INTERNET-DRAFT
draft-ietf-avt-rtcp-feedback-03.txt

Joerg Ott/Uni Bremen TZI
Stephan Wenger/TU Berlin
Shigeru Fukunaga/Oki
Noriyuki Sato/Oki
Koichi Yano/Fast Forward Networks
Akihiro Miyazaki/Matsushita
Koichi Hata/Matsushita
Rolf Hakenberg/Matsushita
Carsten Burmeister/Matsushita
Jose Rey/Matsushita

29 June 2002
Expires December 2002

Extended RTP Profile for RTCP-based Feedback (RTP/AVPF)

Status of this Memo

Abstract

Real-time media streams that use RTP are not resilient against packet losses. Receivers may use the base mechanisms of RTCP to report packet reception statistics and thus allow a sender to adapt its transmission behavior in the mid-term as 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 allow 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.

1. Introduction

Real-time media streams that use RTP are not 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 QoS. However, except for a few payload specific mechanisms [10], RTP makes no provision for timely feedback that would allow a sender to repair the media stream immediately: through retransmissions, retro-active 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 [7], video redundancy coding [11], RTP-level FEC [5], and general considerations on more robust media streams transmission [6]. These mechanisms may be applied pro-actively (thereby increasing the bandwidth of a given media stream). Alternatively, in sufficiently small groups with small 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: to achieve timely feedback, the concepts of Immediate Feedback messages and 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. A small number of general-purpose feedback messages as well as a format for codec and application-specific feedback information are defined as specific RTCP payloads.

1.1 Definitions

The definitions from [1] and [2] apply. In addition, the following definitions are used in this document:

Early RTCP mode:
The mode of operation in which a receiver of a media stream is, statistically, often (but not always) capable of reporting events of interest back to the sender close to their occurrence. In Early RTCP mode, RTCP feedback messages 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.

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 which is carried in RTCP RRs -- back to the sender of the media stream.

Feedback (FB) threshold:
The FB threshold indicates the transition between Immediate Feedback and Early RTCP mode. For a multicast 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.

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 feedback messages are transmitted according to the timing rules defined in this document.

Regular RTCP mode:
Mode of operation in which no preferred transmission of feedback 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.

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

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 [8]

2. RTP and RTCP Packet Formats and Protocol Behavior

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 to convey feedback information are defined in section 6.

RTCP report intervals:
This memo describes three modes of operation which influence the RTCP report intervals (see section 3.2). In regular RTCP mode, all rules from [1] apply. In both Immediate Feedback and Early RTCP modes the minimal interval of five seconds between two RTCP reports is dropped and the rules specified in section 3 apply if RTCP packets containing feedback messages (defined in section 4) 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 5, further consideration is given to the impact of feedback and a sender's reaction to feedback messages.

3. Rules for RTCP Feedback

3.1 Compound RTCP Feedback Packets

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

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

- Feedback 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 feedback messages are newly introduced in this memo.

RTCP Feedback (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.

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

- OPTIONAL encryption prefix that MUST be present if the RTCP message is to be encrypted.
- MANDATORY SR or RR.
- MANDATORY SDES which MUST contain the CNAME item; all other SDES items are OPTIONAL.
- 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 feedback 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 feedback message is sent as part of an Early RTCP packet.

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 feedback message is sent as part of a regularly scheduled RTCP packet or in Regular RTCP mode. It MAY also be used to send RTCP feedback messages in Immediate Feedback or Early RTCP mode.

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 are thus subject to the same bandwidth constraints as other RTCP traffic. 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 media packets) to accommodate algorithms that use FB messages and are sensitive to the feedback timing.

The modified RTCP transmission algorithm can be outlined as follows: Normally, when 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 and 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; in addition, a minimum interval between regular RTCP packets may be enforced. If a receiver detects the need to send an FB message, the receiver waits for a short, random dithering interval (in case of multicast) and then checks whether it has already seen a corresponding FB message from any other receiver (which it can do with all FB messages that are transmitted via multicast; for unicast sessions, there is no such delay). If this is the case then the receiver refrains from sending the FB message and continues to follow the regular RTCP transmission schedule. If the receiver has not yet seen a similar FB message from any other receiver, it checks whether it has recently sent another FB message (without waiting for its regularly scheduled RTCP transmission time). Only if this is not the case, it sends the FB message as part of a (minimal) compound RTCP packet.

