Internet Draft
draft-ietf-avt-rtp-retransmission-08.txt

J. Rey/Matsushita
D. Leon/Nokia
A. Miyazaki/Matsushita
V. Varsa/Nokia
R. Hakenberg/Matsushita

Expires: June 2003 December 2003

RTP Retransmission Payload Format

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.

Copyright Notice

[Note to RFC Editor: This paragraph is to be deleted when this draft is published as an RFC. References in this draft to RFC XXXX should be replaced with the RFC number assigned to this document.]

Abstract

RTP retransmission is an effective packet loss recovery technique for real-time applications with relaxed delay bounds. This document describes an RTP payload format for performing retransmissions. Retransmitted RTP packets are sent in a separate

stream from the original RTP stream. It is assumed that feedback

from receivers to senders is available. In particular, it is assumed that RTCP feedback as defined in the extended RTP profile for RTCP-based feedback (denoted RTP/AVPF), is available in this memo.

Table of Contents

1. Introduction

Packet losses between an RTP sender and receiver may significantly degrade the quality of the received media. Several techniques, such as forward error correction (FEC), retransmissions or interleaving may be considered to increase packet loss resiliency. RFC 2354 [8] discusses the different options.

When choosing a repair technique for a particular application, 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.

However, in the case of multimedia streaming, the user can tolerate an initial latency as part of the session set-up and thus an end-to-end delay of several seconds may be acceptable. Retransmission may thus be considered for such applications.

This document specifies a retransmission method for RTP applicable to unicast and (small) multicast groups: it defines a payload format for retransmitted RTP packets and provides protocol rules for the sender and the receiver involved in retransmissions.

Furthermore, this retransmission payload format was designed for use with the extended RTP profile for RTCP-based feedback, AVPF [1]. It may also be used with other RTP profiles defined in the future.

The AVPF profile allows for more frequent feedback and for early feedback. It defines a small number of general-purpose feedback messages, e.g. ACKs and NACKs, as well as codec and application-specific feedback messages. See [1] for details.

2. Terminology

The following terms are used in this document:

Original packet: refers to an RTP packet which carries user data sent for the first time by an RTP sender.

Original stream: refers to the RTP stream of original packets.

Retransmission packet: refers to an RTP packet which is to be used by the receiver instead of a lost original packet. Such a retransmission packet is said to be associated with the original RTP packet.

Retransmission request: a means by which an RTP receiver is able to request that the RTP sender should send a retransmission packet for a given original packet. Usually, an RTCP NACK packet as

specified in [1] is used as retransmission request for lost packets.

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

Session-multiplexing: scheme by which the original stream and the associated retransmission stream are sent into two different RTP sessions.

SSRC-multiplexing: scheme by which the original stream and the retransmission stream are sent in the same RTP session with different SSRC values.

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 [2].

3. Requirements and design rationale for a retransmission scheme

The use of retransmissions in RTP as a repair method for streaming media is appropriate in those scenarios with relaxed delay bounds and where full reliability is not a requirement. More specifically, RTP retransmission allows to trade-off reliability vs. delay, i.e. the endpoints may give up retransmitting a lost packet after a given buffering time has elapsed. Unlike TCP there is thus no head-of-line blocking caused by RTP retransmissions. The implementer should be aware that in cases where full reliability is required or higher delay and jitter can be tolerated, TCP or other transport options should be considered.

The RTP retransmission scheme defined in this document is designed to fulfil the following set of requirements:

1. It must not break general RTP and RTCP mechanisms.
2. It must be suitable for unicast and small multicast groups.
3. It must work with mixers and translators.
4. It must work with all known payload types.
5. It must not prevent the use of multiple payload types in a session.
6. In order to support the largest variety of payload formats, the RTP receiver must be able to derive how many and which RTP packets were lost as a result of a gap in received RTP sequence numbers. This requirement is referred to as sequence number preservation. Without such a requirement, it would be impossible to use retransmission with payload formats, such as

conversational text [9] or most audio/video streaming applications, that use the RTP sequence number to detect lost packets.

When designing a solution for RTP retransmission, several approaches may be considered for the multiplexing of the original RTP packets and the retransmitted RTP packets.

