Network Working Group
Request for Comments: 4585
Category: Standards Track

J. Ott
Helsinki University of Technology
S. Wenger
Nokia
N. Sato
Oki
C. Burmeister
J. Rey
Matsushita

July 2006

Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/AVPF)

Status of This Memo

Copyright Notice

Abstract

Real-time media streams that use RTP are, to some degree, resilient against packet losses. Receivers may use the base mechanisms of the Real-time Transport Control Protocol (RTCP) to report packet reception statistics and thus allow a sender to adapt its transmission behavior in the mid-term. This is the sole means for feedback and feedback-based error repair (besides a few codec-specific mechanisms). This document defines an extension to the Audio-visual Profile (AVP) that enables receivers to provide, statistically, more immediate feedback to the senders and thus allows for short-term adaptation and efficient feedback-based repair mechanisms to be implemented. This early feedback profile (AVPF) maintains the AVP bandwidth constraints for RTCP and preserves scalability to large groups.

Table of Contents

1. Introduction

Real-time media streams that use RTP are, to some degree, resilient against packet losses. RTP [1] provides all the necessary mechanisms to restore ordering and timing present at the sender to properly reproduce a media stream at a recipient. RTP also provides continuous feedback about the overall reception quality from all receivers -- thereby allowing the sender(s) in the mid-term (in the order of several seconds to minutes) to adapt their coding scheme and transmission behavior to the observed network quality of service (QoS). However, except for a few payload-specific mechanisms [6], RTP makes no provision for timely feedback that would allow a sender to repair the media stream immediately: through retransmissions, retroactive Forward Error Correction (FEC) control, or media-specific mechanisms for some video codecs, such as reference picture selection.

Current mechanisms available with RTP to improve error resilience include audio redundancy coding [13], video redundancy coding [14], RTP-level FEC [11], and general considerations on more robust media streams transmission [12]. These mechanisms may be applied proactively (thereby increasing the bandwidth of a given media stream). Alternatively, in sufficiently small groups with small round-trip times (RTTs), the senders may perform repair on-demand, using the above mechanisms and/or media-encoding-specific approaches. Note that "small group" and "sufficiently small RTT" are both highly application dependent.

This document specifies a modified RTP profile for audio and video conferences with minimal control based upon [1] and [2] by means of two modifications/additions: Firstly, to achieve timely feedback, the concept of Early RTCP messages as well as algorithms allowing for low-delay feedback in small multicast groups (and preventing feedback implosion in large ones) are introduced. Special consideration is given to point-to-point scenarios. Secondly, a small number of general-purpose feedback messages as well as a format for codec- and application-specific feedback information are defined for transmission in the RTCP payloads.

1.1. Definitions

The definitions from RTP/RTCP [1] and the "RTP Profile for Audio and Video Conferences with Minimal Control" [2] apply. In addition, the following definitions are used in this document:

Early RTCP mode:
The mode of operation in that a receiver of a media stream is often (but not always) capable of reporting events of interest back to the sender close to their occurrence. In Early RTCP mode, RTCP packets are transmitted according to the timing rules defined in this document.

Early RTCP packet:
An Early RTCP packet is a packet which is transmitted earlier than would be allowed if following the scheduling algorithm of [1], the reason being an "event" observed by a receiver. Early RTCP packets may be sent in Immediate Feedback and in Early RTCP mode. Sending an Early RTCP packet is also referred to as sending Early Feedback in this document.

Event:
An observation made by the receiver of a media stream that is (potentially) of interest to the sender -- such as a packet loss or packet reception, frame loss, etc. -- and thus useful to be reported back to the sender by means of a feedback message.

Feedback (FB) message:
An RTCP message as defined in this document is used to convey information about events observed at a receiver -- in addition to long-term receiver status information that is carried in RTCP receiver reports (RRs) -- back to the sender of the media stream. For the sake of clarity, feedback message is referred to as FB message throughout this document.

Feedback (FB) threshold:
The FB threshold indicates the transition between Immediate Feedback and Early RTCP mode. For a multiparty scenario, the FB threshold indicates the maximum group size at which, on average, each receiver is able to report each event back to the sender(s) immediately, i.e., by means of an Early RTCP packet without having to wait for its regularly scheduled RTCP interval. This threshold is highly dependent on the type of feedback to be provided, network QoS (e.g., packet loss probability and distribution), codec and packetization scheme in use, the session bandwidth, and application requirements. Note that the algorithms do not depend on all senders and receivers agreeing on the same value for this threshold. It is merely intended to provide conceptual guidance to application designers and is not used in any calculations. For the sake of clarity, the term feedback threshold is referred to as FB threshold throughout this document.

Immediate Feedback mode:
A mode of operation in which each receiver of a media stream is, statistically, capable of reporting each event of interest immediately back to the media stream sender. In Immediate Feedback mode, RTCP FB messages are transmitted according to the timing rules defined in this document.

Media packet:
A media packet is an RTP packet.

Regular RTCP mode:
Mode of operation in which no preferred transmission of FB messages is allowed. Instead, RTCP messages are sent following the rules of [1]. Nevertheless, such RTCP messages may contain feedback information as defined in this document.

Regular RTCP packet:
An RTCP packet that is not sent as an Early RTCP packet.

RTP sender:
An RTP sender is an RTP entity that transmits media packets as well as RTCP packets and receives Regular as well as Early RTCP (i.e., feedback) packets. Note that the RTP sender is a logical role and that the same RTP entity may at the same time act as an RTP receiver.

RTP receiver:
An RTP receiver is an RTP entity that receives media packets as well as RTCP packets and transmits Regular as well as Early RTCP (i.e., feedback) packets. Note that the RTP receiver is a logical role and that the same RTP entity may at the same time act as an RTP sender.

1.2. 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 RFC 2119 [5].

2. RTP and RTCP Packet Formats and Protocol Behavior

2.1. RTP

The rules defined in [2] also apply to this profile except for those rules mentioned in the following:

RTCP packet types:
Two additional RTCP packet types are registered and the corresponding FB messages to convey feedback information are defined in Section 6 of this memo.

RTCP report intervals:
This document describes three modes of operation that influence the RTCP report intervals (see Section 3.2 of this memo). In Regular RTCP mode, all rules from [1] apply except for the recommended minimal interval of five seconds between two RTCP reports from the same RTP entity. In both Immediate Feedback and Early RTCP modes, the minimal interval of five seconds between two RTCP reports is dropped and, additionally, the rules specified in Section 3 of this memo apply if RTCP packets containing FB messages (defined in Section 4 of this memo) are to be transmitted.

The rules set forth in [1] may be overridden by session descriptions specifying different parameters (e.g., for the bandwidth share assigned to RTCP for senders and receivers, respectively). For sessions defined using the Session Description Protocol (SDP) [3], the rules of [4] apply.

Congestion control:
The same basic rules as detailed in [2] apply. Beyond this, in Section 7, further consideration is given to the impact of feedback and a sender’s reaction to FB messages.

2.2. Underlying Transport Protocols

RTP is intended to be used on top of unreliable transport protocols, including UDP and the Datagram Congestion Control Protocol (DCCP). This section briefly describes the specifics beyond plain RTP operation introduced by RTCP feedback as specified in this memo.

UDP: UDP provides best-effort delivery of datagrams for point-to-point as well as for multicast communications. UDP does not support congestion control or error repair. The RTCP-based feedback defined in this memo is able to provide minimal support for limited error repair. As RTCP feedback is not guaranteed to operate on sufficiently small timescales (in the order of RTT),

RTCP feedback is not suitable to support congestion control. This memo addresses both unicast and multicast operation.

DCCP: DCCP [19] provides for congestion-controlled but unreliable datagram flows for unicast communications. With TCP Friendly Rate Control (TFRC)-based [20] congestion control (CCID 3), DCCP is particularly suitable for audio and video communications. DCCP's acknowledgement messages may provide detailed feedback reporting about received and missed datagrams (and thus about congestion).

When running RTP over DCCP, congestion control is performed at the DCCP layer and no additional mechanisms are required at the RTP layer. Furthermore, an RTCP-feedback-capable sender may leverage the more frequent DCCP-based feedback and thus a receiver may refrain from using (additional) Generic Feedback messages where appropriate.

3. Rules for RTCP Feedback

3.1. Compound RTCP Feedback Packets

Two components constitute RTCP-based feedback as described in this document:

- Status reports are contained in sender report (SR)/received report (RR) packets and are transmitted at regular intervals as part of compound RTCP packets (which also include source description (SDES) and possibly other messages); these status reports provide an overall indication for the recent reception quality of a media stream.

- FB messages as defined in this document that indicate loss or reception of particular pieces of a media stream (or provide some other form of rather immediate feedback on the data received). Rules for the transmission of FB messages are newly introduced in this document.

RTCP FB messages are just another RTCP packet type (see Section 4). Therefore, multiple FB messages MAY be combined in a single compound RTCP packet and they MAY also be sent combined with other RTCP packets.

Compound RTCP packets containing FB messages as defined in this document MUST contain RTCP packets in the order defined in [1]:

- OPTIONAL encryption prefix that MUST be present if the RTCP packet(s) is to be encrypted according to Section 9.1 of [1].
- MANDATORY SR or RR.

- o MANDATORY SDES, which MUST contain the CNAME item; all other SDES items are OPTIONAL.
- o One or more FB messages.

The FB message(s) MUST be placed in the compound packet after RR and SDES RTCP packets defined in [1]. The ordering with respect to other RTCP extensions is not defined.

Two types of compound RTCP packets carrying feedback packets are used in this document:

a) Minimal compound RTCP feedback packet

A minimal compound RTCP feedback packet MUST contain only the mandatory information as listed above: encryption prefix if necessary, exactly one RR or SR, exactly one SDES with only the CNAME item present, and the FB message(s). This is to minimize the size of the RTCP packet transmitted to convey feedback and thus to maximize the frequency at which feedback can be provided while still adhering to the RTCP bandwidth limitations.

This packet format SHOULD be used whenever an RTCP FB message is sent as part of an Early RTCP packet. This packet type is referred to as minimal compound RTCP packet in this document.

b) (Full) compound RTCP feedback packet

A (full) compound RTCP feedback packet MAY contain any additional number of RTCP packets (additional RRs, further SDES items, etc.). The above ordering rules MUST be adhered to.

This packet format MUST be used whenever an RTCP FB message is sent as part of a Regular RTCP packet or in Regular RTCP mode. It MAY also be used to send RTCP FB messages in Immediate Feedback or Early RTCP mode. This packet type is referred to as full compound RTCP packet in this document.

