                          Extended RTP Profile for
           Real-time Transport Control Protocol (RTCP)-Based Feedback:
                      Results of the Timing Rule Simulations

Status of This Memo

Copyright Notice

Abstract

   This document describes the results achieved when simulating the
   timing rules of the Extended RTP Profile for Real-time Transport
   Control Protocol (RTCP)-Based Feedback, denoted AVPF. Unicast and
   multicast topologies are considered as well as several protocol and
   environment configurations. The results show that the timing rules
   result in better performance regarding feedback delay and still
   preserve the well-accepted RTP rules regarding allowed bit rates for
   control traffic.

Table of Contents

1. Introduction

   The Real-time Transport Protocol (RTP) is widely used for the
   transmission of real-time or near real-time media data over the
   Internet. While it was originally designed to work well for
   multicast groups in very large scales, its scope is not limited to
   that. More and more applications use RTP for small multicast groups
   (e.g., video conferences) or even unicast (e.g., IP telephony and
   media streaming applications).

   RTP comes together with its companion protocol Real-time Transport
   Control Protocol (RTCP), which is used to monitor the transmission of
   the media data and provide feedback of the reception quality.
   Furthermore, it can be used for loose session control. Having the
   scope of large multicast groups in mind, the rules regarding when to
   send feedback were carefully restricted to avoid feedback explosion
   or feedback-related congestion in the network. RTP and RTCP have
   proven to work well in the Internet, especially in large multicast
   groups, which is shown by their widespread usage today.

   However, the applications that transmit the media data only to small
   multicast groups or unicast may benefit from more frequent feedback.
   The source of the packets may be able to react to changes in the
   reception quality, which may be due to varying network utilization
   (e.g., congestion) or other changes. Possible reactions include
   transmission rate adaptation according to a congestion control
   algorithm or the invocation of error resilience features for the
   media stream (e.g., retransmissions, reference picture selection,
   NEWPRED, etc.).

   As mentioned before, more frequent feedback may be desirable to
   increase the reception quality, but RTP restricts the use of RTCP
   feedback. Hence it was decided to create a new extended RTP profile,
   which redefines some of the RTCP timing rules, but keeps most of the
   algorithms for RTP and RTCP, which have proven to work well. The new
   rules should scale from unicast to multicast, where unicast or small
   multicast applications have the most gain from it. A detailed
   description of the new profile and its timing rules can be found in
   [1].

   This document investigates the new algorithms by the means of
   simulations. We show that the new timing rules scale well and behave
   in a network-friendly manner. Firstly, the key features of the new
   RTP profile that are important for our simulations are roughly
   described in Section 2. After that, we describe in Section 3 the
   environment that is used to conduct the simulations. Section 4
   describes simulation results that show the backwards compatibility to
   RTP and that the new profile is network-friendly in terms of used

bandwidth for RTCP traffic. In Section 5, we show the benefit that
applications could get from implementing the new profile. In Section
6, we investigated the effect of the parameter "l" (used to calculate
the T_dither_max value) upon the algorithm performance, and finally,
in Section 7, we show the performance gain we could get for a special
application, namely, NEWPRED in [6] and [7].

2. Timing Rules of the Extended RTP Profile for RTCP-Based Feedback

    As said above, RTP restricts the usage of RTCP feedback. The main
    restrictions on RTCP are as follows:

    - RTCP messages are sent in compound packets, i.e., every RTCP packet
      contains at least one sender report (SR) or receiver report (RR)
      message and a source description (SDES) message.

    - The RTCP compound packets are sent in time intervals (T_rr), which
      are computed as a function of the average packet size, the number
      of senders and receivers in the group, and the session bandwidth
      (5% of the session bandwidth is used for RTCP messages; this
      bandwidth is shared between all session members, where the senders
      may get a larger share than the receivers.)

    - The average minimum interval between two RTCP packets from the same
      source is 5 seconds.

    We see that these rules prevent feedback explosion and scale well to
    large multicast groups. However, they do not allow timely feedback
    at all. While the second rule scales also to small groups or unicast
    (in this cases the interval might be as small as a few milliseconds),
    the third rule may prevent the receivers from sending feedback
    timely.

    The timing rules to send RTCP feedback from the new RTP profile [1]
    consist of two key components. First, the minimum interval of 5
    seconds is abolished. Second, receivers get one chance during every
    other of their (now quite small) RTCP intervals to send an RTCP
    packet "early", i.e., not according to the calculated interval, but
    virtually immediately. It is important to note that the RTCP
    interval calculation is still inherited from the original RTP
    specification.

    The specification and all the details of the extended timing rules
    can be found in [1]. Rather than describing the algorithms here, we
    reference the original specification [1]. Therefore, we use also the
    same variable names and abbreviations as in [1].

3. Simulation Environment

   This section describes the simulation testbed that was used for the
   investigations and its key features. The extensions to the simulator
   that were necessary are roughly described in the following sections.

3.1. Network Simulator Version 2

   The simulations were conducted using the network simulator version 2
   (ns2). ns2 is an open source project, written in a combination of
   Tool Command Language (TCL) and C++. The scenarios are set up using
   TCL. Using the scripts, it is possible to specify the topologies
   (nodes and links, bandwidths, queue sizes, or error rates for links)
   and the parameters of the "agents", i.e., protocol configurations.
   The protocols themselves are implemented in C++ in the agents, which
   are connected to the nodes. The documentation for ns2 and the newest
   version can be found in [4].

3.2. RTP Agent

   We implemented a new agent, based on RTP/RTCP. RTP packets are sent
   at a constant packet rate with the correct header sizes. RTCP
   packets are sent according to the timing rules of [2] and [3], and
   also its algorithms for group membership maintenance are implemented.
   Sender and receiver reports are sent.

   Further, we extended the agent to support the extended profile [1].
   The use of the new timing rules can be turned on and off via
   parameter settings in TCL.

3.3. Scenarios

   The scenarios that are simulated are defined in TCL scripts. We set
   up several different topologies, ranging from unicast with two
   session members to multicast with up to 25 session members.
   Depending on the sending rates used and the corresponding link
   bandwidths, congestion losses may occur. In some scenarios, bit
   errors are inserted on certain links. We simulated groups with
   RTP/AVP agents, RTP/AVPF agents, and mixed groups.

   The feedback messages are generally NACK messages as defined in [1]
   and are triggered by packet loss.

