David Leon
Viktor Varsa
Nokia

Internet Draft
Document: draft-leon-rtp-retransmission-02.txt
Expires: September 2002
March 2002

RTP retransmission framework

Status of this Memo

Abstract

RTP retransmission is an effective packet loss recovery scheme for real-time applications with relaxed delay bounds.

This document describes an RTP retransmission framework. It defines a payload format for retransmitted packets and recommends rules for sending these packets. Retransmitted RTP packets are sent in a separate stream from the original RTP stream. It is assumed that feedback from receivers to senders indicating the occurred packet losses is available by some means not defined here.

Main changes since draft-leon-rtp-retransmission-01.txt

The draft was updated according to the comments given at IETF#52:

*The draft describes the use of the redundancy payload format (RFC 2198) in order to send retransmission data and original data in the same stream.

*The draft gives some motivations in defining this retransmission framework document instead of using the FEC payload format (RFC 2733)

*Congestion section rewritten so as to state clearly the risks of RTP retransmission. Congestion control rules when using RTP retransmission have been clarified.

Editorial modifications were made throughout the text.

Table of Contents

Abstract...1
Main changes since draft-leon-rtp-retransmission-01.txt...........1
1. Introduction...2
2. Terminology..3
3. Retransmission framework basic principles.......................3
4. Retransmission payload format.................................5
5. Use with the Extended RTP profile for RTCP-based feedback.......5
6. Congestion control and scheme efficiency.......................7
7. Example scenario...8
8. Retransmission piggybacked on original packets.................9
9. SDP usage..9
10. FEC for retransmission....................................10
11. Security consideration....................................11
12. References..11
Author's Addresses...12

1. **Introduction**

Packet losses between an RTP sender and receiver may significantly degrade the quality of the received signal. Several techniques, such as forward error correction (FEC), retransmissions or adaptation of application layer (e.g. video) error resilience techniques based on back-channel messages may be considered to increase the robustness to packet loss. RFC-2354 [1] discusses the different options.

When choosing a technique for a particular system, the tolerable latency of the application has to be taken into account. In the case of multimedia conferencing, the end-to-end delay has to be at most a few hundred milliseconds in order to guarantee interactivity, which usually excludes the use of retransmission. On the other hand, in the case of multimedia streaming, the user can tolerate an initial latency as part of the session setup and thus an end-to-end delay of several seconds is possible. Retransmission may thus be considered for such applications.

This document proposes a retransmission framework. It defines a payload format for retransmitted RTP packets and retransmission rules. This RTP retransmission scheme requires frequent packet loss indication feedback to the RTP entity performing retransmission. The AV profile [2] does not provide this feature and could therefore not be used. However, the retransmission scheme could be run under the extended RTP profile for RTCP-based feedback[3] which defines a NACK message that can be sent as part of a compound RTCP packet.

2. Terminology

The following terms are used in this document:

Original packet: refers to an RTP packet sent by an RTP sender as part of an RTP stream.

Original stream: refers to the stream of original packets.

Retransmission packet: refers to an RTP packet whose payload includes the payload of an already sent original packet. Such a retransmission packet is said to be associated with the original RTP packet whose payload is included in the retransmission packet.

Retransmission request: a means by which an RTP receiver is able to request that the RTP sender send a retransmission packet associated with a given original packet. In [3], a retransmission request is sent as a packet loss indication in a NACK message.

Retransmission stream: the stream of retransmission packets associated to with an original stream.

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 [4].

3. Retransmission framework basic principles

Retransmission packets MUST NOT share the RTP Sequence Number (SN) space with the original stream. The retransmission stream SHOULD use a different RTP session (as defined in RTP [5]) from that of the original stream. When a separate session is used, the original and retransmission streams are sent to different multicast group/unicast addresses and/or port numbers. The retransmission stream has then its own SN space. The retransmission stream MAY also be sent as redundant data piggybacked to the original packets as shown in Section 9. In the latter case, the retransmission stream does not have any SN space.

There are several reasons why the SN space must not be shared:

Since retransmission packets do not share the SN space with the original packets, the receiver is able to distinguish between the loss of original packets and retransmission packets. Otherwise, reliable data loss detection would not be possible. Reliable data loss detection is mandated for example in RTP conversational text [6].