FB messages may also be sent as part of full compound RTCP packets which are interspersed 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 is a hint whether or not a receiver should send early feedback at all and, if so, whether, statistically, all events observed at the receiver can be reported back to the sender in a timely fashion. The current mode of operation is continuously derived independently at each receiver and the receivers do not have to agree on a common mode.

a) Immediate feedback 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 it is supposed/expected to 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 -- again depending on the type of FB used -- the (worst case or observed) frequency of events to report (e.g. frame received, packet lost).

A special case of this is the ACK mode (where positive acknowledgements are used to confirm reception of data) which is restricted to point-to-point communications.

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 (or needed) to report. 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 reproduced media 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) From some group size upwards, it is no longer useful to provide feedback from individual 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 group size threshold can be specified at which this mode starts.

As the feedback algorithm described in this memo scales smoothly, there is no need for an agreement among the participants on the precise values of the respective "thresholds" within the group. Hence the borders between all these modes are allowed to be fluent.

```
     ACK
   feedback
     V
     :<- - - -  NACK feedback - - - ->//
     :
     :   Immediate   ||
     : Feedback mode ||Early RTCP mode   Regular RTCP mode
     :<=============>||<=============>//<=================>
     :               ||
    -+---------------||---------------//------------------> group size
     2               ||
      Application-specific FB Threshold
         = f(data rate, packet loss, codec, ...)
```

Figure 1: Modes of operation

As stated before, the respective 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

The following pieces of state information need to be maintained per receiver (largely taken from [1]). Note that all variables (except for g) are calculated independently at each receiver and so 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 reconsideration.

d) Let Tmin be the minimal interval between RTCP packets as per [1]. Unlike [1], the initial Tmin is set to 1 second, then it 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 following the rules of [1]: T_rr = T (the "calculated interval") with tn = tp + T. Note that a different Tmin is used to compute the "calculated

interval T". T_rr refers to the last value of T that has been computed (because of reconsideration or to determine tn).

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).

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. Note that this value is application-specific.

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 feedback message in response to an event that a certain packet P caused.

k) Let allow_early be a Boolean variable that indicates whether the receiver currently may transmit feedback 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 adjusted (set to FALSE) after early feedback transmission and is reset to TRUE as soon as the next regular RTCP transmission has occurred.

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 regular RTCP packets will not be scheduled T_rr after the last regular RTCP transmission (at tp+T_rr) but only at least T_rr_interval after the last regular RTCP transmission (later than or at tp+T_rr). T_rr_interval does not affect the calculation of T_rr and tp but may lead to regular RTCP packets being suppressed. T_rr_interval does not affect transmission scheduling for Early RTCP packets.

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

NOTE: Providing T_rr_interval as an independent variable is meant to minimize regular feedback (and thus bandwidth consumption) as needed by the application but still allow for more frequent use of 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 Early feedback.

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 a feedback message needs to be sent back to the sender.

To avoid an implosion of immediate feedback packets, the receiver MUST delay the transmission of the RTCP feedback packet by a random amount 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 (which, in turn, is based upon the group size).

For a certain application scenario, a receiver may determine an upper bound for the acceptable local delay of feedback 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 RTCP feedback packet is scheduled, the time slot for the next scheduled (full) compound RTCP packet MUST be updated accordingly to a new tn (which will then be in the order of tn=tp+2*T_rr). This is to ensure that the short term average bandwidth used for RTCP with feedback does not exceed the bandwidth limit that would be used without feedback.

```
          event to
          report
          detected
             |
             |  RTCP feedback range
             |   (T_max_fb_delay)
             vXXXXXXXXXXXXXXXXXXXXXXXXXXX     ) )
   |---+--------+-------------+-----+------------| |--------+--->
       |        |             |     |            ( (        |
       |       t0            te                             |
      tp                                                   tn
                 \_______   ________/
                         \/
                T_dither_max
```

Figure 2: Event report and parameters for Early RTCP scheduling

3.5 Early RTCP Algorithm

Assume an active sender S0 (out of S senders) and a number N of receivers with R being one of these receivers.