RTCP packets that do not contain FB messages are referred to as non-FB RTCP packets. Such packets MUST follow the format rules in [1].

3.2. Algorithm Outline

FB messages are part of the RTCP control streams and thus subject to the RTCP bandwidth constraints. This means, in particular, that it may not be possible to report an event observed at a receiver immediately back to the sender. However, the value of feedback

given to a sender typically decreases over time -- in terms of the media quality as perceived by the user at the receiving end and/or the cost required to achieve media stream repair.

RTP [1] and the commonly used RTP profile [2] specify rules when compound RTCP packets should be sent. This document modifies those rules in order to allow applications to timely report events (e.g., loss or reception of RTP packets) and to accommodate algorithms that use FB messages.

The modified RTCP transmission algorithm can be outlined as follows: As long as no FB messages have to be conveyed, compound RTCP packets are sent following the rules of RTP [1] -- except that the five-second minimum interval between RTCP reports is not enforced. Hence, the interval between RTCP reports is only derived from the average RTCP packet size and the RTCP bandwidth share available to the RTP/RTCP entity. Optionally, a minimum interval between Regular RTCP packets may be enforced.

If a receiver detects the need to send an FB message, it may do so earlier than the next regular RTCP reporting interval (for which it would be scheduled following the above regular RTCP algorithm). Feedback suppression is used to avoid feedback implosion in multiparty sessions: The receiver waits for a (short) random dithering interval to check whether it sees a corresponding FB message from any other receiver reporting the same event. Note that for point-to-point sessions there is no such delay. If a corresponding FB message from another member is received, this receiver refrains from sending the FB message and continues to follow the Regular RTCP transmission schedule. In case the receiver has not yet seen a corresponding FB message from any other member, it checks whether it is allowed to send Early feedback. If sending Early feedback is permissible, the receiver sends the FB message as part of a minimal compound RTCP packet. The permission to send Early feedback depends on the type of the previous RTCP packet sent by this receiver and the time the previous Early feedback message was sent.

FB messages may also be sent as part of full compound RTCP packets, which are transmitted as per [1] (except for the five-second lower bound) in regular intervals.

3.3. Modes of Operation

RTCP-based feedback may operate in one of three modes (Figure 1) as described below. The mode of operation is just an indication of whether or not the receiver will, on average, be able to report all events to the sender in a timely fashion; the mode does not influence the algorithm used for scheduling the transmission of FB messages.

And, depending on the reception quality and the locally monitored state of the RTP session, individual receivers may not (and do not have to) agree on a common perception on the current mode of operation.

a) Immediate Feedback mode: In this mode, the group size is below the FB threshold, which gives each receiving party sufficient bandwidth to transmit the RTCP feedback packets for the intended purpose. This means that, for each receiver, there is enough bandwidth to report each event by means of a virtually "immediate" RTCP feedback packet.

   The group size threshold is a function of a number of parameters including (but not necessarily limited to): the type of feedback used (e.g., ACK vs. NACK), bandwidth, packet rate, packet loss probability and distribution, media type, codec, and the (worst case or observed) frequency of events to report (e.g., frame received, packet lost).

   As a rough estimate, let N be the average number of events to be reported per interval T by a receiver, B the RTCP bandwidth fraction for this particular receiver, and R the average RTCP packet size, then the receiver operates in Immediate Feedback mode as long as N<=B*T/R.

b) Early RTCP mode: In this mode, the group size and other parameters no longer allow each receiver to react to each event that would be worth reporting (or that needed reporting). But feedback can still be given sufficiently often so that it allows the sender to adapt the media stream transmission accordingly and thereby increase the overall media playback quality.

   Using the above notation, Early RTCP mode can be roughly characterized by N > B*T/R as "lower bound". An estimate for an upper bound is more difficult. Setting N=1, we obtain for a given R and B the interval T = R/B as average interval between events to be reported. This information can be used as a hint to determine whether or not early transmission of RTCP packets is useful.

c) Regular RTCP Mode: From some group size upwards, it is no longer useful to provide feedback for individual events from receivers at all -- because of the time scale in which the feedback could be provided and/or because in large groups the sender(s) have no chance to react to individual feedback anymore.

   No precise group size threshold can be specified at which this mode starts but, obviously, this boundary matches the upper bound of the Early RTCP mode as specified in item b) above.

As the feedback algorithm described in this document scales smoothly, there is no need for an agreement among the participants on the precise values of the respective FB thresholds within the group. Hence, the borders between all these modes are soft.

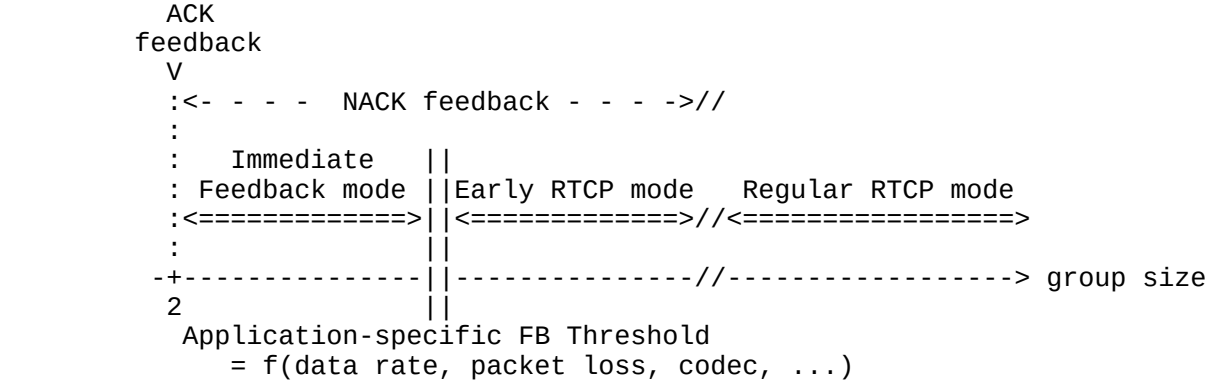

Figure 1: Modes of operation

As stated before, the respective FB thresholds depend on a number of technical parameters (of the codec, the transport, the type of feedback used, etc.) but also on the respective application scenarios. Section 3.6 provides some useful hints (but no precise calculations) on estimating these thresholds.

3.4. Definitions and Algorithm Overview

The following pieces of state information need to be maintained per receiver (largely taken from [1]). Note that all variables (except in item h) below) are calculated independently at each receiver. Therefore, their local values may differ at any given point in time.

a) Let "senders" be the number of active senders in the RTP session.

b) Let "members" be the current estimate of the number of receivers in the RTP session.

c) Let tn and tp be the time for the next (last) scheduled RTCP RR transmission calculated prior to timer reconsideration.

d) Let Tmin be the minimal interval between RTCP packets as per [1]. Unlike in [1], the initial Tmin is set to 1 second to allow for some group size sampling before sending the first RTCP packet. After the first RTCP packet is sent, Tmin is set to 0.

e) Let T_rr be the interval after which, having just sent a regularly scheduled RTCP packet, a receiver would schedule the transmission of its next Regular RTCP packet. This value is obtained following the rules of [1] but with Tmin as defined in this document: T_rr = T (the "calculated interval" as defined in [1]) with tn = tp + T. T_rr always refers to the last value of T that has been computed (because of reconsideration or to determine tn). T_rr is also referred to as Regular RTCP interval in this document.

f) Let t0 be the time at which an event that is to be reported is detected by a receiver.

g) Let T_dither_max be the maximum interval for which an RTCP feedback packet MAY be additionally delayed to prevent implosions in multiparty sessions; the value for T_dither_max is dynamically calculated based upon T_rr (or may be derived by means of another mechanism common across all RTP receivers to be specified in the future). For point-to-point sessions (i.e., sessions with exactly two members with no change in the group size expected, e.g., unicast streaming sessions), T_dither_max is set to 0.

h) Let T_max_fb_delay be the upper bound within which feedback to an event needs to be reported back to the sender to be useful at all. This value is application specific, and no values are defined in this document.

i) Let te be the time for which a feedback packet is scheduled.

j) Let T_fd be the actual (randomized) delay for the transmission of FB message in response to an event at time t0.

k) Let allow_early be a Boolean variable that indicates whether the receiver currently may transmit FB messages prior to its next regularly scheduled RTCP interval tn. This variable is used to throttle the feedback sent by a single receiver. allow_early is set to FALSE after Early feedback transmission and is set to TRUE as soon as the next Regular RTCP transmission takes place.

l) Let avg_rtcp_size be the moving average on the RTCP packet size as defined in [1].

m) Let T_rr_interval be an OPTIONAL minimal interval to be used between Regular RTCP packets. If T_rr_interval == 0, then this variable does not have any impact on the overall operation of the RTCP feedback algorithm. If T_rr_interval != 0, then the next Regular RTCP packet will not be scheduled T_rr after the last Regular RTCP transmission (i.e., at tp+T_rr). Instead, the next Regular RTCP packet will be delayed until at least T_rr_interval

after the last Regular RTCP transmission, i.e., it will be scheduled at or later than tp+T_rr_interval. Note that T_rr_interval does not affect the calculation of T_rr and tp; instead, Regular RTCP packets scheduled for transmission before tp+T_rr_interval will be suppressed if, for example, they do not contain any FB messages. The T_rr_interval does not affect transmission scheduling of Early RTCP packets.

Note: Providing T_rr_interval as an independent variable is meant to minimize Regular RTCP feedback (and thus bandwidth consumption) as needed by the application while additionally allowing the use of more frequent Early RTCP packets to provide timely feedback. This goal could not be achieved by reducing the overall RTCP bandwidth as RTCP bandwidth reduction would also impact the frequency of Early feedback.

n) Let t_rr_last be the point in time at which the last Regular RTCP packet has been scheduled and sent, i.e., has not been suppressed due to T_rr_interval.

o) Let T_retention be the time window for which past FB messages are stored by an AVPF entity. This is to ensure that feedback suppression also works for entities that have received FB messages from other entities prior to noticing the feedback event itself. T_retention MUST be set to at least 2 seconds.

p) Let M*Td be the timeout value for a receiver to be considered inactive (as defined in [1]).

The feedback situation for an event to report at a receiver is depicted in Figure 2 below. At time t0, such an event (e.g., a packet loss) is detected at the receiver. The receiver decides -- based upon current bandwidth, group size, and other application-specific parameters -- that an FB message needs to be sent back to the sender.

