

avtcore WG  
Internet-Draft  
Intended status: Informational  
Expires: 18 September 2025

E. Spr̃ng  
Google  
17 March 2025

RTCP Feedback Message and Request Mechanism for Frame-level  
Acknowledgement  
draft-spr̃ng-avtcore-frame-acknowledgement-00

## Abstract

This document describes a mechanism for signaling which video frames have been received and decoded by a remote peer. It comprises an RTCP feedback message and an RTP header extension used to request said feedback.

One of the main use cases for this data is to implement various forms of Long Term Reference (LTR) reference structures.

## About This Document

This note is to be removed before publishing as an RFC.

The latest revision of this draft can be found at <https://github.com/spr̃ng/rtcp-frame-acknowledgement/blob/main/draft-ietf-avtcore-frame-acknowledgement.md>. Status information for this document may be found at <https://datatracker.ietf.org/doc/draft-spr̃ng-avtcore-frame-acknowledgement/>.

Discussion of this document takes place on the avtcore WG Working Group mailing list (<mailto:avt@ietf.org>), which is archived at <https://datatracker.ietf.org/wg/avtcore>. Subscribe at <https://www.ietf.org/mailman/listinfo/avt/>.

Source for this draft and an issue tracker can be found at <https://github.com/spr̃ng/rtcp-frame-acknowledgement>.

## 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 18 September 2025.

## Copyright Notice

Copyright (c) 2025 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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

## Table of Contents

1. Introduction . . . . .	2
2. Conventions and Definitions . . . . .	3
3. Applicability . . . . .	3
4. Existing Feedback Formats . . . . .	4
5. Requirements . . . . .	4
6. Frame Acknowledgment . . . . .	5
6.1. Frame identifier selection . . . . .	5
6.2. Frame Acknowledgment Request . . . . .	5
6.2.1. Data layout overview . . . . .	6
6.3. Frame Acknowledgment . . . . .	6
6.3.1. Data layout overview . . . . .	6
7. Security Considerations . . . . .	7
8. IANA Considerations . . . . .	8
9. References . . . . .	8
9.1. Normative References . . . . .	8
9.2. Informative References . . . . .	8
Acknowledgments . . . . .	9
Author's Address . . . . .	9

## 1. Introduction

The most common way for realtime video to be transmitted is to encode a pretty much fixed scalability structure, such as those in the W3C [SVC] Scalability mode list.

In such a scenario, the video encoder produces frames "blindly" without real knowledge of what state the remote receiver is in. Using recovery mechanisms such as retransmission, forward error correction and fast-forwarding past skippable frames the receiver is assumed to be able to decode the video. In some cases those methods may not be enough, requiring keyframe requests to be sent as a last resort.

On the other hand, if the encoder is able to reason about which frames have been received and decoded it can be more proactive. One way is to store frames that are known to be received so that they can be later used as guaranteed good references in the case of e.g. large loss events, avoiding the need for potentially large retransmissions etc. Collectively this is often referred to as "Long Term Reference" structures or LTR for short, although the exact structure may vary.

In order to achieve this the sender must be able to reason about the state of the receiver, necessitating the need for feedback signals. In this document a new RTCP message called "Frame Acknowledgement" is introduced as a codec agnostic feedback message for this purpose. Further, an RTP header extension is introduced that allows the sender to actively request feedback on decoding of the associated frame. This allows the sender to both request quick feedback on frames that are important for latency, and enables resilience against loss of feedback packets.

Note that it is allowed to report a frame as decoded even if the decode process is not complete - as long as the receiver guarantees that it will attempt to decode the frame. The rationale for this is that we want to reduce the feedback delay as much as possible. Should the decoding of a frame that has been acknowledged fail, then the receiver **MUST** request a keyframe to recover, even if the failed decoding belongs to a droppable layer.

## 2. Conventions and Definitions

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.

## 3. Applicability

Frame Acknowledgement can be used for video streams in most topologies. It is also designed to be codec agnostic.