Assume further 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 memo).

Assume 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.

Then, receiver R MUST use the following rules for transmitting one or more Feedback messages as minimal or full compound RTCP packet:

3.5.1 Initialization

Initially, R MUST set allow_early = TRUE and t_rr_last = NaN.

Furthermore, the initialization of the RTCP variables as per [1] applies except that the initial value for Tmin is 1.0 seconds.

3.5.2 Early Feedback Transmission

Assume that R has scheduled the last RTCP RR packet for transmission at tp and has scheduled the next transmission (including possible reconsideration) for tn = tp + T_rr. Assume

also that the last T_rr_interval-based transmission (if any) has occurred at t_rr_last (if defined).

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

2. R first checks whether there is still a compound RTCP feedback packet waiting for transmission (scheduled as early or regular RTCP packet).

2.a) If so, the new feedback message MUST be appended to the packet; the scheduling of the waiting RTCP feedback packet MUST remain unchanged. When appending, the feedback information of several RTCP feedback packets SHOULD be merged to produce as few feedback messages as possible. This completes the course of immediate actions to be taken.

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

i) If the session is a unicast session (group size = 2) then T_dither_max = 0.

ii) If the session is a multicast session with potentially more than two group members then

T_dither_max = l * T_rr

with l=0.5.

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 regularly scheduled RTCP packet is within the time bounds for the RTCP FB (t0 + T_dither_max > tn).

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

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

4. R MUST check whether it is allowed to transmit an Early RTCP packet (allow_early == TRUE).

   4.a) If allow_early == FALSE then R MUST check the time for the next scheduled RR:

      1. If tn - t0 < T_max_fb_delay (i.e. if, despite late reception, the feedback could still be useful for the sender) then R MAY create an RTCP FB message for transmission along with the RTCP packet at tn.

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

   This completes the immediate course of actions to be taken.

   4.b) 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. If, while waiting for te, R receives RTCP feedback packets contained in one or more (minimal) compound RTCP packets, R MUST act as follows for each of the RTCP feedback messages in the one or more compound RTCP packets received in the interval [t0 - 1s ; te]:

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

   5.b) If R understands the received feedback message's semantics and the message contents is not a superset of the feedback R wanted to send then R SHOULD transmit its own feedback 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 send its feedback information unchanged, R MAY as well eliminate any redundancy between its own feedback and the feedback received so far.

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

   Note: With rule #3, receiving unknown feedback packets may not lead to feedback suppression at a particular receiver. As a consequence, a given event may cause M different types of feedback packets (which are all appropriate but not the same

and mutually not understood) to be scheduled, and a "large" receiver group may be partitioned into at most M groups. Among members of each of these M groups, feedback suppression will occur following the rules #1 and #2 but no suppression will happen across groups. As a result, O(M) RTCP feedback messages may be received by the sender. Given that these M groups consist of receivers for the same application using the same (set of) codecs in the same RTP session, M is assumed to be small in the general case. Given further that the O(M) feedback packets are randomly distributed over a time interval of T_dither_max, the resulting limited number of extra feedback packets (a) is assumed not to overwhelm the sender and (b) should be conveyed as all contain complementary pieces of information.

Refer to section 4 on the comparison of feedback messages and for which feedback messages MUST be understood by a receiver.

6. Otherwise, when te is reached, R MUST transmit the RTCP packet containing the FB message. 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 and R sends its next regularly scheduled RTCP RR or suppresses it because of T_rr_interval, it MUST set allow_early = TRUE again.