To avoid an implosion of feedback packets in multiparty sessions, the receiver MUST delay the transmission of the RTCP feedback packet by a random amount of time T_fd (with the random number evenly distributed in the interval [0, T_dither_max]). Transmission of the compound RTCP packet MUST then be scheduled for te = t0 + T_fd.

The T_dither_max parameter is derived from the Regular RTCP interval, T_rr, which, in turn, is based upon the group size. A future document may also specify other calculations for T_dither_max (e.g., based upon RTT) if it can be assured that all RTP receivers will use the same mechanism for calculating T_dither_max.

For a certain application scenario, a receiver may determine an upper bound for the acceptable local delay of FB messages: T_max_fb_delay. If an a priori estimation or the actual calculation of T_dither_max indicates that this upper bound MAY be violated (e.g., because T_dither_max > T_max_fb_delay), the receiver MAY decide not to send any feedback at all because the achievable gain is considered insufficient.

If an Early RTCP packet is scheduled, the time slot for the next Regular RTCP packet MUST be updated accordingly to have a new tn (tn=tp+2*T_rr) and a new tp (tp=tp+T_rr) afterwards. This is to ensure that the short-term average RTCP bandwidth used with Early feedback does not exceed the bandwidth used without Early feedback.

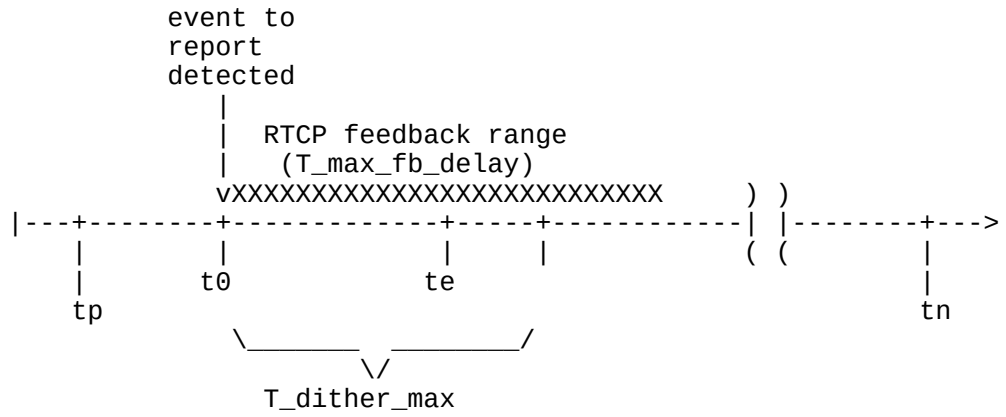

Figure 2: Event report and parameters for Early RTCP scheduling

3.5. AVPF RTCP Scheduling Algorithm

Let S0 be an active sender (out of S senders) and let N be the number of receivers with R being one of these receivers.

Assume that R has verified that using feedback mechanisms is reasonable at the current constellation (which is highly application specific and hence not specified in this document).

Assume further that T_rr_interval is 0, if no minimal interval between Regular RTCP packets is to be enforced, or T_rr_interval is set to some meaningful value, as given by the application. This value then denotes the minimal interval between Regular RTCP packets.

With this, a receiver R MUST use the following rules for transmitting one or more FB messages as minimal or full compound RTCP packet.

3.5.1. Initialization

Initially, R MUST set allow_early = TRUE and t_rr_last = NaN (Not-a-Number, i.e., some invalid value that can be distinguished from a valid time).

Furthermore, the initialization of the RTCP variables as per [1] applies except for the initial value for Tmin. For a point-to-point session, the initial Tmin is set to 0. For a multiparty session, Tmin is initialized to 1.0 seconds.

3.5.2. Early Feedback Transmission

Assume that R had scheduled the last Regular RTCP RR packet for transmission at tp (and sent or suppressed this packet at tp) and has scheduled the next transmission (including possible reconsideration as per [1]) for tn = tp + T_rr. Assume also that the last Regular RTCP packet transmission has occurred at t_rr_last.

The Early Feedback algorithm then comprises the following steps:

1. At time t0, R detects the need to transmit one or more FB messages, e.g., because media "units" need to be ACKed or NACKed, and finds that providing the feedback information is potentially useful for the sender.

2. R first checks whether there is already a compound RTCP packet containing one or more FB messages scheduled for transmission (either as Early or as Regular RTCP packet).

   2a) If so, the new FB message MUST be included in the scheduled packet; the scheduling of the waiting compound RTCP packet MUST remain unchanged. When doing so, the available feedback information SHOULD be merged to produce as few FB messages as possible. This completes the course of immediate actions to be taken.

   2b) If no compound RTCP packet is already scheduled for transmission, a new (minimal or full) compound RTCP packet MUST be created and the minimal interval for T_dither_max MUST be chosen as follows:

      i) If the session is a point-to-point session, then

         T_dither_max = 0.

ii) If the session is a multiparty session, then

T_dither_max = l * T_rr

with l=0.5.

The value for T_dither_max MAY be calculated differently (e.g., based upon RTT), which MUST then be specified in a future document. Such a future specification MUST ensure that all RTP receivers use the same mechanism to calculate T_dither_max.

The values given above for T_dither_max are minimal values. Application-specific feedback considerations may make it worthwhile to increase T_dither_max beyond this value. This is up to the discretion of the implementer.

3. Then, R MUST check whether its next Regular RTCP packet would be within the time bounds for the Early RTCP packet triggered at t0, i.e., if t0 + T_dither_max > tn.

   3a) If so, an Early RTCP packet MUST NOT be scheduled; instead, the FB message(s) MUST be stored to be included in the Regular RTCP packet scheduled for tn. This completes the course of immediate actions to be taken.

   3b) Otherwise, the following steps are carried out.

4. R MUST check whether it is allowed to transmit an Early RTCP packet, i.e., allow_early == TRUE, or not.

   4a) If allow_early == FALSE, then R MUST check the time for the next scheduled Regular RTCP packet:

      1. If tn - t0 < T_max_fb_delay, then the feedback could still be useful for the sender, despite the late reporting. Hence, R MAY create an RTCP FB message to be included in the Regular RTCP packet for transmission at tn.

      2. Otherwise, R MUST discard the RTCP FB message.

      This completes the immediate course of actions to be taken.

   4b) If allow_early == TRUE, then R MUST schedule an Early RTCP packet for te = t0 + RND * T_dither_max with RND being a pseudo random function evenly distributed between 0 and 1.

5. R MUST detect overlaps in FB messages received from other members of the RTP session and the FB messages R wants to send. Therefore, while a member of the RTP session, R MUST continuously monitor the arrival of (minimal) compound RTCP packets and store each FB message contained in these RTCP packets for at least T_retention. When scheduling the transmission of its own FB message following steps 1 through 4 above, R MUST check each of the stored and newly received FB messages from the RTCP packets received during the interval [t0 - T_retention ; te] and act as follows:

   5a) If R understands the received FB message's semantics and the message contents is a superset of the feedback R wanted to send, then R MUST discard its own FB message and MUST re-schedule the next Regular RTCP packet transmission for tn (as calculated before).

   5b) If R understands the received FB message's semantics and the message contents is not a superset of the feedback R wanted to send, then R SHOULD transmit its own FB message as scheduled. If there is an overlap between the feedback information to send and the feedback information received, the amount of feedback transmitted is up to R: R MAY leave its feedback information to be sent unchanged, R MAY as well eliminate any redundancy between its own feedback and the feedback received so far from other session members.

   5c) If R does not understand the received FB message's semantics, R MAY keep its own FB message scheduled as an Early RTCP packet, or R MAY re-schedule the next Regular RTCP packet transmission for tn (as calculated before) and MAY append the FB message to the now regularly scheduled RTCP message.

   Note: With 5c), receiving unknown FB messages may not lead to feedback suppression at a particular receiver. As a consequence, a given event may cause M different types of FB messages (which are all appropriate but not mutually understood) to be scheduled, so that a "large" receiver group may effectively be partitioned into at most M groups. Among members of each of these M groups, feedback suppression will occur following 5a and 5b but no suppression will happen across groups. As a result, O(M) RTCP FB messages may be received by the sender. Hence, there is a chance for a very limited feedback implosion. However, as sender(s) and all receivers make up the same application using the same (set of) codecs in the same RTP session, only little divergence in semantics for FB messages can safely be assumed and, therefore, M is assumed to be small in the general case.

Given further that the O(M) FB messages are randomly distributed over a time interval of T_dither_max, we find that the resulting limited number of extra compound RTCP packets (a) is assumed not to overwhelm the sender and (b) should be conveyed as all contain complementary pieces of information.

6. If R's FB message(s) was not suppressed by other receiver FB messages as per 5, when te is reached, R MUST transmit the (minimal) compound RTCP packet containing its FB message(s). R then MUST set allow_early = FALSE, MUST recalculate tn = tp + 2*T_rr, and MUST set tp to the previous tn. As soon as the newly calculated tn is reached, regardless whether R sends its next Regular RTCP packet or suppresses it because of T_rr_interval, it MUST set allow_early = TRUE again.

3.5.3. Regular RTCP Transmission

Full compound RTCP packets MUST be sent in regular intervals. These packets MAY also contain one or more FB messages. Transmission of Regular RTCP packets is scheduled as follows:

If T_rr_interval == 0, then the transmission MUST follow the rules as specified in Sections 3.2 and 3.4 of this document and MUST adhere to the adjustments of tn specified in Section 3.5.2 (i.e., skip one regular transmission if an Early RTCP packet transmission has occurred). Timer reconsideration takes place when tn is reached as per [1]. The Regular RTCP packet is transmitted after timer reconsideration. Whenever a Regular RTCP packet is sent or suppressed, allow_early MUST be set to TRUE and tp, tn MUST be updated as per [1]. After the first transmission of a Regular RTCP packet, Tmin MUST be set to 0.

If T_rr_interval != 0, then the calculation for the transmission times MUST follow the rules as specified in Sections 3.2 and 3.4 of this document and MUST adhere to the adjustments of tn specified in Section 3.5.2 (i.e., skip one regular transmission if an Early RTCP transmission has occurred). Timer reconsideration takes place when tn is reached as per [1]. After timer reconsideration, the following actions are taken:

1. If no Regular RTCP packet has been sent before (i.e., if t_rr_last == NaN), then a Regular RTCP packet MUST be scheduled. Stored FB messages MAY be included in the Regular RTCP packet. After the scheduled packet has been sent, t_rr_last MUST be set to tn. Tmin MUST be set to 0.