One approach may be to retransmit the RTP packet with its original sequence number and send original and retransmission packets in the same RTP stream. The retransmission packet would then be identical to the original RTP packet, i.e. the same header (and thus same sequence number) and the same payload. However, such an approach is not acceptable because it would corrupt the RTCP statistics. As a consequence, requirement 1 would not be met. Correct RTCP statistics require that for every RTP packet within the RTP stream, the sequence number be increased by one.

Another approach may be to multiplex original RTP packets and retransmission packets in the same RTP stream using different payload type values. With such an approach, the original packets and the retransmission packets would share the same sequence number space. As a result, the RTP receiver would not be able to infer how many and which original packets (which sequence numbers) were lost.

In other words, this approach does not satisfy the sequence number preservation requirement (requirement 6). This in turn implies that requirement 4 would not be met. Interoperability with mixers and translators would also be more difficult if they did not understand this new retransmission payload type in a sender RTP stream. For these reasons, a solution based on payload type multiplexing of original packets and retransmission packets in the same RTP stream is excluded.

Finally, the original and retransmission packets may be sent in two separate streams. These two streams may be multiplexed either by sending them in two different sessions , i.e. session-multiplexing, or in the same session using different SSRC values, i.e. SSRC-multiplexing. Since original and retransmission packets carry media of the same type, the objections in Section 5.2 of RTP [3] to RTP multiplexing do not apply in this case.

Mixers and translators may process the original stream and simply discard the retransmission stream if they are unable to utilise it. Using two separate streams thus satisfies all the requirements listed in this section.

3.1 Multiplexing scheme choice

Session-multiplexing and SSRC-multiplexing have different pros and cons:

Session-multiplexing is based on sending the retransmission stream in a different RTP session (as defined in RTP [3]) from that of the original stream, i.e. the original and retransmission streams are sent to different network addresses and/or port numbers.

Having a separate session allows more flexibility. In multicast, using two separate sessions for the original and the retransmission streams allows a receiver to choose whether or not to subscribe to the RTP session carrying the retransmission stream. The original session may also be single-source multicast while separate unicast sessions are used to convey retransmissions to each of the receivers, which as a result will receive only the retransmission packets they request.

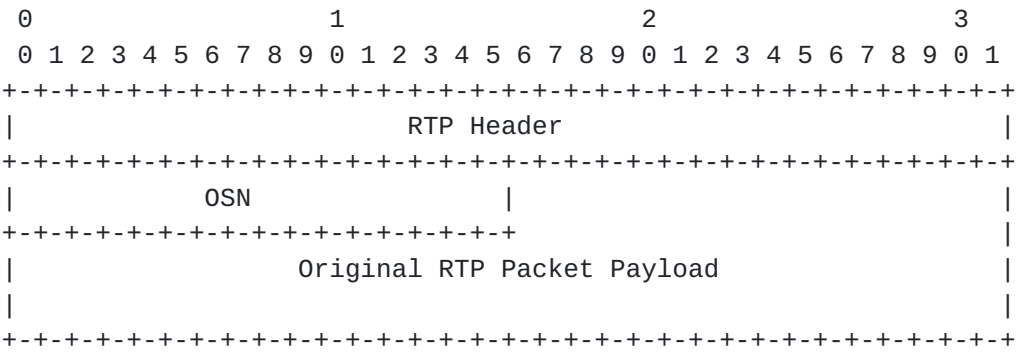
The use of separate sessions also facilitates differential treatment by the network and may simplify processing in mixers, translators and packet caches.

With SSRC-multiplexing, a single session is needed for the original and the retransmission stream. This allows streaming servers and middleware which are involved in a high number of concurrent sessions to minimise their port usage.

This retransmission payload format allows both session-multiplexing and SSRC-multiplexing for unicast sessions. From an implementation point of view, there is little difference between the two approaches. Hence, in order to maximise interoperability, both multiplexing approaches SHOULD be supported by senders and receivers. For multicast sessions, session-multiplexing MUST be used because the association of the original stream and the retransmission stream is problematic if SSRC-multiplexing is used with multicast sessions(see Section 5.3 for motivation).