3.5.3 Regular RTCP Transmission

In regular intervals full compound RTCP packets MUST be sent. These packets MAY also contain one or more feedback messages. Transmission of regular RTCP packets is scheduled as follows:

If T_rr_interval == 0 then the transmission MUST follow the rules as specified by [1] (except for the different Tmin) 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] and the regular RTCP packet is transmitted after timer reconsideration. Whenever a regular RTCP message is sent, allow_early MUST be set to TRUE and tp, tn MUST be updated as per [1]. If this was the first transmission of an 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 [1] (except for the different Tmin) 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:

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

If t_rr_last + T_rr_interval <= tn then a full compound RTCP packet MUST be scheduled. Stored RTCP feedback messages MAY be included in the full compound RTCP packet. t_rr_last MUST be set to tn.

If t_rr_last + T_rr_interval > tn and RTCP feedback messages have been stored and are awaiting transmission, an RTCP packet MUST be scheduled. This RTCP packet MAY be a minimal or a full compound RTCP packet (at the discretion of the implementer) and the compound RTCP packet MUST include the stored RTCP feedback message. t_rr_last MUST remain unchanged.

Otherwise (if t_rr_last + T_rr_interval > tn but no stored RTCP feedback messages are awaiting transmission), no compound RTCP packet MUST be scheduled.

In all the four cases above, allow early MUST be set to TRUE and tp and tn MUST be updated following the rules of [1] except for the five second minimum.

3.5.4 Other Considerations

Furthermore, 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.

Whenever an RTCP packet is sent or received -- minimal or full compound, early or regularly scheduled -- the avg_rtcp_size variable MUST be updated accordingly (see [1]) and the tn MUST be calculated using 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 group size MUST be exactly two participants, i.e. point-to-point communications. Unicast addresses MUST be used in the session description.

For unidirectional as well as bi-directional communication between two parties, 2.5% of the RTP session bandwidth are 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 acknowledgments for 10 events are collected in each feedback 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 RTCP feedback 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 (or similar types of feedback) MUST be used for all groups larger than two. Of course, NACKs MAY be used for point-to-point communications as well.

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

The crucial parameters -- to which virtually all of the above can be reduced -- is the allowed minimal interval between two RTCP reports 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.

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 is used 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.

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 256kbit/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 its own 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 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 9.6kbit/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 packets, this yields a maximum group size of 6-7 receivers if all loss events shall be reported. The rules for transmission of immediate 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 second 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, of course, that all those 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 obviously be based upon (and dynamically revised following) regular 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 provides valuable input to this step, too.

3) If these tests pass, the application has to follow the rules for transmitting Early RTCP packets or regularly scheduled RTCP packets with piggybacked feedback.

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 mechanisms consists of 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, SAP, 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 MAY be used (=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 profile 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 (for use with "a=fmtp:") is defined to indicate the capability of using RTCP feedback as specified in this document: "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 feedback messages MAY be used in this media session for the indicated payload type. If several types of feedback are supported, several "a=rtcp-fb:" lines MUST be used.

If no "rtcp-fb" attribute is specified the RTP receivers SHOULD assume that the RTP senders only support generic NACKs. In addition, 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 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"

- o 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=fmtp:rtcp-fb line);

o 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

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

When used in conjunction with the offer/answer model [18], 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.

RTP senders MUST be prepared to receive any kind of RTCP feedback messages and MUST silently discard all those RTCP feedback 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, SP and CRLF are used as defined in [3].)

```
rtcp-fb-syntax = "a=fmtp:" <format> SP "rtcp-fb" SP rtcp-fb-val
CRLF

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
                   | ; empty

rtcp-fb-ack-param  = SP "rpsi"
                   | SP "app"
                   | SP byte-string
                   | ; empty

rtcp-fb-nack-param = SP "pli"
                   | SP "sli"
                   | SP "rpsi"
                   | SP "app"
                   | 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 3.6.1.2.

Parameters MAY be provided to further distinguish different types of positive acknowledgement feedback. If no parameters are present, the Generic ACK as specified in section 6.2.2 is implied.

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 General 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":

o "pli" indicates the use of Picture Loss Indication feedback as defined in section 6.3.1.
o "sli" indicates the use of Slice Loss Indication feedback as defined in section 6.3.2.
o "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) to make use of feedback provided.

4.3 Unicasting vs. Multicasting

If a media session description indicates unicast addresses for a particular media type (and does not operate in multi-unicast mode with all recipients listed explicitly but still addressed via unicast), the RTCP feedback MAY operate in ACK feedback mode.

If a media session description indicates multicast addresses for a particular media type or a multi-unicast session, ACK feedback mode MUST NOT be used.

4.4 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).

4.5 Examples

Example 1: The following session description indicates a session made up from an audio and a DTMF for point-to-point communication in which the DTMF stream uses Generic ACKs. 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=fmtp:96 rtcp-fb ack
```