In terms of [RFC7667], Point-to-Point is the most straightforward target as it is easiest to reason about a single receiver. A Media Translator or other systems that include a decoder are similarly easy - from the perspective of the sender the middle box is the receiver.

If a Transport Translator is used for Point-to-Multi-Point, then the middlebox must make sure to make valid translations - e.g. by acknowledging a frame only if all recipients of that stream have acknowledged said frame.

#### 4. Existing Feedback Formats

This section provides an overview, for informational purposes, of some existing feedback formats that could be seen as alternatives.

NACK, defined in [RFC4585], provides only requests for packets the receiver is interested in having retransmitted. Absence of feedback is a poor signal for acknowledgement, especially since said feedback can be lost.

[RFC8888] and [TWCC] provide per-packet acknowledgement and so are more useful. A mapping from packet(s) to frame needs to happen but that is not a big problem. However, even if a frame is confirmed to be received there is no guarantee that it gets decoded.

Reference Picture Selection Indication (RPSI) is another existing message, but it puts the logic of requesting a particular reference frame in the receiver - significantly complicating the system especially in Point-to-Multi-Point systems. It is further codec specific, and several modern codecs lack a specification - including AV1 and H.266.

Loss Notification [LNTF] was a proposed RTCP message intended to solve most of these problems, but it lacks resilience against loss of feedback and also cannot handle out-of-order acknowledgements. The latter makes for instance single-SSRC simulcast structures (e.g. SxTx modes in [SVC]) impossible.

#### 5. Requirements

The messages in this proposal are intended to fulfill the following requirements:

1. Codec agnostic The protocol should be general enough to work across all current and future codecs.

2. Payload Invariant The protocol should not depend on data within the encoded bitstream payload. That includes codec specific frame identifiers, feedback requests and feedback messages.
3. Uses Frame Identifiers Explicit marking of frames, rather than using an indirection via packets.
4. Order Invariant The format should not make assumptions about the required decode order of frames.
5. Send-side Controlled The sender explicitly indicates when and for which frames feedback should be sent.
6. Loss Resilient The sender should be able to detect and recover from lost feedback messages.
7. Low Delay The latency should be small, with the sender being able to tune delay vs rate tradeoff.
8. Low Overhead The network overhead in terms of both packet rate and bitrate should be minimized.

## 6. Frame Acknowledgment

### 6.1. Frame identifier selection

In order to request and receive information about decoded frames, we must be able to identify them. Rather than adding new metadata for this purpose alone, we do that by picking the first available option from a list of available sources:

1. The frame\_number from a Dependency Descriptor header extension [DD]

Note: In this draft version, only a single source is allowed. Future versions may add other alternatives. Cases can be made for anything from a new dedicated identification system (similar to [VFTI]) to mappings from codec specific payload data.

### 6.2. Frame Acknowledgment Request

A Frame Acknowledgement Request is an RTP header extension indicating the oldest frame ID the sender is interested in receiving feedback for. The request MUST be done on the media SSRC of video frames in question. The request implies a status request for all frames starting at the given frame ID, up to and including the frame contained in the RTP packet the header extension is attached to - even if that frame is not yet complete. If the extension is attached

to a packet not containing a video frame, the feedback should be up to and including the immediately preceding frame ID.

Note that the Frame ID is a 16 bit counter with rollover, so e.g. a request with Frame ID = 65535 attached to a packet containing Frame ID = 1 is a request for the three frames {65535, 0, 1}.

If a new Frame Acknowledgement Request is sent with an incremented Frame ID, all status values prior to that Frame ID are considered as acknowledged and can be culled by the receiver. A sender MUST NOT request prior to either the last acknowledged Frame ID or start of the stream.