4. Retransmission payload format

The format of a retransmission packet is shown below:



The RTP header usage is as follows:

In the case of session-multiplexing, the same SSRC value MUST be used for the original stream and the retransmission stream. In

the case of an SSRC collision in either the original session or the retransmission session, the RTP specification requires that an RTCP BYE packet MUST be sent in the session where the collision happened. In addition, an RTCP BYE packet MUST also be sent for

the associated stream in its own session. After a new SSRC identifier is obtained, the SSRC of both streams MUST be set to this value.

In the case of SSRC-multiplexing, two different SSRC values MUST be used for the original stream and the retransmission stream as required by RTP. If an SSRC collision is detected for either the original stream or the retransmission stream, the RTP specification requires that an RTCP BYE packet MUST be sent for this stream. An RTCP BYE packet MUST NOT be sent for the associated stream. Therefore, only the stream that experienced SSRC collision will choose a new SSRC value. Refer to Section 5.3 for the implications on the original and retransmission stream SSRC association at the receiver.

For either multiplexing scheme, the sequence number has the standard definition, i.e. it MUST be one higher than the sequence number of the preceding packet sent in the retransmission stream.

The retransmission packet timestamp is set to the original timestamp, i.e. to the timestamp of the original packet. As a consequence, the initial RTP timestamp for the first packet of the retransmission stream is not random but equal to the original timestamp of the first packet that is retransmitted. See the security considerations section in this document for security implications.

Implementers have to be aware that the RTCP jitter value for the retransmission stream does not reflect the actual network jitter since there could be little correlation between the time a packet is retransmitted and its original timestamp.

The payload type is dynamic. Each payload type of the original stream MUST map to a different payload type value in the retransmission stream. Therefore, when multiple payload types are used in the original stream, multiple dynamic payload types will be mapped to the retransmission payload format. See Section 8.1 for the specification of how the mapping between original and retransmission payload types is done with SDP.

As the retransmission packet timestamp carries the original media timestamp, the timestamp clockrate used by the retransmission payload type is the same as the one used by the associated original payload type. Therefore, if an RTP stream carries payload types of different clockrates, this will also be the case for the associated retransmission stream. Note that an RTP stream does not usually carry payload types of different clockrates.

The payload of the RTP retransmission packet comprises the retransmission payload header followed by the payload of the original RTP packet. The length of the retransmission payload header is 2 octets. This payload header contains only one field, OSN (original sequence number), which MUST be set to the sequence

number of the associated original RTP packet. The original RTP packet payload, including any possible payload headers specific to the original payload type, is placed right after the retransmission payload header.

For payload formats that support encoding at multiple rates, instead of retransmitting the same payload as the original RTP packet the sender MAY retransmit the same data encoded at a lower rate. This aims at limiting the bandwidth usage of the retransmission stream. When doing so, the sender MUST ensure that the receiver will still be able to decode the payload of the already sent original packets that might have been encoded based on the payload of the lost original packet. In addition, if the sender chooses to retransmit at a lower rate, the values in the payload header of the original RTP packet may not longer apply to the retransmission packet and may need to be modified in the retransmission packet to reflect the change in rate. The sender should trade-off the decrease in bandwidth usage with the decrease in quality caused by resending at a lower rate.

If the original RTP header carried any profile-specific extensions, the retransmission packet SHOULD include the same extensions immediately following the fixed RTP header as expected by applications running under this profile. In this case, the retransmission payload header is thus placed after the profile-specific extensions.

If the original RTP header carried an RTP header extension, the retransmission packet SHOULD carry the same header extension. This header extension MUST be placed right after the fixed RTP header, as specified in RTP [3]. In this case, the retransmission payload header is thus placed after the header extension.

If the original RTP packet contained RTP padding, that padding MUST be removed before constructing the retransmission packet. If padding of the retransmission packet is needed, padding is performed as with any RTP packets and the padding bit is set.

The marker bit (M), the CSRC count (CC) and the CSRC list of the original RTP header MUST be copied "as is" into the RTP header of the retransmission packet.

5. Association of a retransmission stream to its original stream

5.1 Retransmission session sharing

In the case of session-multiplexing, a retransmission session MUST map to exactly one original session, i.e. the same retransmission

session cannot be used for different original sessions.