Example 2: The following session description indicates a multicast video-only session (using H.263+) with the video source accepting Generic NACKs and Reference Picture Selection. 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
```

```
c=IN IP4 224.2.1.184
a=rtpmap:98 H263-1998/90000
a=fmtp:98 rtcp-fb nack
a=fmtp:98 rtcp-fb nack rpsi
```

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
c=IN IP4 224.2.1.184
a=rtpmap:98 H263-1998/90000
m=video 51372 RTP/AVPF 98
c=IN IP4 224.2.1.184
a=rtpmap:98 H263-1998/90000
a=fmtp:98 rtcp-fb nack
a=fmtp:98 rtcp-fb nack rpsi
```

Note that these two m= lines SHOULD be grouped by some appropriate mechanisms to indicate that both are alternatives actually conveying the same contents. A sample mechanism by which this can be achieved is defined in [14].

5. Interworking and Co-Existence 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 of using either of the two profiles can be mixed in a single session (see e.g. 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.

o 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 under-estimate the group size. Therefore, if AVP entities may be involved in a media session, T_rr_interval SHOULD NOT be used.

o 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 sub-optimal.

o AVPF receivers

AVPF receivers SHOULD send immediate or 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 classified into three categories as follows:

- Transport layer feedback messages
- Payload-specific feedback messages
- Application layer feedback messages

Transport layer feedback 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 general positive acknowledgement (ACK) and a negative acknowledgement (NACK) message are defined.

Payload-specific feedback 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 feedback messages. The definition of specific messages is left to either RTP payload format specifications or to additional feedback format documents.

Application layer feedback 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's 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 feedback messages. From a protocol point of view, an application layer feedback message is treated as a special case of a payload-specific feedback message.

This document defines two transport layer feedback and three (video) payload-specific feedback messages as well as a single container for application layer feedback messages. Additional transport layer and payload specific feedback messages MAY be defined in other documents and MUST be registered through IANA (see section IANA considerations).

