not defined at this time.

## Author:

Andreas Reisenbauer <mailto:andreas.reisenbauer@frequentis.com>

## Change controller:

The IETF PAYLOAD Working Group, or other party as designated by the IESG.

## 8. Mapping to SDP

The information carried in the media type specification has a specific mapping to fields in the Session Description Protocol [RFC4566], which is commonly used to describe RTP sessions. When SDP is used to specify sessions employing the TETRA codec, the mapping is as follows:

## Media Type name:

audio

## Media subtype name:

TETRA

## Required parameters:

none

## Optional parameters:

none

#### Mapping Parameters into SDP

The information carried in the media type specification has a specific mapping to fields in the Session Description Protocol [RFC4566], which is commonly used to describe RTP sessions. When SDP is used to specify sessions employing the TETRA codec, the mapping is as follows:

- \* The media type ("audio") goes in SDP "m=" as the media name.
- \* The media subtype (payload format name) goes in SDP "a=rtpmap" as the encoding name. The RTP clock rate in "a=rtpmap" MUST be 8000.
- \* The parameters "ptime" and "maxptime" go in the SDP "a=ptime" and "a=maxptime" attributes, respectively.
- \* Any remaining parameters go in the SDP "a=fmtp" attribute by copying them directly from the media type parameter string as a semicolon-separated list of parameter=value pairs.

Here is an example SDP session of usage of TETRA:

```
m=audio 49120 RTP/AVP 99
a=rtpmap:99 TETRA/8000
a=maxptime:60
a=ptime:60
```

#### 8.1. Offer/Answer Considerations

The following considerations apply when using SDP Offer-Answer procedures to negotiate the use of TETRA payload in RTP:

- o In most cases, the parameters "maxptime" and "ptime" will not affect interoperability; however, the setting of the parameters can affect the performance of the application. The SDP offer-answer handling of the "ptime" and "maxptime" parameter is described in RFC3264 [RFC3264].
- o Integer multiples of 30ms SHALL be used for ptime. It is recommended to use packet size of 60ms. There is no need that ptime and maxptime parameters are negotiated symmetrically.
- o Any unknown parameter in an offer SHALL be removed in the answer.

#### 8.2. Declarative SDP Considerations

For declarative media, the "ptime" and "maxptime" parameter specify the possible variants used by the sender.

## 9. IANA Considerations

This memo requests that IANA registers [audio/TETRA] from section Section 7.1. The media type is also requested to be added to the IANA registry for "RTP Payload Format MIME types" (<http://www.iana.org/assignments/rtp-parameters>).

## 10. Security Considerations

RTP packets using the payload format defined in this specification are subject to the security considerations discussed in the RTP specification [RFC3550], and in any applicable RTP profile. The main security considerations for the RTP packet carrying the RTP payload format defined within this memo are confidentiality, integrity and source authenticity. Confidentiality is achieved by encryption of the RTP payload. Integrity of the RTP packets through suitable cryptographic integrity protection mechanism. Cryptographic systems may also allow the authentication of the source of the payload. A suitable security mechanism for this RTP payload format should provide confidentiality, integrity protection and at least source authentication capable of determining if an RTP packet is from a member of the RTP session or not.

Note that the appropriate mechanism to provide security to RTP and payloads following this memo may vary. It is dependent on the application, the transport, and the signaling protocol employed. Therefore a single mechanism is not sufficient, although if suitable the usage of SRTP [RFC3711] is recommended. Other mechanism that may be used are IPsec [RFC4301] and TLS [RFC5246] (RTP over TCP), but also other alternatives may exist.

## 11. References

### 11.1. Normative References

[BDBOS-BIP20]

BDBOS, "BIP 20 QOS Dienstguete-Parameter BOS-Interoperabilitaetsprofil fuer Endgeraete zur Nutzung im Digitalfunk BOS; Version 2014-04 - Revision 2", 2014.

[ETSI-TETRA-Codec]

ETSI, "EN 300 395-2; Terrestrial Trunked Radio (TETRA); Speech codec for full-rate traffic channel; Part 2: TETRA codec V1.3.1", 2005, [http://www.etsi.org/deliver/etsi\\_en/300300\\_300399/30039502/01.03.01\\_60/en\\_30039502v010301p.pdf](http://www.etsi.org/deliver/etsi_en/300300_300399/30039502/01.03.01_60/en_30039502v010301p.pdf).

## [ETSI-TETRA-ISI]

ETSI, "TS 100 392-3-8; Terrestrial Trunked Radio (TETRA); Voice plus Data (V+D); Part 3: Interworking at the Inter-System Interface (ISI); Sub-part 8: Generic Speech Format Implementation V1.3.1", 2018, <[https://www.etsi.org/deliver/etsi\\_ts/100300\\_100399/1003920308/01.03.01\\_60/ts\\_1003920308v010301p.pdf](https://www.etsi.org/deliver/etsi_ts/100300_100399/1003920308/01.03.01_60/ts_1003920308v010301p.pdf)>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, DOI 10.17487/RFC3550, July 2003, <<https://www.rfc-editor.org/info/rfc3550>>.

[RFC3551] Schulzrinne, H. and S. Casner, "RTP Profile for Audio and Video Conferences with Minimal Control", STD 65, RFC 3551, DOI 10.17487/RFC3551, July 2003, <<https://www.rfc-editor.org/info/rfc3551>>.

[RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, DOI 10.17487/RFC4566, July 2006, <<https://www.rfc-editor.org/info/rfc4566>>.

[RFC8083] Perkins, C. and V. Singh, "Multimedia Congestion Control: Circuit Breakers for Unicast RTP Sessions", RFC 8083, DOI 10.17487/RFC8083, March 2017, <<https://www.rfc-editor.org/info/rfc8083>>.

[RFC8085] Eggert, L., Fairhurst, G., and G. Shepherd, "UDP Usage Guidelines", BCP 145, RFC 8085, DOI 10.17487/RFC8085, March 2017, <<https://www.rfc-editor.org/info/rfc8085>>.

## 11.2. Informative References

[RFC2736] Handley, M. and C. Perkins, "Guidelines for Writers of RTP Payload Format Specifications", BCP 36, RFC 2736, DOI 10.17487/RFC2736, December 1999, <<https://www.rfc-editor.org/info/rfc2736>>.

[RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, DOI 10.17487/RFC3264, June 2002, <<https://www.rfc-editor.org/info/rfc3264>>.

- [RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, DOI 10.17487/RFC3711, March 2004, <<https://www.rfc-editor.org/info/rfc3711>>.
- [RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, DOI 10.17487/RFC4301, December 2005, <<https://www.rfc-editor.org/info/rfc4301>>.
- [RFC4855] Casner, S., "Media Type Registration of RTP Payload Formats", RFC 4855, DOI 10.17487/RFC4855, February 2007, <<https://www.rfc-editor.org/info/rfc4855>>.
- [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008, <<https://www.rfc-editor.org/info/rfc5246>>.
- [RFC6838] Freed, N., Klensin, J., and T. Hansen, "Media Type Specifications and Registration Procedures", BCP 13, RFC 6838, DOI 10.17487/RFC6838, January 2013, <<https://www.rfc-editor.org/info/rfc6838>>.
- [RFC8088] Westerlund, M., "How to Write an RTP Payload Format", RFC 8088, DOI 10.17487/RFC8088, May 2017, <<https://www.rfc-editor.org/info/rfc8088>>.
- [RMCAT] IETF, "RTP Media Congestion Avoidance Techniques (rmcat) Working Group", 2018, <<https://datatracker.ietf.org/wg/rmcat/about/>>.

## Authors' Addresses

Andreas Reisenbauer  
Frequentis AG  
Innovationsstr. 1  
Vienna 1100  
Austria

Email: [andreas.reisenbauer@frequentis.com](mailto:andreas.reisenbauer@frequentis.com)

Udo Brandhuber  
eurofunk Kappacher GmbH  
Germany

Email: [ubrandhuber@eurofunk.com](mailto:ubrandhuber@eurofunk.com)

Joachim Hagedorn  
Hagedorn Informationssysteme GmbH  
Germany

Email: joachim@hagedorn-infosysteme.de

Klaus-Peter Hoehnsch  
T-Systems International GmbH  
Germany

Email: klaus-peter.hoehnsch@t-systems.com

Stefan Wenk  
Frequentis AG  
Innovationsstr. 1  
Vienna 1100  
Austria

Email: stefan.wenk@frequentis.com

Payload Working Group  
Internet-Draft  
Intended Status: Standards Track

Expires: April 27, 2020

Victor Demjanenko  
John Punaro  
David Satterlee  
VOCAL Technologies, Ltd.  
October 25, 2019

RTP Payload Format for TSVCSIS Codec  
draft-ietf-payload-tsvcsis-04

Status of This Memo

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

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.

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."

Abstract

This document describes the RTP payload format for the Tactical Secure Voice Cryptographic Interoperability Specification (TSVCIS) speech coder. TSVCSIS is a scalable narrowband voice coder supporting varying encoder data rates and fallbacks. It is implemented as an augmentation to the Mixed Excitation Linear Prediction Enhanced (MELPe) speech coder by conveying additional speech coder parameters for enhancing voice quality. TSVCSIS augmented speech data is

processed in conjunction with its temporal matched MELPe 2400 speech data. The RTP packetization of TSVCIIS and MELPe speech coder data is described in detail.

## Table of Contents

1. Introduction . . . . .	2
1.1. Conventions . . . . .	3
2. Background . . . . .	3
3. Payload Format . . . . .	4
3.1. MELPe Bitstream Definitions . . . . .	5
3.1.1. 2400 bps Bitstream Structure . . . . .	6
3.1.2. 1200 bps Bitstream Structure . . . . .	6
3.1.3. 600 bps Bitstream Structure . . . . .	7
3.1.4. Comfort Noise Bitstream Definition . . . . .	8
3.2. TSVCIIS Bitstream Definition . . . . .	8
3.3. Multiple TSVCIIS Frames in an RTP Packet . . . . .	10
3.4. Congestion Control Considerations . . . . .	11
4. Payload Format Parameters . . . . .	11
4.1. Media Type Definitions . . . . .	11
4.2. Mapping to SDP . . . . .	13
4.3. Declarative SDP Considerations . . . . .	15
4.4. Offer/Answer SDP Considerations . . . . .	15
5. Discontinuous Transmissions . . . . .	16
6. Packet Loss Concealment . . . . .	16
7. IANA Considerations . . . . .	16
8. Security Considerations . . . . .	16
10. References . . . . .	17
10.1. Normative References . . . . .	17
10.2. Informative References . . . . .	19
Authors' Addresses . . . . .	19

## 1. Introduction

This document describes how compressed Tactical Secure Voice Cryptographic Interoperability Specification (TSVCIIS) speech as produced by the TSVCIIS codec [TSVCIIS] [NRLVDR] may be formatted for use as an RTP payload. The TSVCIIS speech coder (or TSVCIIS speech aware communications equipment on any intervening transport link) may adjust to restricted bandwidth conditions by reducing the amount of augmented speech data and relying on the underlying MELPe speech coder for the most constrained bandwidth links.

Details are provided for packetizing the TSVCIIS augmented speech data along with MELPe 2400 bps speech parameters in a RTP packet. The sender may send one or more codec data frames per packet, depending on the application scenario or based on transport network conditions,



bandwidth restrictions, delay requirements, and packet loss tolerance.

### 1.1. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

Best current practices for writing an RTP payload format specification were followed [RFC2736] [RFC8088].

## 2. Background

The MELP speech coder was developed by the US military as an upgrade from the LPC-based CELP standard vocoder for low-bitrate communications [MELP]. ("LPC" stands for "Linear-Predictive Coding", and "CELP" stands for "Code-Excited Linear Prediction".) MELP was further enhanced and subsequently adopted by NATO as MELPe for use by its members and Partnership for Peace countries for military and other governmental communications as international NATO Standard STANAG 4591 [MELPE].

The Tactical Secure Voice Cryptographic Interoperability Specification (TSVCIS) is a specification written by the Tactical Secure Voice Working Group (TSVWG) for enabling all modern tactical secure voice devices to be interoperable across the Department of Defense [TSVCIS]. One of the most important aspects is that the voice modes defined in TSVCIS are based on specific fixed rates of Naval Research Lab's (NRL's) Variable Data Rate (VDR) Vocoder which uses the MELPe standard as its base [NRLVDR]. A complete TSVCIS speech frame consists of MELPe speech parameters and corresponding TSVCIS augmented speech data.

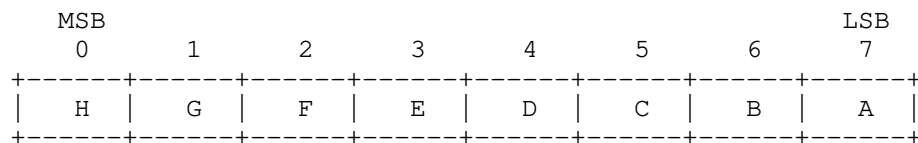
In addition to the augmented speech data, the TSVCIS specification identifies which speech coder and framing bits are to be encrypted, and how they are protected by forward error correction (FEC) techniques (using block codes). At the RTP transport layer, only the speech-coder-related bits need to be considered and are conveyed in unencrypted form. In most IP-based network deployments, standard link encryption methods (SRTP, VPNs, FIPS 140 link encryptors or Type 1 Ethernet encryptors) would be used to secure the RTP speech contents.

TSVCIS augmented speech data is derived from the signal processing and data already performed by the MELPe speech coder. For the

purposes of this specification, only the general parameter nature of TSVCIIS will be characterized. Depending on the bandwidth available (and FEC requirements), a varying number of TSVCIIS-specific speech coder parameters need to be transported. These are first byte-packed and then conveyed from encoder to decoder.

Byte packing of TSVCIIS speech data into packed parameters is processed as per the following example:

Three-bit field: bits A, B, and C (A is MSB, C is LSB)  
 Five-bit field: bits D, E, F, G, and H (D is MSB, H is LSB)



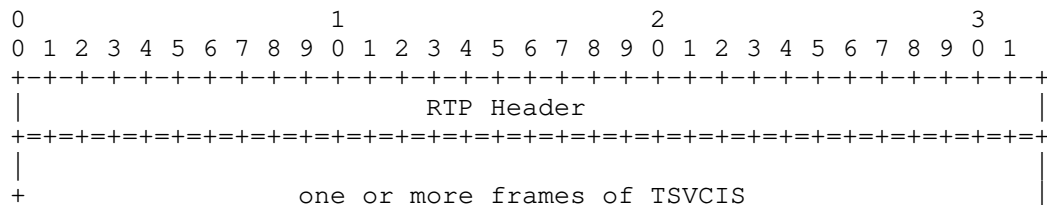
This packing method places the three-bit field "first" in the lowest bits followed by the next five-bit field. Parameters may be split between octets with the most significant bits in the earlier octet. Any unfilled bits in the last octet MUST be filled with zero.

In order to accommodate a varying amount of TSVCIIS augmented speech data, it is only necessary to specify the number of octets containing the packed TSVCIIS parameters. The encoding to do so is presented in Section 3.2. TSVCIIS specifically uses the NRL VDR in two configurations using 15 and 35 packed octet parameters [TSVCIIS].

### 3. Payload Format

The TSVCIIS codec augments the standard MELP 2400, 1200 and 600 bitrates and hence uses 22.5, 67.5, or 90 ms frames with a sampling rate clock of 8 kHz, so the RTP timestamp MUST be in units of 1/8000 of a second.

The RTP payload for TSVCIIS has the format shown in Figure 1. No additional header specific to this payload format is needed. This format is intended for situations where the sender and the receiver send one or more codec data frames per packet.



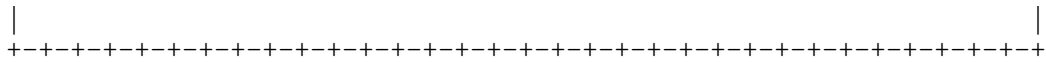


Figure 1: Packet Format Diagram

The RTP header of the packetized encoded TSVCIIS speech has the expected values as described in [RFC3550]. The usage of the M bit SHOULD be as specified in the applicable RTP profile -- for example, [RFC3551], where [RFC3551] specifies that if the sender does not suppress silence (i.e., sends a frame on every frame interval), the M bit will always be zero. When more than one codec data frame is present in a single RTP packet, the timestamp specified is that of the oldest data frame represented in the RTP packet.

The assignment of an RTP payload type for this new packet format is outside the scope of this document and will not be specified here. It is expected that the RTP profile for a particular class of applications will assign a payload type for this encoding, or if that is not done, then a payload type in the dynamic range shall be chosen by the sender.

### 3.1. MELPe Bitstream Definitions

The TCVCIIS speech coder includes all three MELPe coder rates used as base speech parameters or as speech coders for bandwidth restricted links. RTP packetization of MELPe follows RFC 8130 and is repeated here for all three MELPe rates [RFC8130] with its recommendations now regarded as requirements. The bits previously labeled as RSVA, RSVB, and RSVC in RFC 8130 SHOULD be filled with rate coding, CODA, CODB, and CODC, as shown in Table 1 (compatible with Table 7 in Section 3.3 of [RFC8130]).

Coder Bitrate	CODA	CODB	CODC	Length
2400 bps	0	0	N/A	7
1200 bps	1	0	0	11
600 bps	0	1	N/A	7
Comfort Noise	1	0	1	2
TSVCIIS data	1	1	N/A	var.

Table 1: TSVCIIS/MELPe Frame Bitrate Indicators and Frame Length

The total number of bits used to describe one MELPe frame of 2400 bps speech is 54, which fits in 7 octets (with two rate code bits). For MELPe 1200 bps speech, the total number of bits used is 81, which fits in 11 octets (with three rate code bits and four unused bits). For MELPe 600 bps speech, the total number of bits used is 54, which fits in 7 octets (with two rate code bits). The comfort noise frame consists of 13 bits, which fits in 2 octets (with three rate code bits). TSVCIS packed parameters will use the last code combination in a trailing byte as discussed in Section 3.2.

It should be noted that CODB for MELPe 600 bps mode MAY deviate from the value in Table 1 when bit 55 is used as an end-to-end framing bit. Frame decoding would remain distinct as CODA being zero on its own would indicate a 7-byte frame for either 2400 or 600 bps rate and the use of 600 bps speech coding could be deduced from the RTP timestamp (and anticipated by the SDP negotiations).

### 3.1.1. 2400 bps Bitstream Structure

The 2400 bps MELPe RTP payload is constructed as per Figure 2. Note that CODA MUST be filled with 0 and CODB SHOULD be filled with 0 as per Section 3.1. CODB MAY contain an end-to-end framing bit if required by the endpoints.