3.4. Topologies

   Mainly, four different topologies are simulated to show the key
   features of the extended profile. However, for some specific
   simulations we used different topologies. This is then indicated in
   the description of the simulation results. The main four topologies
   are named after the number of participating RTP agents, i.e., T-2,
   T-4, T-8, and T-16, where T-2 is a unicast scenario, T-4 contains
   four agents, etc. Figure 1 below illustrates the main topologies.
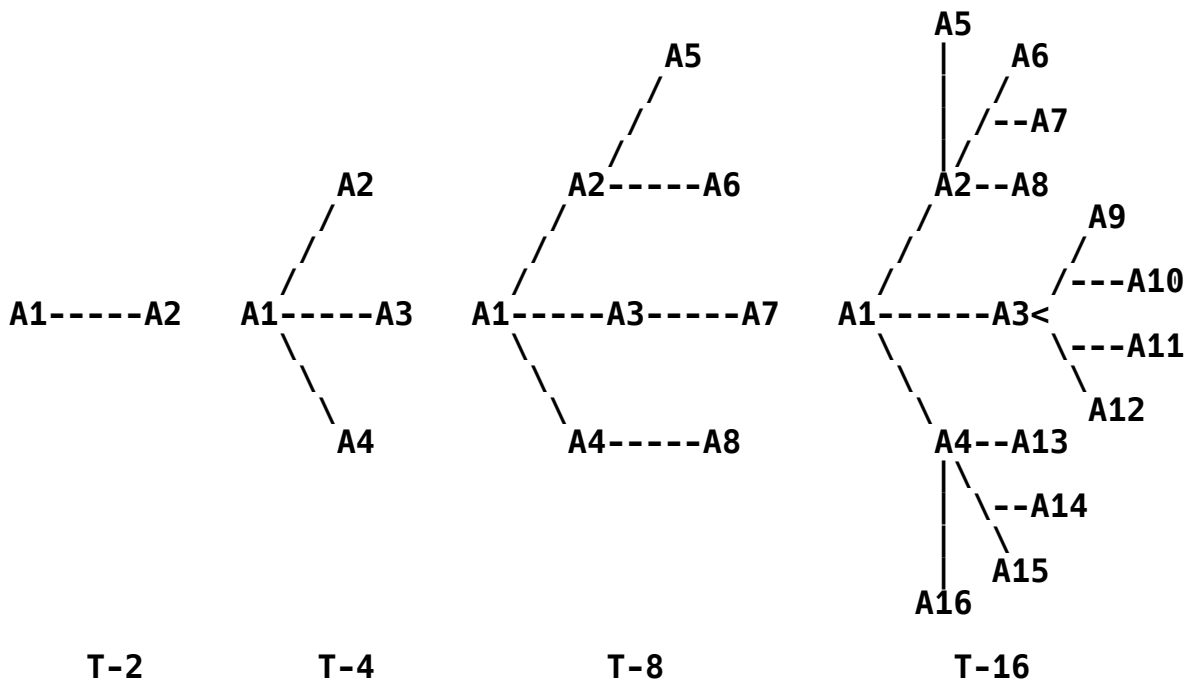
```
                                                            A5
                                               A5           |   A6
                                               /            |  /
                                              /             | /--A7
                           A2           A2-----A6           |/
                           /            /                   A2--A8
                          /            /                    /        A9
                         /            /                    /        /
     A1-----A2     A1-----A3     A1-----A3-----A7     A1------A3<   /---A10
                     \            \                    \        \---A11
                      \            \                    \         \
                       \            \                    \        A12
                        A4           A4-----A8           A4--A13
                                                         |    \--A14
                                                         |      \
                                                         |      A15
                                                         A16

        T-2            T-4            T-8                T-16
```

                      Figure 1: Simulated topologies

4. RTCP Bit Rate Measurements

   The new timing rules allow more frequent RTCP feedback for small
   multicast groups. In large groups, the algorithm behaves similarly
   to the normal RTCP timing rules. While it is generally good to
   have more frequent feedback, it cannot be allowed at all to
   increase the bit rate used for RTCP above a fixed limit, i.e., 5%
   of the total RTP bandwidth according to RTP. This section shows
   that the new timing rules keep RTCP bandwidth usage under the 5%
   limit for all investigated scenarios, topologies, and group sizes.
   Furthermore, we show that mixed groups (some members using
   AVP, some AVPF) can be allowed and that each session member behaves

fairly according to its corresponding specification. Note that
other values for the RTCP bandwidth limit may be specified using
the RTCP bandwidth modifiers as in [10].

4.1. Unicast

First we measured the RTCP bandwidth share in the unicast topology
T-2. Even for a fixed topology and group size, there are several
protocol parameters that are varied to simulate a large range of
different scenarios. We varied the configurations of the agents
in the sense that the agents may use AVP or AVPF. Thereby it
is possible that one agent uses AVP and the other AVPF in one RTP
session. This is done to test the backwards compatibility of the
AVPF profile.

Next, we consider scenarios where no losses occur. In this case,
both RTP session members transmit the RTCP compound packets at
regular intervals, calculated as T_rr, if they use AVPF, and
use a minimum interval of 5 seconds (on average) if they implement
AVP. No early packets are sent, because the need to send early
feedback is not given. Still it is important to see that not more
than 5% of the session bandwidth is used for RTCP and that AVP and
AVPF members can coexist without interference. The results can
be found in Table 1.

Session Bandwidth	Send Agents	Rec. Agents	AVP Agents	AVPF Agents	Used RTCP Bit Rate (% of session bw)		
					A1	A2	sum
2 Mbps	1	2	-	1,2	2.42	2.56	4.98
2 Mbps	1,2	-	-	1,2	2.49	2.49	4.98
2 Mbps	1	2	1	2	0.01	2.49	2.50
2 Mbps	1,2	-	1	2	0.01	2.48	2.49
2 Mbps	1	2	1,2	-	0.01	0.01	0.02
2 Mbps	1,2	-	1,2	-	0.01	0.01	0.02
200 kbps	1	2	-	1,2	2.42	2.56	4.98
200 kbps	1,2	-	-	1,2	2.49	2.49	4.98
200 kbps	1	2	1	2	0.06	2.49	2.55
200 kbps	1,2	-	1	2	0.08	2.50	2.58
200 kbps	1	2	1,2	-	0.06	0.06	0.12
200 kbps	1,2	-	1,2	-	0.08	0.08	0.16
20 kbps	1	2	-	1,2	2.44	2.54	4.98
20 kbps	1,2	-	-	1,2	2.50	2.51	5.01
20 kbps	1	2	1	2	0.58	2.48	3.06
20 kbps	1,2	-	1	2	0.77	2.51	3.28
20 kbps	1	2	1,2	-	0.58	0.61	1.19
20 kbps	1,2	-	1,2	-	0.77	0.79	1.58

Table 1: Unicast simulations without packet loss

We can see that in configurations where both agents use the new
timing rules each of them uses, at most, about 2.5% of the session
bandwidth for RTP, which sums up to 5% of the session bandwidth for
both. This is achieved regardless of the agent being a sender or a
receiver. In the cases where agent A1 uses AVP and agent A2 AVPF,
the total RTCP session bandwidth decreases. This is because agent A1
can send RTCP packets only with an average minimum interval of 5
seconds. Thus, only a small fraction of the session bandwidth is
used for its RTCP packets. For a high-bit-rate session (session
bandwidth = 2 Mbps), the fraction of the RTCP packets from agent A1
is as small as 0.01%. For smaller session bandwidths, the fraction
increases because the same amount of RTCP data is sent. The
bandwidth share that is used by RTCP packets from agent A2 is not
different from what was used, when both agents implemented the AVPF.
Thus, the interaction of AVP and AVPF agents is not problematic in
these scenarios at all.

