

31 October 2002  
Expires April 2003

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

### Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of [Section 10 of RFC 2026](#). Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at  
<http://www.ietf.org/ietf/1id-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at  
<http://www.ietf.org/shadow.html>.

### 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:

Ott et al.

Expires April 2003

[Page 3]

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:

- o 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.
- o 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 [\[1\]](#):

- o OPTIONAL encryption prefix that MUST be present if the RTCP message is to be encrypted.
- o 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 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  $N \leq B \cdot T/R$ .

Ott et al.

Expires April 2003

[Page 7]

- 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  $N > B \cdot T / R$  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 soft.

```

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  $t_n$  and  $t_p$  be the time for the next (last) scheduled RTCP RR transmission calculated prior to reconsideration.
- d) Let  $T_{min}$  be the minimal interval between RTCP packets as per [\[1\]](#). Unlike [\[1\]](#), the initial  $T_{min}$  is set to 1 second (to allow for some group size sampling before sending the first RTCP packet), 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  $t_n = t_p + T$ . Note that  $T_{min}$  as defined in this specification 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  $t_n$ ).
- f) Let  $t_0$  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). For point-to-point sessions,  $T_{dither\_max}$  is set to 0 and hence no additional delay is introduced.
- 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  $t_e$  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 at time  $t_0$ .
- 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  $t_n$ . 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  $T_{rr\_interval} \neq 0$  then regular RTCP packets will not be scheduled  $T_{rr}$  after the last regular RTCP transmission (at  $t_p + T_{rr}$ ) but only at least  $T_{rr\_interval}$  after the last regular RTCP transmission (later than or at  $t_p + T_{rr}$ ).  $T_{rr\_interval}$  does not affect the calculation of  $T_{rr}$  and  $t_p$  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.

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

The feedback situation for an event to report at a receiver is depicted in figure 2 below. At time  $t_0$ , 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 in multicast sessions, 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  $t_e = t_0 + T_{fd}$ .

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  $t_n$  (which will then be in the order of  $t_n = t_p + 2 * T_{rr}$ ). 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.

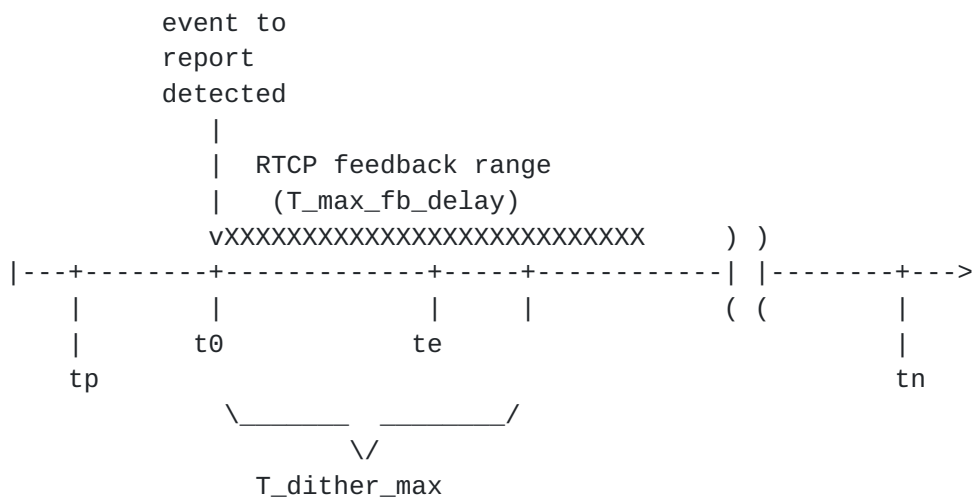


Figure 2: Event report and parameters for Early RTCP scheduling

### 3.5 Early RTCP Algorithm

Assume an active sender  $S_0$  (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  $T_{min}$ . For a unicast session, the initial  $T_{min}$  is set to 0. For a multicast session,  $T_{min}$  is initialized to 1.0 seconds.

### 3.5.2 Early Feedback Transmission

Assume that R has scheduled the last RTCP RR packet for transmission at  $t_p$  and has scheduled the next transmission (including possible reconsideration) for  $t_n = t_p + T_{rr}$ . Assume also that the last  $T_{rr\_interval}$ -based transmission (if any) has occurred at  $t_{rr\_last}$  (if defined).

1. At time  $t_0$ , 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 already a compound RTCP packet containing an RTCP feedback message scheduled for transmission (as early or regular RTCP packet).

- 2.a) If so, the new feedback message MUST be included in the scheduled packet; the scheduling of the waiting RTCP feedback packet MUST remain unchanged. When doing so, 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 scheduled for 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 would be within the time bounds for the RTCP FB ( $t_0 +$   
 $T\_dither\_max > t_n$ ).

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  $t_n$ . 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  $t_n - t_0 < T\_max\_fb\_delay$  (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  $t_n$ .

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  $t_e = t_0 + RND * T\_dither\_max$  with RND being a pseudo  
random function evenly distributed between 0 and 1.

5. R MUST continuously monitor the received RTCP feedback packets  
contained in one or more (minimal) compound RTCP packets and keep  
each of these packets for at least  $T\_retention$ . When scheduling  
the transmission of an RTCP feedback message, R MUST check each of





the RTCP feedback messages in the one or more compound RTCP packets received in the interval  $[t_0 - T_{\text{retention}}; t_e]$  and act as follows:

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  $t_n$  (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  $t_n$  (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  $T_{\text{dither\_max}}$ , 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  $t_e$  is reached, R MUST transmit the RTCP packet

containing the FB message. R then MUST set allow\_early = FALSE,

Ott et al.

Expires April 2003

[Page 14]

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  $T_{min}$ ) 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,  $T_{min}$  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  $T_{min}$ ) 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  $tn$ .  $T_{min}$  MUST be set to 0.

Otherwise, an 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 for the following alternatives:

If  $t_{rr\_last} + T_{rr\_current\_interval} \leq tn$  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  $t\_rr\_last + T\_rr\_current\_interval > tn$  and RTCP feedback 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 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  $t\_rr\_last + T\_rr\_current\_interval > tn$  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  $T\_rr\_interval \neq 0$  then the timeout calculation for RTP/AVPF entities (section 6.3.5 of [1]) MUST be modified to use  $T\_rr\_interval$  instead of  $T_{min}$  for computing  $T_d$ .

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 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 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  $N \leq B \cdot T / R$ .

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  $N > B \cdot T / R$  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 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.

Ott et al.

Expires April 2003

[Page 18]

- 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 mechanism 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 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 feedback 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 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=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=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 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=rtcp-fb:96 ack
```