MSB							LSB
0	1	2	3	4	5	6	7
B_08	B_07	B_06	B_05	B_04	B_03	B_02	B_01
B_16	B_15	B_14	B_13	B_12	B_11	B_10	B_09
B_24	B_23	B_22	B_21	B_20	B_19	B_18	B_17
B_32	B_31	B_30	B_29	B_28	B_27	B_26	B_25
B_40	B_39	B_38	B_37	B_36	B_35	B_34	B_33
B_48	B_47	B_46	B_45	B_44	B_43	B_42	B_41
CODA	CODB	B_54	B_53	B_52	B_51	B_50	B_49

Figure 2: Packed MELPe 2400 bps Payload Octets

### 3.1.2. 1200 bps Bitstream Structure

The 1200 bps MELPe RTP payload is constructed as per Figure 3. Note that CODA, CODB, and CODC MUST be filled with 1, 0, and 0

respectively as per Section 3.1. RSV0 MUST be coded as 0.

MSB								LSB
0	1	2	3	4	5	6	7	
B_08	B_07	B_06	B_05	B_04	B_03	B_02	B_01	
B_16	B_15	B_14	B_13	B_12	B_11	B_10	B_09	
B_24	B_23	B_22	B_21	B_20	B_19	B_18	B_17	
B_32	B_31	B_30	B_29	B_28	B_27	B_26	B_25	
B_40	B_39	B_38	B_37	B_36	B_35	B_34	B_33	
B_48	B_47	B_46	B_45	B_44	B_43	B_42	B_41	
B_56	B_55	B_54	B_53	B_52	B_51	B_50	B_49	
B_64	B_63	B_62	B_61	B_60	B_59	B_58	B_57	
B_72	B_71	B_70	B_69	B_68	B_67	B_66	B_65	
B_80	B_79	B_78	B_77	B_76	B_75	B_74	B_73	
CODA	CODB	CODC	RSV0	RSV0	RSV0	RSV0	B_81	

Figure 3: Packed MELPe 1200 bps Payload Octets

### 3.1.3. 600 bps Bitstream Structure

The 600 bps MELPe RTP payload is constructed as per Figure 4. Note CODA MUST be filled with 0 and CODB SHOULD be filled with 1 as per Section 3.1. CODB MAY contain an end-to-end framing bit if required by the endpoints.

MSB								LSB
0	1	2	3	4	5	6	7	
B_08	B_07	B_06	B_05	B_04	B_03	B_02	B_01	
B_16	B_15	B_14	B_13	B_12	B_11	B_10	B_09	
B_24	B_23	B_22	B_21	B_20	B_19	B_18	B_17	
B_32	B_31	B_30	B_29	B_28	B_27	B_26	B_25	

B_40	B_39	B_38	B_37	B_36	B_35	B_34	B_33
+-----+	+-----+	+-----+	+-----+	+-----+	+-----+	+-----+	+-----+
B_48	B_47	B_46	B_45	B_44	B_43	B_42	B_41
+-----+	+-----+	+-----+	+-----+	+-----+	+-----+	+-----+	+-----+
CODA	CODB	B_54	B_53	B_52	B_51	B_50	B_49
+-----+	+-----+	+-----+	+-----+	+-----+	+-----+	+-----+	+-----+

Figure 4: Packed MELPe 600 bps Payload Octets

## 3.1.4. Comfort Noise Bitstream Definition

The comfort noise MELPe RTP payload is constructed as per Figure 5. Note that CODA, CODB, and CODC MUST be filled with 1, 0, and 1 respectively as per Section 3.1.

MSB							LSB
0	1	2	3	4	5	6	7
+-----+	+-----+	+-----+	+-----+	+-----+	+-----+	+-----+	+-----+
B_08	B_07	B_06	B_05	B_04	B_03	B_02	B_01
+-----+	+-----+	+-----+	+-----+	+-----+	+-----+	+-----+	+-----+
CODA	CODB	CODC	B_13	B_12	B_11	B_10	B_09
+-----+	+-----+	+-----+	+-----+	+-----+	+-----+	+-----+	+-----+

Figure 5: Packed MELPe Comfort Noise Payload Octets

## 3.2. TSVCIS Bitstream Definition

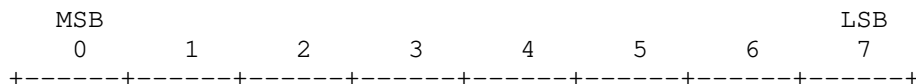
The TSVCIS augmented speech data as packed parameters MUST be placed immediately after a corresponding MELPe 2400 bps payload in the same RTP packet. The packed parameters are counted in octets (TC). The preferred placement SHOULD be used for TSVCIS payloads with TC less than or equal to 77 octets, and is shown in Figure 6. In the preferred placement, a single trailing octet SHALL be appended to include a two-bit rate code, CODA and CODB, (both bits set to one) and a six-bit modified count (MTC). The special modified count value of all ones (representing a MTC value of 63) SHALL NOT be used for this format as it is used as the indicator for the alternate packing format shown next. In a standard implementation, the TSVCIS speech coder uses a minimum of 15 octets for parameters in octet packed form. The modified count (MTC) MUST be reduced by 15 from the full octet count (TC). Computed MTC = TC-15. This accommodates a maximum of 77 parameter octets (maximum value of MTC is 62, 77 is the sum of 62+15).

MSB							LSB
0	1	2	3	4	5	6	7
+-----+	+-----+	+-----+	+-----+	+-----+	+-----+	+-----+	+-----+

1		T008		T007		T006		T005		T004		T003		T002		T001	
2		T016		T015		T014		T013		T012		T011		T010		T009	
3		T024		T023		T022		T021		T020		T019		T018		T017	
4		T032		T031		T030		T029		T028		T027		T026		T025	
5		T040		T039		T038		T037		T036		T035		T034		T033	
6		T048		T047		T046		T045		T044		T043		T042		T041	
7		T056		T055		T054		T053		T052		T051		T050		T049	
8		T064		T063		T062		T061		T060		T059		T058		T057	
9		T072		T071		T070		T069		T068		T067		T066		T065	
10		T080		T079		T078		T077		T076		T075		T074		T073	
11		T088		T087		T086		T085		T084		T083		T082		T081	
12		T096		T095		T094		T093		T092		T091		T090		T089	
13		T104		T103		T102		T101		T100		T099		T098		T097	
14		T112		T111		T110		T109		T108		T107		T106		T105	
15		T120		T119		T118		T117		T116		T115		T114		T113	
TC+1		CODA		CODB		modified octet count											

Figure 6: Preferred Packed TSVCIIS Payload Octets

In order to accommodate all other NRL VDR configurations, an alternate parameter placement MUST use two trailing bytes as shown in Figure 7. The last trailing byte MUST be filled with a two-bit rate code, CODA and CODB, (both bits set to one) and its six-bit count field MUST be filled with ones. The second to last trailing byte MUST contain the parameter count (TC) in octets (a value from 1 and 255, inclusive). The value of zero SHALL be considered as reserved.



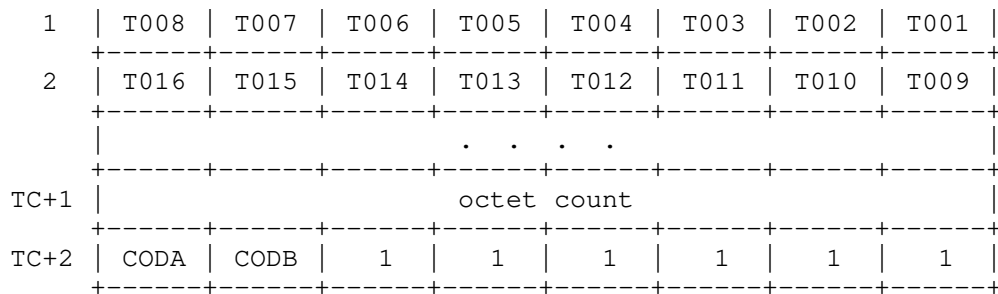


Figure 7: Length Unrestricted Packed TSVCIIS Payload Octets

### 3.3. Multiple TSVCIIS Frames in an RTP Packet

A TSVCIIS RTP packet payload consists of zero or more consecutive TSVCIIS coder frames (each consisting of MELPe 2400 and TSVCIIS coder data), with the oldest frame first, followed by zero or one MELPe comfort noise frame. The presence of a comfort noise frame can be determined by its rate code bits in its last octet.

The default packetization interval is one coder frame (22.5, 67.5, or 90 ms) according to the coder bitrate (2400, 1200, or 600 bps). For some applications, a longer packetization interval is used to reduce the packet rate.

A TSVCIIS RTP packet without coder and comfort noise frames MAY be used periodically by an endpoint to indicate connectivity by an otherwise idle receiver.

TSVCIIS coder frames in a single RTP packet MAY have varying TSVCIIS parameter octet counts. Its packed parameter octet count (length) is indicated in the trailing byte(s). All MELPe frames in a single RTP packet MUST be of the same coder bitrate. For all MELPe coder frames, the coder rate bits in the trailing byte identify the contents and length as per Table 1.

It is important to observe that senders have the following additional restrictions:

Senders SHOULD NOT include more TSVCIIS or MELPe frames in a single RTP packet than will fit in the MTU of the RTP transport protocol.

Frames MUST NOT be split between RTP packets.

It is RECOMMENDED that the number of frames contained within an RTP packet be consistent with the application. For example, in telephony and other real-time applications where delay is important, then the



fewer frames per packet the lower the delay, whereas for bandwidth-constrained links or delay-insensitive streaming messaging applications, more than one frame per packet or many frames per packet would be acceptable.

Information describing the number of frames contained in an RTP packet is not transmitted as part of the RTP payload. The way to determine the number of TSVCIIS/MELPe frames is to identify each frame type and length thereby counting the total number of octets within the RTP packet.

### 3.4. Congestion Control Considerations

The target bitrate of TSVCIIS can be adjusted at any point in time, thus allowing congestion management. Furthermore, the amount of encoded speech or audio data encoded in a single packet can be used for congestion control, since the packet rate is inversely proportional to the packet duration. A lower packet transmission rate reduces the amount of header overhead but at the same time increases latency and loss sensitivity, so it ought to be used with care.

Since UDP does not provide congestion control, applications that use RTP over UDP SHOULD implement their own congestion control above the UDP layer [RFC8085] and MAY also implement a transport circuit breaker [RFC8083]. Work in the RMCAT working group [RMCAT] describes the interactions and conceptual interfaces necessary between the application components that relate to congestion control, including the RTP layer, the higher-level media codec control layer, and the lower-level transport interface, as well as components dedicated to congestion control functions.

## 4. Payload Format Parameters

This RTP payload format is identified using the TSVCIIS media subtype, which is registered in accordance with RFC 4855 [RFC4855] and per the media type registration template from RFC 6838 [RFC6838].

### 4.1. Media Type Definitions

Type name: audio

Subtype name: TSVCIIS

Required parameters: N/A

Optional parameters:

ptime: the recommended length of time (in milliseconds) represented by the media in a packet. It SHALL use the nearest rounded-up ms integer packet duration. For TSVCIIS, this corresponds to the following values: 23, 45, 68, 90, 112, 135, 156, and 180. Larger values can be used as long as they are properly rounded. See Section 6 of RFC 4566 [RFC4566].

maxptime: the maximum length of time (in milliseconds) that can be encapsulated in a packet. It SHALL use the nearest rounded-up ms integer packet duration. For TSVCIIS, this corresponds to the following values: 23, 45, 68, 90, 112, 135, 156, and 180. Larger values can be used as long as they are properly rounded. See Section 6 of RFC 4566 [RFC4566].

bitrate: specifies the MELPe coder bitrates supported. Possible values are a comma-separated list of rates from the following set: 2400, 1200, 600. The modes are listed in order of preference; first is preferred. If "bitrate" is not present, the fixed coder bitrate of 2400 MUST be used.

tcmax: specifies the TSVCIIS maximum value for TC supported or desired ranging from 1 to 255. If "tcmax" is not present, a default value of 35 is used.

Encoding considerations: This media subtype is framed and binary; see Section 4.8 of RFC 6838 [RFC6838].

Security considerations: Please see Section 8 of RFC XXXX.

[EDITOR NOTE - please replace XXXX with the RFC number of this document.]

Interoperability considerations: N/A

Published specification: [TSVCIIS]

Applications that use this media type: N/A

Fragment identifier considerations: N/A

Additional information:

Clock Rate (Hz): 8000  
Channels: 1

Deprecated alias names for this type: N/A

Magic number(s): N/A

File extension(s): N/A

Macintosh file type code(s): N/A

Person & email address to contact for further information:

Victor Demjanenko, Ph.D.  
VOCAL Technologies, Ltd.  
520 Lee Entrance, Suite 202  
Buffalo, NY 14228  
United States of America  
Phone: +1 716 688 4675  
Email: victor.demjanenko@vocal.com

Intended usage: COMMON

Restrictions on usage: The media subtype depends on RTP framing and hence is only defined for transfer via RTP [RFC3550]. Transport within other framing protocols is not defined at this time.

Author: Victor Demjanenko

Change controller: IETF, contact <avt@ietf.org>

Provisional registration? (standards tree only): No

#### 4.2. Mapping to SDP

The mapping of the above-defined payload format media subtype and its parameters SHALL be done according to Section 3 of RFC 4855 [RFC4855].

The information carried in the media type specification has a specific mapping to fields in the Session Description Protocol (SDP) [RFC4566], which is commonly used to describe RTP sessions. When SDP is used to specify sessions employing the TSVCIIS codec, the mapping is as follows:

- o The media type ("audio") goes in SDP "m=" as the media name.
- o The media subtype (payload format name) goes in SDP "a=rtpmap" as the encoding name.
- o The parameter "bitrate" goes in the SDP "a=fmtp" attribute by copying it as a "bitrate=<value>" string.
- o The parameter "tcmx" goes in the SDP "a=fmtp" attribute by copying it as a "tcmx=<value>" string.

- o The parameters "ptime" and "maxptime" go in the SDP "a=ptime" and "a=maxptime" attributes, respectively.

When conveying information via SDP, the encoding name SHALL be "TSVCIS" (the same as the media subtype).

An example of the media representation in SDP for describing TSVCIIS might be:

```
m=audio 49120 RTP/AVP 96
a=rtpmap:96 TSVCIIS/8000
```

The optional media type parameter "bitrate", when present, MUST be included in the "a=fmtp" attribute in the SDP, expressed as a media type string in the form of a semicolon-separated list of parameter=value pairs. The string "value" can be one or more of 2400, 1200, and 600, separated by commas (where each bitrate value indicates the corresponding MELPe coder). An example of the media representation in SDP for describing TSVCIIS when all three coder bitrates are supported might be:

```
m=audio 49120 RTP/AVP 96
a=rtpmap:96 TSVCIIS/8000
a=fmtp:96 bitrate=2400,600,1200
```

The optional media type parameter "tcmax", when present, MUST be included in the "a=fmtp" attribute in the SDP, expressed as a media type string in the form of a semicolon-separated list of parameter=value pairs. The string "value" is an integer number in the range of 1 to 255 representing the maximum number of TSVCIIS parameter octets supported. An example of the media representation in SDP for describing TSVCIIS with a maximum of 101 octets supported is as follows:

```
m=audio 49120 RTP/AVP 96
a=rtpmap:96 TSVCIIS/8000
a=fmtp:96 tcmax=101
```

The parameter "ptime" cannot be used for the purpose of specifying the TSVCIIS operating mode, due to the fact that for certain values it will be impossible to distinguish which mode is about to be used (e.g., when ptime=68, it would be impossible to distinguish if the packet is carrying one frame of 67.5 ms or three frames of 22.5 ms).

Note that the payload format (encoding) names are commonly shown in upper case. Media subtypes are commonly shown in lower case. These names are case insensitive in both places. Similarly, parameter names are case insensitive in both the media subtype name and the

default mapping to the SDP a=fmtp attribute.

#### 4.3. Declarative SDP Considerations

For declarative media, the "bitrate" parameter specifies the possible bitrates used by the sender. Multiple TSVCIS rtpmap values (such as 97, 98, and 99, as used below) MAY be used to convey TSVCIS-coded voice at different bitrates. The receiver can then select an appropriate TSVCIS codec by using 97, 98, or 99.

```
m=audio 49120 RTP/AVP 97 98 99
a=rtpmap:97 TSVCIS/8000
a=fmtp:97 bitrate=2400
a=rtpmap:98 TSVCIS/8000
a=fmtp:98 bitrate=1200
a=rtpmap:99 TSVCIS/8000
a=fmtp:99 bitrate=600
```

For declarative media, the "tcmax" parameter specifies the maximum number of TSVCIS packed parameter octets used by the sender or the sender's communications channel.

#### 4.4. Offer/Answer SDP Considerations

In the Offer/Answer model [RFC3264], "bitrate" is a bidirectional parameter. Both sides MUST use a common "bitrate" value or values. The offer contains the bitrates supported by the offerer, listed in its preferred order. The answerer MAY agree to any bitrate by listing the bitrate first in the answerer response. Additionally, the answerer MAY indicate any secondary bitrate or bitrates that it supports. The initial bitrate used by both parties SHALL be the first bitrate specified in the answerer response.

For example, if offerer bitrates are "2400,600" and answer bitrates are "600,2400", the initial bitrate is 600. If other bitrates are provided by the answerer, any common bitrate between the offer and answer MAY be used at any time in the future. Activation of these other common bitrates is beyond the scope of this document.

The use of a lower bitrate is often important for a case such as when one endpoint utilizes a bandwidth-constrained link (e.g., 1200 bps radio link or slower), where only the lower coder bitrate will work.

In the Offer/Answer model [RFC3264], "tcmax" is a bidirectional parameter. Both sides SHOULD use a common "tcmax" value. The offer contains the tcmax supported by the offerer. The answerer MAY agree to any tcmax equal or less than this value by stating the desired tcmax in the answerer response. The answerer alternatively MAY

identify its own tcmax and rely on TSVCIIS ignoring any augmented data it cannot use.

## 5. Discontinuous Transmissions

A primary application of TSVCIIS is for radio communications of voice conversations, and discontinuous transmissions are normal. When TSVCIIS is used in an IP network, TSVCIIS RTP packet transmissions may cease and resume frequently. RTP synchronization source (SSRC) sequence number gaps indicate lost packets to be filled by Packet Loss Concealment (PLC), while abrupt loss of RTP packets indicates intended discontinuous transmissions. Resumption of voice transmission SHOULD be indicated by the RTP marker bit (M) set to 1.

If a TSVCIIS coder so desires, it may send a MELPe comfort noise frame as per Appendix B of [SCIP210] prior to ceasing transmission. A receiver may optionally use comfort noise during its silence periods. No SDP negotiations are required.

## 6. Packet Loss Concealment

TSVCIIS packet loss concealment (PLC) uses the special properties and coding for the pitch/voicing parameter of the MELPe 2400 bps coder. The PLC erasure indication utilizes any of the errored encodings of a non-voiced frame as identified in Table 1 of [MELPE]. For the sake of simplicity, it is preferred that a code value of 3 for the pitch/voicing parameter be used. Hence, set bits P0 and P1 to one and bits P2, P3, P4, P5, and P6 to zero.

When using PLC in 1200 bps or 600 bps mode, the MELPe 2400 bps decoder is called three or four times, respectively, to cover the loss of a low bitrate MELPe frame.