In our second unicast experiment, we show that the allowed RTCP
bandwidth share is not exceeded, even if packet loss occurs. We
simulated a constant byte error rate (BYER) on the link. The byte
errors are inserted randomly according to a uniform distribution.

Packets with byte errors are discarded on the link; hence the
receiving agents will not see the loss immediately. The agents
detect packet loss by a gap in the sequence number.

When an AVPF agent detects a packet loss, the early feedback
procedure is started. As described in AVPF [1], in unicast
T_dither_max is always zero, hence an early packet can be sent
immediately if allow_early is true. If the last packet was already
an early one (i.e., allow_early = false), the feedback might be
appended to the next regularly scheduled receiver report. The
max_feedback_delay parameter (which we set to 1 second in our
simulations) determines if that is allowed.

The results are shown in Table 2, where we can see that there is no
difference in the RTCP bandwidth share, whether or not losses occur.
This is what we expected, because even though the RTCP packet size
grows and early packets are sent, the interval between the packets
increases and thus the RTCP bandwidth stays the same. Only the RTCP
bandwidth of the agents that use the AVP increases slightly. This is
because the interval between the packets is still 5 seconds (in
average), but the packet size increased because of the feedback that
is appended.

Session Bandwidth	Send Agents	Rec. Agents	AVP Agents	AVPF Agents	Used RTCP Bit Rate (% of session bw)		
					A1	A2	sum
2 Mbps	1	2	-	1,2	2.42	2.56	4.98
2 Mbps	1,2	-	-	1,2	2.49	2.49	4.98
2 Mbps	1	2	1	2	0.01	2.49	2.50
2 Mbps	1,2	-	1	2	0.01	2.48	2.49
2 Mbps	1	2	1,2	-	0.01	0.02	0.03
2 Mbps	1,2	-	1,2	-	0.01	0.01	0.02
200 kbps	1	2	-	1,2	2.42	2.56	4.98
200 kbps	1,2	-	-	1,2	2.50	2.49	4.99
200 kbps	1	2	1	2	0.06	2.50	2.56
200 kbps	1,2	-	1	2	0.08	2.49	2.57
200 kbps	1	2	1,2	-	0.06	0.07	0.13
200 kbps	1,2	-	1,2	-	0.09	0.08	0.17
20 kbps	1	2	-	1,2	2.42	2.57	4.99
20 kbps	1,2	-	-	1,2	2.52	2.51	5.03
20 kbps	1	2	1	2	0.58	2.54	3.12
20 kbps	1,2	-	1	2	0.83	2.43	3.26
20 kbps	1	2	1,2	-	0.58	0.73	1.31
20 kbps	1,2	-	1,2	-	0.86	0.84	1.70

Table 2: Unicast simulations with packet loss

4.2. Multicast

Next, we investigated the RTCP bandwidth share in multicast
scenarios; i.e., we simulated the topologies T-4, T-8, and T-16 and
measured the fraction of the session bandwidth that was used for RTCP
packets. Again we considered different situations and protocol
configurations (e.g., with or without bit errors, groups with AVP
and/or AVPF agents, etc.). For reasons of readability, we present
only selected results. For a documentation of all results, see [5].

The simulations of the different topologies in scenarios where no
losses occur (neither through bit errors nor through congestion) show
a similar behavior as in the unicast case. For all group sizes, the
maximum RTCP bit rate share used is 5.06% of the session bandwidth in
a simulation of 16 session members in a low-bit-rate scenario
(session bandwidth = 20 kbps) with several senders. In all other
scenarios without losses, the RTCP bit rate share used is below that.
Thus, the requirement that not more than 5% of the session bit rate
should be used for RTCP is fulfilled with reasonable accuracy.

Simulations where bit errors are randomly inserted in RTP and RTCP
packets and the corrupted packets are discarded give the same
results. The 5% rule is kept (at maximum 5.07% of the session
bandwidth is used for RTCP).

Finally, we conducted simulations where we reduced the link bandwidth
and thereby caused congestion-related losses. These simulations are
different from the previous bit error simulations, in that the losses
occur more in bursts and are more correlated, also between different
agents. The correlation and "burstiness" of the packet loss is due
to the queuing discipline in the routers we simulated; we used simple
FIFO queues with a drop-tail strategy to handle congestion. Random
Early Detection (RED) queues may enhance the performance, because the
burstiness of the packet loss might be reduced; however, this is not
the subject of our investigations, but is left for future study. The
delay between the agents, which also influences RTP and RTCP packets,
is much more variable because of the added queuing delay. Still the
RTCP bit rate share used does not increase beyond 5.09% of the
session bandwidth. Thus, also for these special cases the
requirement is fulfilled.

4.3. Summary of the RTCP Bit Rate Measurements

We have shown that for unicast and reasonable multicast scenarios,
feedback implosion does not happen. The requirement that at maximum
5% of the session bandwidth is used for RTCP is fulfilled for all
investigated scenarios.

5. Feedback Measurements

   In this section we describe the results of feedback delay
   measurements, which we conducted in the simulations. Therefore, we
   use two metrics for measuring the performance of the algorithms;
   these are the "mean waiting time" (MWT) and the number of feedback
   packets that are sent, suppressed, or not allowed. The waiting time
   is the time, measured at a certain agent, between the detection of a
   packet loss event and the time when the corresponding feedback is
   sent. Assuming that the value of the feedback decreases with its
   delay, we think that the mean waiting time is a good metric to
   measure the performance gain we could get by using AVPF instead of
   AVP.

   The feedback an RTP/AVPF agent wants to send can be either sent or
   not sent. If it was not sent, this could be due to feedback
   suppression (i.e., another receiver already sent the same feedback)
   or because the feedback was not allowed (i.e., the max_feedback_delay
   was exceeded). We traced for every detected loss, if the agent sent
   the corresponding feedback or not and if not, why. The more feedback
   was not allowed, the worse the performance of the algorithm.
   Together with the waiting times, this gives us a good hint of the
   overall performance of the scheme.

5.1. Unicast

   In the unicast case, the maximum dithering interval T_dither_max is
   fixed and set to zero. This is because it does not make sense for a
   unicast receiver to wait for other receivers if they have the same
   feedback to send. But still feedback can be delayed or might not be
   permitted to be sent at all. The regularly scheduled packets are
   spaced according to T_rr, which depends in the unicast case mainly on
   the session bandwidth.