If retransmission session sharing were allowed, it would be a problem for receivers, since they would receive retransmissions

for original sessions they might not have joined. For example, a receiver wishing to receive only audio would receive also retransmitted video packets if an audio and video session shared the same retransmission session.

5.2 CNAME use

In both the session-multiplexing and the SSRC-multiplexing cases, a sender MUST use the same CNAME for an original stream and its associated retransmission stream.

5.3 Association at the receiver

A receiver receiving multiple original and retransmission streams needs to associate each retransmission stream with its original stream. The association is done differently depending on whether session-multiplexing or SSRC-multiplexing is used.

If session-multiplexing is used, the receiver associates the two streams having the same SSRC in the two sessions. Note that the payload type field cannot be used to perform the association as several media streams may have the same payload type value. The two sessions are themselves associated out-of-band. See Section 8 for how the grouping of the two sessions is done with SDP.

If SSRC-multiplexing is used, the receiver should first of all look for two streams that have the same CNAME in the session. In some cases, the CNAME may not be enough to determine the association as multiple original streams in the same session may share the same CNAME. For example, there can be in the same video session multiple video streams mapping to different SSRCs and still using the same CNAME and possibly the same PT values. Each (or some) of these streams may have an associated retransmission stream.

In this case, in order to find out the association between original and retransmission streams having the same CNAME, the receiver SHOULD behave as follows.

The association can generally be resolved when the receiver receives a retransmission packet matching a retransmission request which had been sent earlier. Upon reception of a retransmission packet whose original sequence number has been previously requested, the receiver can derive that the SSRC of the retransmission packet is associated to the sender SSRC from which the packet was requested.

However, this mechanism might fail if there are two outstanding requests for the same packet sequence number in two different

original streams of the session. Note that since the initial packet sequence numbers are random, the probability of having two outstanding requests for the same packet sequence number would be very small. Nevertheless, in order to avoid ambiguity in the

unicast case, the receiver MUST NOT have two outstanding requests for the same packet sequence number in two different original streams before the association is resolved. In multicast, this ambiguity cannot be completely avoided, because another receiver may have requested the same sequence number from another stream. Therefore, SSRC-multiplexing MUST NOT be used in multicast sessions.

If the receiver discovers that two senders are using the same SSRC or if it receives an RTCP BYE packet, it MUST stop requesting retransmissions for that SSRC. Upon reception of original RTP packets with a new SSRC, the receiver MUST perform the SSRC association again as described in this section.

6. Use with the extended RTP profile for RTCP-based feedback

This section gives general hints for the usage of this payload format with the extended RTP profile for RTCP-based feedback, denoted AVPF [1]. Note that the general RTCP send and receive rules and the RTCP packet format as specified in RTP apply, except for the changes that the AVPF profile introduces. In short, the AVPF profile relaxes the RTCP timing rules and specifies additional general-purpose RTCP feedback messages. See [1] for details.

6.1 RTCP at the sender

In the case of session-multiplexing, Sender Report (SR) packets for the original stream are sent in the original session and SR packets for the retransmission stream are sent in the retransmission session according to the rules of RTP.

In the case of SSRC-multiplexing, SR packets for both original and retransmission streams are sent in the same session according to the rules of RTP. The original and retransmission streams are seen, as far the RTCP bandwidth calculation is concerned, as independent senders belonging to the same RTP session and are thus equally sharing the RTCP bandwidth assigned to senders.

Note that in both cases, session- and SSRC-multiplexing, BYE packets MUST still be sent for both streams as specified in RTP. In other words, it is not enough to send BYE packets for the original stream only.

6.2 RTCP Receiver Reports

In the case of session-multiplexing, the receiver will send report blocks for the original stream and the retransmission stream in

separate Receiver Report (RR) packets belonging to separate RTP sessions. RR packets reporting on the original stream are sent in the original RTP session while RR packets reporting on the retransmission stream are sent in the retransmission session. The

RTCP bandwidth for these two sessions may be chosen independently (for example through RTCP bandwidth modifiers [4]).

In the case of SSRC-multiplexing, the receiver sends report blocks for the original and the retransmission streams in the same RR packet since there is a single session.