## 7. IANA Considerations

This memo requests that IANA registers TSVCIIS as specified in Section 4.1. The media type is also requested to be added to the IANA registry for "RTP Payload Format MIME types" (<http://www.iana.org/assignments/rtp-parameters>).

## 8. Security Considerations

RTP packets using the payload format defined in this specification are subject to the security considerations discussed in the RTP specification [RFC3550] and in any applicable RTP profile such as RTP/AVP [RFC3551], RTP/AVPF [RFC4585], RTP/SAVP [RFC3711], or RTP/SAVPF [RFC5124]. However, as discussed in [RFC7202], it is not an RTP payload format's responsibility to discuss or mandate what

solutions are used to meet such basic security goals as confidentiality, integrity, and source authenticity for RTP in general. This responsibility lies with anyone using RTP in an application. They can find guidance on available security mechanisms and important considerations in [RFC7201]. Applications SHOULD use one or more appropriate strong security mechanisms. The rest of this section discusses the security-impacting properties of the payload format itself.

This RTP payload format and the TSVCIIS decoder, to the best of our knowledge, do not exhibit any significant non-uniformity in the receiver-side computational complexity for packet processing and thus are unlikely to pose a denial-of-service threat due to the receipt of pathological data. Additionally, the RTP payload format does not contain any active content.

Please see the security considerations discussed in [RFC6562] regarding Voice Activity Detect (VAD) and its effect on bitrates.

## 10. References

### 10.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, May 2017, <<http://www.rfc-editor.org/info/rfc8174>>.
- [RFC2736] Handley, M. and C. Perkins, "Guidelines for Writers of RTP Payload Format Specifications", BCP 36, RFC 2736, DOI 10.17487/RFC2736, December 1999, <<http://www.rfc-editor.org/info/rfc2736>>.
- [RFC8088] Westerlund, M., "How to Write an RTP Payload Format", RFC 8088, DOI 10.17487/RFC8088, May 2017, <<http://www.rfc-editor.org/info/rfc8088>>.
- [RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, DOI 10.17487/RFC3264, June 2002, <<http://www.rfc-editor.org/info/rfc3264>>.

- [RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, DOI 10.17487/RFC3550, July 2003, <<http://www.rfc-editor.org/info/rfc3550>>.
- [RFC3551] Schulzrinne, H. and S. Casner, "RTP Profile for Audio and Video Conferences with Minimal Control", STD 65, RFC 3551, DOI 10.17487/RFC3551, July 2003, <<http://www.rfc-editor.org/info/rfc3551>>.
- [RFC8130] Demjanenko, V., and D. Satterlee, "RTP Payload Format for the Mixed Excitation Linear Prediction Enhanced (MELPe) Codec", RFC 8130, DOI 10.tbd/RFC8130, March 2017, <<http://www.rfc-editor.org/info/rfc8130>>.
- [RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, DOI 10.17487/RFC3711, March 2004, <<http://www.rfc-editor.org/info/rfc3711>>.
- [RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, DOI 10.17487/RFC4566, July 2006, <<http://www.rfc-editor.org/info/rfc4566>>.
- [RFC4855] Casner, S., "Media Type Registration of RTP Payload Formats", RFC 4855, DOI 10.17487/RFC4855, February 2007, <<http://www.rfc-editor.org/info/rfc4855>>.
- [RFC5124] Ott, J. and E. Carrara, "Extended Secure RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/SAVPF)", RFC 5124, DOI 10.17487/RFC5124, February 2008, <<http://www.rfc-editor.org/info/rfc5124>>.
- [RFC6562] Perkins, C. and JM. Valin, "Guidelines for the Use of Variable Bit Rate Audio with Secure RTP", RFC 6562, DOI 10.17487/RFC6562, March 2012, <<http://www.rfc-editor.org/info/rfc6562>>.
- [RFC6838] Freed, N., Klensin, J., and T. Hansen, "Media Type Specifications and Registration Procedures", BCP 13, RFC 6838, DOI 10.17487/RFC6838, January 2013, <<http://www.rfc-editor.org/info/rfc6838>>.
- [RFC8083] Perkins, C. and V. Singh, "Multimedia Congestion Control: Circuit Breakers for Unicast RTP Sessions", RFC 8083, DOI 10.17487/RFC8083, March 2017, <<http://www.rfc-editor.org/info/rfc8083>>.



- [RFC8085] Eggert, L., Fairhurst, G., and G. Shepherd, "UDP Usage Guidelines", RFC 8085, DOI 10.17487/RFC8085, March 2017, <<http://www.rfc-editor.org/info/rfc8085>>.
- [NRLVDR] Heide, D., Cohen, A., Lee, Y., and T. Moran, "Universal Vocoder Using Variable Data Rate Vocoding", Naval Research Lab, NRL/FR/5555-13-10,239, June 2013.
- [MELP] Department of Defense Telecommunications Standard, "Analog-to-Digital Conversion of Voice by 2,400 Bit/Second Mixed Excitation Linear Prediction (MELP)", MIL-STD-3005, December 1999.
- [MELPE] North Atlantic Treaty Organization (NATO), "The 600 Bit/S, 1200 Bit/S and 2400 Bit/S NATO Interoperable Narrow Band Voice Coder", STANAG No. 4591, January 2006.
- [SCIP210] National Security Agency, "SCIP Signaling Plan", SCIP-210, December 2007.

#### 10.2. Informative References

- [TSVCIS] National Security Agency, "Tactical Secure Voice Cryptographic Interoperability Specification (TSVCIS) Version 3.1", NSA 09-01A, March 2019.
- [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, DOI 10.17487/RFC4585, July 2006, <<http://www.rfc-editor.org/info/rfc4585>>.
- [RFC7201] Westerlund, M. and C. Perkins, "Options for Securing RTP Sessions", RFC 7201, DOI 10.17487/RFC7201, April 2014, <<http://www.rfc-editor.org/info/rfc7201>>.
- [RFC7202] Perkins, C. and M. Westerlund, "Securing the RTP Framework: Why RTP Does Not Mandate a Single Media Security Solution", RFC 7202, DOI 10.17487/RFC7202, April 2014, <<http://www.rfc-editor.org/info/rfc7202>>.
- [RMCAT] IETF, RTP Media Congestion Avoidance Techniques (rmcat) Working Group, <<https://datatracker.ietf.org/wg/rmcat/about/>>.

#### Authors' Addresses

Victor Demjanenko, Ph.D.

VOCAL Technologies, Ltd.  
520 Lee Entrance, Suite 202  
Buffalo, NY 14228  
United States of America

Phone: +1 716 688 4675  
Email: victor.demjanenko@vocal.com

John Punaro  
VOCAL Technologies, Ltd.  
520 Lee Entrance, Suite 202  
Buffalo, NY 14228  
United States of America

Phone: +1 716 688 4675  
Email: john.punaro@vocal.com

David Satterlee  
VOCAL Technologies, Ltd.  
520 Lee Entrance, Suite 202  
Buffalo, NY 14228  
United States of America

Phone: +1 716 688 4675  
Email: david.satterlee@vocal.com

Payload Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: January 25, 2020

J. Uberti  
S. Holmer  
M. Flodman  
Google  
J. Lennox  
8x8 / Jitsi  
D. Hong  
Vidyo  
July 24, 2019

RTP Payload Format for VP9 Video  
draft-ietf-payload-vp9-07

Abstract

This memo describes an RTP payload format for the VP9 video codec. The payload format has wide applicability, as it supports applications from low bit-rate peer-to-peer usage, to high bit-rate video conferences. It includes provisions for temporal and spatial scalability.

Status of This Memo

This Internet-Draft is submitted 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 <https://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 January 25, 2020.

Copyright Notice

Copyright (c) 2019 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 (<https://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 . . . . .	2
2. Conventions, Definitions and Acronyms . . . . .	3
3. Media Format Description . . . . .	3
4. Payload Format . . . . .	5
4.1. RTP Header Usage . . . . .	5
4.2. VP9 Payload Descriptor . . . . .	7
4.2.1. Scalability Structure (SS): . . . . .	11
4.3. VP9 Payload Header . . . . .	13
4.4. Frame Fragmentation . . . . .	13
4.5. Scalable encoding considerations . . . . .	13
4.6. Examples of VP9 RTP Stream . . . . .	14
4.6.1. Reference picture use for scalable structure . . . . .	14
5. Feedback Messages and Header Extensions . . . . .	15
5.1. Reference Picture Selection Indication (RPSI) . . . . .	15
5.2. Full Intra Request (FIR) . . . . .	15
5.3. Layer Refresh Request (LRR) . . . . .	15
5.4. Frame Marking . . . . .	16
6. Payload Format Parameters . . . . .	17
6.1. Media Type Definition . . . . .	17
6.2. SDP Parameters . . . . .	19
6.2.1. Mapping of Media Subtype Parameters to SDP . . . . .	19
6.2.2. Offer/Answer Considerations . . . . .	20
7. Security Considerations . . . . .	20
8. Congestion Control . . . . .	21
9. IANA Considerations . . . . .	21
10. Acknowledgments . . . . .	21
11. References . . . . .	21
11.1. Normative References . . . . .	21
11.2. Informative References . . . . .	23
Authors' Addresses . . . . .	23

## 1. Introduction

This memo describes an RTP payload specification applicable to the transmission of video streams encoded using the VP9 video codec [VP9-BITSTREAM]. The format described in this document can be used both in peer-to-peer and video conferencing applications.

TODO: VP9 description. Please see [VP9-BITSTREAM].

## 2. Conventions, Definitions and Acronyms

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 [RFC2119].

TODO: Cite terminology from [VP9-BITSTREAM].

## 3. Media Format Description

The VP9 codec can maintain up to eight reference frames, of which up to three can be referenced by any new frame.

VP9 also allows a frame to use another frame of a different resolution as a reference frame. (Specifically, a frame may use any references whose width and height are between 1/16th that of the current frame and twice that of the current frame, inclusive.) This allows internal resolution changes without requiring the use of key frames.

These features together enable an encoder to implement various forms of coarse-grained scalability, including temporal, spatial and quality scalability modes, as well as combinations of these, without the need for explicit scalable coding tools.

Temporal layers define different frame rates of video; spatial and quality layers define different and possibly dependent representations of a single input frame. Spatial layers allow a frame to be encoded at different resolutions, whereas quality layers allow a frame to be encoded at the same resolution but at different qualities (and thus with different amounts of coding error). VP9 supports quality layers as spatial layers without any resolution changes; hereinafter, the term "spatial layer" is used to represent both spatial and quality layers.

This payload format specification defines how such temporal and spatial scalability layers can be described and communicated.

Temporal and spatial scalability layers are associated with non-negative integer IDs. The lowest layer of either type has an ID of 0, and is sometimes referred to as the "base" temporal or spatial layer.

Layers are designed (and MUST be encoded) such that if any layer, and all higher layers, are removed from the bitstream along either of the two dimensions, the remaining bitstream is still correctly decodable.

For terminology, this document uses the term "frame" to refer to a single encoded VP9 frame for a particular resolution/quality, and "picture" to refer to all the representations (frames) at a single instant in time. A picture thus consists of one or more frames, encoding different spatial layers.

Within a picture, a frame with spatial layer ID equal to SID, where  $SID > 0$ , can depend on a frame of the same picture with a lower spatial layer ID. This "inter-layer" dependency can result in additional coding gain compared to the case where only traditional "inter-picture" dependency is used, where a frame depends on previously coded frame in time. For simplicity, this payload format assumes that, within a picture and if inter-layer dependency is used, a spatial layer SID frame can depend only on the immediately previous spatial layer SID-1 frame, when  $S > 0$ . Additionally, if inter-picture dependency is used, a spatial layer SID frame is assumed to only depend on a previously coded spatial layer SID frame.

Given above simplifications for inter-layer and inter-picture dependencies, a flag (the D bit described below) is used to indicate whether a spatial layer SID frame depends on the spatial layer SID-1 frame. Given the D bit, a receiver only needs to additionally know the inter-picture dependency structure for a given spatial layer frame in order to determine its decodability. Two modes of describing the inter-picture dependency structure are possible: "flexible mode" and "non-flexible mode". An encoder can only switch between the two on the first packet of a key frame with temporal layer ID equal to 0.

In flexible mode, each packet can contain up to 3 reference indices, which identify all frames referenced by the frame transmitted in the current packet for inter-picture prediction. This (along with the D bit) enables a receiver to identify if a frame is decodable or not and helps it understand the temporal layer structure. Since this is signaled in each packet it makes it possible to have very flexible temporal layer hierarchies and patterns which are changing dynamically.

In non-flexible mode, the inter-picture dependency (the reference indices) of a Picture Group (PG) MUST be pre-specified as part of the scalability structure (SS) data. In this mode, each packet has an index to refer to one of the described pictures in the PG, from which the pictures referenced by the picture transmitted in the current packet for inter-picture prediction can be identified.

(Editor's Note: A "Picture Group", as used in this document, is not the same thing as the term "Group of Pictures" as it is traditionally used in video coding, i.e. to mean an independently-

decoadable run of pictures beginning with a keyframe. Suggestions for better terminology are welcome.)

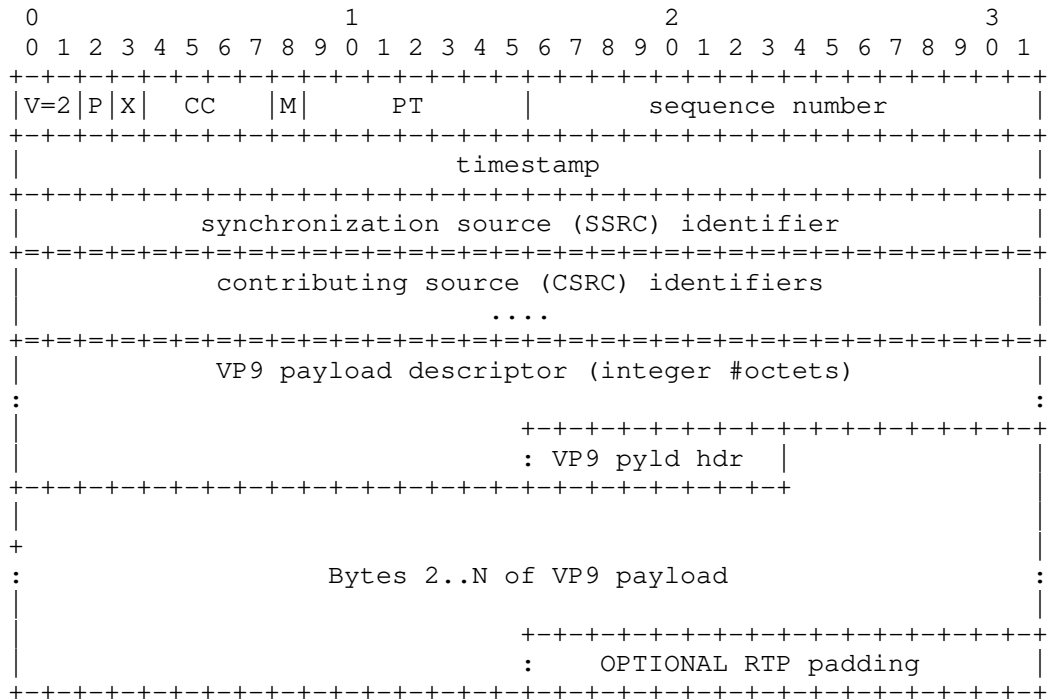
The SS data can also be used to specify the resolution of each spatial layer present in the VP9 stream for both flexible and non-flexible modes.

#### 4. Payload Format

This section describes how the encoded VP9 bitstream is encapsulated in RTP. To handle network losses usage of RTP/AVPF [RFC4585] is RECOMMENDED. All integer fields in the specifications are encoded as unsigned integers in network octet order.

##### 4.1. RTP Header Usage

The general RTP payload format for VP9 is depicted below.



The VP9 payload descriptor and VP9 payload header will be described in Section 4.2 and Section 4.3. OPTIONAL RTP padding MUST NOT be included unless the P bit is set. The figure specifically shows the format for the first packet in a frame. Subsequent packets will not contain the VP9 payload header, and will have later octets in the frame payload.

Figure 1

**Marker bit (M):** MUST be set to 1 for the final packet of the highest spatial layer frame (the final packet of the picture), and 0 otherwise. Unless spatial scalability is in use for this picture, this will have the same value as the E bit described below. Note this bit MUST be set to 1 for the target spatial layer frame if a stream is being rewritten to remove higher spatial layers.

**Payload Type (PT):** In line with the policy in Section 3 of [RFC3551], applications using the VP9 RTP payload profile MUST assign a dynamic payload type number to be used in each RTP session and provide a mechanism to indicate the mapping. See



Section 6.2 for the mechanism to be used with the Session Description Protocol (SDP) [RFC4566].

**Timestamp:** The RTP timestamp indicates the time when the input frame was sampled, at a clock rate of 90 kHz. If the input picture is encoded with multiple layer frames, all of the frames of the picture **MUST** have the same timestamp.

If a frame has the VP9 `show_frame` field set to 0 (i.e., it is meant only to populate a reference buffer, without being output) its timestamp **MAY** alternately be set to be the same as the subsequent frame with `show_frame` equal to 1. (This will be convenient for playing out pre-encoded content packaged with VP9 "superframes", which typically bundle `show_frame==0` frames with a subsequent `show_frame==1` frame.) Every frame with `show_frame==1`, however, **MUST** have a unique timestamp modulo the  $2^{32}$  wrap of the field.

The remaining RTP Fixed Header Fields (V, P, X, CC, sequence number, SSRC and CSRC identifiers) are used as specified in Section 5.1 of [RFC3550].

#### 4.2. VP9 Payload Descriptor

In flexible mode (with the F bit below set to 1), The first octets after the RTP header are the VP9 payload descriptor, with the following structure.

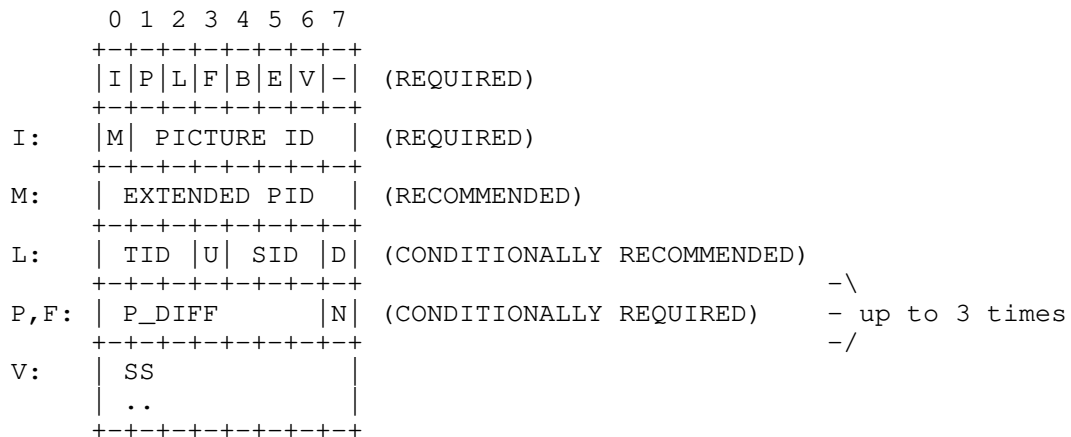


Figure 2

In non-flexible mode (with the F bit below set to 0), The first octets after the RTP header are the VP9 payload descriptor, with the following structure.