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
```

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

Ott et al.

Expires April 2003

[Page 24]

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 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 larger than five seconds .

#### 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 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.

NOTE: Proper processing of some feedback messages at the media sender side may require the sender to know which payload type the feedback 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 DTFM) or when codec changes occur, the payload type information may need to be conveyed explicitly as part of the feedback message. This applies to all payload-specific as well as application layer feedback messages. It is up to the specification of a feedback message to define how payload type information is transmitted.

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:

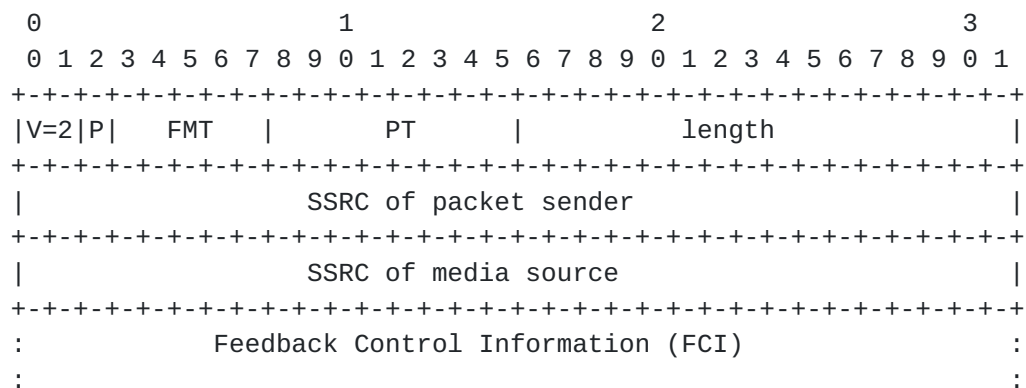


Figure 3: Common Packet Format for Feedback Messages

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): 5 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 at least one feedback message in the FCI field. Sections 6.2 and 6.3 define for each FCI type, whether or not multiple feedback messages MAY be contained in a single FCI field. If multiple feedback messages are contained in one FCI field, they MUST be of the same type as. If multiple types of feedback need to be conveyed, then several RTCP feedback packets MUST be generated and SHOULD be 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:	unassigned
1:	Generic NACK
2:	Generic ACK
3-30:	unassigned
31:	reserved for future expansion of the sequence number space

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

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):

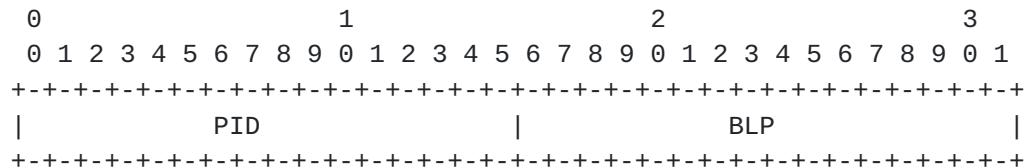


Figure 4: Syntax for the Generic NACK message

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.

The length of the feedback 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).



The Generic ACK message is identified by PT=RTPFB and FMT=2. The FCI field MUST contain at least one and MAY contain more than one Generic ACK.

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



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.

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.

The semantics of this field depends on the value of the R-bit.

```
#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  $PID_x = 380$  and  $PID_y = 422$  have been received, the Generic ACK would contain  $PID = PID_x = 380$  and  $\#packets = PID_y - PID_x = 42$ . 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.

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

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

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

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. 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 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.

The semantics of this feedback 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 than predicted (inter)









MUST be set to '0' upon transmission and MUST be ignored upon reception.