   Table 3 shows the mean waiting times (MWTs) measured in seconds for
   some configurations of the unicast topology T-2. The number of
   feedback packets that are sent or discarded is listed also (feedback
   sent (sent) or feedback discarded (disc)). We do not list suppressed
   packets, because for the unicast case feedback suppression does not
   apply. In the simulations, agent A1 was a sender and agent A2 was a
   pure receiver.

| Session | | Feedback Statistics | | | | | |
| Bandwidth | PLR | AVP | | | AVPF | | |
		sent	disc	MWT	sent	disc	MWT
2 Mbps	0.001	781	0	2.604	756	0	0.015
2 Mbps	0.01	7480	0	2.591	7548	2	0.006
2 Mbps	cong.	25	0	2.557	1741	0	0.001
20 kbps	0.001	79	0	2.472	74	2	0.034
20 kbps	0.01	780	0	2.605	709	64	0.163
20 kbps	cong.	780	0	2.590	687	70	0.162

Table 3: Feedback statistics for the unicast simulations

From the table above we see that the mean waiting time can be
decreased dramatically by using AVPF instead of AVP. While the
waiting times for agents using AVP is always around 2.5 seconds (half
the minimum interval average), it can be decreased to a few ms for
most of the AVPF configurations.

In the configurations with high session bandwidth, normally all
triggered feedback is sent. This is because more RTCP bandwidth is
available. There are only very few exceptions, which are probably
due to more than one packet loss within one RTCP interval, where the
first loss was by chance sent quite early. In this case, it might be
possible that the second feedback is triggered after the early packet
was sent, but possibly too early to append it to the next regularly
scheduled report, because of the limitation of the
max_feedback_delay. This is different for the cases with a small
session bandwidth, where the RTCP bandwidth share is quite low and
T_{rr} thus larger. After an early packet was sent, the time to the
next regularly scheduled packet can be very high. We saw that in
some cases the time was larger than the max_feedback_delay, and in
these cases the feedback is not allowed to be sent at all.

With a different setting of max_feedback_delay, it is possible to
have either more feedback that is not allowed and a decreased mean
waiting time or more feedback that is sent but an increased waiting
time. Thus, the parameter should be set with care according to the
application's needs.

5.2. Multicast

In this section, we describe some measurements of feedback statistics
in the multicast simulations. We picked out certain characteristic
and representative results. We considered the topology T-16.
Different scenarios and applications are simulated for this topology.
The parameters of the different links are set as follows. The agents
A2, A3, and A4 are connected to the middle node of the multicast

tree, i.e., agent A1, via high bandwidth and low-delay links. The
other agents are connected to the nodes 2, 3, and 4 via different
link characteristics. The agents connected to node 2 represent
mobile users. They suffer in certain configurations from a certain
byte error rate on their access links and the delays are high. The
agents that are connected to node 3 have low-bandwidth access links,
but do not suffer from bit errors. The last agents, which are
connected to node 4, have high bandwidth and low delay.

5.2.1. Shared Losses vs. Distributed Losses

   In our first investigation, we wanted to see the effect of the loss
   characteristic on the algorithm's performance. We investigate the
   cases where packet loss occurs for several users simultaneously
   (shared losses) or totally independently (distributed losses). We
   first define agent A1 to be the sender. In the case of shared
   losses, we inserted a constant byte error rate on one of the middle
   links, i.e., the link between A1 and A2. In the case of distributed
   losses, we inserted the same byte error rate on all links downstream
   of A2.

   These scenarios are especially interesting because of the feedback
   suppression algorithm. When all receivers share the same loss, it is
   only necessary for one of them to send the loss report. Hence if a
   member receives feedback with the same content that it has scheduled
   to be sent, it suppresses the scheduled feedback. Of course, this
   suppressed feedback does not contribute to the mean waiting times.
   So we expect reduced waiting times for shared losses, because the
   probability is high that one of the receivers can send the feedback
   more or less immediately. The results are shown in the following
   table.

	Feedback Statistics									
	Shared Losses					Distributed Losses				
Agent	sent	fbsp	disc	sum	MWT	sent	fbsp	disc	sum	MWT
A2	274	351	25	650	0.267	-	-	-	-	-
A5	231	408	11	650	0.243	619	2	32	653	0.663
A6	234	407	9	650	0.235	587	2	32	621	0.701
A7	223	414	13	650	0.253	594	6	41	641	0.658
A8	188	443	19	650	0.235	596	1	32	629	0.677

             Table 4: Feedback statistics for multicast simulations

   Table 4 shows the feedback statistics for the simulation of a large
   group size. All 16 agents of topology T-16 joined the RTP session.
   However, only agent A1 acts as an RTP sender; the other agents are
   pure receivers. Only 4 or 5 agents suffer from packet loss, i.e.,

A2, A5, A6, A7, and A8 for the case of shared losses and A5, A6, A7,
and A8 in the case of distributed losses. Since the number of
session members is the same for both cases, T_rr is also the same on
the average. Still the mean waiting times are reduced by more than
50% in the case of shared losses. This proves our assumption that
shared losses enhance the performance of the algorithm, regardless of
the loss characteristic.

The feedback suppression mechanism seems to be working quite well.
Even though some feedback is sent from different receivers (i.e.,
1150 loss reports are sent in total and only 650 packets were lost,
resulting in loss reports being received on the average 1.8 times),
most of the redundant feedback was suppressed. That is, 2023 loss
reports were suppressed from 3250 individual detected losses, which
means that more than 60% of the feedback was actually suppressed.

6. Investigations on "l"

In this section, we want to investigate the effect of the parameter
"l" on the T_dither_max calculation in RTP/AVPF agents. We
investigate the feedback suppression performance as well as the
report delay for three sample scenarios.

For all receivers, the T_dither_max value is calculated as
T_dither_max = l * T_rr, with l = 0.5. The rationale for this is
that, in general, if the receiver has no round-trip time (RTT)
estimation, it does not know how long it should wait for other
receivers to send feedback. The feedback suppression algorithm would
certainly fail if the time selected is too short. However, the
waiting time is increased unnecessarily (and thus the value of the
feedback is decreased) in case the chosen value is too large.
Ideally, the optimum time value could be found for each case, but
this is not always feasible. On the other hand, it is not dangerous
if the optimum time is not used. A decreased feedback value and a
failure of the feedback suppression mechanism do not hurt the network
stability. We have shown for the cases of distributed losses that
the overall bandwidth constraints are kept in any case and thus we
could only lose some performance by choosing the wrong time value.
On the other hand, a good measure for T_dither_max is the RTCP
interval T_rr. This value increases with the number of session
members. Also, we know that we can send feedback at least every
T_rr. Thus, increasing T_dither max beyond T_rr would certainly make
no sense. So by choosing T_rr/2, we guarantee that at least
sometimes (i.e., when a loss is detected in the first half of the
interval between two regularly scheduled RTCP packets) we are allowed
to send early packets. Because of the randomness of T_dither, we
still have a good chance of sending the early packet in time.