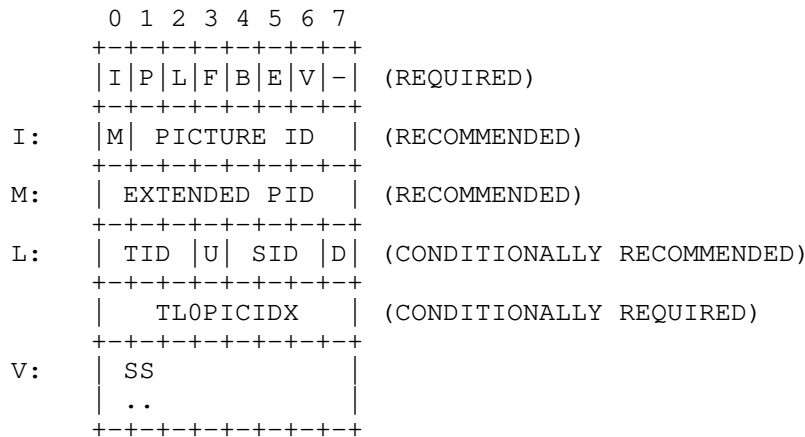


Figure 3

- I: Picture ID (PID) present. When set to one, the OPTIONAL PID MUST be present after the mandatory first octet and specified as below. Otherwise, PID MUST NOT be present. If the SS field was present in the stream's most recent start of a keyframe (i.e., non-flexible scalability mode is in use), then the PID MUST also be present in every packet.
- P: Inter-picture predicted frame. When set to zero, the frame does not utilize inter-picture prediction. In this case, up-switching to a current spatial layer's frame is possible from directly lower spatial layer frame. P SHOULD also be set to zero when encoding a layer synchronization frame in response to an LRR [I-D.ietf-avtext-ldrr] message (see Section 5.3). When P is set to zero, the TID field (described below) MUST also be set to 0 (if present). Note that the P bit does not forbid intra-picture, inter-layer prediction from earlier frames of the same picture, if any.
- L: Layer indices present. When set to one, the one or two octets following the mandatory first octet and the PID (if present) is as described by "Layer indices" below. If the F bit (described below) is set to 1 (indicating flexible mode), then only one octet is present for the layer indices. Otherwise if the F bit is set to 0 (indicating non-flexible mode), then two octets are present for the layer indices.

- F: Flexible mode. F set to one indicates flexible mode and if the P bit is also set to one, then the octets following the mandatory first octet, the PID, and layer indices (if present) are as described by "Reference indices" below. This MUST only be set to 1 if the I bit is also set to one; if the I bit is set to zero, then this MUST also be set to zero and ignored by receivers. The value of this F bit MUST only change on the first packet of a key picture. A key picture is a picture whose base spatial layer frame is a key frame, and which thus completely resets the encoder state. This packet will have its P bit equal to zero, SID or D bit (described below) equal to zero, and B bit (described below) equal to 1.
- B: Start of a frame. MUST be set to 1 if the first payload octet of the RTP packet is the beginning of a new VP9 frame, and MUST NOT be 1 otherwise. Note that this frame might not be the first frame of a picture.
- E: End of a frame. MUST be set to 1 for the final RTP packet of a VP9 frame, and 0 otherwise. This enables a decoder to finish decoding the frame, where it otherwise may need to wait for the next packet to explicitly know that the frame is complete. Note that, if spatial scalability is in use, more frames from the same picture may follow; see the description of the M bit above.
- V: Scalability structure (SS) data present. When set to one, the OPTIONAL SS data MUST be present in the payload descriptor. Otherwise, the SS data MUST NOT be present.
- : Bit reserved for future use. MUST be set to zero and MUST be ignored by the receiver.

The mandatory first octet is followed by the extension data fields that are enabled:

- M: The most significant bit of the first octet is an extension flag. The field MUST be present if the I bit is equal to one. If set, the PID field MUST contain 15 bits; otherwise, it MUST contain 7 bits. See PID below.

Picture ID (PID): Picture ID represented in 7 or 15 bits, depending on the M bit. This is a running index of the pictures. The field MUST be present if the I bit is equal to one. If M is set to zero, 7 bits carry the PID; else if M is set to one, 15 bits carry the PID in network byte order. The sender may choose between a 7- or 15-bit index. The PID SHOULD start on a random number, and MUST wrap after reaching the maximum ID. The receiver MUST NOT

assume that the number of bits in PID stay the same through the session.

In the non-flexible mode (when the F bit is set to 0), this PID is used as an index to the picture group (PG) specified in the SS data below. In this mode, the PID of the key frame corresponds to the first specified frame in the PG. Then subsequent PIDs are mapped to subsequently specified frames in the PG (modulo N\_G, specified in the SS data below), respectively.

All frames of the same picture MUST have the same PID value.

Frames (and their corresponding pictures) with the VP9 show\_frame field equal to 0 MUST have distinct PID values from subsequent pictures with show\_frame equal to 1. Thus, a Picture as defined in this specification is different than a VP9 Superframe.

All frames of the same picture MUST have the same value for show\_frame.

**Layer indices:** This information is optional but recommended whenever encoding with layers. For both flexible and non-flexible modes, one octet is used to specify a layer frame's temporal layer ID (TID) and spatial layer ID (SID) as shown both in Figure 2 and Figure 3. Additionally, a bit (U) is used to indicate that the current frame is a "switching up point" frame. Another bit (D) is used to indicate whether inter-layer prediction is used for the current frame.

In the non-flexible mode (when the F bit is set to 0), another octet is used to represent temporal layer 0 index (TL0PICIDX), as depicted in Figure 3. The TL0PICIDX is present so that all minimally required frames - the base temporal layer frames - can be tracked.

The TID and SID fields indicate the temporal and spatial layers and can help middleboxes and endpoints quickly identify which layer a packet belongs to.

**TID:** The temporal layer ID of current frame. In the case of non-flexible mode, if PID is mapped to a picture in a specified PG, then the value of TID MUST match the corresponding TID value of the mapped picture in the PG.

**U:** Switching up point. If this bit is set to 1 for the current picture with temporal layer ID equal to TID, then "switch up" to a higher frame rate is possible as subsequent higher temporal layer pictures will not depend on any picture before

the current picture (in coding order) with temporal layer ID greater than TID.

SID: The spatial layer ID of current frame. Note that frames with spatial layer SDI > 0 may be dependent on decoded spatial layer SID-1 frame within the same picture. Different frames of the same picture MUST have distinct spatial layer IDs, and frames' spatial layers MUST appear in increasing order within the frame.

D: Inter-layer dependency used. MUST be set to one if current spatial layer SID frame depends on spatial layer SID-1 frame of the same picture. MUST only be set to zero if current spatial layer SID frame does not depend on spatial layer SID-1 frame of the same picture. For the base layer frame (with SID equal to 0), this D bit MUST be set to zero.

TLOPICIDX: 8 bits temporal layer zero index. TLOPICIDX is only present in the non-flexible mode (F = 0). This is a running index for the temporal base layer pictures, i.e., the pictures with TID set to 0. If TID is larger than 0, TLOPICIDX indicates which temporal base layer picture the current picture depends on. TLOPICIDX MUST be incremented when TID is equal to 0. The index SHOULD start on a random number, and MUST restart at 0 after reaching the maximum number 255.

Reference indices: When P and F are both set to one, indicating a non-key frame in flexible mode, then at least one reference index has to be specified as below. Additional reference indices (total of up to 3 reference indices are allowed) may be specified using the N bit below. When either P or F is set to zero, then no reference index is specified.

P\_DIFF: The reference index (in 7 bits) specified as the relative PID from the current picture. For example, when P\_DIFF=3 on a packet containing the picture with PID 112 means that the picture refers back to the picture with PID 109. This calculation is done modulo the size of the PID field, i.e., either 7 or 15 bits.

N: 1 if there is additional P\_DIFF following the current P\_DIFF.

#### 4.2.1. Scalability Structure (SS):

The scalability structure (SS) data describes the resolution of each frame within a picture as well as the inter-picture dependencies for a picture group (PG). If the VP9 payload descriptor's "V" bit is

set, the SS data is present in the position indicated in Figure 2 and Figure 3.

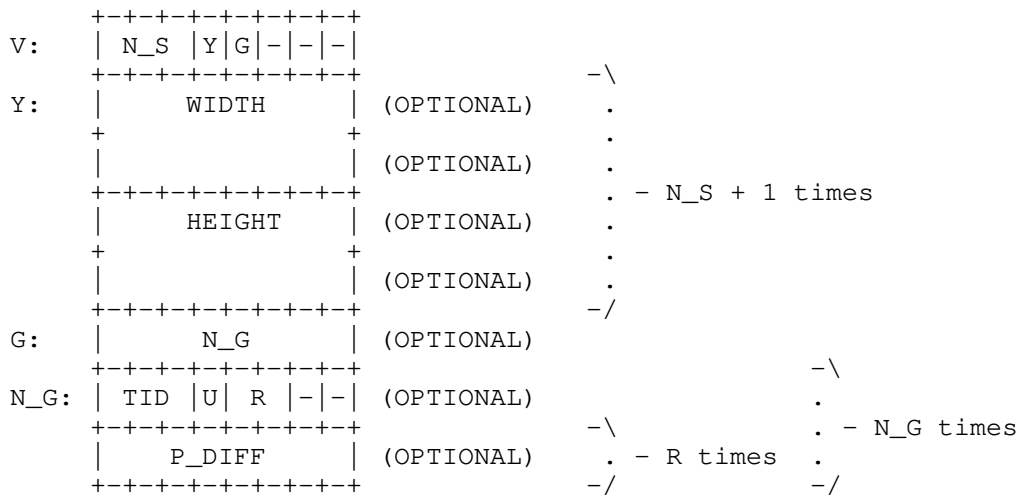


Figure 4

N\_S: N\_S + 1 indicates the number of spatial layers present in the VP9 stream.

Y: Each spatial layer's frame resolution present. When set to one, the OPTIONAL WIDTH (2 octets) and HEIGHT (2 octets) MUST be present for each layer frame. Otherwise, the resolution MUST NOT be present.

G: PG description present flag.

-: Bit reserved for future use. MUST be set to zero and MUST be ignored by the receiver.

N\_G: N\_G indicates the number of pictures in a Picture Group (PG). If N\_G is greater than 0, then the SS data allows the inter-picture dependency structure of the VP9 stream to be pre-declared, rather than indicating it on the fly with every packet. If N\_G is greater than 0, then for N\_G pictures in the PG, each picture's temporal layer ID (TID), switch up point (U), and the R reference indices (P\_DIFFs) are specified.

The first picture specified in the PG MUST have TID set to 0.

G set to 0 or N\_G set to 0 indicates that either there is only one temporal layer or no fixed inter-picture dependency information is present going forward in the bitstream.

Note that for a given picture, all frames follow the same inter-picture dependency structure. However, the frame rate of each spatial layer can be different from each other and this can be controlled with the use of the D bit described above. The specified dependency structure in the SS data **MUST** be for the highest frame rate layer.

In a scalable stream sent with a fixed pattern, the SS data **SHOULD** be included in the first packet of every key frame. This is a packet with P bit equal to zero, SID or D bit equal to zero, and B bit equal to 1. The SS data **MUST** only be changed on the picture that corresponds to the first picture specified in the previous SS data's PG (if the previous SS data's N\_G was greater than 0).

#### 4.3. VP9 Payload Header

TODO: need to describe VP9 payload header.

#### 4.4. Frame Fragmentation

VP9 frames are fragmented into packets, in RTP sequence number order, beginning with a packet with the B bit set, and ending with a packet with the E bit set. There is no mechanism for finer-grained access to parts of a VP9 frame.

#### 4.5. Scalable encoding considerations

In addition to the use of reference frames, VP9 has several additional forms of inter-frame dependencies, largely involving probability tables for the entropy and tree encoders. In VP9 syntax, the syntax element "error\_resilient\_mode" resets this additional inter-frame data, allowing a frame's syntax to be decoded independently.

Due to the requirements of scalable streams, a VP9 encoder producing a scalable stream needs to ensure that a frame does not depend on a previous frame (of the same or a previous picture) that can legitimately be removed from the stream. Thus, a frame that follows a removable frame (in full decode order) **MUST** be encoded with "error\_resilient\_mode" to true.

For spatially-scalable streams, this means that "error\_resilient\_mode" needs to be turned on for the base spatial layer; it can however be turned off for higher spatial layers,

assuming they are sent with inter-layer dependency (i.e. with the "D" bit set). For streams that are only temporally-scalable without spatial scalability, "error\_resilient\_mode" can additionally be turned off for any picture that immediately follows a temporal layer 0 frame.

#### 4.6. Examples of VP9 RTP Stream

TODO: Examples of packet layouts

##### 4.6.1. Reference picture use for scalable structure

As discussed in Section 3, the VP9 codec can maintain up to eight reference frames, of which up to three can be referenced or updated by any new frame. This section illustrates one way that a scalable structure (with three spatial layers and three temporal layers) can be constructed using these reference frames.

Temporal	Spatial	References	Updates
0	0	0	0
0	1	0, 1	1
0	2	1, 2	2
2	0	0	6
2	1	1, 6	7
2	2	2, 7	–
1	0	0	3
1	1	1, 3	4
1	2	2, 4	5
2	0	3	6
2	1	4, 6	7
2	2	5, 7	–

Example scalability structure



This structure is constructed such that the "U" bit can always be set.

## 5. Feedback Messages and Header Extensions

### 5.1. Reference Picture Selection Indication (RPSI)

The reference picture selection index is a payload-specific feedback message defined within the RTCP-based feedback format. The RPSI message is generated by a receiver and can be used in two ways. Either it can signal a preferred reference picture when a loss has been detected by the decoder -- preferably then a reference that the decoder knows is perfect -- or, it can be used as positive feedback information to acknowledge correct decoding of certain reference pictures. The positive feedback method is useful for VP9 used for point to point (unicast) communication. The use of RPSI for VP9 is preferably combined with a special update pattern of the codec's two special reference frames -- the golden frame and the altref frame -- in which they are updated in an alternating leapfrog fashion. When a receiver has received and correctly decoded a golden or altref frame, and that frame had a PictureID in the payload descriptor, the receiver can acknowledge this simply by sending an RPSI message back to the sender. The message body (i.e., the "native RPSI bit string" in [RFC4585]) is simply the PictureID of the received frame.

Note: because all frames of the same picture must have the same inter-picture reference structure, there is no need for a message to specify which frame is being selected.

### 5.2. Full Intra Request (FIR)

The Full Intra Request (FIR) [RFC5104] RTCP feedback message allows a receiver to request a full state refresh of an encoded stream.

Upon receipt of an FIR request, a VP9 sender MUST send a picture with a keyframe for its spatial layer 0 layer frame, and then send frames without inter-picture prediction (P=0) for any higher layer frames.

### 5.3. Layer Refresh Request (LRR)

The Layer Refresh Request [I-D.ietf-avtext-lrr] allows a receiver to request a single layer of a spatially or temporally encoded stream to be refreshed, without necessarily affecting the stream's other layers.

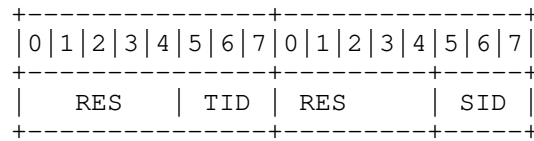


Figure 5

Figure 5 shows the format of LRR's layer index fields for VP9 streams. The two "RES" fields MUST be set to 0 on transmission and ignored on reception. See Section 4.2 for details on the TID and SID fields.

Identification of a layer refresh frame can be derived from the reference IDs of each frame by backtracking the dependency chain until reaching a point where only decodable frames are being referenced. Therefore it's recommended for both the flexible and the non-flexible mode that, when upgrade frames are being encoded in response to a LRR, those packets should contain layer indices and the reference fields so that the decoder or an MCU can make this derivation.

Example:

LRR {1,0}, {2,1} is sent by an MCU when it is currently relaying {1,0} to a receiver and which wants to upgrade to {2,1}. In response the encoder should encode the next frames in layers {1,1} and {2,1} by only referring to frames in {1,0}, or {0,0}.

In the non-flexible mode, periodic upgrade frames can be defined by the layer structure of the SS, thus periodic upgrade frames can be automatically identified by the picture ID.

#### 5.4. Frame Marking

The Frame Marking RTP header extension [I-D.ietf-avtext-framemarking] is a mechanism to provide information about frames of video streams in a largely codec-independent manner. However, for its extension for scalable codecs, the specific manner in which codec layers are identified needs to be specified specifically for each codec. This section defines how frame marking is used with VP9.

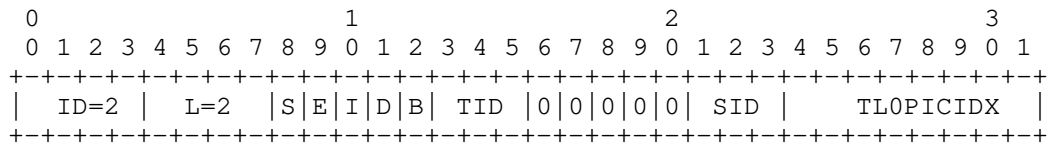


Figure 6

When this header extension is used with VP9, the TID and SID fields MUST match the values in the packet which the header extension is attached to; see Section 4.2 for details on these fields.

See [I-D.ietf-avtext-framemarking] for explanations of the other fields, which are generic.

## 6. Payload Format Parameters

This payload format has two optional parameters.

### 6.1. Media Type Definition

This registration is done using the template defined in [RFC6838] and following [RFC4855].

Type name: video

Subtype name: VP9

Required parameters: None.

Optional parameters:

These parameters are used to signal the capabilities of a receiver implementation. If the implementation is willing to receive media, both parameters MUST be provided. These parameters MUST NOT be used for any other purpose.

**max-fr:** The value of max-fr is an integer indicating the maximum frame rate in units of frames per second that the decoder is capable of decoding.

**max-fs:** The value of max-fs is an integer indicating the maximum frame size in units of macroblocks that the decoder is capable of decoding.

The decoder is capable of decoding this frame size as long as the width and height of the frame in macroblocks are less than  $\text{int}(\text{sqrt}(\text{max-fs} * 8))$  – for instance, a max-fs of 1200 (capable

of supporting 640x480 resolution) will support widths and heights up to 1552 pixels (97 macroblocks).

profile-id: The value of profile-id is an integer indicating the default coding profile, the subset of coding tools that may have been used to generate the stream or that the receiver supports). Table 1 lists all of the profiles defined in section 7.2 of [VP9-BITSTREAM] and the corresponding integer values to be used.