RTP Timestamp (TS) estimation of missing original packets is necessary at the receiver in order to decide whether a retransmission is useful or not. A retransmission is useful if the retransmission packet as a response to the retransmission request is estimated to arrive at the receiver before the time it is used for playback (playout time). The missing original packet TS can be estimated from TS of packets preceding and following the SN gap caused by the missing packet in the original stream.
The fact that RTP streams for original and retransmission packets do not share the same SN space guarantees that the RTP timestamp estimation method is reliable. Reliability would be sacrificed if original and retransmission packets were sent in the same RTP stream. The TS estimate for a lost retransmission packet ,when calculated as above, would then be incorrect. This is because the retransmission packet usually has a smaller "out-of-order" TS than the TS of the consecutive original packets.

There are some additional advantages when the RTP stream is sent to a separate session vs. piggybacked data:

There are no modifications to the payload format of the original packets. Retransmission can thus work with any existing payload format.

When the retransmission stream is sent to a multicast RTP session, receivers may choose to subscribe or not to subscribe to the RTP session carrying the retransmission stream. Thus, a multicast streaming application can use retransmission and still be backwards compatible as receivers which do not implement the retransmission payload format only join the RTP session carrying the original stream. A scenario where the original session is multicast and separate unicast sessions carry the retransmission stream to each participant, is also possible. In this scenario a receiver can receive retransmission packets only for original packets that it has itself requested, and not the retransmission packets that are sent as responses to retransmission requests from other receivers.

5. Mixers, translators and packet caches may be able to separate retransmission packets from original packets at an RTP session level and process them differently if necessary.

6. As a consequence of having separate RTP sessions for original and retransmission streams, there are also separate RTCP streams and statistics for these two sessions. There is thus no corruption of the original stream RTCP statistics. The RTP sender is able to know the packet loss and jitter of the original stream. It can thus

estimate what the quality of the received signal would be without the use of retransmission.

4. Retransmission payload format

The payload format of a retransmission packet is shown below.

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         RTP Header                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|E|   OPT       |            OSN                |               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+               |
|                  Original RTP Packet Payload                  |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

RTP header usage:
The same SSRC value SHOULD be used for the original stream and the retransmission stream. In the RTP header, SN has the standard definition, i.e. it MUST be one higher than the SN of the preceding retransmission packet. The payload type is dynamic and indicates the use of the retransmission payload format. All other fields of the RTP header have the same value as in the original RTP packet.

The retransmission packet payload carries an E bit, an OPT field (7 bits) and an OSN field (2 bytes) followed by the original RTP packet payload. The E bit is an extension bit for future-proofing. It MUST be set to zero. The OPT field is the payload type of the original packet payload that is being retransmitted. The OSN field is the SN of the original packet that originally carried the same payload.

5. Use with the Extended RTP profile for RTCP-based feedback

When the Extended RTP profile for RTCP-based feedback [4] is used, it is RECOMMENDED that receivers send retransmission requests according to the rules in this section. The rules described hereafter aim at limiting the maximum delay before a retransmission request can be sent and are compliant to the more general rules described in the profile itself.

5.1 Sending rules for RTCP-based feedback

The minimum RTCP report interval is computed according to the allowed RTCP bandwidth of a given session (obtained explicitly or calculated as a percentage of the RTP session bandwidth). The NACK message format defined in the profile is used. Early RTCP packets SHOULD NOT be used, and the NACK message SHOULD always be appended to a regularly scheduled compound RTCP packet that is sent at every RTCP report interval T_rr.

Any other rules to send feedback MAY be used as long as they are compliant with the profile in use. In particular, a receiver MAY send NACK messages in early RTCP packets. However, in that case the time when the next RTCP packet following this early RTCP packet can be sent could be too late to report a loss occurring right after the early RTCP packet was sent. Sending RTCP packets at regular intervals guarantees that the delay between detecting an original packet loss and being able to send a NACK message for that packet is no longer than the regular RTCP interval.