The AVPF profile specifies that the calculation of T_dither_max, as
given above, is common to session members having an RTT estimation
and to those not having it. If this were not so, participants using
different calculations for T_dither_max might also have very
different mean waiting times before sending feedback, which
translates into different reporting priorities. For example, in a
scenario where T_rr = 1 s and the RTT = 100 ms, receivers using the
RTT estimation would, on average, send more feedback than those not
using it. This might partially cancel out the feedback suppression
mechanism and even cause feedback implosion. Also note that, in a
general case where the losses are shared, the feedback suppression
mechanism works if the feedback packets from each receiver have
enough time to reach each of the other ones before the calculated
T_dither_max seconds. Therefore, in scenarios of very high bandwidth
(small T_rr), the calculated T_dither_max could be much smaller than
the propagation delay between receivers, which would translate into a
failure of the feedback suppression mechanism. In these cases, one
solution could be to limit the bandwidth available to receivers (see
[10]) such that this does not happen. Another solution could be to
develop a mechanism for feedback suppression based on the RTT
estimation between senders. This will not be discussed here and may
be the subject of another document. Note, however, that a really
high bandwidth media stream is not that likely to rely on this kind
of error repair in the first place.

In the following, we define three representative sample scenarios.
We use the topology from the previous section, T-16. Most of the
agents contribute only little to the simulations, because we
introduced an error rate only on the link between the sender A1 and
the agent A2.

The first scenario represents those cases, where losses are shared
between two agents. One agent is located upstream on the path
between the other agent and the sender. Therefore, agent A2 and
agent A5 see the same losses that are introduced on the link between
the sender and agent A2. Agents A6, A7, and A8 do not join the RTP
session. From the other agents, only agents A3 and A9 join. All
agents are pure receivers, except A1, which is the sender.

The second scenario also represents cases where losses are shared
between two agents, but this time the agents are located on different
branches of the multicast tree. The delays to the sender are roughly
of the same magnitude. Agents A5 and A6 share the same losses.
Agents A3 and A9 join the RTP session, but are pure receivers and do
not see any losses.

Finally, in the third scenario, the losses are shared between two
agents, A5 and A6. The same agents as in the second scenario are

active. However, the delays of the links are different. The delay
of the link between agents A2 and A5 is reduced to 20 ms and between
A2 and A6 to 40 ms.

All agents beside agent A1 are pure RTP receivers. Thus, these
agents do not have an RTT estimation to the source. T_dither_max is
calculated with the above given formula, depending only on T_rr and
l, which means that all agents should calculate roughly the same
T_dither_max.

6.1. Feedback Suppression Performance

The feedback suppression rate for an agent is defined as the ratio of
the total number of feedback packets not sent out of the total number
of feedback packets the agent intended to send (i.e., the sum of sent
and not sent). The reasons for not sending a packet include: the
receiver already saw the same loss reported in a receiver report
coming from another session member or the max_feedback_delay
(application-specific) was surpassed.

The results for the feedback suppression rate of the agent Af that is
further away from the sender are depicted in Table 5. In general, it
can be seen that the feedback suppression rate increases as l
increases. However there is a threshold, depending on the
environment, from which the additional gain is not significant
anymore.

| | Feedback Suppression Rate | | |
l	Scen. 1	Scen. 2	Scen. 3
0.10	0.671	0.051	0.089
0.25	0.582	0.060	0.210
0.50	0.524	0.114	0.361
0.75	0.523	0.180	0.370
1.00	0.523	0.204	0.369
1.25	0.506	0.187	0.372
1.50	0.536	0.213	0.414
1.75	0.526	0.215	0.424
2.00	0.535	0.216	0.400
3.00	0.522	0.220	0.405
4.00	0.522	0.220	0.405

Table 5: Fraction of feedback that was suppressed at agent (Af) of
the total number of feedback messages the agent wanted to send

Similar results can be seen in Table 6 for the agent An that is
nearer to the sender.

l	Feedback Suppression Rate		
	Scen. 1	Scen. 2	Scen. 3
0.10	0.056	0.056	0.090
0.25	0.063	0.055	0.166
0.50	0.116	0.099	0.255
0.75	0.141	0.141	0.312
1.00	0.179	0.175	0.352
1.25	0.206	0.176	0.361
1.50	0.193	0.193	0.337
1.75	0.197	0.204	0.341
2.00	0.207	0.207	0.368
3.00	0.196	0.203	0.359
4.00	0.196	0.203	0.359

Table 6: Fraction of feedback that was suppressed at agent (An) of
the total number of feedback messages the agent wanted to send

The rate of feedback suppression failure is depicted in Table 7. The
trend of additional performance increase is not significant beyond a
certain threshold. Dependence on the scenario is noticeable here as
well.

l	Feedback Suppr. Failure Rate		
	Scen. 1	Scen. 2	Scen. 3
0.10	0.273	0.893	0.822
0.25	0.355	0.885	0.624
0.50	0.364	0.787	0.385
0.75	0.334	0.679	0.318
1.00	0.298	0.621	0.279
1.25	0.289	0.637	0.267
1.50	0.274	0.595	0.249
1.75	0.274	0.580	0.235
2.00	0.258	0.577	0.233
3.00	0.282	0.577	0.236
4.00	0.282	0.577	0.236

Table 7: The ratio of feedback suppression failures.

Summarizing the feedback suppression results, it can be said that in
general the feedback suppression performance increases as l
increases. However, beyond a certain threshold, depending on
environment parameters such as propagation delays or session
bandwidth, the additional increase is not significant anymore. This
threshold is not uniform across all scenarios; a value of l=0.5 seems
to produce reasonable results with acceptable (though not optimal)
overhead.

6.2. Loss Report Delay

   In this section, we show the results for the measured report delay
   during the simulations of the three sample scenarios. This
   measurement is a metric of the performance of the algorithms, because
   the value of the feedback for the sender typically decreases with the
   delay of its reception. The loss report delay is measured as the
   time at the sender between sending a packet and receiving the first
   corresponding loss report.

```
               |       |       Mean Loss Report Delay       |
               |   l   | Scen. 1 | Scen. 2 | Scen. 3 |
               +-------+---------+---------+---------+
               | 0.10  |  0.124  |  0.282  |  0.210  |
               | 0.25  |  0.168  |  0.266  |  0.234  |
               | 0.50  |  0.243  |  0.264  |  0.284  |
               | 0.75  |  0.285  |  0.286  |  0.325  |
               | 1.00  |  0.329  |  0.305  |  0.350  |
               | 1.25  |  0.351  |  0.329  |  0.370  |
               | 1.50  |  0.361  |  0.363  |  0.388  |
               | 1.75  |  0.360  |  0.387  |  0.392  |
               | 2.00  |  0.367  |  0.412  |  0.400  |
               | 3.00  |  0.368  |  0.507  |  0.398  |
               | 4.00  |  0.368  |  0.568  |  0.398  |
```

        Table 8: The mean loss report delay, measured at the sender.