If no profile-id is present, Profile 0 MUST be inferred.

Informative note: See Table 2 for capabilities of coding profiles defined in section 7.2 of [VP9-BITSTREAM].

Encoding considerations:

This media type is framed in RTP and contains binary data; see Section 4.8 of [RFC6838].

Security considerations: See Section 7 of RFC xxxx.

[RFC Editor: Upon publication as an RFC, please replace "XXXX" with the number assigned to this document and remove this note.]

Interoperability considerations: None.

Published specification: VP9 bitstream format [VP9-BITSTREAM] and RFC XXXX.

[RFC Editor: Upon publication as an RFC, please replace "XXXX" with the number assigned to this document and remove this note.]

Applications which use this media type:

For example: Video over IP, video conferencing.

Fragment identifier considerations: N/A.

Additional information: None.

Person & email address to contact for further information:

TODO [Pick a contact]

Intended usage: COMMON

Restrictions on usage:

This media type depends on RTP framing, and hence is only defined for transfer via RTP [RFC3550].

Author: TODO [Pick a contact]

Change controller:

IETF Payload Working Group delegated from the IESG.

Profile	profile-id
0	0
1	1
2	2
3	3

Table 1: Table 1. Table of profile-id integer values representing the VP9 profile corresponding to the set of coding tools supported.

Profile	Bit Depth	SRGB Colorspace	Chroma Subsampling
0	8	No	YUV 4:2:0
1	8	Yes	YUV 4:2:0, 4:4:0 or 4:4:4
2	10 or 12	No	YUV 4:2:0
3	10 or 12	Yes	YUV 4:2:0, 4:4:0 or 4:4:4

Table 2: Table 2. Table of profile capabilities.

## 6.2. SDP Parameters

The receiver MUST ignore any fmtp parameter unspecified in this memo.

### 6.2.1. Mapping of Media Subtype Parameters to SDP

The media type video/VP9 string is mapped to fields in the Session Description Protocol (SDP) [RFC4566] as follows:

- o The media name in the "m=" line of SDP MUST be video.
- o The encoding name in the "a=rtpmap" line of SDP MUST be VP9 (the media subtype).
- o The clock rate in the "a=rtpmap" line MUST be 90000.

- o The parameters "max-fs", and "max-fr", MUST be included in the "a=fmtp" line of SDP if SDP is used to declare receiver capabilities. These parameters are expressed as a media subtype string, in the form of a semicolon separated list of parameter=value pairs.
- o The OPTIONAL parameter profile-id, when present, SHOULD be included in the "a=fmtp" line of SDP. This parameter is expressed as a media subtype string, in the form of a parameter=value pair. When the parameter is not present, a value of 0 MUST be used for profile-id.

#### 6.2.1.1. Example

An example of media representation in SDP is as follows:

```
m=video 49170 RTP/AVPF 98
a=rtpmap:98 VP9/90000
a=fmtp:98 max-fr=30; max-fs=3600; profile-id=0;
```

#### 6.2.2. Offer/Answer Considerations

When VP9 is offered over RTP using SDP in an Offer/Answer model [RFC3264] for negotiation for unicast usage, the following limitations and rules apply:

- o The parameter identifying a media format configuration for VP9 is profile-id. This media format configuration parameter MUST be used symmetrically; that is, the answerer MUST either maintain all configuration parameters or remove the media format (payload type) completely if one or more of the parameter values are not supported.
- o To simplify the handling and matching of these configurations, the same RTP payload type number used in the offer SHOULD also be used in the answer, as specified in [RFC3264]. An answer MUST NOT contain the payload type number used in the offer unless the configuration is exactly the same as in the offer.

### 7. Security Considerations

RTP packets using the payload format defined in this specification are subject to the security considerations discussed in the RTP specification [RFC3550], and in any applicable RTP profile such as RTP/AVP [RFC3551], RTP/AVPF [RFC4585], RTP/SAVP [RFC3711], or RTP/SAVPF [RFC5124]. SAVPF [RFC5124]. However, as "Securing the RTP Protocol Framework: Why RTP Does Not Mandate a Single Media Security Solution" [RFC7202] discusses, it is not an RTP payload format's

responsibility to discuss or mandate what solutions are used to meet the basic security goals like confidentiality, integrity and source authenticity for RTP in general. This responsibility lays on anyone using RTP in an application. They can find guidance on available security mechanisms in Options for Securing RTP Sessions [RFC7201]. Applications SHOULD use one or more appropriate strong security mechanisms. The rest of this security consideration section discusses the security impacting properties of the payload format itself.

This RTP payload format and its media decoder do not exhibit any significant non-uniformity in the receiver-side computational complexity for packet processing, and thus are unlikely to pose a denial-of-service threat due to the receipt of pathological data. Nor does the RTP payload format contain any active content.

## 8. Congestion Control

Congestion control for RTP SHALL be used in accordance with RFC 3550 [RFC3550], and with any applicable RTP profile; e.g., RFC 3551 [RFC3551]. The congestion control mechanism can, in a real-time encoding scenario, adapt the transmission rate by instructing the encoder to encode at a certain target rate. Media aware network elements MAY use the information in the VP9 payload descriptor in Section 4.2 to identify non-reference frames and discard them in order to reduce network congestion. Note that discarding of non-reference frames cannot be done if the stream is encrypted (because the non-reference marker is encrypted).

## 9. IANA Considerations

The IANA is requested to register the following values:  
- Media type registration as described in Section 6.1.

## 10. Acknowledgments

Alex Eleftheriadis, Yuki Ito, Won Kap Jang, Sergio Garcia Murillo, Roi Sasson, Timothy Terriberry, Emircan Uysaler, and Thomas Volkert commented on the development of this document and provided helpful comments and feedback.

## 11. References

### 11.1. Normative References

- [I-D.ietf-avtext-framemarking]  
Zanaty, M., Berger, E., and S. Nandakumar, "Frame Marking RTP Header Extension", draft-ietf-avtext-framemarking-09 (work in progress), March 2019.
- [I-D.ietf-avtext-lrr]  
Lennox, J., Hong, D., Uberti, J., Holmer, S., and M. Flodman, "The Layer Refresh Request (LRR) RTCP Feedback Message", draft-ietf-avtext-lrr-07 (work in progress), July 2017.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, DOI 10.17487/RFC3264, June 2002, <<https://www.rfc-editor.org/info/rfc3264>>.
- [RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, DOI 10.17487/RFC3550, July 2003, <<https://www.rfc-editor.org/info/rfc3550>>.
- [RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, DOI 10.17487/RFC4566, July 2006, <<https://www.rfc-editor.org/info/rfc4566>>.
- [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, DOI 10.17487/RFC4585, July 2006, <<https://www.rfc-editor.org/info/rfc4585>>.
- [RFC4855] Casner, S., "Media Type Registration of RTP Payload Formats", RFC 4855, DOI 10.17487/RFC4855, February 2007, <<https://www.rfc-editor.org/info/rfc4855>>.
- [RFC5104] Wenger, S., Chandra, U., Westerlund, M., and B. Burman, "Codec Control Messages in the RTP Audio-Visual Profile with Feedback (AVPF)", RFC 5104, DOI 10.17487/RFC5104, February 2008, <<https://www.rfc-editor.org/info/rfc5104>>.



- [RFC6838] Freed, N., Klensin, J., and T. Hansen, "Media Type Specifications and Registration Procedures", BCP 13, RFC 6838, DOI 10.17487/RFC6838, January 2013, <<https://www.rfc-editor.org/info/rfc6838>>.
- [VP9-BITSTREAM]  
Grange, A., de Rivaz, P., and J. Hunt, "VP9 Bitstream & Decoding Process Specification", Version 0.6, March 2016, <<https://storage.googleapis.com/downloads.webmproject.org/docs/vp9/vp9-bitstream-specification-v0.6-20160331-draft.pdf>>.

## 11.2. Informative References

- [RFC3551] Schulzrinne, H. and S. Casner, "RTP Profile for Audio and Video Conferences with Minimal Control", STD 65, RFC 3551, DOI 10.17487/RFC3551, July 2003, <<https://www.rfc-editor.org/info/rfc3551>>.
- [RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, DOI 10.17487/RFC3711, March 2004, <<https://www.rfc-editor.org/info/rfc3711>>.
- [RFC5124] Ott, J. and E. Carrara, "Extended Secure RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/SAVPF)", RFC 5124, DOI 10.17487/RFC5124, February 2008, <<https://www.rfc-editor.org/info/rfc5124>>.
- [RFC7201] Westerlund, M. and C. Perkins, "Options for Securing RTP Sessions", RFC 7201, DOI 10.17487/RFC7201, April 2014, <<https://www.rfc-editor.org/info/rfc7201>>.
- [RFC7202] Perkins, C. and M. Westerlund, "Securing the RTP Framework: Why RTP Does Not Mandate a Single Media Security Solution", RFC 7202, DOI 10.17487/RFC7202, April 2014, <<https://www.rfc-editor.org/info/rfc7202>>.

## Authors' Addresses

Justin Uberti  
Google, Inc.  
747 6th Street South  
Kirkland, WA 98033  
USA

Email: [justin@uberti.name](mailto:justin@uberti.name)

Stefan Holmer  
Google, Inc.  
Kungsbron 2  
Stockholm 111 22  
Sweden

Email: holmer@google.com

Magnus Flodman  
Google, Inc.  
Kungsbron 2  
Stockholm 111 22  
Sweden

Email: mflodman@google.com

Jonathan Lennox  
8x8, Inc. / Jitsi  
1350 Broadway  
New York, NY 10018  
US

Email: jonathan.lennox@8x8.com

Danny Hong  
Vidyo, Inc.  
433 Hackensack Avenue  
Seventh Floor  
Hackensack, NJ 07601  
US

Email: danny@vidyo.com

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: March 26, 2020

S. Zhao  
S. Wenger  
Tencent  
September 23, 2019

RTP Payload Format for Versatile Video Coding (VVC)  
draft-zhao-avtcore-rtp-vvc-00

Abstract

This memo describes an RTP payload format for the video coding standard ITU-T Recommendation H.266 and ISO/IEC International Standard 23090-3, both also known as Versatile Video Coding (VVC) and developed by the Joint Video Experts Team (JVET). The RTP payload format allows for packetization of one or more Network Abstraction Layer (NAL) units in each RTP packet payload as well as fragmentation of a NAL unit into multiple RTP packets. The payload format has wide applicability in videoconferencing, Internet video streaming, and high-bitrate entertainment-quality video, among others.

Status of This Memo

This Internet-Draft is submitted 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 <https://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 March 26, 2020.

Copyright Notice

Copyright (c) 2019 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 (<https://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
1.1. Overview of the VVC Codec . . . . .	3
1.1.1. Coding-Tool Features (informative) . . . . .	3
1.1.2. Systems and Transport Interfaces . . . . .	6
1.1.3. Parallel Processing Support (informative) . . . . .	10
1.1.4. NAL Unit Header . . . . .	10
1.2. Overview of the Payload Format . . . . .	11
2. Conventions . . . . .	11
3. Definitions and Abbreviations . . . . .	12
3.1. Definitions . . . . .	12
3.1.1. Definitions from the VVC Specification . . . . .	12
3.1.2. Definitions Specific to This Memo . . . . .	12
3.2. Abbreviations . . . . .	12
4. RTP Payload Format . . . . .	12
4.1. RTP Header Usage . . . . .	12
4.2. Payload Header Usage . . . . .	14
4.3. Payload Structures . . . . .	15
4.3.1. Single NAL Unit Packets . . . . .	15
4.3.2. Aggregation Packets (APs) . . . . .	16
4.3.3. Fragmentation Units . . . . .	21
4.4. Decoding Order Number . . . . .	24
5. Packetization Rules . . . . .	25
6. De-packetization Process . . . . .	26
7. Payload Format Parameters . . . . .	28
8. Use with Feedback Messages . . . . .	28
8.1. Picture Loss Indication (PLI) . . . . .	28
8.2. Slice Loss Indication (SLI) . . . . .	29
8.3. Reference Picture Selection Indication (RPSI) . . . . .	29
8.4. Full Intra Request (FIR) . . . . .	29
9. Security Considerations . . . . .	30
10. Congestion Control . . . . .	31
11. IANA Considerations . . . . .	32
12. Acknowledgements . . . . .	32
13. References . . . . .	32
13.1. Normative References . . . . .	32
13.2. Informative References . . . . .	34
Appendix A. Change History . . . . .	35
Authors' Addresses . . . . .	35

## 1. Introduction

The VVC specification, formally published as both ITU-T Recommendation H.266 and ISO/IEC International Standard 23090-23 [ISO23090-3], is planned for ratification in mid 2020. A draft that's currently in the approval process of ISO/IEC can be found as [VVC]. H.266 is reported to provide significant coding efficiency gains over H.265 [H.265] and earlier video codec formats.

This memo describes an RTP payload format for [VVC]. It shares its basic design with the NAL unit-based RTP payload formats of [RFC7798], [RFC6184] and [RFC6190]. With respect to design philosophy, security, congestion control, and overall implementation complexity, it has similar properties to those earlier payload format specifications. This is a conscious choice, as at least RFC 6184 is widely deployed and generally known in the relevant implementer communities. Certain mechanisms known from RFC 6190 were incorporated as [VVC] version 1 supports all temporal, spatial, and SNR scalability.

### 1.1. Overview of the VVC Codec

[VVC] and H.265 share a similar hybrid video codec design. In this memo, we provide a very brief overview of those features of [VVC] that are, in some form, addressed by the payload format specified herein. Implementers have to read, understand, and apply the ITU-T/ISO/IEC specifications pertaining to [VVC] to arrive at interoperable, well-performing implementations.

Conceptually, both [VVC] and HEVC include a Video Coding Layer (VCL), which is often used to refer to the coding-tool features, and a Network Abstraction Layer (NAL), which is often used to refer to the systems and transport interface aspects of the codecs.

#### 1.1.1. Coding-Tool Features (informative)

Coding tool features are described below with occasional reference to the coding tool set of HEVC, which is believed to be well known in the community.

Similar to earlier hybrid-video-coding-based standards, including HEVC, the following basic video coding design is employed by [VVC]. A prediction signal is first formed by either intra- or motion-compensated prediction, and the residual (the difference between the original and the prediction) is then coded. The gains in coding efficiency are achieved by redesigning and improving almost all parts of the codec over earlier designs. In addition, VVC includes several tools to make the implementation on parallel architectures easier.

Finally, VVC includes temporal, spatial, and SNR scalability as well as multiview coding support.

#### Coding blocks and transform structure

Among major coding-tool differences between HEVC and [VVC], one of the important improvements is the more flexible coding tree structure in VVC, i.e., multi-type tree. In addition to quadtree, binary and ternary trees are also supported, which contributes significant improvement in coding efficiency. Moreover, the maximum size of Coding Tree Unit (CTU) is increased from 64x64 to 128x128. To improve the coding efficiency of chroma signal, luma chroma separated trees at CTU level may be employed for intra-slices. As to transform, the square transforms in HEVC are extended to non-square transforms for rectangular blocks resulted from binary and ternary tree splits. Besides, [VVC] supports multiple transform sets (MTS), including DCT-2, DST-7, and DCT-8 as well as the non-separable secondary transform. The transforms used in [VVC] can have different sizes with support for larger transform sizes. For DCT-2, the transform sizes range from 2x2 to 64x64, and for DST-7 and DCT-8, the transform sizes range from 4x4 to 32x32. In addition, [VVC] also support sub-block transform for both intra and inter coded blocks. For intra coded blocks, intra sub-partitioning (ISP) may be used to allow sub-block based intra prediction and transform. For inter blocks, sub-block transform may be used assuming that only a part of an inter-block has non-zero transform coefficients.

#### Entropy coding

Similar to HEVC, [VVC] uses a single entropy-coding engine, which is based on Context Adaptive Binary Arithmetic Coding (CABAC) [CABAC], but with the support of multi-window sizes. The window sizes can be initialized differently for different context models. Due to such a design, it has more efficient adaptation speed and better coding efficiency. A joint chroma residual coding scheme is applied to further exploit the correlation between the residuals of two colour components. In [VVC], different residual coding schemes are applied for regular transform coefficients and residual samples generated using transform-skip mode.

#### In-loop filtering

[VVC] has more feature supports in loop filters than HEVC. The deblocking filter in [VVC] is similar to HEVC but operates at a smaller grid. After deblocking and sample adaptive offset (SAO), an adaptive loop filter (ALF) may be used. As a Wiener filter, ALF reduces distortion of decoded pictures. Besides, [VVC] introduces a new module before deblocking called luma mapping with chroma scaling

to fully utilize the dynamic range of signal so that rate-distortion performance of both SDR and HDR content is improved.

#### Motion prediction and coding

Compared to HEVC, [VVC] introduces several improvements in this area. First, there is the Adaptive motion vector resolution (AMVR), which can save bit cost for motion vectors by adaptively signaling motion vector resolution. Then the Affine motion compensation is included to capture complicated motion like zooming and rotation. Meanwhile, prediction refinement with the optical flow with affine mode (PROF) is further deployed to mimic affine motion at the pixel level. Thirdly the decoder side motion vector refinement (DMVR) is a method to derive MV vector at decoder side so that fewer bits may be spent on motion vectors. Bi-directional optical flow (BDOF) is a similar method to DMVR but at 4x4 sub-block level. Another difference is that DMVR is based on block matching while BDOF derives MVs with equations. Furthermore, merge with motion vector difference (MMVD) is a special mode, which further signals a limited set of motion vector differences on top of merge mode. In addition to MMVD, there are another three types of special merge modes, i.e., sub-block merge, triangle, and combined intra-/inter- prediction (CIIP). Sub-block merge list includes one candidate of sub-block temporal motion vector prediction (SbTMVP) and up to four candidates of affine motion vectors. Triangle is based on triangular block motion compensation. CIIP combines intra- and inter- predictions with weighting. Moreover, weighting in bi-prediction has more flexibility than HEVC. Adaptive weighting may be employed with a block-level tool called bi-prediction with CU based weighting (BCW).

#### Intra prediction and intra-coding

To capture the diversified local image texture directions with finer granularity, [VVC] supports 65 angular directions instead of 33 directions in HEVC. The intra mode coding is based on a 6 most probable mode scheme, and the 6 most probable modes are derived using the neighboring intra prediction directions. In addition, to deal with the different distributions of intra prediction angles for different block aspect ratios, a wide-angle intra prediction (WAIP) scheme is applied in [VVC] by including intra prediction angles beyond those present in HEVC. Unlike HEVC which only allows using the most adjacent line of reference samples for intra prediction, [VVC] also allows using two further reference lines, as known as multi-reference-line (MRL) intra prediction. The additional reference lines can be only used for 6 most probable intra prediction modes. To capture the strong correlation between different colour components, in [VVC], a cross-component linear mode (CCLM) is utilized which assumes a linear relationship between the luma sample

values and their associated chroma samples. For intra prediction, [VVC] also applies a position-dependent prediction combination (PDPC) for refining the prediction samples closer to the intra prediction block boundary. Matrix-based intra-prediction (MIP) modes are also used in [VVC] which generates an up to 8x8 intra prediction block using a weighted sum of downsampled neighboring reference samples, and the weightings are hardcoded constants.