#### 6.2.1. Data layout overview

```

0                               1                               2
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| ID   | len=1 |                               Frame ID   |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

##### 6.2.1.1. Frame ID (16 bits)

The earliest Frame ID that feedback is requested for.

#### 6.3. Frame Acknowledgment

##### 6.3.1. Data layout overview

Short feedback message (L = 0):

```

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 | FMT=12 | PT = 205 | length |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                               SSRC of RTCP packet sender |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Start Frame ID | L | length | status + pad |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| ... |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Long feedback message (L = 1):

```

      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| FMT=12 | PT = 205 | length |
+-----+-----+-----+-----+-----+-----+-----+-----+
| SSRC of RTCP packet sender |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Start Frame ID |L| length |
+-----+-----+-----+-----+-----+-----+-----+-----+
| status + pad . . . |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

#### 6.3.1.1. Start Frame ID (16 bits)

The first Frame ID in this feedback.

#### 6.3.1.2. L (1 bit)

If the number frames in the feedback vector is < 128, then L = 0.  
Otherwise L = 1.

#### 6.3.1.3. length (7 bits or 15 bits)

An unsigned integer denoting how many consecutive frames this message contains feedback for. The last Frame ID is thus Frame ID + length - 1.

If L = 0, length is 7 bits, otherwise length is 15 bits.

#### 6.3.1.4. status (N bit)

A bit vector of the length specified in the length field above. For each bit position, the Frame ID is incremented by one.

A value of 0 indicates the frame has not been received and decoded.  
A value of 1 indicates the frame has been received and decoded.

### 7. Security Considerations

The messages in this proposal may expose a small amount of data, namely the number of frames that have been sent, and potentially in an indirect way which frames the sender sees as important for recovery.

This data should however not pose any significant privacy or security risks.

## 8. IANA Considerations

The RTP header extension needs to have a URI identifier assigned by IANA. See [IANAEXT].

The RTCP message uses PT = 205 (RTPFB, Generic RTP Feedback). As of writing, the next available FMT value is 12. A dedicated ID needs to be assigned by IANA. See [IANARTCP].

## 9. References

### 9.1. Normative References

- [DD] AOM, "Dependency Descriptor RTP Header Extension", n.d., <<https://aomediacodec.github.io/av1-rtp-spec/#dependency-descriptor-rtp-header-extension>>.
- [IANAEXT] IANA, "RTP Compact Header Extensions", n.d., <<https://www.iana.org/assignments/rtp-parameters/rtp-parameters.xhtml#rtp-parameters-10>>.
- [IANARTCP] IANA, "FMT Values for RTPFB Payload Types", n.d., <<https://www.iana.org/assignments/rtp-parameters/rtp-parameters.xhtml#rtp-parameters-4>>.
- [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/rfc/rfc2119>>.
- [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/rfc/rfc8174>>.

### 9.2. Informative References

- [LNTF] "RTCP feedback Message for Loss Notification", n.d., <<https://www.ietf.org/archive/id/draft-majali-avtcore-lntf-feedback-message-00.html>>.
- [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/rfc/rfc4585>>.

- [RFC7667] Westerlund, M. and S. Wenger, "RTP Topologies", RFC 7667, DOI 10.17487/RFC7667, November 2015, <<https://www.rfc-editor.org/rfc/rfc7667>>.
- [RFC8888] Sarker, Z., Perkins, C., Singh, V., and M. Ramalho, "RTP Control Protocol (RTCP) Feedback for Congestion Control", RFC 8888, DOI 10.17487/RFC8888, January 2021, <<https://www.rfc-editor.org/rfc/rfc8888>>.
- [SVC] W3C, "Scalable Video Coding (SVC) Extension for WebRTC", n.d., <<https://www.w3.org/TR/webrtc-svc>>.
- [TWCC] "RTP Extensions for Transport-wide Congestion Control", n.d., <<https://datatracker.ietf.org/doc/html/draft-holmer-rmcat-transport-wide-cc-extensions-01>>.
- [VFTI] "Video Frame Tracking Id", n.d., <<http://www.webrtc.org/experiments/rtp-hdrext/video-frame-tracking-id>>.

#### Acknowledgments

#### Author's Address

Erik Sprang  
Google  
Email: [sprang@google.com](mailto:sprang@google.com)