   As can be seen from Table 8, the delay increases, in general, as l
   increases. Also, a similar effect as for the feedback suppression
   performance is present: beyond a certain threshold, the additional
   increase in delay is not significant anymore. The threshold is
   environment dependent and seems to be related to the threshold, where
   the feedback suppression gain would not increase anymore.

6.3. Summary of "l" Investigations

   We have shown experimentally that the performance of the feedback
   suppression mechanisms increases as l increases. The same applies
   for the report delay, which also increases as l increases. This
   leads to a threshold where both the performance and the delay do not
   increase any further. The threshold is dependent upon the
   environment.

   So finding an optimum value of l is not possible because it is always
   a trade-off between delay and feedback suppression performance. With
   l=0.5, we think that a trade-off was found that is acceptable for
   typical applications and environments.

7. Applications Using AVPF

   NEWPRED is one of the error resilience tools, which is defined in
   both ISO/IEC MPEG-4 visual part and ITU-T H.263. NEWPRED achieves
   fast error recovery using feedback messages. We simulated the
   behavior of NEWPRED in the network simulator environment as described
   above and measured the waiting time statistics, in order to verify
   that the extended RTP profile for RTCP-based feedback (AVPF) [1] is
   appropriate for the NEWPRED feedback messages. Simulation results,
   which are presented in the following sections, show that the waiting
   time is small enough to get the expected performance of NEWPRED.

7.1. NEWPRED Implementation in NS2

   The agent that performs the NEWPRED functionality, called NEWPRED
   agent, is different from the RTP agent we described above. Some of
   the added features and functionalities are described in the following
   points:

   Application Feedback
      The "Application Layer Feedback Messages" format is used to
      transmit the NEWPRED feedback messages. Thereby the NEWPRED
      functionality is added to the RTP agent. The NEWPRED agent
      creates one NACK message for each lost segment of a video frame,
      and then assembles multiple NACK messages corresponding to the
      segments in the same video frame into one Application Layer
      Feedback Message. Although there are two modes, namely, NACK mode
      and ACK mode, in NEWPRED [6][7], only NACK mode is used in these
      simulations. In this simulation, the RTP layer doesn't generate
      feedback messages. Instead, the decoder (NEWPRED) generates a
      NACK message when the segment cannot be decoded because the data
      hasn't arrived or loss of reference picture has occurred. Those
      conditions are detected in the decoder with frame number, segment
      number, and existence of reference pictures in the decoder.

   The parameters of NEWPRED agent are as follows:

        f: Frame Rate(frames/sec)
      seg: Number of segments in one video frame
       bw: RTP session bandwidth(kbps)

   Generation of NEWPRED's NACK Messages
      The NEWPRED agent generates NACK messages when segments are lost.

   a. The NEWPRED agent generates multiple NACK messages per one
      video frame when multiple segments are lost. These are
      assembled into one Feedback Control Information (FCI) message
      per video frame. If there is no lost segment, no message is
      generated and sent.

   b. The length of one NACK message is 4 bytes. Let num be the
      number of NACK messages in one video frame (1 <= num <= seg).
      Thus, 12+4*num bytes is the size of the low-delay RTCP feedback
      message in a compound RTCP packet.

Measurements
   We defined two values to be measured:

   - Recovery time
     The recovery time is measured as the time between the detection
     of a lost segment and reception of a recovered segment. We
     measured this "recovery time" for each lost segment.

   - Waiting time
     The waiting time is the additional delay due to the feedback
     limitation of RTP.

Figure 2 depicts the behavior of a NEWPRED agent when a loss occurs.

The recovery time is approximated as follows:

   (Recovery time) = (Waiting time) +
                     (Transmission time for feedback message) +
                     (Transmission time for media data)

Therefore, the waiting time is derived as follows:

   (Waiting time) = (Recovery time) - (Round-trip delay), where

   (Round-trip delay) = (Transmission time for feedback message) +
                         (Transmission time for media data)

```
         Picture Reference                          |: Picture Segment
                                                    %: Lost Segment
                       _____
                      /                   \
                  v/ \  / \  / \  / \       \
                  v   \v   \v   \v   \       \
 Sender   ---|----|----|----|----|---^|------------->
                  \    \    \        / ^ \
                   \    \    \      /   / \
                    \    \    v    /   /   \
                     \    \   x   /   /     \
                      \    \ Lost/   /       \
                       \    \   x   /         \
 _____                  v    x  / NACK         v
 Receiver ---------------|----%===-%----%----%----|----->
                        |-a-|
                        |------- b  -------|

                         a: Waiting time
                         b: Recover time (%: Video segments are lost)
```
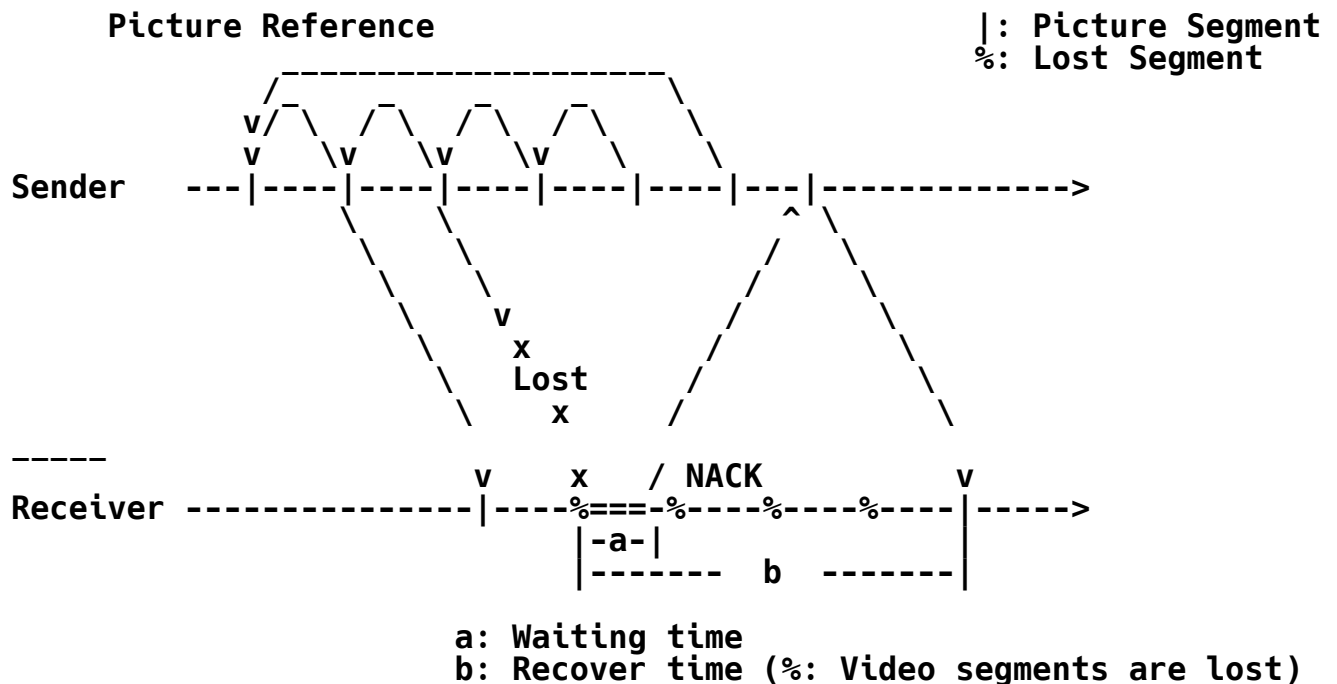
   Figure 2: Relation between the measured values at the NEWPRED agent