#### Other coding-tool feature

[VVC] introduces dependent quantization (DQ) to reduce quantization error by state-based switching between two quantizers.

#### 1.1.2. Systems and Transport Interfaces

[VVC] inherits the basic systems and transport interfaces designs from HEVC and H.264. These include the NAL-unit-based syntax structure, the hierarchical syntax and data unit structure, the Supplemental Enhancement Information (SEI) message mechanism, and the video buffering model based on the Hypothetical Reference Decoder (HRD). The scalability features of [VVC] are conceptually similar to the scalable variant of HEVC known as SHVC. The hierarchical syntax and data unit structure consists of parameter sets at various levels (decoder, sequence (including layers), sequence (per layer), picture), slice-level header parameters, and lower-level parameters.

Below described are a number of key components that influenced the Network Abstraction Layer design of VVC as well as this memo.

#### Decoder parameter set

The Decoder parameter set includes parameters that stay constant for the lifetime of a Video Bitstream, which in IETF terms can translate to the lifetime of a session. Decoder parameter sets can include profile, level, and sub-profile information to determine a maximum complexity interop point that is guaranteed to be never exceeded, even if splicing of video sequences occurs within a session. It further optionally includes constraint flags, which indicate that the video bitstream will be constraint of the use of certain features as indicated by the values of those flags. With this, a bitstream can be labelled as not using certain tools, which allows among other things for resource allocation in a decoder implementation. As all parameter sets, also the decoder parameter set is required to be present when first referenced, and it is necessarily referenced by the very first picture in a video sequence, implying that it has to be sent among the first NAL units in the bitstream (see section xxx below). While multiple DPSs can be in the bitstream, the value of



the syntax elements therein cannot be inconsistent when being referenced.

#### Video parameter set

The Video Parameter Set (VPS) includes decoding dependency or information for reference picture set construction of enhancement layers. The VPS provides a "big picture" of a scalable sequence, including what types of operation points are provided, the profile, tier, and level of the operation points, and some other high-level properties of the bitstream that can be used as the basis for session negotiation and content selection, etc. (see Section xxx).

#### Sequence parameter set

The Sequence Parameter Set (SPS) contains syntax elements pertaining to a coded video sequence (CVS), which is a group of pictures, starting with a random access point, and followed by pictures that may depend on each other and the random access point picture. In MPEG-2, the equivalent of a CVS was a Group of Pictures (GOP), which normally started with an I frame and was followed by P and B frames. While more complex in its options of random access points, [VVC] retains this basic concept. In many TV-like applications, a CVS contains a few hundred milliseconds to a few seconds of video. In video conferencing (without switching MCUs involved), a CVS can be as long in duration as the whole session.

#### Picture and Adaptation parameter set

The Picture Parameter Set and the Adaptation Parameter Set (PPS and APS, respectively) carry information pertaining to a single picture. The PPS contains information that is likely to stay constant from picture to picture—at least for pictures for a certain type—whereas the APS contains information, such as adaptive loop filter coefficients, that are likely to change from picture to picture.

#### Profile, tier, and level

The profile, tier, and level syntax structure can be included in all DPS, VPS, and SPS. Somewhat oversimplified, they can be viewed to provide information about maximum bitstream complexity in the dimensions of tools used (profile), sample count (level), and maximum bitrate (tier). Level and tier are onion shaped, in that a decoder that can decode a certain level or tier can also decode lower levels or tiers. Profiles are not necessarily onion shaped and do not necessarily form a hierarchy. Therefore, the profile\_tier\_level structure in the video bitstream contains a bitmask which allows an encoder to mark a bitstream to be compatible with multiple profiles.

### Sub-Profiles

Within the [VVC] specification, a sub-profile is simply a 32 bit number coded according to ITU-T Rec. T.35, that does not carry a semantic. It is carried in the `profile_tier_level` structure and hence (potentially) present in the DPS, VPS, and SPS. External registration bodies can register a T.35 codepoint with ITU-T registration authorities and associate with their registration a description of bitstream complexity restrictions beyond the profiles defined by ITU-T and ISO/IEC. This would allow encoder manufacturers to label the bitstreams generated by their encoder as complying with such sub-profile. It is expected that upstream standardization organizations (such as: DVB and ATSC), as well as large walled-garden video services will take advantage of this labelling system. In contrast to "normal" profiles, it is expected that sub-profiles may indicate encoder choices traditionally left open in the (decoder-centric) video coding specs, such as GOP structures, minimum/maximum QP values, and the mandatory use of certain tools or SEI messages.

### Constraint Flags

The `profile_tier_level` structure optionally carries a considerable number of constraint flags, which an encoder can use to indicate to a decoder that it will not use a certain tool or technology. They were included in reaction to a perceived market need for labelling a bitstream as not exercising a certain tool that has become commercially unviable.

### Temporal scalability support

Edt. note: this section may need adjustment as JVET work on bitstream extraction is in progress.

[VVC] includes support of temporal scalability, by inclusion of the signaling of `TemporalId` in the NAL unit header, the restriction that pictures of a particular temporal sub-layer cannot be used for inter prediction reference by pictures of a lower temporal sub-layer, the sub-bitstream extraction process, and the requirement that each sub-bitstream extraction output be a conforming bitstream. Media-Aware Network Elements (MANEs) can utilize the `TemporalId` in the NAL unit header for stream adaptation purposes based on temporal scalability.

### Spatial, SNR, View Scalability

[VVC] includes support for spatial, SNR, and View scalability. Scalable video coding is widely considered to have technical benefits and enrich services for various video applications. Until recently, however, the functionality has not been included in the main profiles

of video codecs and not wide deployed due to additional costs. In VVC, however, all those forms of scalability are supported natively through the signaling of the `layer_id` in the NAL unit header, the VPS which associates layers with given `layer_ids` to each other, reference picture selection, reference picture resampling for spatial scalability, and a number of other mechanisms not relevant for this memo. Scalability support can be implemented in a single decoding "loop" and is widely considered a comparatively lightweight operation.

#### Spatial Scalability

With the existence of Reference Picture Resampling, likely in the "main" profile of VVC, the additional burden for scalability support is just a minor modification of the high-level syntax (HLS). In technical aspects, the inter-layer prediction is employed in a scalable system to improve the coding efficiency of the enhancement layers. In addition to the spatial and temporal motion-compensated predictions that are available in a single-layer codec, the inter-layer prediction in [VVC] uses the resampled video data of the reconstructed reference picture from a reference layer to predict the current enhancement layer. Then, the resampling process for inter-layer prediction is performed at the block-level, by modifying the existing interpolation process for motion compensation. It means that no additional resampling process is needed to support scalability.

#### SNR Scalability>

SNR scalability is similar to Spatial Scalability except that the resampling factors are 1:1--in other words, there is no change in resolution, but there is inter-layer prediction.

#### View Scalability>

#### Placeholder

#### SEI Messages

Supplementary Enhancement Information (SEI) messages are codepoints in the bitstream that do not influence the decoding process as specified in the [VVC] spec, but address issues of representation/rendering of the decoded bitstream, label the bitstream for certain applications, among other, similar tasks. The overall concept of SEI messages and many of the messages themselves has been inherited from the H.264 and HEVC specs. In the [VVC] environment, some of the SEI messages considered to be generally useful also in other video coding technologies have been moved out of the main specification into a

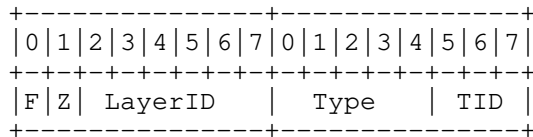
companion document (TO DO: add reference once ITU designation is known).

#### 1.1.3. Parallel Processing Support (informative)

Compared to HEVC [RFC7798], the [VVC] design to support parallelization offers numerous improvements. Some of those improvements are still undergoing changes in JVET. Information, to the extent relevant for this memo, will be added in future versions of this memo as the standardization in JVET progresses and the technology stabilizes.

#### 1.1.4. NAL Unit Header

[VVC] maintains the NAL unit concept of HEVC with modifications. VVC uses a two-byte NAL unit header, as shown in Figure 1. The payload of a NAL unit refers to the NAL unit excluding the NAL unit header.



The Structure of the [VVC] NAL Unit Header.

Figure 1

The semantics of the fields in the NAL unit header are as specified in [VVC] and described briefly below for convenience. In addition to the name and size of each field, the corresponding syntax element name in [VVC] is also provided.

F: 1 bit

forbidden\_zero\_bit. Required to be zero in [VVC]. Note that the inclusion of this bit in the NAL unit header was to enable transport of [VVC] video over MPEG-2 transport systems (avoidance of start code emulations) [MPEG2S]. In the context of this memo the value 1 may be used to indicate a syntax violation, e.g., for a NAL unit resulted from aggregating a number of fragmented units of a NAL unit but missing the last fragment, as described in Section TBD.

Z: 1 bit

nuh\_reserved\_zero\_bit. Required to be zero in [VVC], and reserved for future extensions by ITU-T and ISO/IEC. This memo does not

overload the "Z" bit for local extensions, as a) overloading the "F" bit is sufficient and b) to preserve the usefulness of this memo to possible future versions of [VVC].

LayerId: 6 bits

nuh\_layer\_id. Identifies the layer a NAL unit belongs to, wherein a layer may be, e.g., a spatial scalable layer, a quality scalable layer .

Type: 6 bits

nal\_unit\_type. This field specifies the NAL unit type as defined in Table 7-1 of [VVC]. For a reference of all currently defined NAL unit types and their semantics, please refer to Section 7.4.2.2 in [VVC].

TID: 3 bits

nuh\_temporal\_id\_plus1. This field specifies the temporal identifier of the NAL unit plus 1. The value of TemporalId is equal to TID minus 1. A TID value of 0 is illegal to ensure that there is at least one bit in the NAL unit header equal to 1, so to enable independent considerations of start code emulations in the NAL unit header and in the NAL unit payload data.

## 1.2. Overview of the Payload Format

This payload format defines the following processes required for transport of [VVC] coded data over RTP [RFC3550]:

- o Usage of RTP header with this payload format
- o Packetization of [VVC] coded NAL units into RTP packets using three types of payload structures: a single NAL unit packet, aggregation packet, and fragment unit
- o Transmission of HEVC NAL units of the same bitstream within a single RTP stream.
- o Media type parameters to be used with the Session Description Protocol (SDP) [RFC4566]

## 2. Conventions

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 BCP 14 [RFC2119]. In

this document, the above key words will convey that interpretation only when in ALL CAPS. Lowercase uses of these words are not to be interpreted as carrying the significance described in RFC 2119. This specification uses the notion of setting and clearing a bit when bit fields are handled. Setting a bit is the same as assigning that bit the value of 1 (On). Clearing a bit is the same as assigning that bit the value of 0 (Off).

### 3. Definitions and Abbreviations

#### 3.1. Definitions

This document uses the terms and definitions of [VVC]. Section 3.1.1 lists relevant definitions from [VVC] for convenience. Section 3.1.2 provides definitions specific to this memo.

##### 3.1.1. Definitions from the VVC Specification

Placeholder

##### 3.1.2. Definitions Specific to This Memo

Placeholder

#### 3.2. Abbreviations

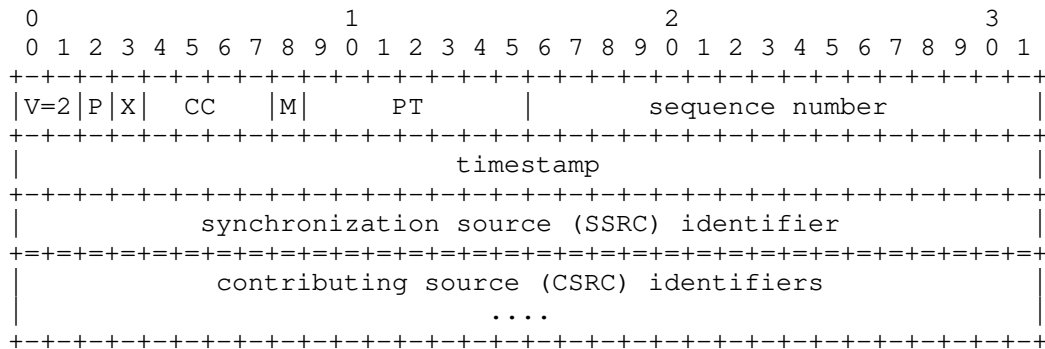
Placeholder

### 4. RTP Payload Format

#### 4.1. RTP Header Usage

The format of the RTP header is specified in [RFC3550] (reprinted as Figure 2 for convenience). This payload format uses the fields of the header in a manner consistent with that specification.

The RTP payload (and the settings for some RTP header bits) for aggregation packets and fragmentation units are specified in Sections 4.4.2 and 4.4.3, respectively.



RTP Header According to [RFC3550]

Figure 2

The RTP header information to be set according to this RTP payload format is set as follows:

Marker bit (M): 1 bit

Set for the last packet of the access unit, carried in the current RTP stream. This is in line with the normal use of the M bit in video formats to allow an efficient playout buffer handling.

The informative note below needs updating once the NAL unit type table is stable in the [VVC] spec

Informative note: The content of a NAL unit does not tell whether or not the NAL unit is the last NAL unit, in decoding order, of an access unit. An RTP sender implementation may obtain this information from the video encoder. If, however, the implementation cannot obtain this information directly from the encoder, e.g., when the bitstream was pre-encoded, and also there is no timestamp allocated for each NAL unit, then the sender implementation can inspect subsequent NAL units in decoding order to determine whether or not the NAL unit is the last NAL unit of an access unit as follows. A NAL unit is determined to be the last NAL unit of an access unit if it is the last NAL unit of the bitstream. A NAL unit nalux is also determined to be the last NAL unit of an access unit if both the following conditions are true: 1) the next VCL NAL unit naluy in decoding order has the high-order bit of the first byte after its NAL unit header equal to 1, and 2) all NAL units between nalux and naluy, when present, have nal\_unit\_type in the range of 32 to 35, inclusive, equal to 39, or in the ranges of 41 to 44, inclusive, or 48 to 55, inclusive.

Payload Type (PT): 7 bits

The assignment of an RTP payload type for this new packet format is outside the scope of this document and will not be specified here. The assignment of a payload type has to be performed either through the profile used or in a dynamic way.

Sequence Number (SN): 16 bits

Set and used in accordance with [RFC3550] .

Timestamp: 32 bits

The RTP timestamp is set to the sampling timestamp of the content. A 90 kHz clock rate MUST be used. If the NAL unit has no timing properties of its own (e.g., parameter set and SEI NAL units), the RTP timestamp MUST be set to the RTP timestamp of the coded picture of the access unit in which the NAL unit (according to Section xxx of [VVC]) is included. Receivers MUST use the RTP timestamp for the display process, even when the bitstream contains picture timing SEI messages or decoding unit information SEI messages as specified in [VVC]. However, this does not mean that picture timing SEI messages in the bitstream should be discarded, as picture timing SEI messages may contain frame-field information that is important in appropriately rendering interlaced video.

Synchronization source (SSRC): 32 bits

Used to identify the source of the RTP packets. When using SRST, by definition a single SSRC is used for all parts of a single bitstream.

#### 4.2. Payload Header Usage

The first two bytes of the payload of an RTP packet are referred to as the payload header. The payload header consists of the same fields (F, Z, LayerId, Type, and TID) as the NAL unit header as shown in Section 1.1.4, irrespective of the type of the payload structure.

The TID value indicates (among other things) the relative importance of an RTP packet, for example, because NAL units belonging to higher temporal sub-layers are not used for the decoding of lower temporal sub-layers. A lower value of TID indicates a higher importance. More-important NAL units MAY be better protected against transmission losses than less-important NAL units.



For Discussion: quite possibly something similar can be said for the Layer\_id in layered coding, but perhaps not in multiview coding. (The relevant part of the spec is relatively new, therefore the soft language). However, for serious layer pruning, interpretation of the VPS is required. We can add language about the need for starteful interpretation of LayerID vis-a-vis stateless interpretation of TID later.

#### 4.3. Payload Structures

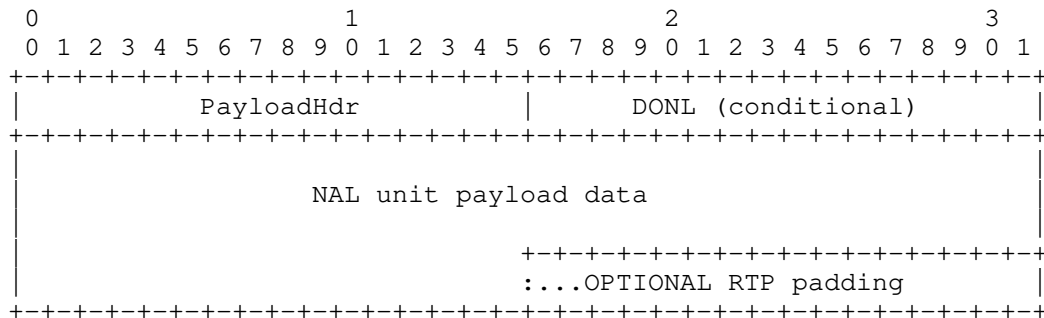
Four different types of RTP packet payload structures are specified. A receiver can identify the type of an RTP packet payload through the Type field in the payload header.

The four different payload structures are as follows:

- o Single NAL unit packet: Contains a single NAL unit in the payload, and the NAL unit header of the NAL unit also serves as the payload header. This payload structure is specified in Section 4.4.1.
- o Aggregation Packet (AP): Contains more than one NAL unit within one access unit. This payload structure is specified in Section 4.4.2.
- o Fragmentation Unit (FU): Contains a subset of a single NAL unit. This payload structure is specified in Section 4.4.3.

##### 4.3.1. Single NAL Unit Packets

A single NAL unit packet contains exactly one NAL unit, and consists of a payload header (denoted as PayloadHdr), a conditional 16-bit DONL field (in network byte order), and the NAL unit payload data (the NAL unit excluding its NAL unit header) of the contained NAL unit, as shown in Figure 3.