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

6.1 Common Packet Format for Feedback Message

All feedback message MUST use a common packet format that is depicted in figure 3:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|V=2|P|0|  FMT  |       PT      |          length               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  SSRC of packet sender                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  SSRC of media source                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:            Feedback Control Information (FCI)                 :
:                                                               :
```

Figure 3: Common Packet Format for Feedback Messages

The various 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 which are not part of the control information but are included in the length field.

Feedback message type (FMT): 4 bits
    This field identifies the type of the feedback message and is interpreted relative to the RTCP message type (transport, 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 which identifies the packet as being an RTCP Feedback Message. Two values are defined (TBA. by IANA):

Name	Value	Brief Description
RTPFB	205	Transport layer feedback message
PSFB	206	Payload-specific feedback 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 feedback message for each type of feedback (further FCI contents MAY be specified in further documents). Each RTCP feedback packet MUST contain exactly one FCI field of the types defined in sections 6.2 and 6.3. If multiple FCI fields (even of the same type) need to be conveyed, then several RTCP feedback packets MUST be generated and concatenated in the same compound RTCP packet.

6.2 Transport Layer Feedback Messages

Transport Layer Feedback messages are identified by the value RTPFB as RTCP message type.

Two general purpose transport layer feedback messages are defined so far: General ACK and General NACK. They are identified by means of the FMT parameter as follows:

0: reserved
1: Generic NACK
2: Generic ACK
3-15: reserved

The following two subsections define the packet formats for these messages.

6.2.1 Generic NACK

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

The Generic NACK packet 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.

The Feedback control information (FCI) field has the following Syntax (figure 4):

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |            PID                |             BLP               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 4: Syntax for the Generic NACK message

Packet ID (PID): 16 bits

The PID field is used to specify a lost packet. Typically, the RTP sequence number is used for PID as the default format, but RTP Payload Formats may decide to identify a packet differently.

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 [10]. 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.

6.2.2 Generic ACK

The Generic ACK message is identified by PT=RTPFB and FMT=2.

The Generic ACK packet is used to indicate that one or several RTP packets were received correctly. The received packet(s) are identified by the means of a packet identifier and a bit mask. ACKing of a range of consecutive packets is also possible.

The Feedback control information (FCI) field has the following syntax:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              PID              |R|       BLP/#packets          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 5: Syntax for the Generic ACK message

Packet ID (1st PID): 16 bits
This PID field is used to specify a correctly received packet. Typically, the RTP sequence number is used for PID as the default format, but RTP Payload Formats may decide to identify a packet differently.

Range of ACKs (R): 1 bit
The R-bit indicates that a range of consecutive packets are received correctly. If R=1 then the PID field specifies the first packet of that range and the next field (BLP/#packets) will carry the number of packets being acknowledged. If R=0 then PID specifies the first packet to be acknowledged and BLP/#packets provides a bit mask to selectively indicate individual packets that are acknowledged.

Bit mask of lost packets (BLP)/#packets (PID): 15 bits
The semantics of this field depends on the value of the R-bit.

If R=1, this field is used to identify the number of additional packets of to be acknowledged:

    #packets = <highest seq# to be ACKed> - <PID>

That is, #packets MUST indicate the number of packet to be ACKed minus one. In particular, if only a single packet is to be ACKed and R=1 then #packets MUST be set to 0x0000.

Example: If all packets between and including PIDx = 380 and PIDy = 422 have been received, the Generic ACK would contain PID = PIDx = 380 and #packets = PIDy - PID = 42. In case the PID wraps around, modulo arithmetic is used to calculate the number of packets.

If R=0, this field carries a bit mask. The BLP allows for reporting reception of any of the 15 RTP packets immediately following the RTP packet indicated by the PID. The BLP's definition is identical to that given in [10] except that, here, BLP is only 15 bits wide. Denoting the BLP's least significant bit as bit 1, and its most significant bit as bit 15, then bit i of the bitmask is set to 1 if the receiver has received RTP packet number PID+i (modulo 2^16) and decides to

ACK this packet; bit i is set to 0 otherwise. If only the packet indicated by PID is to be ACKed and R=0 then BLP MUST be set to 0x0000.

6.3 Payload Specific Feedback Messages

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

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