2. Otherwise, a temporary value T_rr_current_interval is calculated as follows:

   T_rr_current_interval = RND*T_rr_interval

   with RND being a pseudo random function evenly distributed between 0.5 and 1.5. This dithered value is used to determine one of the following alternatives:

   2a) If t_rr_last + T_rr_current_interval <= tn, then a Regular RTCP packet MUST be scheduled. Stored RTCP FB messages MAY be included in the Regular RTCP packet. After the scheduled packet has been sent, t_rr_last MUST be set to tn.

   2b) If t_rr_last + T_rr_current_interval > tn and RTCP FB messages have been stored and are awaiting transmission, an RTCP packet MUST be scheduled for transmission at tn. This RTCP packet MAY be a minimal or a Regular RTCP packet (at the discretion of the implementer), and the compound RTCP packet MUST include the stored RTCP FB message(s). t_rr_last MUST remain unchanged.

   2c) Otherwise (if t_rr_last + T_rr_current_interval > tn but no stored RTCP FB messages are awaiting transmission), the compound RTCP packet MUST be suppressed (i.e., it MUST NOT be scheduled). t_rr_last MUST remain unchanged.

In all the four cases above (1, 2a, 2b, and 2c), allow_early MUST be set to TRUE (possibly after sending the Regular RTCP packet) and tp and tn MUST be updated following the rules of [1] except for the five second minimum.

3.5.4. Other Considerations

If T_rr_interval != 0, then the timeout calculation for RTP/AVPF entities (Section 6.3.5 of [1]) MUST be modified to use T_rr_interval instead of Tmin for computing Td and thus M*Td for timing out RTP entities.

Whenever a compound RTCP packet is sent or received -- minimal or full compound, Early or Regular -- the avg_rtcp_size variable MUST be updated accordingly (see [1]) and subsequent computations of tn MUST use the new avg_rtcp_size.

3.6. Considerations on the Group Size

This section provides some guidelines to the group sizes at which the various feedback modes may be used.

3.6.1. ACK Mode

The RTP session MUST have exactly two members and this group size MUST NOT grow, i.e., it MUST be point-to-point communications. Unicast addresses SHOULD be used in the session description.

For unidirectional as well as bi-directional communication between two parties, 2.5% of the RTP session bandwidth is available for RTCP traffic from the receivers including feedback. For a 64-kbit/s stream this yields 1,600 bit/s for RTCP. If we assume an average of 96 bytes (=768 bits) per RTCP packet, a receiver can report 2 events per second back to the sender. If acknowledgements for 10 events are collected in each FB message, then 20 events can be acknowledged per second. At 256 kbit/s, 8 events could be reported per second; thus, the ACKs may be sent in a finer granularity (e.g., only combining three ACKs per FB message).

From 1 Mbit/s upwards, a receiver would be able to acknowledge each individual frame (not packet!) in a 30-fps video stream.

ACK strategies MUST be defined to work properly with these bandwidth limitations. An indication whether or not ACKs are allowed for a session and, if so, which ACK strategy should be used, MAY be conveyed by out-of-band mechanisms, e.g., media-specific attributes in a session description using SDP.

3.6.2. NACK Mode

Negative acknowledgements (and the other types of feedback exhibiting similar reporting characteristics) MUST be used for all sessions with a group size that may grow larger than two. Of course, NACKs MAY be used for point-to-point communications as well.

Whether or not the use of Early RTCP packets should be considered depends upon a number of parameters including session bandwidth, codec, special type of feedback, and number of senders and receivers.

The most important parameters when determining the mode of operation are the allowed minimal interval between two compound RTCP packets (T_rr) and the average number of events that presumably need reporting per time interval (plus their distribution over time, of course). The minimum interval can be derived from the available RTCP bandwidth and the expected average size of an RTCP packet. The

number of events to report (e.g., per second) may be derived from the packet loss rate and sender's rate of transmitting packets. From these two values, the allowable group size for the Immediate Feedback mode can be calculated.

As stated in Section 3.3:

> Let N be the average number of events to be reported per interval T by a receiver, B the RTCP bandwidth fraction for this particular receiver, and R the average RTCP packet size, then the receiver operates in Immediate Feedback mode as long as N<=B*T/R.

The upper bound for the Early RTCP mode then solely depends on the acceptable quality degradation, i.e., how many events per time interval may go unreported.

As stated in Section 3.3:

> Using the above notation, Early RTCP mode can be roughly characterized by N > B*T/R as "lower bound". An estimate for an upper bound is more difficult. Setting N=1, we obtain for a given R and B the interval T = R/B as average interval between events to be reported. This information can be used as a hint to determine whether or not early transmission of RTCP packets is useful.

Example: If a 256-kbit/s video with 30 fps is transmitted through a network with an MTU size of some 1,500 bytes, then, in most cases, each frame would fit in into one packet leading to a packet rate of 30 packets per second. If 5% packet loss occurs in the network (equally distributed, no inter-dependence between receivers), then each receiver will, on average, have to report 3 packets lost each two seconds. Assuming a single sender and more than three receivers, this yields 3.75% of the RTCP bandwidth allocated to the receivers and thus 9.6 kbit/s. Assuming further a size of 120 bytes for the average compound RTCP packet allows 10 RTCP packets to be sent per second or 20 in two seconds. If every receiver needs to report three lost packets per two seconds, this yields a maximum group size of 6-7 receivers if all loss events are reported. The rules for transmission of Early RTCP packets should provide sufficient flexibility for most of this reporting to occur in a timely fashion.

Extending this example to determine the upper bound for Early RTCP mode could lead to the following considerations: assume that the underlying coding scheme and the application (as well as the tolerant users) allow on the order of one loss without repair per two seconds. Thus, the number of packets to be reported by each receiver decreases to two per two seconds and increases the group size to 10. Assuming further that some number of packet losses are correlated, feedback

traffic is further reduced and group sizes of some 12 to 16 (maybe even 20) can be reasonably well supported using Early RTCP mode. Note that all these considerations are based upon statistics and will fail to hold in some cases.

3.7. Summary of Decision Steps

3.7.1. General Hints

Before even considering whether or not to send RTCP feedback information, an application has to determine whether this mechanism is applicable:

1) An application has to decide whether -- for the current ratio of packet rate with the associated (application-specific) maximum feedback delay and the currently observed round-trip time (if available) -- feedback mechanisms can be applied at all.

   This decision may be based upon (and dynamically revised following) RTCP reception statistics as well as out-of-band mechanisms.

2) The application has to decide -- for a certain observed error rate, assigned bandwidth, frame/packet rate, and group size -- whether (and which) feedback mechanisms can be applied.

   Regular RTCP reception statistics provide valuable input to this step, too.

3) If the application decides to send feedback, the application has to follow the rules for transmitting Early RTCP packets or Regular RTCP packets containing FB messages.

4) The type of RTCP feedback sent should not duplicate information available to the sender from a lower layer transport protocol. That is, if the transport protocol provides negative or positive acknowledgements about packet reception (such as DCCP), the receiver should avoid repeating the same information at the RTCP layer (i.e., abstain from sending Generic NACKs).

3.7.2. Media Session Attributes

Media sessions are typically described using out-of-band mechanisms to convey transport addresses, codec information, etc., between sender(s) and receiver(s). Such a mechanism is two-fold: a format used to describe a media session and another mechanism for transporting this description.

In the IETF, the Session Description Protocol (SDP) is currently used to describe media sessions while protocols such as SIP, Session Announcement Protocol (SAP), Real Time Streaming Protocol (RTSP), and HTTP (among others) are used to convey the descriptions.

A media session description format MAY include parameters to indicate that RTCP feedback mechanisms are supported in this session and which of the feedback mechanisms MAY be applied.

To do so, the profile "AVPF" MUST be indicated instead of "AVP". Further attributes may be defined to show which type(s) of feedback are supported.

Section 4 contains the syntax specification to support RTCP feedback with SDP. Similar specifications for other media session description formats are outside the scope of this document.

4. SDP Definitions

This section defines a number of additional SDP parameters that are used to describe a session. All of these are defined as media-level attributes.

4.1. Profile Identification

The AV profile defined in [4] is referred to as "AVP" in the context of, e.g., the Session Description Protocol (SDP) [3]. The profile specified in this document is referred to as "AVPF".

Feedback information following the modified timing rules as specified in this document MUST NOT be sent for a particular media session unless the description for this session indicates the use of the "AVPF" profile (exclusively or jointly with other AV profiles).

4.2. RTCP Feedback Capability Attribute

A new payload format-specific SDP attribute is defined to indicate the capability of using RTCP feedback as specified in this document: "a=rtcp-fb". The "rtcp-fb" attribute MUST only be used as an SDP media attribute and MUST NOT be provided at the session level. The "rtcp-fb" attribute MUST only be used in media sessions for which the "AVPF" is specified.

The "rtcp-fb" attribute SHOULD be used to indicate which RTCP FB messages MAY be used in this media session for the indicated payload type. A wildcard payload type ("*") MAY be used to indicate that the RTCP feedback attribute applies to all payload types. If several types of feedback are supported and/or the same feedback shall be

specified for a subset of the payload types, several "a=rtcp-fb" lines MUST be used.

If no "rtcp-fb" attribute is specified, the RTP receivers MAY send feedback using other suitable RTCP feedback packets as defined for the respective media type. The RTP receivers MUST NOT rely on the RTP senders reacting to any of the FB messages. The RTP sender MAY choose to ignore some feedback messages.

If one or more "rtcp-fb" attributes are present in a media session description, the RTCP receivers for the media session(s) containing the "rtcp-fb"

- MUST ignore all "rtcp-fb" attributes of which they do not fully understand the semantics (i.e., where they do not understand the meaning of all values in the "a=rtcp-fb" line);

- SHOULD provide feedback information as specified in this document using any of the RTCP feedback packets as specified in one of the "rtcp-fb" attributes for this media session; and

- MUST NOT use other FB messages than those listed in one of the "rtcp-fb" attribute lines.

When used in conjunction with the offer/answer model [8], the offerer MAY present a set of these AVPF attributes to its peer. The answerer MUST remove all attributes it does not understand as well as those it does not support in general or does not wish to use in this particular media session. The answerer MUST NOT add feedback parameters to the media description and MUST NOT alter values of such parameters. The answer is binding for the media session, and both offerer and answerer MUST only use feedback mechanisms negotiated in this way. Both offerer and answerer MAY independently decide to send RTCP FB messages of only a subset of the negotiated feedback mechanisms, but they SHOULD react properly to all types of the negotiated FB messages when received.