A receiver should compute an estimate of the delay D to receive a retransmission packet after a NACK message has been sent. This estimate may be obtained from past observations, RTCP report round-trip time if available or any other means.

The minimum receiver buffering delay B (i.e. the time between a packet is received and its payload is used at the receiver) is the RTCP reporting interval added to the delay estimate, which guarantees that a retransmission packet as a response to a sent retransmission request can be received before its payload is used.

i.e.
B>T_rr+D

It can be seen that the needed receiver buffering delay is dependent on the amount of RTCP traffic allowed in the session. It is illustrated in Section 7 that moderate RTCP feedback traffic is enough to perform retransmission with reasonable receiver buffering delay.

5.2 Receiver algorithm for generating retransmission requests

A receiver needs to maintain a list of missing original packet sequence numbers (SN) since its last sent RTCP packet. A receiver needs also to store for each missing original packet an estimated RTP timestamp (TS) as described in Section 3.

At the next scheduled RTCP packet time, the receiver estimates based on the estimated TS of the missing original packet if a retransmission packet as a response to a NACK message that would be sent in the current RTCP packet could arrive at the receiver before its playout time or not. If not, it removes the packet SN from its list of missing packets. The receiver then sends retransmission requests for the missing original packets by adding the SNs of the missing original packets to a NACK message in the scheduled RTCP compound packet (see usage of the PID and BLP fields of the NACK message format in [4]). If needed for signalling of multiple missing SN, multiple NACK messages can be added to the RTCP compound packet..

If the retransmission stream is sent to a multicast session, the receiver SHOULD listen to NACK messages from other receivers. If a

NACK message for the SN of a missing packet at the receiver has been sent by another receiver, the receiver SHOULD ignore that SN in its list of missing packets and refrain from sending a retransmission request for that SN.

The same retransmission request may be resent by the receiver if no retransmission packet with OSN equal to the missing original packet SN was received after an estimated retransmission reception time. This increases robustness to the loss of a NACK message or of a retransmission packet. The repeated retransmission request may be sent at the earliest at the next RTCP report scheduled time. Repeated retransmission requests MUST be sent in the original RTP session and not in the retransmission RTP session. The number of repeated retransmission requests that may be sent for a given missing original packet SN depends on the ratio of the receiver buffering delay and RTCP session bandwidth.

The receiver should upon reception of a retransmission packet remove the SN corresponding to the original packet (OSN in the retransmission payload format) from the list of missing SNs.

5.3 RTCP sending rules in the retransmission RTP session

When the original stream and the retransmission stream are carried in separate RTP sessions, the retransmission stream has its own RTCP stream as well. The amount of RTCP sent for the retransmission stream is computed as a fraction of the retransmission RTP session bandwidth. Since, the retransmission traffic is limited, the overhead caused by the additional RTCP packets in the retransmission RTP session is moderate. For example, assuming a 64 kbps original RTP session and on average 3% packet loss and all lost original packets retransmitted once, the bandwidth of the retransmission RTP session would be about 2 kbps. In this case the recommended RTCP traffic for the retransmission RTP session would be 0.1 kbps.

Early RTCP packets and RTCP feedback messages SHOULD NOT be used in the retransmission RTP session.

6. Congestion control

RTP retransmission poses a risk of increased network congestion. In a best-effort environment, packet loss is caused by congestion. Reacting to loss by retransmitting older packets in addition to the current data would thus further increase congestion. Implementations SHOULD follow the recommendations below in order to use retransmission.

The RTP profile under which the retransmission scheme is used defines an appropriate congestion control mechanism in different

environments. Following the rules under this profile, an RTP application can determine its acceptable bitrate and packet rate in order to be fair to other applications such as TCP flows and other RTP flows.

If an RTP application uses retransmission, the acceptable packet rate and bitrate SHOULD include both the original and retransmitted data. This will guarantee that an application using retransmission should be fair to any other RTP or TCP flows. Such a rule would translate in practice into the following actions:

If enhanced service is used it should made sure that the total bitrate and packet rate do not exceed that of the requested service. It should be further monitored that the requested services are actually delivered. In a best-effort environment, the sender SHOULD NOT send retransmission packets without reducing the packet rate and bitrate of the original stream (for example by encoding the data at a lower rate).