6.3 Retransmission requests

The NACK feedback message format defined in the AVPF profile SHOULD be used by receivers to send retransmission requests. Whether a receiver chooses to request a packet or not is an implementation issue. An actual receiver implementation should take into account such factors as the tolerable application delay, the network environment and the media type.

The receiver should generally assess whether the retransmitted packet would still be useful at the time it is received. The timestamp of the missing packet can be estimated from the timestamps of packets preceding and/or following the sequence number gap caused by the missing packet in the original stream. In most cases, some form of linear estimate of the timestamp is good enough.

Furthermore, a receiver should compute an estimate of the round-trip time (RTT) to the sender. This can be done, for example, by measuring the retransmission delay to receive a retransmission packet after a NACK has been sent for that packet. This estimate may also be obtained from past observations, RTCP report round-trip time if available or any other means. A standard mechanism for the receiver to estimate the RTT is specified in RTP Extended Reports [11].

The receiver should not send a retransmission request as soon as it detects a missing sequence number but should add some extra delay to compensate for packet reordering. This extra delay may, for example, be based on past observations of the experienced packet reordering.

To increase the robustness to the loss of a NACK or of a retransmission packet, a receiver may send a new NACK for the same packet. This is referred to as multiple retransmissions. Before sending a new NACK for a missing packet, the receiver should rely on a timer to be reasonably sure that the previous retransmission attempt has failed in order to avoid unnecessary retransmissions.

NACKs MUST be sent only for the original RTP stream. Otherwise, if a receiver wanted to perform multiple retransmissions by sending a NACK in the retransmission stream, it would not be able

to know the original sequence number and a timestamp estimation of the packet it requests.

6.4 Timing rules

The NACK feedback message may be sent in a regular full compound RTCP packet or in an early RTCP packet, as per AVPF [1]. Sending a NACK in an early packet allows to react more quickly to a given packet loss. However, in that case if a new packet loss occurs right after the early RTCP packet was sent, the receiver will then have to wait for the next regular RTCP compound packet after the early packet. Sending NACKs only in regular RTCP compound decreases the maximum delay between detecting an original packet loss and being able to send a NACK for that packet. Implementers should consider the possible implications of this fact for the application being used.

Furthermore, receivers may make use of the minimum interval between regular RTCP compound packets. This interval can be used to keep regular receiver reporting down to a minimum, while still allowing receivers to send early RTCP packets during periods requiring more frequent feedback, e.g. times of higher packet loss rate. Note that although RTCP packets may be suppressed because they do not contain NACKs, the same RTCP bandwidth as if they were sent needs to be available. See AVPF [1] for details on the use of the minimum interval.

7. Congestion control

RTP retransmission poses a risk of increasing network congestion. In a best-effort environment, packet loss is caused by congestion. Reacting to loss by retransmission of older data without decreasing the rate of the original stream 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 the profile, an RTP application can determine its acceptable bitrate and packet rate in order to be fair to other TCP or RTP flows.

If an RTP application uses retransmission, the acceptable packet rate and bitrate includes both the original and retransmitted data. This guarantees that an application using retransmission achieves the same fairness as one that does not. Such a rule would translate in practice into the following actions:

If enhanced service is used, it should be 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 selectively 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 packet loss rate within acceptable parameters. Packet loss is considered acceptable if a TCP flow across the same network path and experiencing the same network conditions would achieve, on a reasonable timescale, an average throughput, that is not less than the one the RTP flow achieves. If the packet loss rate exceeds an acceptable level, it SHOULD be concluded that congestion is not kept under control and retransmission SHOULD NOT then be used. It may further be necessary to adapt the transmission rate (or the number of layers subscribed for a layered multicast session), or to arrange for the receiver to leave the session.

8. Retransmission Payload Format MIME type registration

8.1 Introduction

The following MIME subtype name and parameters are introduced in this document: "rtx", "rtx-time" and "apt".

The binding used for the retransmission stream to the payload type number is indicated by an rtpmap attribute. The MIME subtype name used in the binding is "rtx".

The "apt" (associated payload type) parameter MUST be used to map the retransmission payload type to the associated original stream payload type. If multiple original payload types are used, then multiple "apt" parameters MUST be included to map each original payload type to a different retransmission payload type.

An OPTIONAL payload-format-specific parameter, "rtx-time", indicates the maximum time a sender will keep an original RTP packet in its buffers available for retransmission. This time starts with the first transmission of the packet.