RTP senders MUST be prepared to receive any kind of RTCP FB messages and MUST silently discard all those RTCP FB messages that they do not understand.

The syntax of the "rtcp-fb" attribute is as follows (the feedback types and optional parameters are all case sensitive):

(In the following ABNF, fmt, SP, and CRLF are used as defined in [3].)

```
rtcp-fb-syntax = "a=rtcp-fb:" rtcp-fb-pt SP rtcp-fb-val CRLF

rtcp-fb-pt         = "*"   ; wildcard: applies to all formats
                   / fmt   ; as defined in SDP spec

rtcp-fb-val        = "ack" rtcp-fb-ack-param
                   / "nack" rtcp-fb-nack-param
                   / "trr-int" SP 1*DIGIT
                   / rtcp-fb-id rtcp-fb-param

rtcp-fb-id         = 1*(alpha-numeric / "-" / "_")

rtcp-fb-param      = SP "app" [SP byte-string]
                   / SP token [SP byte-string]
                   / ; empty

rtcp-fb-ack-param  = SP "rpsi"
                   / SP "app" [SP byte-string]
                   / SP token [SP byte-string]
                   / ; empty

rtcp-fb-nack-param = SP "pli"
                   / SP "sli"
                   / SP "rpsi"
                   / SP "app" [SP byte-string]
                   / SP token [SP byte-string]
                   / ; empty
```

The literals of the above grammar have the following semantics:

Feedback type "ack":

This feedback type indicates that positive acknowledgements for feedback are supported.

The feedback type "ack" MUST only be used if the media session is allowed to operate in ACK mode as defined in Section 3.6.1.

Parameters MUST be provided to further distinguish different types of positive acknowledgement feedback.

The parameter "rpsi" indicates the use of Reference Picture Selection Indication feedback as defined in Section 6.3.3.

If the parameter "app" is specified, this indicates the use of application layer feedback. In this case, additional parameters following "app" MAY be used to further differentiate various types of application layer feedback. This document does not define any parameters specific to "app".

Further parameters for "ack" MAY be defined in other documents.

Feedback type "nack":

This feedback type indicates that negative acknowledgements for feedback are supported.

The feedback type "nack", without parameters, indicates use of the Generic NACK feedback format as defined in Section 6.2.1.

The following three parameters are defined in this document for use with "nack" in conjunction with the media type "video":

- "pli" indicates the use of Picture Loss Indication feedback as defined in Section 6.3.1.

- "sli" indicates the use of Slice Loss Indication feedback as defined in Section 6.3.2.

- "rpsi" indicates the use of Reference Picture Selection Indication feedback as defined in Section 6.3.3.

"app" indicates the use of application layer feedback. Additional parameters after "app" MAY be provided to differentiate different types of application layer feedback. No parameters specific to "app" are defined in this document.

Further parameters for "nack" MAY be defined in other documents.

Other feedback types <rtcp-fb-id>:

Other documents MAY define additional types of feedback; to keep the grammar extensible for those cases, the rtcp-fb-id is introduced as a placeholder. A new feedback scheme name MUST to be unique (and thus MUST be registered with IANA). Along with a new name, its semantics, packet formats (if necessary), and rules for its operation MUST be specified.

Regular RTCP minimum interval "trr-int":

The attribute "trr-int" is used to specify the minimum interval T_rr_interval between two Regular (full compound) RTCP packets in milliseconds for this media session. If "trr-int" is not specified, a default value of 0 is assumed.

Note that it is assumed that more specific information about application layer feedback (as defined in Section 6.4) will be conveyed as feedback types and parameters defined elsewhere. Hence, no further provision for any types and parameters is made in this document.

Further types of feedback as well as further parameters may be defined in other documents.

It is up to the recipients whether or not they send feedback information and up to the sender(s) (how) to make use of feedback provided.

4.3. RTCP Bandwidth Modifiers

The standard RTCP bandwidth assignments as defined in [1] and [2] MAY be overridden by bandwidth modifiers that explicitly define the maximum RTCP bandwidth. For use with SDP, such modifiers are specified in [4]: "b=RS:<bw>" and "b=RR:<bw>" MAY be used to assign a different bandwidth (measured in bits per second) to RTP senders and receivers, respectively. The precedence rules of [4] apply to determine the actual bandwidth to be used by senders and receivers.

Applications operating knowingly over highly asymmetric links (such as satellite links) SHOULD use this mechanism to reduce the feedback rate for high bandwidth streams to prevent deterministic congestion of the feedback path(s).

4.4. Examples

Example 1: The following session description indicates a session made up from audio and DTMF [18] for point-to-point communication in which the DTMF stream uses Generic NACKs. This session description could be contained in a SIP INVITE, 200 OK, or ACK message to indicate that its sender is capable of and willing to receive feedback for the DTMF stream it transmits.

```
v=0
o=alice 3203093520 3203093520 IN IP4 host.example.com
s=Media with feedback
t=0 0
```

```
c=IN IP4 host.example.com
m=audio 49170 RTP/AVPF 0 96
a=rtpmap:0 PCMU/8000
a=rtpmap:96 telephone-event/8000
a=fmtp:96 0-16
a=rtcp-fb:96 nack
```

This allows sender and receiver to provide reliable transmission of DTMF events in an audio session. Assuming a 64-kbit/s audio stream with one receiver, the receiver has 2.5% RTCP bandwidth available for the negative acknowledgement stream, i.e., 250 bytes per second or some 2 RTCP feedback messages every second. Hence, the receiver can individually communicate up to two missing DTMF audio packets per second.

Example 2: The following session description indicates a multicast video-only session (using either H.261 or H.263+) with the video source accepting Generic NACKs for both codecs and Reference Picture Selection for H.263. Such a description may have been conveyed using the Session Announcement Protocol (SAP).

```
v=0
o=alice 3203093520 3203093520 IN IP4 host.example.com
s=Multicast video with feedback
t=3203130148 3203137348
m=audio 49170 RTP/AVP 0
c=IN IP4 224.2.1.183
a=rtpmap:0 PCMU/8000
m=video 51372 RTP/AVPF 98 99
c=IN IP4 224.2.1.184
a=rtpmap:98 H263-1998/90000
a=rtpmap:99 H261/90000
a=rtcp-fb:* nack
a=rtcp-fb:98 nack rpsi
```

The sender may use an incoming Generic NACK as a hint to send a new intra-frame as soon as possible (congestion control permitting). Receipt of a Reference Picture Selection Indication (RPSI) message allows the sender to avoid sending a large intra-frame; instead it may continue to send inter-frames, however, choosing the indicated frame as new encoding reference.

Example 3: The following session description defines the same media session as example 2 but allows for mixed-mode operation of AVP and AVPF RTP entities (see also next section). Note that both media descriptions use the same addresses; however, two m= lines are needed to convey information about both applicable RTP profiles.

```
v=0
o=alice 3203093520 3203093520 IN IP4 host.example.com
s=Multicast video with feedback
t=3203130148 3203137348
m=audio 49170 RTP/AVP 0
c=IN IP4 224.2.1.183
a=rtpmap:0 PCMU/8000
m=video 51372 RTP/AVP 98 99
c=IN IP4 224.2.1.184
a=rtpmap:98 H263-1998/90000
a=rtpmap:99 H261/90000
m=video 51372 RTP/AVPF 98 99
c=IN IP4 224.2.1.184
a=rtpmap:98 H263-1998/90000
a=rtpmap:99 H261/90000
a=rtcp-fb:* nack
a=rtcp-fb:98 nack rpsi
```

Note that these two m= lines SHOULD be grouped by some appropriate mechanism to indicate that both are alternatives actually conveying the same contents. A sample framework by which this can be achieved is defined in [10].

In this example, the RTCP feedback-enabled receivers will gain an occasional advantage to report events earlier back to the sender (which may benefit the entire group). On average, however, all RTP receivers will provide the same amount of feedback. The interworking between AVP and AVPF entities is discussed in depth in the next section.

5. Interworking and Coexistence of AVP and AVPF Entities

The AVPF profile defined in this document is an extension of the AVP profile as defined in [2]. Both profiles follow the same basic rules (including the upper bandwidth limit for RTCP and the bandwidth assignments to senders and receivers). Therefore, senders and receivers using either of the two profiles can be mixed in a single session (see Example 3 in Section 4.5).

AVP and AVPF are defined in a way that, from a robustness point of view, the RTP entities do not need to be aware of entities of the respective other profile: they will not disturb each other's functioning. However, the quality of the media presented may suffer.

The following considerations apply to senders and receivers when used in a combined session.

- AVP entities (senders and receivers)

  AVP senders will receive RTCP feedback packets from AVPF receivers and ignore these packets. They will see occasional closer spacing of RTCP messages (e.g., violating the five-second rule) by AVPF entities. As the overall bandwidth constraints are adhered to by both types of entities, they will still get their share of the RTCP bandwidth. However, while AVP entities are bound by the five-second rule, depending on the group size and session bandwidth, AVPF entities may provide more frequent RTCP reports than AVP ones will. Also, the overall reporting may decrease slightly as AVPF entities may send bigger compound RTCP packets (due to the extra RTCP packets).

  If T_rr_interval is used as lower bound between Regular RTCP packets, T_rr_interval is sufficiently large (e.g., T_rr_interval > M*Td as per Section 6.3.5 of [1]), and no Early RTCP packets are sent by AVPF entities, AVP entities may accidentally time out those AVPF group members and hence underestimate the group size. Therefore, if AVP entities may be involved in a media session, T_rr_interval SHOULD NOT be larger than five seconds.

- AVPF entities (senders and receivers)

  If the dynamically calculated T_rr is sufficiently small (e.g., less than one second), AVPF entities may accidentally time out AVP group members and hence underestimate the group size. Therefore, if AVP entities may be involved in a media session, T_rr_interval SHOULD be used and SHOULD be set to five seconds.

  In conclusion, if AVP entities may be involved in a media session and T_rr_interval is to be used, T_rr_interval SHOULD be set to five seconds.

- AVPF senders

  AVPF senders will receive feedback information only from AVPF receivers. If they rely on feedback to provide the target media quality, the quality achieved for AVP receivers may be suboptimal.

- AVPF receivers

  AVPF receivers SHOULD send Early RTCP feedback packets only if all sending entities in the media session support AVPF. AVPF receivers MAY send feedback information as part of regularly scheduled compound RTCP packets following the timing rules of

[1] and [2] also in media sessions operating in mixed mode. However, the receiver providing feedback MUST NOT rely on the sender reacting to the feedback at all.