Payload Type: 7 bits

This field contains the Payload Type of the RTP packet (i.e. the codec) the SLI message refers to.

MBZ: 8 bits

MUST be set to '0' upon transmission and MUST be ignored upon reception.

First: 16 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: 16 bits

The number of lost macroblocks, in scan order as discussed above.

PictureID: 16 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 than 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. 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 [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 than 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:

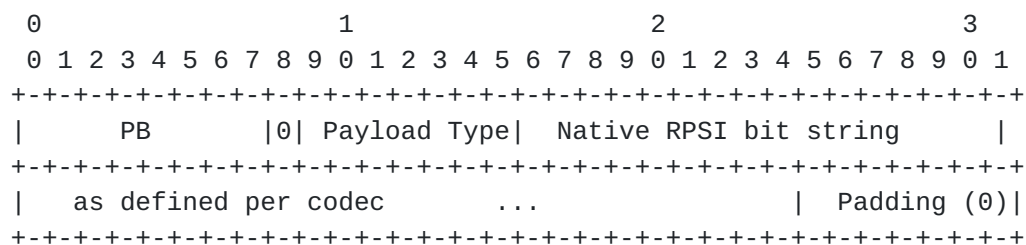


Figure 7: Syntax of the Reference Picture Selection Indication (RPSI)

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: 8 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**

RPS is even more critical to delay than 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. There MUST be exactly one Application Layer Feedback message contained in the FCI field.

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 32-bit-aligned, padding bits and bytes must be added. Identification of padding is up to the application layer and not defined in this specification.

The application layer feedback 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 feedback message specification MUST define a way to communicate the payload type information as part of the application layer feedback message itself.

## **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]; as a prerequisite, an appropriate combination of those two profiles (an "SAVPF") needs to be specified.

## 9. IANA Considerations

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

Contact: Joerg Ott  
mailto:jo@acm.org  
tel:+49-421-201-7028

The feedback profile as an extension to the profile for audio-visual conferences with minimal control needs to be 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: See this document  
Reference: This document

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: See this document  
Reference: This document  
Values: See this document and registrations below

A new registry needs to be 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: This document.

Value name: nack  
Long name: Negative Acknowledgement  
Reference: This document.

Value name: trr-int  
Long name: Minimal receiver report interval  
Reference: This document.

Value name: app  
Long name: Application-defined paramater  
Reference: This document.

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 needs to be 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: This document.

Value name: pli  
Long name: Picture Loss Indication  
Usable with: nack  
Reference: This document.

Value name: rpsi  
Long name: Reference Picture Selection Indication  
Usable with: ack, nack  
Reference: This document.

Value name: app  
Long name: Application layer feedback  
Usable with: ack, nack  
Reference: This document.



Further entries may be registered on a first-come first serve basis. Each registrations 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 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.

RTP RTCP Control Packet types (PT):

Name: RTPFB  
Long name: Generic RTP Feedback  
Value: 205  
Reference: This document.

Name: PSFB  
Long name: Payload-specific  
Value: 206  
Reference: This document.

As AVPF defines additional RTCP payload types, the corresponding "reserved" RTP payload type space (72--76, as defined in [\[2\]](#)), needs to be expanded accordingly (to cover the range 72--78).

A new sub-registry needs to be 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 three format (FMT) values are initially registered:

Name: Generic NACK  
Long name: Generic negative acknowledgement  
Value: 1  
Reference: This document.

Name: Generic ACK  
Long name: Generic positive acknowledgement  
Value: 2  
Reference: This document.



Name: Extension  
Long name: Reserved for future extensions  
Value: 31  
Reference: This document.

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

Name: PLI  
Long name: Picture Loss Indication  
Value: 1  
Reference: This document.

Name: SLI  
Long name: Slice Loss Indication  
Value: 2  
Reference: This document.

Name: RPSI  
Long name: Reference Picture Selection Indication  
Value: 3  
Reference: This document.

Name: AFB  
Long name: Application Layer Feedback  
Value: 1  
Reference: This document.

Name: Extension  
Long name: Reserved for future extensions.  
Value: 31  
Reference: This document.

Further entries may be registered on a first-come first serve basis. Each registration needs to indicate the FMT value, if there is a specific feedback message to go into the FCI field, and whether or not multiple feedback 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 feedback 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 thank Magnus Westerlund for his review and his valuable suggestions; Shigeru Fukunaga, Koichi Yano, Akihiro Miyazaki, and Rolf Hakenberg for the contributions on for feedback message formats and semantics; and Andreas Buesching for his feedback based upon implementation and testing.

## **11. Full Copyright Statement**

Copyright (C) The Internet Society (2001). All Rights Reserved. This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works.

However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE."





## **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  
Skr. FR 6-3  
Franklinstr. 28-29  
D-10587 Berlin  
Germany

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

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

Jose Rey  
Panasonic European Laboratories GmbH  
Monzastr. 4c, 63225 Langen, Germany  
Tel. +49-(0)6103-766-134  
Fax. +49-(0)6103-766-166  
Mail hakenberg@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. Turlatti 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-05.txt](#), Work in Progress, June 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.