The syntax is as follows:

a=fmtp:<number> apt=<apt-value>;rtx-time=<rtx-time-val>

where,

<number>: indicates the dynamic payload type number assigned to the retransmission payload format in an rtpmap attribute.

<apt-value>: the value of the original stream payload type to

which this retransmission stream payload type is associated.

<rtx-time-val>: specifies the time in milliseconds (measured from the time a packet was first sent) that a sender keeps an

RTP packet in its buffers available for retransmission. The absence of the rtx-time parameter for a retransmission stream means that the maximum retransmission time is not defined, but MAY be negotiated by other means.

8.2 Registration of audio/rtx

MIME type: audio

MIME subtype: rtx

Required parameters:

rate: the RTP timestamp clockrate is equal to the RTP timestamp clockrate of the media that is retransmitted.

apt: associated payload type. The value of this parameter is the payload type of the associated original stream.

Optional parameters:

rtx-time: indicates the time in milliseconds (measured from the time a packet was first sent) that the sender keeps an RTP packet in its buffers available for retransmission.

Encoding considerations: this type is only defined for transfer via RTP.

Security considerations: see Section 12 of RFC XXXX

Interoperability considerations: none

Published specification: RFC XXXX

Applications which use this media type: multimedia streaming applications

Additional information: none

Person & email address to contact for further information:
rey@panasonic.de
david.leon@nokia.com
avt@ietf.org

Intended usage: COMMON

Author/Change controller:
Jose Rey

David Leon
IETF AVT WG

8.3 Registration of video/rtx

MIME type: video

MIME subtype: rtx

Required parameters:

rate: the RTP timestamp clockrate is equal to the RTP timestamp clockrate of the media that is retransmitted.

apt: associated payload type. The value of this parameter is the payload type of the associated original stream.

Optional parameters:

rtx-time: indicates the time in milliseconds (measured from the time a packet was first sent) that the sender keeps an RTP packet in its buffers available for retransmission.

Encoding considerations: this type is only defined for transfer via RTP.

Security considerations: see Section 12 of RFC XXXX

Interoperability considerations: none

Published specification: RFC XXXX

Applications which use this media type: multimedia streaming applications

Additional information: none

Person & email address to contact for further information:
rey@panasonic.de
david.leon@nokia.com
avt@ietf.org

Intended usage: COMMON

Author/Change controller:
Jose Rey
David Leon
IETF AVT WG

8.4 Registration of text/rtx

MIME type: text

MIME subtype: rtx

Required parameters:

rate: the RTP timestamp clockrate is equal to the RTP timestamp clockrate of the media that is retransmitted.

apt: associated payload type. The value of this parameter is the payload type of the associated original stream.

Optional parameters:

rtx-time: indicates the time in milliseconds (measured from the time a packet was first sent) that the sender keeps an RTP packet in its buffers available for retransmission.

Encoding considerations: this type is only defined for transfer via RTP.

Security considerations: see Section 12 of RFC XXXX

Interoperability considerations: none

Published specification: RFC XXXX

Applications which use this media type: multimedia streaming applications

Additional information: none

Person & email address to contact for further information:
rey@panasonic.de
david.leon@nokia.com
avt@ietf.org

Intended usage: COMMON

Author/Change controller:
Jose Rey
David Leon
IETF AVT WG

8.5 Registration of application/rtx

MIME type: application

MIME subtype: rtx

Required parameters:

rate: the RTP timestamp clockrate is equal to the RTP timestamp clockrate of the media that is retransmitted.

apt: associated payload type. The value of this parameter is the payload type of the associated original stream.

Optional parameters:

rtx-time: indicates the time in milliseconds (measured from the time a packet was first sent) that the sender keeps an RTP packet in its buffers available for retransmission.

Encoding considerations: this type is only defined for transfer via RTP.