6. Format of RTCP Feedback Messages

This section defines the format of the low-delay RTCP feedback messages. These messages are classified into three categories as follows:

- Transport layer FB messages
- Payload-specific FB messages
- Application layer FB messages

Transport layer FB messages are intended to transmit general purpose feedback information, i.e., information independent of the particular codec or the application in use. The information is expected to be generated and processed at the transport/RTP layer. Currently, only a generic negative acknowledgement (NACK) message is defined.

Payload-specific FB messages transport information that is specific to a certain payload type and will be generated and acted upon at the codec "layer". This document defines a common header to be used in conjunction with all payload-specific FB messages. The definition of specific messages is left either to RTP payload format specifications or to additional feedback format documents.

Application layer FB messages provide a means to transparently convey feedback from the receiver's to the sender's application. The information contained in such a message is not expected to be acted upon at the transport/RTP or the codec layer. The data to be exchanged between two application instances is usually defined in the application protocol specification and thus can be identified by the application so that there is no need for additional external information. Hence, this document defines only a common header to be used along with all application layer FB messages. From a protocol point of view, an application layer FB message is treated as a special case of a payload-specific FB message.

Note: Proper processing of some FB messages at the media sender side may require the sender to know which payload type the FB message refers to. Most of the time, this knowledge can likely be derived from a media stream using only a single payload type. However, if several codecs are used simultaneously (e.g., with audio and DTMF) or when codec changes occur, the payload type information may need to be conveyed explicitly as part of the FB message. This applies to all

payload-specific as well as application layer FB messages. It is up to the specification of an FB message to define how payload type information is transmitted.

This document defines two transport layer and three (video) payload-specific FB messages as well as a single container for application layer FB messages. Additional transport layer and payload-specific FB messages MAY be defined in other documents and MUST be registered through IANA (see Section 9, "IANA Considerations").

The general syntax and semantics for the above RTCP FB message types are described in the following subsections.

6.1. Common Packet Format for Feedback Messages

All FB messages MUST use a common packet format that is depicted in Figure 3:

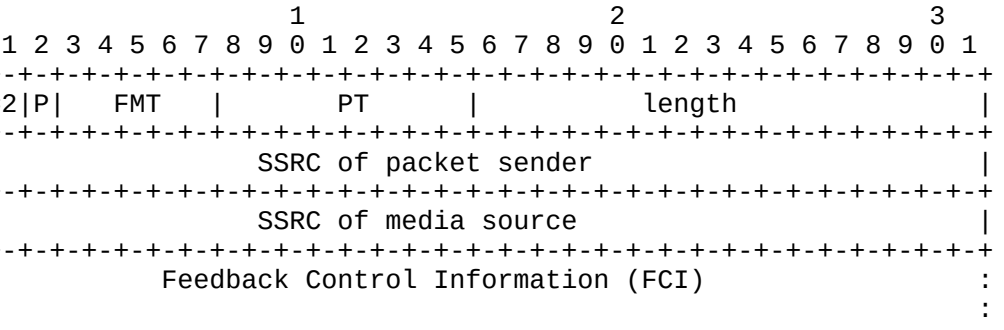

Figure 3: Common Packet Format for Feedback Messages

The fields V, P, SSRC, and length are defined in the RTP specification [2], the respective meaning being summarized below:

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

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

Feedback message type (FMT): 5 bits
This field identifies the type of the FB message and is interpreted relative to the type (transport layer, payload-specific, or application layer feedback). The values for each of the three feedback types are defined in the respective sections below.

Payload type (PT): 8 bits
This is the RTCP packet type that identifies the packet as being an RTCP FB message. Two values are defined by the IANA:

Name	Value	Brief Description
RTPFB	205	Transport layer FB message
PSFB	206	Payload-specific FB message

Length: 16 bits
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 [3].

SSRC of packet sender: 32 bits
The synchronization source identifier for the originator of this packet.

SSRC of media source: 32 bits
The synchronization source identifier of the media source that this piece of feedback information is related to.

Feedback Control Information (FCI): variable length
The following three sections define which additional information MAY be included in the FB message for each type of feedback: transport layer, payload-specific, or application layer feedback. Note that further FCI contents MAY be specified in further documents.

Each RTCP feedback packet MUST contain at least one FB message in the FCI field. Sections 6.2 and 6.3 define for each FCI type, whether or not multiple FB messages MAY be compressed into a single FCI field. If this is the case, they MUST be of the same type, i.e., same FMT. If multiple types of feedback messages, i.e., several FMTs, need to be conveyed, then several RTCP FB messages MUST be generated and SHOULD be concatenated in the same compound RTCP packet.

6.2. Transport Layer Feedback Messages

Transport layer FB messages are identified by the value RTPFB as RTCP message type.

A single general purpose transport layer FB message is defined in this document: Generic NACK. It is identified by means of the FMT parameter as follows:

```
0:    unassigned
1:    Generic NACK
2-30: unassigned
31:   reserved for future expansion of the identifier number space
```

The following subsection defines the formats of the FCI field for this type of FB message. Further generic feedback messages MAY be defined in the future.

6.2.1. Generic NACK

The Generic NACK message is identified by PT=RTPFB and FMT=1.

The FCI field MUST contain at least one and MAY contain more than one Generic NACK.

The Generic NACK is used to indicate the loss of one or more RTP packets. The lost packet(s) are identified by the means of a packet identifier and a bit mask.

Generic NACK feedback SHOULD NOT be used if the underlying transport protocol is capable of providing similar feedback information to the sender (as may be the case, e.g., with DCCP).

The Feedback Control Information (FCI) field has the following Syntax (Figure 4):

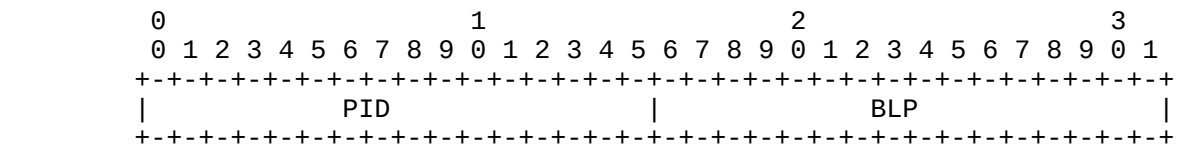

Figure 4: Syntax for the Generic NACK message

Packet ID (PID): 16 bits
The PID field is used to specify a lost packet. The PID field refers to the RTP sequence number of the lost packet.

bitmask of following lost packets (BLP): 16 bits
The BLP allows for reporting losses of any of the 16 RTP packets immediately following the RTP packet indicated by the PID. The BLP's definition is identical to that given in [6]. Denoting the BLP's least significant bit as bit 1, and its most significant bit as bit 16, then bit i of the bit mask is set to 1 if the receiver has not received RTP packet number (PID+i) (modulo 2^16) and indicates this packet is lost; bit i is set to 0 otherwise. Note that the sender MUST NOT assume that a receiver has received a packet because its bit mask was set to 0. For example, the least significant bit of the BLP would be set to 1 if the packet corresponding to the PID and the following packet have been lost. However, the sender cannot infer that packets PID+2 through PID+16 have been received simply because bits 2 through 15 of the BLP are 0; all the sender knows is that the receiver has not reported them as lost at this time.

The length of the FB message MUST be set to 2+n, with n being the number of Generic NACKs contained in the FCI field.

The Generic NACK message implicitly references the payload type through the sequence number(s).

6.3. Payload-Specific Feedback Messages

Payload-Specific FB messages are identified by the value PT=PSFB as RTCP message type.

Three payload-specific FB messages are defined so far plus an application layer FB message. They are identified by means of the FMT parameter as follows:

```
0:     unassigned
1:     Picture Loss Indication (PLI)
2:     Slice Loss Indication (SLI)
3:     Reference Picture Selection Indication (RPSI)
4-14:  unassigned
15:    Application layer FB (AFB) message
16-30: unassigned
31:    reserved for future expansion of the sequence number space
```

The following subsections define the FCI formats for the payload-specific FB messages, Section 6.4 defines FCI format for the application layer FB message.

6.3.1. Picture Loss Indication (PLI)

The PLI FB message is identified by PT=PSFB and FMT=1.

There MUST be exactly one PLI contained in the FCI field.

6.3.1.1. Semantics

With the Picture Loss Indication message, a decoder informs the encoder about the loss of an undefined amount of coded video data belonging to one or more pictures. When used in conjunction with any video coding scheme that is based on inter-picture prediction, an encoder that receives a PLI becomes aware that the prediction chain may be broken. The sender MAY react to a PLI by transmitting an intra-picture to achieve resynchronization (making this message effectively similar to the FIR message as defined in [6]); however, the sender MUST consider congestion control as outlined in Section 7, which MAY restrict its ability to send an intra frame.

Other RTP payload specifications such as RFC 2032 [6] already define a feedback mechanism for some for certain codecs. An application supporting both schemes MUST use the feedback mechanism defined in this specification when sending feedback. For backward compatibility reasons, such an application SHOULD also be capable to receive and react to the feedback scheme defined in the respective RTP payload format, if this is required by that payload format.

6.3.1.2. Message Format

PLI does not require parameters. Therefore, the length field MUST be 2, and there MUST NOT be any Feedback Control Information.

The semantics of this FB message is independent of the payload type.

6.3.1.3. Timing Rules

The timing follows the rules outlined in Section 3. In systems that employ both PLI and other types of feedback, it may be advisable to follow the Regular RTCP RR timing rules for PLI, since PLI is not as delay critical as other FB types.

6.3.1.4. Remarks

PLI messages typically trigger the sending of full intra-pictures. Intra-pictures are several times larger then predicted (inter-) pictures. Their size is independent of the time they are generated. In most environments, especially when employing bandwidth-limited links, the use of an intra-picture implies an allowed delay that is a

significant multitude of the typical frame duration. An example: If the sending frame rate is 10 fps, and an intra-picture is assumed to be 10 times as big as an inter-picture, then a full second of latency has to be accepted. In such an environment, there is no need for a particular short delay in sending the FB message. Hence, waiting for the next possible time slot allowed by RTCP timing rules as per [2] with Tmin=0 does not have a negative impact on the system performance.

6.3.2. Slice Loss Indication (SLI)

The SLI FB message is identified by PT=PSFB and FMT=2.

The FCI field MUST contain at least one and MAY contain more than one SLI.