In addition, the sender MAY retransmit only the packets that it deems important and ignore NACK messages for other packets in order to limit the bitrate

These congestion control mechanisms should keep the congestion level under an acceptable limit. This would in turn guarantee that the packet loss should be moderate. Too high a packet loss would mean that congestion is not kept under control and retransmission should then not be used.

7. Example scenario of unicast streaming

Let us assume that the RTP session bandwidth is 64 kbps and the receiver buffering delay is 3 seconds. The network round trip time is 500 ms and Receiver RTCP packets are sent at 2-second intervals from the receiver including NACK messages for missing original packet SNs. With these time limitations, when the receiver algorithm for generating retransmission requests described in Section 5.2 is followed, only one NACK message but no repeated NACK messages can be sent for the same original packet loss. Let us assume an original packet rate of 50 packets per second (1 packet every 20 ms). At this packet rate in a packet loss distribution worst case scenario where every 17th original packet is lost (i.e. 5.88% uniform packet loss), a maximum of 6 NACK messages need to be appended to the RTCP compound packet that is sent every 2 seconds. The 6 NACK messages can report any packet losses in an SN range of 6*17=102, thus the number of required NACK messages would not increase with higher packet loss rates. The overhead required per RTCP compound packet for the 6 NACK messages is 36 bytes which carries 6 PID and 6 BLP fields in addition to the feedback message headers.

The total compound RTCP packets size is thus: IPv4 (20) + UDP (8)+ RR(header 8 + report 24)+ CNAME(12)+ NACK (36) = 108 bytes. The needed receiver RTCP bandwidth would then be 0.432 kbps. This RTCP

bandwidth is well below the maximum allowed RTCP bandwidth that would be allowed when using [4]. From the receiver to the sender the

RTCP bandwidth limit in an RTP session with 64 kbps is about 0.25*64 = 1.6 kbps.

8. Retransmission piggybacked onto original packets

The RTP payload format for redundant audio data [7] may be used to send retransmitted data within the same RTP stream. It may either be used instead of the retransmission payload format or complementarily to the retransmission payload format.

Original packet payloads for which a retransmission request was received from the receiver may be sent piggybacked to other original packets as if they were redundant data. Although this may be suitable for some applications, such a scheme is not generally applicable, as the RTP payload format for redundant audio data can not carry all the RTP header fields of the retransmitted data which are necessary to implement a complete retransmission scheme. In particular, the receiver would not know the original SN of the retransmitted data. This fact is very limiting as some applications like conversational text [6] could not use the retransmitted data without the SN.

On the other hand, the payload format for redundant audio and the retransmission payload format may be used complementarily by sending the retransmission payload format as a redundant encoding of the original data. Note that although the original SN and TS which are necessary for the retransmission scheme are preserved, the original marker bit and CSRC list are lost.

Upon a receiver retransmission request, the sender SHOULD retransmit only the original data of the packet if the requested packet contains both original and redundant (retransmitted) data. Multiple retransmission should be performed by requesting multiple times the original packet SN.

9. SDP usage

The binding to the payload type number is indicated by an rtpmap attribute. The name used in the binding is "RTX". Two example SDP descriptions are shown below.

Retransmission stream as a separate stream:

```
c=IN IP4 113.3.12.11
m=video 49170 RTP/AVPF 96
a=rtpmap:96 H263-1998/90000
a=fmtp:96 rtcp-fb nack
m=video 49172 RTP/AVPF 97
a=rtpmap:97 RTX/90000
```

Use with the redundancy payload format:

```
c=IN IP4 113.3.12.11
m=video 49170 RTP/AVPF 121 96 97
a=rtpmap:121 red/90000
a=rtpmap:96 H263-1998/90000
a=fmtp:96 rtcp-fb nack
a=rtpmap:97 RTX/90000
a=fmtp:121 96/97
```

In this case, although the retransmission payload format is specified as a possible encoding for this stream, retransmission packets MUST NOT be sent by themselves but piggybacked as redundant data.