7.2. Simulation

   We conducted two simulations (Simulation A and Simulation B). In
   Simulation A, the packets are dropped with a fixed packet loss rate
   on a link between two NEWPRED agents. In Simulation B, packet loss
   occurs due to congestion from other traffic sources, i.e., ftp
   sessions.

7.2.1. Simulation A - Constant Packet Loss Rate

   The network topology used for this simulation is shown in Figure 3.

```
              Link 1         Link 2         Link 3
    +--------+      +------+       +------+      +--------+
    | Sender |------|Router|-------|Router|------|Receiver|
    +--------+      +------+       +------+      +--------+
        10(msec)        x(msec)        10(msec)
```
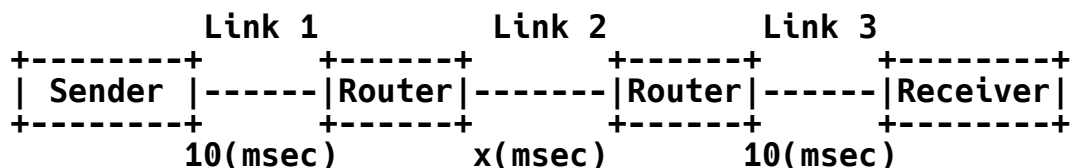
        Figure 3: Network topology that is used for Simulation A

   Link1 and link3 are error free, and each link delay is 10 msec.
   Packets may get dropped on link2. The packet loss rates (Plr) and
   link delay (D) are as follows:

         D [ms] = {10, 50, 100, 200, 500}
         Plr   = {0.005, 0.01, 0.02, 0.03, 0.05, 0.1, 0.2}

   Session bandwidth, frame rate, and the number of segments are shown
   in Table 9.

```
         +------------+---------+-------------+-----+
         |Parameter ID| bw(kbps) |f (frame/sec)| seg |
         +------------+---------+-------------+-----+
         | 32k-4-3    |    32   |      4      |  3  |
         | 32k-5-3    |    32   |      5      |  3  |
         | 64k-5-3    |    64   |      5      |  3  |
         | 64k-10-3   |    64   |     10      |  3  |
         | 128k-10-6  |   128   |     10      |  6  |
         | 128k-15-6  |   128   |     15      |  6  |
         | 384k-15-6  |   384   |     15      |  6  |
         | 384k-30-6  |   384   |     30      |  6  |
         | 512k-30-6  |   512   |     30      |  6  |
         | 1000k-30-9 |  1000   |     30      |  9  |
         | 2000k-30-9 |  2000   |     30      |  9  |
         +------------+---------+-------------+-----+
```

            Table 9: Parameter sets of the NEWPRED agents

   Figure 4 shows the key values of the result (packet loss rate vs.
   mean of waiting time).

   When the packet loss rate is 5% and the session bandwidth is 32 kbps,
   the waiting time is around 400 msec, which is just allowable for
   reasonable NEWPRED performance.

   When the packet loss rate is less than 1%, the waiting time is less
   than 200 msec. In such a case, the NEWPRED allows as much as
   200-msec additional link delay.

   When the packet loss rate is less than 5% and the session bandwidth
   is 64 kbps, the waiting time is also less than 200 msec.

   In 128-kbps cases, the result shows that when the packet loss rate is
   20%, the waiting time is around 200 msec. In cases with more than
   512-kbps session bandwidth, there is no significant delay. This
   means that the waiting time due to the feedback limitation of RTCP is
   negligible for the NEWPRED performance.

Bandwidth	Packet Loss Rate =						
	0.005	0.01	0.02	0.03	0.05	0.10	0.20
32k	130- 180	200- 250	230- 320	280- 390	350- 430	470- 610	560- 780
64k	80- 130	100- 150	120- 180	150- 190	180- 210	210- 300	290- 400
128k	60- 70	70- 80	90- 100	110- 120	130- 140	170- 190	190- 240
384k	30- 50	30- 50	30- 50	40- 50	50- 60	50- 70	50- 90
512k	< 50	< 50	< 50	< 50	< 50	< 50	< 60
1000k	< 50	< 50	< 50	< 50	< 50	< 50	< 55
2000k	< 30	< 30	< 30	< 30	< 30	< 35	< 35

Figure 4: The result of simulation A

7.2.2. Simulation B - Packet Loss Due to Congestion

The configurations of link1, link2, and link3 are the same as in
Simulation A except that link2 is also error-free, regarding bit
errors. However, in addition, some FTP agents are deployed to
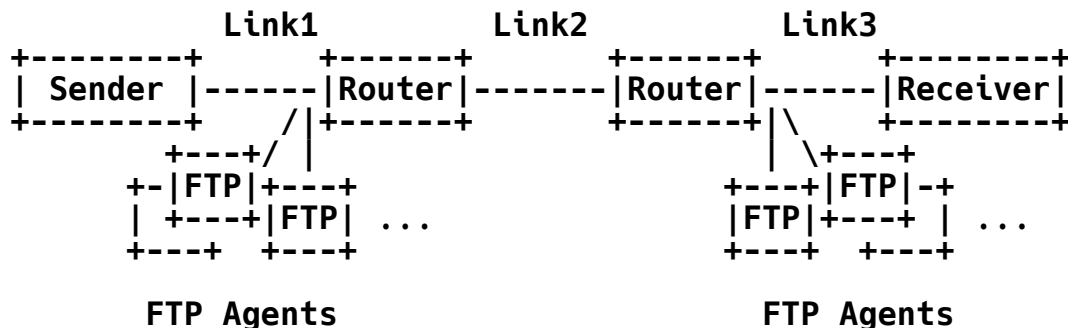overload link2. See Figure 5 for the simulation topology.

```
                Link1            Link2           Link3
    +--------+       +------+        +------+       +--------+
    | Sender |------|Router|-------|Router|------|Receiver|
    +--------+    / |+------+        +------+|\     +--------+
       +---+/   |                    | \+---+
    +-|FTP|+---+                  +---+|FTP|-+
    | +---+|FTP| ...              |FTP|+---+ | ...
    +---+  +---+                  +---+  +---+

        FTP Agents                       FTP Agents
```

Figure 5: Network Topology of Simulation B

The parameters are defined as for Simulation A with the following
values assigned:

D[ms] ={10, 50, 100, 200, 500} 32 FTP agents are deployed at each
edge, for a total of 64 FTP agents active.