6.3.2.1. Semantics

With the Slice Loss Indication, a decoder can inform an encoder that it has detected the loss or corruption of one or several consecutive macroblock(s) in scan order (see below). This FB message MUST NOT be used for video codecs with non-uniform, dynamically changeable macroblock sizes such as H.263 with enabled Annex Q. In such a case, an encoder cannot always identify the corrupted spatial region.

6.3.2.2. Format

The Slice Loss Indication uses one additional FCI field, the content of which is depicted in Figure 6. The length of the FB message MUST be set to 2+n, with n being the number of SLIs contained in the FCI field.

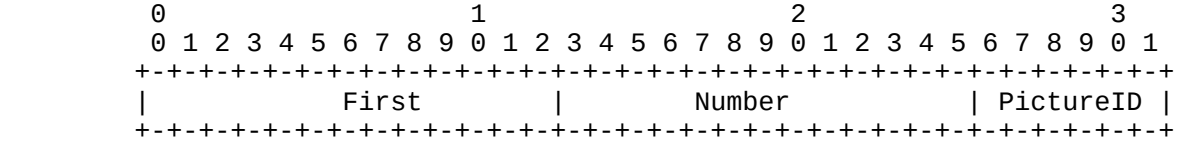

Figure 6: Syntax of the Slice Loss Indication (SLI)

First: 13 bits
The macroblock (MB) address of the first lost macroblock. The MB numbering is done such that the macroblock in the upper left corner of the picture is considered macroblock number 1 and the number for each macroblock increases from left to right and then from top to bottom in raster-scan order (such that if there is a total of N macroblocks in a picture, the bottom right macroblock is considered macroblock number N).

Number: 13 bits
The number of lost macroblocks, in scan order as discussed above.

PictureID: 6 bits
The six least significant bits of the codec-specific identifier that is used to reference the picture in which the loss of the macroblock(s) has occurred. For many video codecs, the PictureID is identical to the Temporal Reference.

The applicability of this FB message is limited to a small set of video codecs; therefore, no explicit payload type information is provided.

6.3.2.3. Timing Rules

The efficiency of algorithms using the Slice Loss Indication is reduced greatly when the Indication is not transmitted in a timely fashion. Motion compensation propagates corrupted pixels that are not reported as being corrupted. Therefore, the use of the algorithm discussed in Section 3 is highly recommended.

6.3.2.4. Remarks

The term Slice is defined and used here in the sense of MPEG-1 -- a consecutive number of macroblocks in scan order. More recent video coding standards sometimes have a different understanding of the term Slice. In H.263 (1998), for example, a concept known as "rectangular slice" exists. The loss of one rectangular slice may lead to the necessity of sending more than one SLI in order to precisely identify the region of lost/damaged MBs.

The first field of the FCI defines the first macroblock of a picture as 1 and not, as one could suspect, as 0. This was done to align this specification with the comparable mechanism available in ITU-T Rec. H.245 [24]. The maximum number of macroblocks in a picture (2**13 or 8192) corresponds to the maximum picture sizes of most of the ITU-T and ISO/IEC video codecs. If future video codecs offer larger picture sizes and/or smaller macroblock sizes, then an additional FB message has to be defined. The six least significant bits of the Temporal Reference field are deemed to be sufficient to indicate the picture in which the loss occurred.

The reaction to an SLI is not part of this specification. One typical way of reacting to an SLI is to use intra refresh for the affected spatial region.

Algorithms were reported that keep track of the regions affected by motion compensation, in order to allow for a transmission of Intra macroblocks to all those areas, regardless of the timing of the FB (see H.263 (2000) Appendix I [17] and [15]). Although the timing of the FB is less critical when those algorithms are used than if they are not, it has to be observed that those algorithms correct large parts of the picture and, therefore, have to transmit much higher data volume in case of delayed FBs.

6.3.3. Reference Picture Selection Indication (RPSI)

The RPSI FB message is identified by PT=PSFB and FMT=3.

There MUST be exactly one RPSI contained in the FCI field.

6.3.3.1. Semantics

Modern video coding standards such as MPEG-4 visual version 2 [16] or H.263 version 2 [17] allow using older reference pictures than the most recent one for predictive coding. Typically, a first-in-first-out queue of reference pictures is maintained. If an encoder has learned about a loss of encoder-decoder synchronicity, a known-as-correct reference picture can be used. As this reference picture is temporally further away then usual, the resulting predictively coded picture will use more bits.

Both MPEG-4 and H.263 define a binary format for the "payload" of an RPSI message that includes information such as the temporal ID of the damaged picture and the size of the damaged region. This bit string is typically small (a couple of dozen bits), of variable length, and self-contained, i.e., contains all information that is necessary to perform reference picture selection.

Both MPEG-4 and H.263 allow the use of RPSI with positive feedback information as well. That is, pictures (or Slices) are reported that were decoded without error. Note that any form of positive feedback MUST NOT be used when in a multiparty session (reporting positive feedback about individual reference pictures at RTCP intervals is not expected to be of much use anyway).

6.3.3.2. Format

The FCI for the RPSI message follows the format depicted in Figure 7:

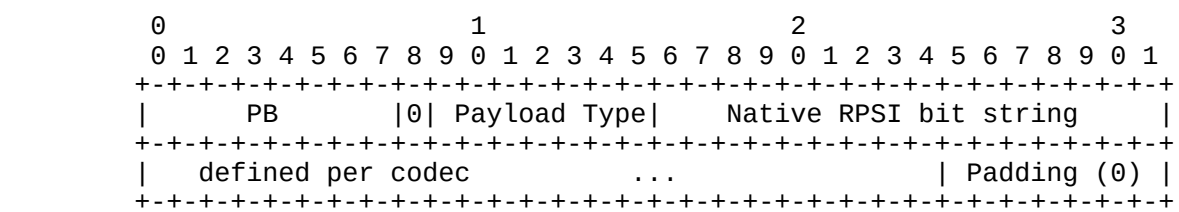

Figure 7: Syntax of the Reference Picture Selection Indication (RPSI)

PB: 8 bits
The number of unused bits required to pad the length of the RPSI message to a multiple of 32 bits.

0: 1 bit
MUST be set to zero upon transmission and ignored upon reception.

Payload Type: 7 bits
Indicates the RTP payload type in the context of which the native RPSI bit string MUST be interpreted.

Native RPSI bit string: variable length
The RPSI information as natively defined by the video codec.

Padding: #PB bits
A number of bits set to zero to fill up the contents of the RPSI message to the next 32-bit boundary. The number of padding bits MUST be indicated by the PB field.

6.3.3.3. Timing Rules

RPSI is even more critical to delay than algorithms using SLI. This is because the older the RPSI message is, the more bits the encoder has to spend to re-establish encoder-decoder synchronicity. See [15] for some information about the overhead of RPSI for certain bit rate/frame rate/loss rate scenarios.

Therefore, RPSI messages should typically be sent as soon as possible, employing the algorithm of Section 3.

6.4. Application Layer Feedback Messages

Application layer FB messages are a special case of payload-specific messages and are identified by PT=PSFB and FMT=15. There MUST be exactly one application layer FB message contained in the FCI field, unless the application layer FB message structure itself allows for stacking (e.g., by means of a fixed size or explicit length indicator).

These messages are used to transport application-defined data directly from the receiver's to the sender's application. The data that is transported is not identified by the FB message. Therefore, the application MUST be able to identify the message payload.

Usually, applications define their own set of messages, e.g., NEWPRED messages in MPEG-4 [16] (carried in RTP packets according to RFC 3016 [23]) or FB messages in H.263/Annex N, U [17] (packetized as per RFC 2429 [14]). These messages do not need any additional information from the RTCP message. Thus, the application message is simply placed into the FCI field as follows and the length field is set accordingly.

Application Message (FCI): variable length
   This field contains the original application message that should be transported from the receiver to the source. The format is application dependent. The length of this field is variable. If the application data is not 32-bit aligned, padding bits and bytes MUST be added to achieve 32-bit alignment. Identification of padding is up to the application layer and not defined in this specification.

The application layer FB message specification MUST define whether or not the message needs to be interpreted specifically in the context of a certain codec (identified by the RTP payload type). If a reference to the payload type is required for proper processing, the application layer FB message specification MUST define a way to communicate the payload type information as part of the application layer FB message itself.

7. Early Feedback and Congestion Control

In the previous sections, the FB messages were defined as well as the timing rules according to which to send these messages. The way to react to the feedback received depends on the application using the feedback mechanisms and hence is beyond the scope of this document.

However, across all applications, there is a common requirement for (TCP-friendly) congestion control on the media stream as defined in [1] and [2] when operating in a best-effort network environment.

It should be noted that RTCP feedback itself is insufficient for congestion control purposes as it is likely to operate at much slower timescales than other transport layer feedback mechanisms (that usually operate in the order of RTT). Therefore, additional mechanisms are required to perform proper congestion control.

A congestion control algorithm that shares the available bandwidth reasonably fairly with competing TCP connections, e.g., TFRC [7], MUST be used to determine the data rate for the media stream within the bounds of the RTP sender's and the media session's capabilities if the RTP/AVPF session is transmitted in a best-effort environment.

8. Security Considerations

RTP packets transporting information with the proposed payload format are subject to the security considerations discussed in the RTP specification [1] and in the RTP/AVP profile specification [2]. This profile does not specify any additional security services.

This profile modifies the timing behavior of RTCP and eliminates the minimum RTCP interval of five seconds and allows for earlier feedback to be provided by receivers. Group members of the associated RTP session (possibly pretending to represent a large number of entities) may disturb the operation of RTCP by sending large numbers of RTCP packets thereby reducing the RTCP bandwidth available for Regular RTCP reporting as well as for Early FB messages. (Note that an entity need not be a member of a multicast group to cause these effects.) Similarly, malicious members may send very large RTCP messages, thereby increasing the avg_rtcp_size variable and reducing the effectively available RTCP bandwidth.

Feedback information may be suppressed if unknown RTCP feedback packets are received. This introduces the risk of a malicious group member reducing Early feedback by simply transmitting payload-specific RTCP feedback packets with random contents that are not recognized by any receiver (so they will suppress feedback) or by the sender (so no repair actions will be taken).

A malicious group member can also report arbitrary high loss rates in the feedback information to make the sender throttle the data transmission and increase the amount of redundancy information or take other action to deal with the pretended packet loss (e.g., send fewer frames or decrease audio/video quality). This may result in a degradation of the quality of the reproduced media stream.