```
0:    reserved
1:    Picture Loss Indication (PLI)
2:    Slice Lost Indication (SLI)
3:    Reference Picture Selection Indication (RPSI)
4-14: reserved
15:   Application layer feedback message
```

The following subsections define the packet formats for the payload-specific messages, section 6.4 defines the application layer feedback message.

6.3.1 Picture Loss Indication (PLI)

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

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 effectively similar to the FIR as defined in [10]); 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 [10] 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.

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 feedback message. Hence waiting for the next possible time slot allowed by RTCP timing rules as per [2] does not have a negative impact on the system performance.

6.3.2 Slice Lost Indication (SLI)

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

6.3.2.1 Semantics

With the Slice Lost 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 feedback 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 Lost Indication uses one additional PCI field the content of which is depicted in figure 6. The length of the feedback message MUST be set to 3.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            First        |  Number                 | PictureID |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 6: Syntax of the Slice Lost 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 a 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..

6.3.2.3 Timing Rules

The efficiency of algorithms using the Slice Lost 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" exist. 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 H.245. 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 feedback 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 a SLI is not part of this specification. One typical way of reacting to a 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 [13] and [15]). While, when those algorithms are used, the timing of the FB is less critical then without, 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 feedback message is identified by PT=PSFB and FMT=3.

6.3.3.1 Semantics

Modern video coding standards such as MPEG-4 visual version 2 [12] or H.263 version 2 [13] allow to use 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.

Note that 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 multicast environment (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:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      PB       |   Native RPSI bit string defined per codec    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              ...                |  Padding (0)|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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.

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

RPS is even more critical to delay then algorithms using SLI. This is due to the fact that the older the RPS 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 RPS for certain bit rate/frame rate/loss rate scenarios.

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

6.4 Application Layer Feedback Messages

Application Layer Feedback Messages are a special case of payload-specific messages and identified by PT=PSFB and FMT=15.

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 feedback message. Therefore, the application MUST be able to identify the messages payload.

Usually, applications define their own set of messages, e.g. NEWPRED messages in MPEG-4 or feedback messages in H.263/Annex N, U. 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 byte aligned, padding bits must be added. Identification of padding bits is up to the application layer and not defined in this specification.

7. Early Feedback and Congestion Control

In the previous sections, the feedback 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.

Low delay feedback supports the use of congestion control algorithms in two ways:

- o The potentially more frequent RTCP messages allow the sender to monitor the network state more closely than with regular RTCP and therefore enable reacting to upcoming congestion in a more timely fashion.

- o The feedback messages themselves may convey additional information as input to congestion control algorithms and thus improve reaction over conventional RTCP. (For example, ACK-based feedback may even allow to construct closed loop algorithms and NACK-based systems may provide further information on the packet loss distribution.)

A congestion control algorithm that shares the available bandwidth fair with competing TCP connections, e.g. TFRC [16], SHOULD be used to determine the data rate for the media stream (if the low delay RTP session is transmitted in a best effort environment).

RTCP feedback messages or RTCP SR/RR packets that indicate recent packet loss MUST NOT lead to a (mid-term) increase in the transmission data rate and SHOULD lead to a (short-term) decrease of the transmission data rate. Such messages SHOULD cause the sender to adjust the transmission data rate to the order of the throughput TCP would achieve under similar conditions (e.g. using TFRC).

RTCP feedback messages or RTCP SR/RR packets that indicate no recent packet loss MAY cause the sender to increase the transmission data rate to roughly the throughput TCP would achieve under similar conditions (e.g. using TFRC).

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 feedback messages. (Note that an entity need not be member of a multicast group to cause these effects.)

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 neither recognized by any receiver (so they will suppress feedback) nor 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 conservative 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 [17].

9. IANA Considerations

The feedback profile as an extension to the profile for audio-visual conferences with minimal control needs to be registered: "RTP/AVPF".

For the Session Description Protocol, the following "fmtp:" attribute needs to be registered: "rtcp-fb".

Along with "rtcp-fb", the feedback types "ack" and "nack" need to be registered. Also, the value "trr-int" needs to be registered.

Along with "nack", the feedback type parameters "sli" and "pli" need to be registered.

Along with "ack" and "nack", the feedback type parameters "rpsi" and "app" need to be registered.

Two RTCP Control Packet Types: for the class of transport layer feedback messages ("RTPFB") and for the class of payload-specific feedback messages ("PSFB"). Section 6 suggests RTPFB=205 and PSFB=206 to be added to the RTCP registry.

Within the RTPFB range, three format (FMT) values need to be registered:

```
 1:    General NACK
 2:    General ACK
```

Within the PSFB range, five format (FMT) values need to be registered:

```
 1:    Picture Loss Indication (PLI)
 2:    Slice Loss Indication (SLI)
 3:    Reference Picture Selection Indication (SLI)
15:    Application layer feedback (AFB)
```

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.

11. Full Copyright Statement

12. Authors' Addresses

Joerg Ott {sip,mailto}:jo@tzi.org
Uni Bremen TZI
MZH 5180
Bibliothekstr. 1
D-28359 Bremen
Germany

Stephan Wenger stewe@cs.tu-berlin.de
TU Berlin
Sekr. FR 6-3
Franklinstr. 28-29
D-10587 Berlin
Germany

Shigeru Fukunaga
Oki Electric Industry Co., Ltd.
1-2-27 Shiromi, Chuo-ku, Osaka 540-6025 Japan
Tel. +81 6 6949 5101
Fax. +81 6 6949 5108
Mail fukunaga444@oki.com

Noriyuki Sato
Oki Electric Industry Co., Ltd.
1-2-27 Shiromi, Chuo-ku, Osaka 540-6025 Japan
Tel. +81 6 6949 5101
Fax. +81 6 6949 5108
Mail sato652@oki.com

Koichi Yano
FastForward Networks,
75 Hawthorne St. #601
San Francisco, CA 94105
Tel. +1.415.430.2500

Akihiro Miyazaki
Matsushita Electric Industrial Co., Ltd
1006, Kadoma, Kadoma City, Osaka, Japan
Tel. +81-6-6900-9192
Fax. +81-6-6900-9193
Mail akihiro@isl.mei.co.jp

Koichi Hata
Matsushita Electric Industrial Co., Ltd
1006, Kadoma, Kadoma City, Osaka, Japan
Tel. +81-6-6900-9192
Fax. +81-6-6900-9193
Mail hata@isl.mei.co.jp

Rolf Hakenberg
Panasonic European Laboratories GmbH
Monzastr. 4c, 63225 Langen, Germany
Tel. +49-(0)6103-766-162
Fax. +49-(0)6103-766-166
Mail hakenberg@panasonic.de

Carsten Burmeister
Panasonic European Laboratories GmbH
Monzastr. 4c, 63225 Langen, Germany
Tel. +49-(0)6103-766-263
Fax. +49-(0)6103-766-166
Mail burmeister@panasonic.de

11. Bibliography

[1] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP - A Transport Protocol for Real-time Applications," Internet Draft, draft-ietf-avt-rtp-new-11.txt, Work in Progress, November 2001.

[2] H. Schulzrinne and S. Casner, "RTP Profile for Audio and Video Conferences with Minimal Control," Internet Draft draft-ietf-avt-profile-new-12.txt, November 2001.

[3] M. Handley, V. Jacobson, and Colin Perkins, "SDP: Session Description Protocol", Internet Draft draft-ietf-mmusic-sdp-new-10.txt, May 2002.

[4] S. Casner, "SDP Bandwidth Modifiers for RTCP Bandwidth", Internet Draft draft-ietf-avt-rtcp-bw-05.txt, November 2001.

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

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

[7] C. Perkins, I. Kouvelas, O. Hodson, V. Hardman, M. Handley, J.C. Bolot, A. Vega-Garcia, and S. Fosse-Parisis, "RTP Payload for Redundant Audio Data," RFC 2198, September 1997.

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

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

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

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

[12] ISO/IEC 14496-2:1999/Amd.1:2000, "Information technology - Coding of audio-visual objects - Part2: Visual", July 2000.

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

[14] G. Camarillo, J. Holler, G. Eriksson, H. Schulzrinne, "Grouping of media lines in SDP," Internet Draft, draft-ietf-mmusic-fid-05.txt, Work in Progress, September 2001.

[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] M. Handley, J. Padhye, S. Floyd, J. Widmer, "TCP friendly Rate Control (TFRC): Protocol Specification," Internet Draft, draft-ietf-tsvwg-tfrc-03.txt, Work in Progress, July 2001.

[17] M. Baugher, R. Blom, E. Carrarra, D. McGrew, M. Naslund, K. Norrman, D. Oran, "The Secure Real-Time Transport Protocol," Internet Draft, draft-ietf-avt-srtp-04.txt, Work in Progress, May 2002.

[18] J. Rosenberg and H. Schulzrinne, "An offer/answer model with SDP," Internet Draft draft-ietf-mmusic-sdp-offer-answer-02.txt, February 2002.