10. FEC for retransmission

It is possible to retransmit the payload of an original packet by sending a FEC packet as defined in [8] instead of using the retransmission payload format. The base SN of the FEC header is the sequence number of the original packet that carried the same payload and the mask is set to zero. There are some advantages in using the FEC payload format in particular in multicast sessions as a single FEC packet may repair the loss of different packets at different receivers. In a multicast session, FEC may be combined with retransmission requests as described in [9] in order to achieve scalable reliable multicast transmission.

There are however some motivations to define the retransmission payload format in order to perform retransmission instead of using the FEC payload format:

*The retransmission payload format has reduced overhead (3 bytes instead of 12 bytes).

*In the retransmission payload format, the RTP header TS is the actual TS of the (retransmitted) media carried in the packets. This is in line with RTP [5] specifications. This is useful in particular for RTP mixers/translators which process the TS. On the other hand, the FEC payload format uses the RTP transmission time (as the FEC packet should be computed over data with different TS).

*Assuming that no retransmission payload format were defined, the FEC payload format would then be used in different ways: sending computed FEC packets proactively for correction of lost packets at the receiver without requiring NACK messages (referred hereafter as

proactive FEC) or sending retransmitted data upon receipt of a NACK message from the receiver.
Although they could use the same payload format, these two repair mechanisms are different. For a given amount of overhead, they offer a different residual packet loss vs. latency trade-off.

Retransmission provides higher packet loss recovery at the expense of higher delay. If a sender uses proactive FEC and none of the packets protected by a given FEC packet is lost, there is then no use of the FEC packet at the receiver. On the other hand, data which are retransmitted are known to have been lost.
Therefore, a receiver may wish to use only retransmission and not receive any proactive FEC from the sender in order to trade-off itself the buffering delay, the data-loss rate and the overhead.
There is thus a motivation to let the receiver decide at session setup whether the sender may send proactive FEC or retransmission only.
If the same payload format were used for these different purposes, the receiver would not know at session establishment which repair mechanism is used by the sender (without defining out-of-band signalling). The sender would decide by itself whether FEC or retransmission (or both) is used and the receiver would not know until receiving the FEC packets whether proactive FEC or retransmission is used.
On the other hand, having different payload formats or at least different out of band signalling for these two repair mechanisms (e.g. through defining the binding name 'RTX' SDP rtpmap attribute) would allow the receiver to choose which mechanism to use (among those supported by the sender). The receiver would then have an explicit way to tell the sender not to send proactive FEC.
Furthermore, if the receiver did not know at session establishment whether it will receive FEC packets or retransmission, it would have to be prepared to receive FEC and thus be able to perform FEC packet recovery operations. Implementers could thus not choose to implement only a FEC or retransmission scheme.

11. Security consideration

The security considerations of the profile under which the retransmission scheme is used should be applied.

12. References

1 Perkins, C., Hodson, O., "Options for Repair of Streaming Media", RFC 2354, June 1998.

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

3 J Ott, S Wenger, S Fukunaga, N Sato, K Yano, M Akihiro, H Koichi, R Hakenberg, C. Burmeister "Extended RTP profile for RTCP-based feedback", November 2001

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

5 Schulzrinne, H., Casner, S., Frederick, R. and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", Internet Draft draft-ietf-avt-rtp-new-11.txt, November 2001.

6 Hellstrom, J, "RTP for conversational text", RFC 2793, May 2000

7 Perkins, C., Kouvelas, I., Hodson, O., Hardman, V., Handley, M., Bolot, J.C, Vega-Garcia, A, Fosse-Parisis, S., "RTP payload for redundant audio data", RFC 2198, September 1997.

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

9 Nonnenmacher, Biersack, E, Towsley, D,. "Parity-based loss recovery for reliable multicast transmission", ACM SIGCOMM'97, Cannes, France, September 1997

Author's Addresses

David Leon
Nokia Research Center
6000 Connection Drive Phone: 1-972-374-1860
Irving, TX. USA Email: david.leon@nokia.com

Viktor Varsa
Nokia Research Center
6000 Connection Drive Phone: 1-972-374-1861
Irving, TX. USA Email: viktor.varsa@nokia.com