## The Structure of a Single NAL Unit Packet

Figure 3

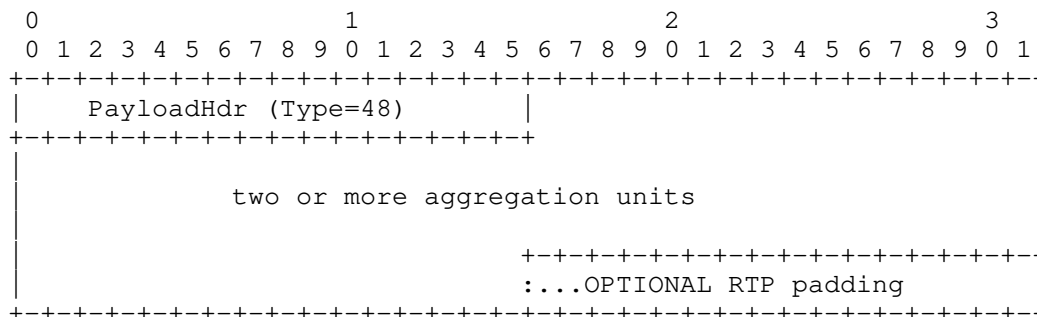
The DONL field, when present, specifies the value of the 16 least significant bits of the decoding order number of the contained NAL unit. If `sprop-max-don-diff` is greater than 0 for any of the RTP streams, the DONL field MUST be present, and the variable DON for the contained NAL unit is derived as equal to the value of the DONL field. Otherwise (`sprop-max-don-diff` is equal to 0 for all the RTP streams), the DONL field MUST NOT be present.

#### 4.3.2. Aggregation Packets (APs)

Aggregation Packets (APs) are introduced to enable the reduction of packetization overhead for small NAL units, such as most of the non-VCL NAL units, which are often only a few octets in size.

An AP aggregates NAL units within one access unit. Each NAL unit to be carried in an AP is encapsulated in an aggregation unit. NAL units aggregated in one AP are in NAL unit decoding order.

An AP consists of a payload header (denoted as PayloadHdr) followed by two or more aggregation units, as shown in Figure 4.



## The Structure of an Aggregation Packet

Figure 4

The fields in the payload header are set as follows. The F bit MUST be equal to 0 if the F bit of each aggregated NAL unit is equal to zero; otherwise, it MUST be equal to 1. The Type field MUST be equal to 48.

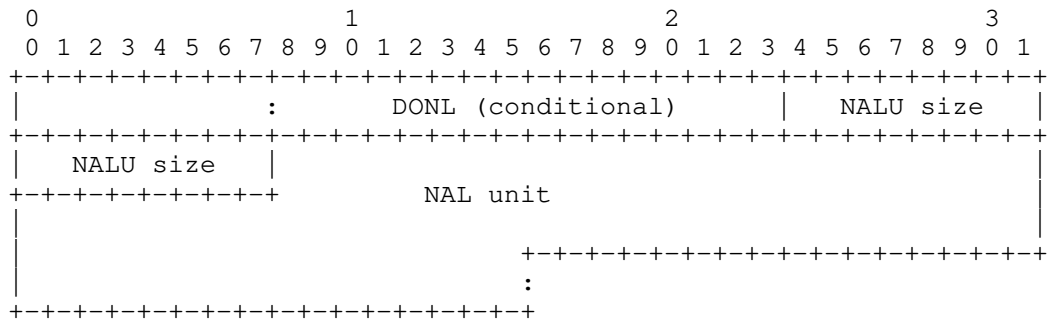
NOTE: double check #48 against post-geneva [VVC] spec

The value of LayerId MUST be equal to the lowest value of LayerId of all the aggregated NAL units. The value of TID MUST be the lowest value of TID of all the aggregated NAL units.

Informative note: All VCL NAL units in an AP have the same TID value since they belong to the same access unit. However, an AP may contain non-VCL NAL units for which the TID value in the NAL unit header may be different than the TID value of the VCL NAL units in the same AP.

An AP MUST carry at least two aggregation units and can carry as many aggregation units as necessary; however, the total amount of data in an AP obviously MUST fit into an IP packet, and the size SHOULD be chosen so that the resulting IP packet is smaller than the MTU size so to avoid IP layer fragmentation. An AP MUST NOT contain FUs specified in Section 4.4.3. APs MUST NOT be nested; i.e., an AP must not contain another AP.

The first aggregation unit in an AP consists of a conditional 16-bit DONL field (in network byte order) followed by a 16-bit unsigned size information (in network byte order) that indicates the size of the NAL unit in bytes (excluding these two octets, but including the NAL unit header), followed by the NAL unit itself, including its NAL unit header, as shown in Figure 5.



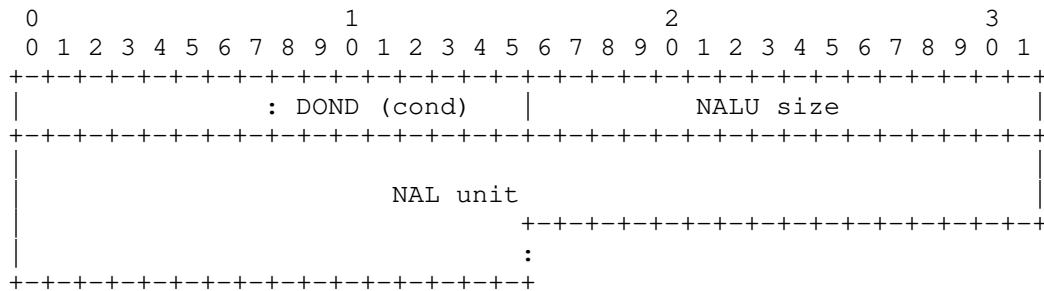
The Structure of the First Aggregation Unit in an AP

Figure 5

The DONL field, when present, specifies the value of the 16 least significant bits of the decoding order number of the aggregated NAL unit.

If `sprop-max-don-diff` is greater than 0 for any of the RTP streams, the DONL field MUST be present in an aggregation unit that is the first aggregation unit in an AP, and the variable DON for the aggregated NAL unit is derived as equal to the value of the DONL field. Otherwise (`sprop-max-don-diff` is equal to 0 for all the RTP streams), the DONL field MUST NOT be present in an aggregation unit that is the first aggregation unit in an AP.

An aggregation unit that is not the first aggregation unit in an AP consists of a conditional 8-bit DONL field followed by a 16-bit unsigned size information (in network byte order) that indicates the size of the NAL unit in bytes (excluding these two octets, but including the NAL unit header), followed by the NAL unit itself, including its NAL unit header, as shown in Figure 6.



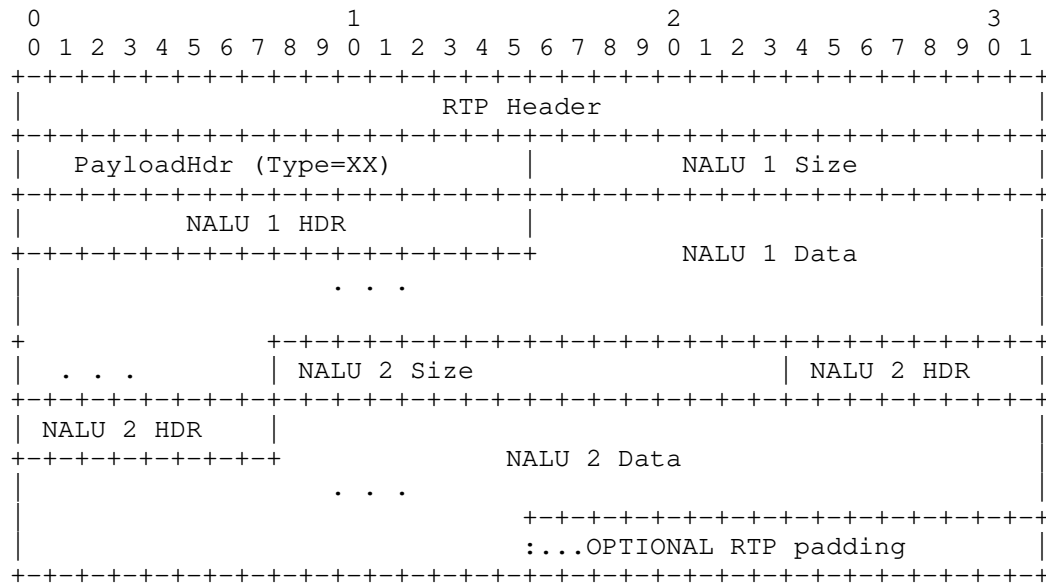
The Structure of an Aggregation Unit That Is Not the First  
Aggregation Unit in an AP

Figure 6

When present, the DOND field plus 1 specifies the difference between the decoding order number values of the current aggregated NAL unit and the preceding aggregated NAL unit in the same AP.

If `sprop-max-don-diff` is greater than 0 for any of the RTP streams, the DOND field MUST be present in an aggregation unit that is not the first aggregation unit in an AP, and the variable DON for the aggregated NAL unit is derived as equal to the DON of the preceding aggregated NAL unit in the same AP plus the value of the DOND field plus 1 modulo 65536. Otherwise (`sprop-max-don-diff` is equal to 0 for all the RTP streams), the DOND field MUST NOT be present in an aggregation unit that is not the first aggregation unit in an AP, and in this case the transmission order and decoding order of NAL units carried in the AP are the same as the order the NAL units appear in the AP.

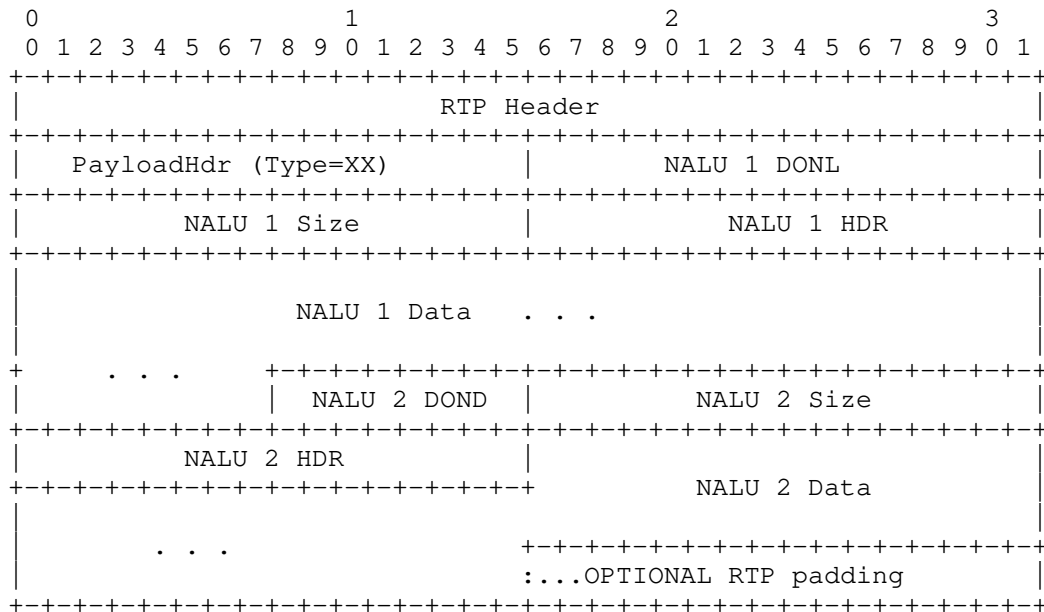
Figure 7 presents an example of an AP that contains two aggregation units, labeled as 1 and 2 in the figure, without the DONL and DOND fields being present.



An Example of an AP Packet Containing Two Aggregation Units without the DONL and DOND Fields

Figure 7

Figure 8 presents an example of an AP that contains two aggregation units, labeled as 1 and 2 in the figure, with the DONL and DOND fields being present.



### An Example of an AP Containing Two Aggregation Units with the DONL and DOND Fields

Figure 8

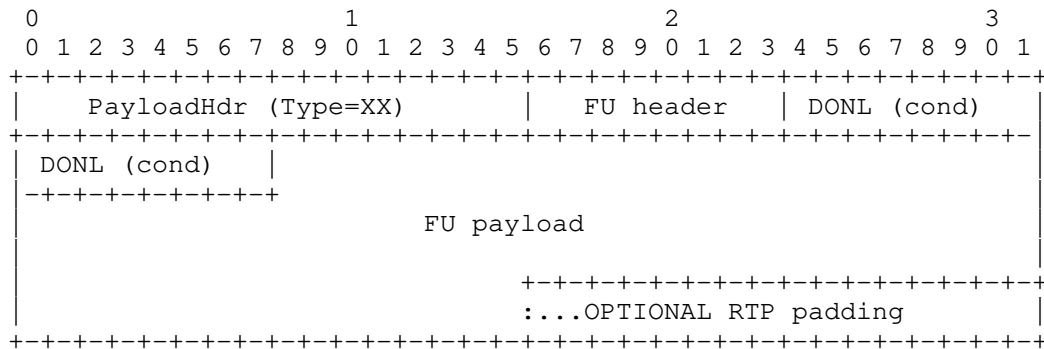
### 4.3.3. Fragmentation Units

Fragmentation Units (FUs) are introduced to enable fragmenting a single NAL unit into multiple RTP packets, possibly without cooperation or knowledge of the HEVC [RFC7798] encoder. A fragment of a NAL unit consists of an integer number of consecutive octets of that NAL unit. Fragments of the same NAL unit **MUST** be sent in consecutive order with ascending RTP sequence numbers (with no other RTP packets within the same RTP stream being sent between the first and last fragment).

When a NAL unit is fragmented and conveyed within FUs, it is referred to as a fragmented NAL unit. APs MUST NOT be fragmented. FUs MUST NOT be nested; i.e., an FU must not contain a subset of another FU.

The RTP timestamp of an RTP packet carrying an FU is set to the NALU-time of the fragmented NAL unit.

An FU consists of a payload header (denoted as PayloadHdr), an FU header of one octet, a conditional 16-bit DONL field (in network byte order), and an FU payload, as shown in Figure 9.

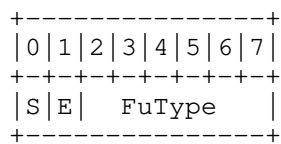


The Structure of an FU

Figure 9

The fields in the payload header are set as follows. The Type field MUST be equal to XX. The fields F, LayerId, and TID MUST be equal to the fields F, LayerId, and TID, respectively, of the fragmented NAL unit.

The FU header consists of an S bit, an E bit, and a 6-bit FuType field, as shown in Figure 10.



The Structure of FU Header

Figure 10

The semantics of the FU header fields are as follows:

S: 1 bit

When set to 1, the S bit indicates the start of a fragmented NAL unit, i.e., the first byte of the FU payload is also the first byte of the payload of the fragmented NAL unit. When the FU payload is not the start of the fragmented NAL unit payload, the S bit MUST be set to 0.

E: 1 bit



When set to 1, the E bit indicates the end of a fragmented NAL unit, i.e., the last byte of the payload is also the last byte of the fragmented NAL unit. When the FU payload is not the last fragment of a fragmented NAL unit, the E bit MUST be set to 0.

FuType: 6 bits

The field FuType MUST be equal to the field Type of the fragmented NAL unit.

The DONL field, when present, specifies the value of the 16 least significant bits of the decoding order number of the fragmented NAL unit.

If sprop-max-don-diff is greater than 0 for any of the RTP streams, and the S bit is equal to 1, the DONL field MUST be present in the FU, and the variable DON for the fragmented NAL unit is derived as equal to the value of the DONL field. Otherwise (sprop-max-don-diff is equal to 0 for all the RTP streams, or the S bit is equal to 0), the DONL field MUST NOT be present in the FU.

A non-fragmented NAL unit MUST NOT be transmitted in one FU; i.e., the Start bit and End bit must not both be set to 1 in the same FU header.

The FU payload consists of fragments of the payload of the fragmented NAL unit so that if the FU payloads of consecutive FUs, starting with an FU with the S bit equal to 1 and ending with an FU with the E bit equal to 1, are sequentially concatenated, the payload of the fragmented NAL unit can be reconstructed. The NAL unit header of the fragmented NAL unit is not included as such in the FU payload, but rather the information of the NAL unit header of the fragmented NAL unit is conveyed in F, LayerId, and TID fields of the FU payload headers of the FUs and the FuType field of the FU header of the FUs. An FU payload MUST NOT be empty.

If an FU is lost, the receiver SHOULD discard all following fragmentation units in transmission order corresponding to the same fragmented NAL unit, unless the decoder in the receiver is known to be prepared to gracefully handle incomplete NAL units.

A receiver in an endpoint or in a MANE MAY aggregate the first n-1 fragments of a NAL unit to an (incomplete) NAL unit, even if fragment n of that NAL unit is not received. In this case, the forbidden\_zero\_bit of the NAL unit MUST be set to 1 to indicate a syntax violation.

#### 4.4. Decoding Order Number

For each NAL unit, the variable AbsDon is derived, representing the decoding order number that is indicative of the NAL unit decoding order.

Let NAL unit  $n$  be the  $n$ -th NAL unit in transmission order within an RTP stream.

If `sprop-max-don-diff` is equal to 0 for all the RTP streams carrying the HEVC bitstream, `AbsDon[n]`, the value of AbsDon for NAL unit  $n$ , is derived as equal to  $n$ .

Otherwise (`sprop-max-don-diff` is greater than 0 for any of the RTP streams), `AbsDon[n]` is derived as follows, where `DON[n]` is the value of the variable DON for NAL unit  $n$ :

- o If  $n$  is equal to 0 (i.e., NAL unit  $n$  is the very first NAL unit in transmission order), `AbsDon[0]` is set equal to `DON[0]`.
- o Otherwise ( $n$  is greater than 0), the following applies for derivation of `AbsDon[n]`:

If `DON[n] == DON[n-1]`,  
    `AbsDon[n] = AbsDon[n-1]`

If (`DON[n] > DON[n-1]` and `DON[n] - DON[n-1] < 32768`),  
    `AbsDon[n] = AbsDon[n-1] + DON[n] - DON[n-1]`

If (`DON[n] < DON[n-1]` and `DON[n-1] - DON[n] >= 32768`),  
    `AbsDon[n] = AbsDon[n-1] + 65536 - DON[n-1] + DON[n]`

If (`DON[n] > DON[n-1]` and `DON[n] - DON[n-1] >= 32768`),  
    `AbsDon[n] = AbsDon[n-1] - (DON[n-1] + 65536 - DON[n])`

If (`DON[n] < DON[n-1]` and `DON[n-1] - DON[n] < 32768`),  
    `AbsDon[n] = AbsDon[n-1] - (DON[n-1] - DON[n])`

For any two NAL units  $m$  and  $n$ , the following applies:

- o `AbsDon[n]` greater than `AbsDon[m]` indicates that NAL unit  $n$  follows NAL unit  $m$  in NAL unit decoding order.
- o When `AbsDon[n]` is equal to `AbsDon[m]`, the NAL unit decoding order of the two NAL units can be in either order.

- o AbsDon[n] less than AbsDon[m] indicates that NAL unit n precedes NAL unit m in decoding order.

Informative note: When two consecutive NAL units in the NAL unit decoding order have different values of AbsDon, the absolute difference between the two AbsDon values may be greater than or equal to 1.