Finally, a malicious group member can act as a large number of group members and thereby obtain an artificially large share of the Early feedback bandwidth and reduce the reactivity of the other group members -- possibly even causing them to no longer operate in Immediate or Early feedback mode and thus undermining the whole purpose of this profile.

Senders as well as receivers SHOULD behave conservatively when observing strange reporting behavior. For excessive failure reporting from one or a few receivers, the sender MAY decide to no longer consider this feedback when adapting its transmission behavior for the media stream. In any case, senders and receivers SHOULD still adhere to the maximum RTCP bandwidth but make sure that they are capable of transmitting at least regularly scheduled RTCP packets. Senders SHOULD carefully consider how to adjust their transmission bandwidth when encountering strange reporting behavior; they MUST NOT increase their transmission bandwidth even if ignoring suspicious feedback.

Attacks using false RTCP packets (Regular as well as Early ones) can be avoided by authenticating all RTCP messages. This can be achieved by using the AVPF profile together with the Secure RTP profile as defined in [22]; as a prerequisite, an appropriate combination of those two profiles (an "SAVPF") is being specified [21]. Note that, when employing group authentication (as opposed to source authentication), the aforementioned attacks may be carried out by malicious or malfunctioning group members in possession of the right keying material.

9. IANA Considerations

The following contact information shall be used for all registrations included here:

Contact: Joerg Ott
mailto:jo@acm.org
tel:+358-9-451-2460

The feedback profile as an extension to the profile for audio-visual conferences with minimal control has been registered for the Session Description Protocol (specifically the type "proto"): "RTP/AVPF".

SDP Protocol ("proto"):

Name:	RTP/AVPF
Long form:	Extended RTP Profile with RTCP-based Feedback
Type of name:	proto
Type of attribute:	Media level only
Purpose:	RFC 4585
Reference:	RFC 4585

SDP Attribute ("att-field"):

Attribute name:	rtcp-fb
Long form:	RTCP Feedback parameter
Type of name:	att-field
Type of attribute:	Media level only
Subject to charset:	No
Purpose:	RFC 4585
Reference:	RFC 4585
Values:	See this document and registrations below

A new registry has been set up for the "rtcp-fb" attribute, with the following registrations created initially: "ack", "nack", "trr-int", and "app" as defined in this document.

Initial value registration for the attribute "rtcp-fb"

Value name:	ack
Long name:	Positive acknowledgement
Reference:	RFC 4585.

Value name:	nack
Long name:	Negative Acknowledgement
Reference:	RFC 4585.

Value name:	trr-int
Long name:	Minimal receiver report interval
Reference:	RFC 4585.

Value name:	app
Long name:	Application-defined parameter
Reference:	RFC 4585.

Further entries may be registered on a first-come first-serve basis. Each new registration needs to indicate the parameter name and the syntax of possible additional arguments. For each new registration, it is mandatory that a permanent, stable, and publicly accessible document exists that specifies the semantics of the registered parameter, the syntax and semantics of its parameters as well as

corresponding feedback packet formats (if needed). The general registration procedures of [3] apply.

For use with both "ack" and "nack", a joint sub-registry has been set up that initially registers the following values:

Initial value registration for the attribute values "ack" and "nack":

Value name:	sli
Long name:	Slice Loss Indication
Usable with:	nack
Reference:	RFC 4585.

Value name:	pli
Long name:	Picture Loss Indication
Usable with:	nack
Reference:	RFC 4585.

Value name:	rpsi
Long name:	Reference Picture Selection Indication
Usable with:	ack, nack
Reference:	RFC 4585.

Value name:	app
Long name:	Application layer feedback
Usable with:	ack, nack
Reference:	RFC 4585.

Further entries may be registered on a first-come first-serve basis. Each registration needs to indicate the parameter name, the syntax of possible additional arguments, and whether the parameter is applicable to "ack" or "nack" feedback or both or some different "rtcp-fb" attribute parameter. For each new registration, it is mandatory that a permanent, stable, and publicly accessible document exists that specifies the semantics of the registered parameter, the syntax and semantics of its parameters as well as corresponding feedback packet formats (if needed). The general registration procedures of [3] apply.

Two RTCP Control Packet Types: for the class of transport layer FB messages ("RTPFB") and for the class of payload-specific FB messages ("PSFB"). Per Section 6, RTPFB=205 and PSFB=206 have been added to the RTCP registry.

RTP RTCP Control Packet types (PT):

```
  Name:          RTPFB
  Long name:     Generic RTP Feedback
  Value:         205
  Reference:     RFC 4585.

  Name:          PSFB
  Long name:     Payload-specific
  Value:         206
  Reference:     RFC 4585.
```

As AVPF defines additional RTCP payload types, the corresponding "reserved" RTP payload type space (72-76, as defined in [2]), has been expanded accordingly.

A new sub-registry has been set up for the FMT values for both the RTPFB payload type and the PSFB payload type, with the following registrations created initially:

Within the RTPFB range, the following two format (FMT) values are initially registered:

```
  Name:          Generic NACK
  Long name:     Generic negative acknowledgement
  Value:         1
  Reference:     RFC 4585.

  Name:          Extension
  Long name:     Reserved for future extensions
  Value:         31
  Reference:     RFC 4585.
```

Within the PSFB range, the following five format (FMT) values are initially registered:

```
  Name:          PLI
  Long name:     Picture Loss Indication
  Value:         1
  Reference:     RFC 4585.

  Name:          SLI
  Long name:     Slice Loss Indication
  Value:         2
  Reference:     RFC 4585.
```

```
Name:           RPSI
Long name:      Reference Picture Selection Indication
Value:          3
Reference:      RFC 4585.

Name:           AFB
Long name:      Application Layer Feedback
Value:          15
Reference:      RFC 4585.

Name:           Extension
Long name:      Reserved for future extensions.
Value:          31
Reference:      RFC 4585.
```

Further entries may be registered following the "Specification Required" rules as defined in RFC 2434 [9]. Each registration needs to indicate the FMT value, if there is a specific FB message to go into the FCI field, and whether or not multiple FB messages may be stacked in a single FCI field. For each new registration, it is mandatory that a permanent, stable, and publicly accessible document exists that specifies the semantics of the registered parameter as well as the syntax and semantics of the associated FB message (if any). The general registration procedures of [3] apply.

10. Acknowledgements

This document is a product of the Audio-Visual Transport (AVT) Working Group of the IETF. The authors would like to thank Steve Casner and Colin Perkins for their comments and suggestions as well as for their responsiveness to numerous questions. The authors would also like to particularly thank Magnus Westerlund for his review and his valuable suggestions and Shigeru Fukunaga for the contributions on FB message formats and semantics.

We would also like to thank Andreas Buesching and people at Panasonic for their simulations and the first independent implementations of the feedback profile.

11. References

11.1. Normative References

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

[2] Schulzrinne, H. and S. Casner, "RTP Profile for Audio and Video Conferences with Minimal Control", STD 65, RFC 3551, July 2003.

[3] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006.

[4] Casner, S., "Session Description Protocol (SDP) Bandwidth Modifiers for RTP Control Protocol (RTCP) Bandwidth", RFC 3556, July 2003.

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

[6] Turletti, T. and C. Huitema, "RTP Payload Format for H.261 Video Streams", RFC 2032, October 1996.

[7] Handley, M., Floyd, S., Padhye, J., and J. Widmer, "TCP Friendly Rate Control (TFRC): Protocol Specification", RFC 3448, January 2003.

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

[9] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 2434, October 1998.

11.2. Informative References

[10] Camarillo, G., Eriksson, G., Holler, J., and H. Schulzrinne, "Grouping of Media Lines in the Session Description Protocol (SDP)", RFC 3388, December 2002.

[11] Perkins, C. and O. Hodson, "Options for Repair of Streaming Media", RFC 2354, June 1998.

[12] Rosenberg, J. and H. Schulzrinne, "An RTP Payload Format for Generic Forward Error Correction", RFC 2733, December 1999.

[13] 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, September 1997.

[14] Bormann, C., Cline, L., Deisher, G., Gardos, T., Maciocco, C., Newell, D., Ott, J., Sullivan, G., Wenger, S., and C. Zhu, "RTP Payload Format for the 1998 Version of ITU-T Rec. H.263 Video (H.263+)", RFC 2429, October 1998.

[15] B. Girod, N. Faerber, "Feedback-based error control for mobile video transmission", Proceedings IEEE, Vol. 87, No. 10, pp. 1707 - 1723, October, 1999.

[16] ISO/IEC 14496-2:2001/Amd.1:2002, "Information technology - Coding of audio-visual objects - Part2: Visual", 2001.

[17] ITU-T Recommendation H.263, "Video Coding for Low Bit Rate Communication", November 2000.

[18] Schulzrinne, H. and S. Petrack, "RTP Payload for DTMF Digits, Telephony Tones and Telephony Signals", RFC 2833, May 2000.

[19] Kohler, E., Handley, M., and S. Floyd, "Datagram Congestion Control Protocol (DCCP)", RFC 4340, March 2006.

[20] Handley, M., Floyd, S., Padhye, J., and J. Widmer, "TCP Friendly Rate Control (TFRC): Protocol Specification", RFC 3448, January 2003.

[21] Ott, J. and E. Carrara, "Extended Secure RTP Profile for RTCP-based Feedback (RTP/SAVPF)", Work in Progress, December 2005.

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

[23] Kikuchi, Y., Nomura, T., Fukunaga, S., Matsui, Y., and H. Kimata, "RTP Payload Format for MPEG-4 Audio/Visual Streams", RFC 3016, November 2000.

[24] ITU-T Recommendation H.245, "Control protocol for multimedia communication", May 2006.

Authors' Addresses

Joerg Ott
Helsinki University of Technology (TKK)
Networking Laboratory
PO Box 3000
FIN-02015 TKK
Finland

EMail: jo@acm.org

Stephan Wenger
Nokia Research Center
P.O. Box 100
33721 Tampere
Finland

EMail: stewe@stewe.org

Noriyuki Sato
Oki Electric Industry Co., Ltd.
1-16-8 Chuo, Warabi-city, Saitama 335-8510
Japan

Phone: +81 48 431 5932
Fax: +81 48 431 9115
EMail: sato652@oki.com

Carsten Burmeister
Panasonic R&D Center Germany GmbH

EMail: carsten.burmeister@eu.panasonic.com

Jose Rey
Panasonic R&D Center Germany GmbH
Monzastr. 4c
D-63225 Langen, Germany

EMail: jose.rey@eu.panasonic.com

Full Copyright Statement

Intellectual Property

Acknowledgement

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).