The sets of session bandwidth, frame rate, and the number of segments
are the same as in Simulation A (Table 9).

We provide the results for the cases with 64 FTP agents, because
these are the cases where packet losses could be detected to be
stable. The results are similar to those for Simulation A except for
a constant additional offset of 50..100 ms. This is due to the delay
incurred by the routers' buffers.

7.3. Summary of Application Simulations

We have shown that the limitations of RTP AVPF profile do not
generate such high delay in the feedback messages that the
performance of NEWPRED is degraded for sessions from 32 kbps to 2
Mbps. We could see that the waiting time increases with a decreasing
session bandwidth and/or an increasing packet loss rate. The cause
of the packet loss is not significant; congestion and constant packet
loss rates behave similarly. Still we see that for reasonable
conditions and parameters the AVPF is well suited to support the
feedback needed for NEWPRED. For more information about NEWPRED, see
[8] and [9].

8. Summary

The new RTP profile AVPF was investigated regarding performance and
potential risks to the network stability. Simulations were conducted
using the network simulator ns2, simulating unicast and several
differently sized multicast topologies. The results were shown in
this document.

Regarding the network stability, it was important to show that the
new profile does not lead to any feedback implosion or use more
bandwidth than it is allowed. We measured the bandwidth that was
used for RTCP in relation to the RTP session bandwidth. We have
shown that, more or less exactly, 5% of the session bandwidth is used
for RTCP, in all considered scenarios. Other RTCP bandwidth values
could be set using the RTCP bandwidth modifiers [10]. The scenarios
included unicast with and without errors, differently sized multicast
groups, with and without errors or congestion on the links. Thus, we
can say that the new profile behaves in a network-friendly manner in
the sense that it uses only the allowed RTCP bandwidth, as defined by
RTP.

Secondly, we have shown that receivers using the new profile
experience a performance gain. This was measured by capturing the
delay that the sender sees for the received feedback. Using the new
profile, this delay can be decreased by orders of magnitude.

   In the third place, we investigated the effect of the parameter "l"
   on the new algorithms. We have shown that there does not exist an
   optimum value for it but only a trade-off can be achieved. The
   influence of this parameter is highly environment-specific and a
   trade-off between performance of the feedback suppression algorithm
   and the experienced delay has to be met. The recommended value of
   l=0.5 given in this document seems to be reasonable for most
   applications and environments.

9. Security Considerations

   This document describes the simulation work carried out to verify the
   correct working of the RTCP timing rules specified in the AVPF
   profile [1]. Consequently, security considerations concerning these
   timing rules are described in that document.

10. Normative References

   [1] Ott, J., Wenger, S., Sato, N., Burmeister, C., and J. Rey,
         "Extended RTP Profile for Real-time Transport Control Protocol
         (RTCP)-Based Feedback (RTP/AVPF)", RFC 4585, July 2006.

11. Informative References

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

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

   [4] Network Simulator Version 2 - ns-2, available from
         http://www.isi.edu/nsnam/ns.

   [5] C. Burmeister, T. Klinner, "Low Delay Feedback RTCP - Timing
         Rules Simulation Results". Technical Report of the Panasonic
         European Laboratories, September 2001, available from:
         http://www.informatik.uni-bremen.de/~jo/misc/
         SimulationResults-A.pdf.

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

   [7] ITU-T Recommendation, H.263. Video encoding for low bitrate
         communication. 1998.

   [8] S. Fukunaga, T. Nakai, and H. Inoue, "Error Resilient Video
         Coding by Dynamic Replacing of Reference Pictures", IEEE Global
         Telecommunications Conference (GLOBECOM), pp.1503-1508, 1996.

   [9] H. Kimata, Y. Tomita, H. Yamaguchi, S. Ichinose, T. Ichikawa,
         "Receiver-Oriented Real-Time Error Resilient Video Communication
         System: Adaptive Recovery from Error Propagation in Accordance
         with Memory Size at Receiver", Electronics and Communications in
         Japan, Part 1, vol. 84, no. 2, pp.8-17, 2001.

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

Authors' Addresses

   Carsten Burmeister
   Panasonic R&D Center Germany GmbH
   Monzastr. 4c
   D-63225 Langen, Germany

   EMail: carsten.burmeister@eu.panasonic.com

   Rolf Hakenberg
   Panasonic R&D Center Germany GmbH
   Monzastr. 4c
   D-63225 Langen, Germany

   EMail: rolf.hakenberg@eu.panasonic.com

   Akihiro Miyazaki
   Matsushita Electric Industrial Co., Ltd
   1006, Kadoma, Kadoma City, Osaka, Japan

   EMail: miyazaki.akihiro@jp.panasonic.com

   Joerg Ott
   Helsinki University of Technology, Networking Laboratory
   PO Box 3000, 02015 TKK, Finland

   EMail: jo@acm.org

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

   EMail: sato652@oki.com

   Shigeru Fukunaga
   Oki Electric Industry Co., Ltd.
   2-5-7 Hommachi, Chuo-ku, Osaka 541-0053 Japan

   EMail: fukunaga444@oki.com

Full Copyright Statement

Intellectual Property

   The IETF takes no position regarding the validity or scope of any
   Intellectual Property Rights or other rights that might be claimed to
   pertain to the implementation or use of the technology described in
   this document or the extent to which any license under such rights
   might or might not be available; nor does it represent that it has
   made any independent effort to identify any such rights. Information
   on the procedures with respect to rights in RFC documents can be
   found in BCP 78 and BCP 79.

   Copies of IPR disclosures made to the IETF Secretariat and any
   assurances of licenses to be made available, or the result of an
   attempt made to obtain a general license or permission for the use of
   such proprietary rights by implementers or users of this
   specification can be obtained from the IETF on-line IPR repository at
   http://www.ietf.org/ipr.

   The IETF invites any interested party to bring to its attention any
   copyrights, patents or patent applications, or other proprietary
   rights that may cover technology that may be required to implement
   this standard. Please address the information to the IETF at
   ietf-ipr@ietf.org.

Acknowledgement