Informative note: There are multiple reasons to allow for the absolute difference of the values of AbsDon for two consecutive NAL units in the NAL unit decoding order to be greater than one. An increment by one is not required, as at the time of associating values of AbsDon to NAL units, it may not be known whether all NAL units are to be delivered to the receiver. For example, a gateway may not forward VCL NAL units of higher sub-layers or some SEI NAL units when there is congestion in the network. In another example, the first intra-coded picture of a pre-encoded clip is transmitted in advance to ensure that it is readily available in the receiver, and when transmitting the first intra-coded picture, the originator does not exactly know how many NAL units will be encoded before the first intra-coded picture of the pre-encoded clip follows in decoding order. Thus, the values of AbsDon for the NAL units of the first intra-coded picture of the pre-encoded clip have to be estimated when they are transmitted, and gaps in values of AbsDon may occur. Another example is MRST or MRMT with sprop-max-don-diff greater than 0, where the AbsDon values must indicate cross-layer decoding order for NAL units conveyed in all the RTP streams.

## 5. Packetization Rules

The following packetization rules apply:

- o If sprop-max-don-diff is greater than 0 for any of the RTP streams, the transmission order of NAL units carried in the RTP stream MAY be different than the NAL unit decoding order and the NAL unit output order. Otherwise (sprop-max-don-diff is equal to 0 for all the RTP streams), the transmission order of NAL units carried in the RTP stream MUST be the same as the NAL unit decoding order and, when tx-mode is equal to "MRST" or "MRMT", MUST also be the same as the NAL unit output order.
- o A NAL unit of a small size SHOULD be encapsulated in an aggregation packet together with one or more other NAL units in order to avoid the unnecessary packetization overhead for small NAL units. For example, non-VCL NAL units such as access unit delimiters, parameter sets, or SEI NAL units are typically small

and can often be aggregated with VCL NAL units without violating MTU size constraints.

- o Each non-VCL NAL unit SHOULD, when possible from an MTU size match viewpoint, be encapsulated in an aggregation packet together with its associated VCL NAL unit, as typically a non-VCL NAL unit would be meaningless without the associated VCL NAL unit being available.
- o For carrying exactly one NAL unit in an RTP packet, a single NAL unit packet MUST be used.

## 6. De-packetization Process

The general concept behind de-packetization is to get the NAL units out of the RTP packets in an RTP stream and all RTP streams the RTP stream depends on, if any, and pass them to the decoder in the NAL unit decoding order.

The de-packetization process is implementation dependent. Therefore, the following description should be seen as an example of a suitable implementation. Other schemes may be used as well, as long as the output for the same input is the same as the process described below. The output is the same when the set of output NAL units and their order are both identical. Optimizations relative to the described algorithms are possible.

All normal RTP mechanisms related to buffer management apply. In particular, duplicated or outdated RTP packets (as indicated by the RTP sequences number and the RTP timestamp) are removed. To determine the exact time for decoding, factors such as a possible intentional delay to allow for proper inter-stream synchronization must be factored in.

NAL units with NAL unit type values in the range of 0 to XX, inclusive, may be passed to the decoder. NAL-unit-like structures with NAL unit type values in the range of XX to XX, inclusive, MUST NOT be passed to the decoder.

The receiver includes a receiver buffer, which is used to compensate for transmission delay jitter within individual RTP streams and across RTP streams, to reorder NAL units from transmission order to the NAL unit decoding order, and to recover the NAL unit decoding order in MRST or MRMT, when applicable. In this section, the receiver operation is described under the assumption that there is no transmission delay jitter within an RTP stream and across RTP streams. To make a difference from a practical receiver buffer that is also used for compensation of transmission delay jitter, the

receiver buffer is hereafter called the de-packetization buffer in this section. Receivers should also prepare for transmission delay jitter; that is, either reserve separate buffers for transmission delay jitter buffering and de-packetization buffering or use a receiver buffer for both transmission delay jitter and de-packetization. Moreover, receivers should take transmission delay jitter into account in the buffering operation, e.g., by additional initial buffering before starting of decoding and playback.

When `sprop-max-don-diff` is equal to 0 for all the received RTP streams, the de-packetization buffer size is zero bytes, and the process described in the remainder of this paragraph applies. When there is only one RTP stream received, the NAL units carried in the single RTP stream are directly passed to the decoder in their transmission order, which is identical to their decoding order. When there is more than one RTP stream received, the NAL units carried in the multiple RTP streams are passed to the decoder in their NTP timestamp order. When there are several NAL units of different RTP streams with the same NTP timestamp, the order to pass them to the decoder is their dependency order, where NAL units of a dependee RTP stream are passed to the decoder prior to the NAL units of the dependent RTP stream. When there are several NAL units of the same RTP stream with the same NTP timestamp, the order to pass them to the decoder is their transmission order.

Informative note: The mapping between RTP and NTP timestamps is conveyed in RTCP SR packets. In addition, the mechanisms for faster media timestamp synchronization discussed in [RFC6051] may be used to speed up the acquisition of the RTP-to-wall-clock mapping.

When `sprop-max-don-diff` is greater than 0 for any the received RTP streams, the process described in the remainder of this section applies.

There are two buffering states in the receiver: initial buffering and buffering while playing. Initial buffering starts when the reception is initialized. After initial buffering, decoding and playback are started, and the buffering-while-playing mode is used.

Regardless of the buffering state, the receiver stores incoming NAL units, in reception order, into the de-packetization buffer. NAL units carried in RTP packets are stored in the de-packetization buffer individually, and the value of `AbsDon` is calculated and stored for each NAL unit. When `MRST` or `MRMT` is in use, NAL units of all RTP streams of a bitstream are stored in the same de-packetization buffer. When NAL units carried in any two RTP streams are available to be placed into the de-packetization buffer, those NAL units

carried in the RTP stream that is lower in the dependency tree are placed into the buffer first. For example, if RTP stream A depends on RTP stream B, then NAL units carried in RTP stream B are placed into the buffer first.

Initial buffering lasts until condition A (the difference between the greatest and smallest AbsDon values of the NAL units in the de-packetization buffer is greater than or equal to the value of sprop-max-don-diff of the highest RTP stream) or condition B (the number of NAL units in the de-packetization buffer is greater than the value of sprop-depack-buf-nalus) is true.

After initial buffering, whenever condition A or condition B is true, the following operation is repeatedly applied until both condition A and condition B become false:

- o The NAL unit in the de-packetization buffer with the smallest value of AbsDon is removed from the de-packetization buffer and passed to the decoder.

When no more NAL units are flowing into the de-packetization buffer, all NAL units remaining in the de-packetization buffer are removed from the buffer and passed to the decoder in the order of increasing AbsDon values.

## 7. Payload Format Parameters

Placeholder

## 8. Use with Feedback Messages

The following subsections define the use of the Picture Loss Indication (PLI), Slice Lost Indication (SLI), Reference Picture Selection Indication (RPSI), and Full Intra Request (FIR) feedback messages with HEVC. The PLI, SLI, and RPSI messages are defined in [RFC4585], and the FIR message is defined in [RFC5104].

### 8.1. Picture Loss Indication (PLI)

As specified in RFC 4585, Section 6.3.1, the reception of a PLI by a media sender indicates "the loss of an undefined amount of coded video data belonging to one or more pictures". Without having any specific knowledge of the setup of the bitstream (such as use and location of in-band parameter sets, non-IDR decoder refresh points, picture structures, and so forth), a reaction to the reception of an PLI by a [VVC] sender SHOULD be to send an IDR picture and relevant parameter sets; potentially with sufficient redundancy so to ensure correct reception. However, sometimes information about the

bitstream structure is known. For example, state could have been established outside of the mechanisms defined in this document that parameter sets are conveyed out of band only, and stay static for the duration of the session. In that case, it is obviously unnecessary to send them in-band as a result of the reception of a PLI. Other examples could be devised based on a priori knowledge of different aspects of the bitstream structure. In all cases, the timing and congestion control mechanisms of RFC 4585 MUST be observed.

#### 8.2. Slice Loss Indication (SLI)

For further study. Maybe remove as there are no known implementations of SDLI in H.265 based systems

#### 8.3. Reference Picture Selection Indication (RPSI)

Feedback-based reference picture selection has been shown as a powerful tool to stop temporal error propagation for improved error resilience [Girod99] [Wang05]. In one approach, the decoder side tracks errors in the decoded pictures and informs the encoder side that a particular picture that has been decoded relatively earlier is correct and still present in the decoded picture buffer; it requests the encoder to use that correct picture-availability information when encoding the next picture, so to stop further temporal error propagation. For this approach, the decoder side should use the RPSI feedback message.

Encoders can encode some long-term reference pictures as specified in [VVC] for purposes described in the previous paragraph without the need of a huge decoded picture buffer. As shown in [Wang05], with a flexible reference picture management scheme, as in [VVC], even a decoded picture buffer size of two picture storage buffers would work for the approach described in the previous paragraph.

the text below is copy-paste from RFC 7798. If we keep the RPSI message, it needs adaptation to the [VVC] syntax. Doing so shouldn't be too hard as the [VVC] reference picture mechanism is not too different from the H.265 one.

#### 8.4. Full Intra Request (FIR)

The purpose of the FIR message is to force an encoder to send an independent decoder refresh point as soon as possible (observing, for example, the congestion-control-related constraints set out in RFC 5104).

Upon reception of a FIR, a sender MUST send an IDR picture. Parameter sets MUST also be sent, except when there is a priori

knowledge that the parameter sets have been correctly established. A typical example for that is an understanding between sender and receiver, established by means outside this document, that parameter sets are exclusively sent out-of-band.

## 9. Security Considerations

The scope of this Security Considerations section is limited to the payload format itself and to one feature of [VVC] that may pose a particularly serious security risk if implemented naively. The payload format, in isolation, does not form a complete system. Implementers are advised to read and understand relevant security-related documents, especially those pertaining to RTP (see the Security Considerations section in [RFC3550] ), and the security of the call-control stack chosen (that may make use of the media type registration of this memo). Implementers should also consider known security vulnerabilities of video coding and decoding implementations in general and avoid those.

Within this RTP payload format, and with the exception of the user data SEI message as described below, no security threats other than those common to RTP payload formats are known. In other words, neither the various media-plane-based mechanisms, nor the signaling part of this memo, seems to pose a security risk beyond those common to all RTP-based systems.

RTP packets using the payload format defined in this specification are subject to the security considerations discussed in the RTP specification [RFC3550] , and in any applicable RTP profile such as RTP/AVP [RFC3551] , RTP/AVPF [RFC4585] , RTP/SAVP [RFC3711] , or RTP/SAVPF [RFC5124] . However, as "Securing the RTP Framework: Why RTP Does Not Mandate a Single Media Security Solution" [RFC7202] discusses, it is not an RTP payload format's responsibility to discuss or mandate what solutions are used to meet the basic security goals like confidentiality, integrity and source authenticity for RTP in general. This responsibility lays on anyone using RTP in an application. They can find guidance on available security mechanisms and important considerations in "Options for Securing RTP Sessions" [RFC7201] . Applications SHOULD use one or more appropriate strong security mechanisms. The rest of this section discusses the security impacting properties of the payload format itself.

Because the data compression used with this payload format is applied end-to-end, any encryption needs to be performed after compression. A potential denial-of-service threat exists for data encodings using compression techniques that have non-uniform receiver-end computational load. The attacker can inject pathological datagrams into the bitstream that are complex to decode and that cause the



receiver to be overloaded. [VVC] is particularly vulnerable to such attacks, as it is extremely simple to generate datagrams containing NAL units that affect the decoding process of many future NAL units. Therefore, the usage of data origin authentication and data integrity protection of at least the RTP packet is RECOMMENDED, for example, with SRTP [RFC3711] .

Like HEVC [RFC7798], [VVC] includes a user data Supplemental Enhancement Information (SEI) message. This SEI message allows inclusion of an arbitrary bitstring into the video bitstream. Such a bitstring could include JavaScript, machine code, and other active content. [VVC] leaves the handling of this SEI message to the receiving system. In order to avoid harmful side effects organization the user data SEI message, decoder implementations cannot naively trust its content. For example, it would be a bad and insecure implementation practice to forward any JavaScript a decoder implementation detects to a web browser. The safest way to deal with user data SEI messages is to simply discard them, but that can have negative side effects on the quality of experience by the user.

End-to-end security with authentication, integrity, or confidentiality protection will prevent a MANE from performing media-aware operations other than discarding complete packets. In the case of confidentiality protection, it will even be prevented from discarding packets in a media-aware way. To be allowed to perform such operations, a MANE is required to be a trusted entity that is included in the security context establishment.

#### 10. Congestion Control

Congestion control for RTP SHALL be used in accordance with RTP [RFC3550] and with any applicable RTP profile, e.g., AVP [RFC3551] . If best-effort service is being used, an additional requirement is that users of this payload format MUST monitor packet loss to ensure that the packet loss rate is within an acceptable range. Packet loss is considered acceptable if a TCP flow across the same network path, and experiencing the same network conditions, would achieve an average throughput, measured on a reasonable timescale, that is not less than all RTP streams combined is achieving. This condition can be satisfied by implementing congestion-control mechanisms to adapt the transmission rate, the number of layers subscribed for a layered multicast session, or by arranging for a receiver to leave the session if the loss rate is unacceptably high.

The bitrate adaptation necessary for obeying the congestion control principle is easily achievable when real-time encoding is used, for example, by adequately tuning the quantization parameter.

However, when pre-encoded content is being transmitted, bandwidth adaptation requires the pre-coded bitstream to be tailored for such adaptivity. The key mechanisms available in [VVC] are temporal scalability, and spatial/SNR scalability. A media sender can remove NAL units belonging to higher temporal sub-layers (i.e., those NAL units with a high value of TID) or higher spatio-SNR layers (as indicated by interpreting the VPS) until the sending bitrate drops to an acceptable range.

Above mechanisms generally work within a defined profile and level and, therefore, no renegotiation of the channel is required. Only when non-downgradable parameters (such as profile) are required to be changed does it become necessary to terminate and restart the RTP stream(s). This may be accomplished by using different RTP payload types.

MANES MAY remove certain unusable packets from the RTP stream when that RTP stream was damaged due to previous packet losses. This can help reduce the network load in certain special cases. For example, MANES can remove those FUs where the leading FUs belonging to the same NAL unit have been lost or those dependent slice segments when the leading slice segments belonging to the same slice have been lost, because the trailing FUs or dependent slice segments are meaningless to most decoders. MANES can also remove higher temporal scalable layers if the outbound transmission (from the MANE's viewpoint) experiences congestion.

## 11. IANA Considerations

Placeholder

## 12. Acknowledgements

Large parts of this specification share text with the RTP payload format for HEVC [RFC7798], RFC 7798. We thank the authors of that specification for their excellent work. We also thank BD Choi for his contribution towards the [VVC] descriptive text.

## 13. References

### 13.1. Normative References

[ISO23090-3]

ISO and IEC, "Versatile video coding -- not yet published", August 2020.

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, DOI 10.17487/RFC3550, July 2003, <<https://www.rfc-editor.org/info/rfc3550>>.
- [RFC3551] Schulzrinne, H. and S. Casner, "RTP Profile for Audio and Video Conferences with Minimal Control", STD 65, RFC 3551, DOI 10.17487/RFC3551, July 2003, <<https://www.rfc-editor.org/info/rfc3551>>.
- [RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, DOI 10.17487/RFC3711, March 2004, <<https://www.rfc-editor.org/info/rfc3711>>.
- [RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, DOI 10.17487/RFC4566, July 2006, <<https://www.rfc-editor.org/info/rfc4566>>.
- [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, DOI 10.17487/RFC4585, July 2006, <<https://www.rfc-editor.org/info/rfc4585>>.
- [RFC5104] Wenger, S., Chandra, U., Westerlund, M., and B. Burman, "Codec Control Messages in the RTP Audio-Visual Profile with Feedback (AVPF)", RFC 5104, DOI 10.17487/RFC5104, February 2008, <<https://www.rfc-editor.org/info/rfc5104>>.
- [RFC5124] Ott, J. and E. Carrara, "Extended Secure RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/SAVPF)", RFC 5124, DOI 10.17487/RFC5124, February 2008, <<https://www.rfc-editor.org/info/rfc5124>>.
- [VVC] ITU-T, "Versatile video coding - JVET-O2001-vE, available from [http://phenix.it-sudparis.eu/jvet/doc\\_end\\_user/documents/15\\_Gothenburg/wg11/JVET-O2001-v14.zip](http://phenix.it-sudparis.eu/jvet/doc_end_user/documents/15_Gothenburg/wg11/JVET-O2001-v14.zip)", August 2019.

## 13.2. Informative References

- [CABAC] Sole, J., Joshi, R., Nguyen, N., Ji, T., Karczewicz, M., Clare, G., Henry, F., and A. Duenas, "Transform coefficient coding in HEVC", IEEE Transactions on Circuits and Systems for Video Technology Vol. 22 No. 12 pp. 1765-1777, DOI 10.1109/TCSVT.2012.2223055, December 2012.
- [Girod99] Girod, B. and F. Faerber, "Feedback-based error control for mobile video transmission", Proceedings of the IEEE Vol. 87, No. 10, pp. 1707-1723, DOI 10.1109/5.790632, October 1999.
- [MPEG2S] ISO/IEC, "Information technology - Generic coding of moving pictures and associated audio information - Part 1: Systems", ISO International Standard 13818-1, 2013.
- [RFC6051] Perkins, C. and T. Schierl, "Rapid Synchronisation of RTP Flows", RFC 6051, DOI 10.17487/RFC6051, November 2010, <<https://www.rfc-editor.org/info/rfc6051>>.
- [RFC6184] Wang, Y., Even, R., Kristensen, T., and R. Jesup, "RTP Payload Format for H.264 Video", RFC 6184, DOI 10.17487/RFC6184, May 2011, <<https://www.rfc-editor.org/info/rfc6184>>.
- [RFC6190] Wenger, S., Wang, Y., Schierl, T., and A. Eleftheriadis, "RTP Payload Format for Scalable Video Coding", RFC 6190, DOI 10.17487/RFC6190, May 2011, <<https://www.rfc-editor.org/info/rfc6190>>.
- [RFC7201] Westerlund, M. and C. Perkins, "Options for Securing RTP Sessions", RFC 7201, DOI 10.17487/RFC7201, April 2014, <<https://www.rfc-editor.org/info/rfc7201>>.
- [RFC7202] Perkins, C. and M. Westerlund, "Securing the RTP Framework: Why RTP Does Not Mandate a Single Media Security Solution", RFC 7202, DOI 10.17487/RFC7202, April 2014, <<https://www.rfc-editor.org/info/rfc7202>>.
- [RFC7798] Wang, Y., Sanchez, Y., Schierl, T., Wenger, S., and M. Hannuksela, "RTP Payload Format for High Efficiency Video Coding (HEVC)", RFC 7798, DOI 10.17487/RFC7798, March 2016, <<https://www.rfc-editor.org/info/rfc7798>>.

Appendix A. Change History

draft-zhao-payload-rtp-vvc-00 ..... initial version

Authors' Addresses

Shuai Zhao  
Tencent  
2747 Park Blvd.  
Palo Alto, CA 94306  
US

Email: shuaiizhao@tencent.com

Stephan Wenger  
Tencent  
2747 Park Blvd.  
Palo Alto, CA 94306  
US

Email: stewe@stewe.org