Security considerations: see Section 12 of RFC XXXX

Interoperability considerations: none

Published specification: RFC XXXX

Applications which use this media type: multimedia streaming applications

Additional information: none

Person & email address to contact for further information:
rey@panasonic.de
david.leon@nokia.com
avt@ietf.org

Intended usage: COMMON

Author/Change controller:
Jose Rey
David Leon
IETF AVT WG

8.6 Mapping to SDP

The information carried in the MIME media type specification has a specific mapping to fields in SDP [5], which is commonly used to describe RTP sessions. When SDP is used to specify retransmissions for an RTP stream, the mapping is done as follows:

- The MIME types ("video"), ("audio"), ("text") and ("application") go in the SDP "m=" as the media name.

- The MIME subtype ("rtx") goes in SDP "a=rtpmap" as the encoding name. The RTP clock rate in "a=rtpmap" MUST be that of the retransmission payload type. See Section 4 for details on this.

- The AVPF profile-specific parameters "ack" and "nack" go in SDP

"a=rtcp-fb". Several SDP "a=rtcp-fb" are used for several types of feedback. See the AVPF profile [1] for details.

- The retransmission payload-format-specific parameters "apt" and "rtx-time" go in the SDP "a=fmtp" as a semicolon separated list of parameter=value pairs.

- Any remaining parameters go in the SDP "a=fmtp" attribute by copying them directly from the MIME media type string as a semicolon separated list of parameter=value pairs.

In the following sections some example SDP descriptions are presented. In some of these examples, long lines are folded to meet the column width constraints of this document; the backslash ("\") at the end of a line and the carriage return that follows it should be ignored.

8.7 SDP description with session-multiplexing

In the case of session-multiplexing, the SDP description contains one media specification "m" line per RTP session. The SDP MUST provide the grouping of the original and associated retransmission sessions' "m" lines, using the Flow Identification (FID) semantics defined in RFC 3388 [6].

The following example specifies two original, AMR and MPEG-4, streams on ports 49170 and 49174 and their corresponding retransmission streams on ports 49172 and 49176, respectively:

```
v=0
o=mascha 2980675221 2980675778 IN IP4 host.example.net
c=IN IP4 192.0.2.0
a=group:FID 1 2
a=group:FID 3 4
m=audio 49170 RTP/AVPF 96
a=rtpmap:96 AMR/8000
a=fmtp:96 octet-align=1
a=rtcp-fb:96 nack
a=mid:1
m=audio 49172 RTP/AVPF 97
a=rtpmap:97 rtx/8000
a=fmtp:97 apt=96;rtx-time=3000
a=mid:2
m=video 49174 RTP/AVPF 98
a=rtpmap:98 MP4V-ES/90000
a=rtcp-fb:98 nack
a=fmtp:98 profile-level-id=8;config=010100000120008840066820209\
0A21F
a=mid:3
m=video 49176 RTP/AVPF 99
a=rtpmap:99 rtx/90000
```

```
a=fmtp:99 apt=98;rtx-time=3000
a=mid:4
```

A special case of the SDP description is a description that contains only one original session "m" line and one retransmission session "m" line, the grouping is then obvious and FID semantics MAY be omitted in this special case only.

This is illustrated in the following example, which is an SDP description for a single original MPEG-4 stream and its corresponding retransmission session:

```
v=0
o=mascha 2980675221 2980675778 IN IP4 host.example.net
c=IN IP4 192.0.2.0
m=video 49170 RTP/AVPF 96
a=rtpmap:96 MP4V-ES/90000
a=rtcp-fb:96 nack
a=fmtp:96 profile-level-id=8;config=01010000012000884006682C209\
0A21F
m=video 49172 RTP/AVPF 97
a=rtpmap:97 rtx/90000
a=fmtp:97 apt=96;rtx-time=3000
```

8.8 SDP description with SSRC-multiplexing

The following is an example of an SDP description for an RTP video session using SSRC-multiplexing with similar parameters as in the single-session example above:

```
v=0
o=mascha 2980675221 2980675778 IN IP4 host.example.net
c=IN IP4 192.0.2.0
m=video 49170 RTP/AVPF 96 97
a=rtpmap:96 MP4V-ES/90000
a=rtcp-fb:96 nack
a=fmtp:96 profile-level-id=8;config=01010000012000884006682C209\
0A21F
a=rtpmap:97 rtx/90000
a=fmtp:97 apt=96;rtx-time=3000
```

9. RTSP considerations

The Real-time Streaming Protocol (RTSP), RFC 2326 [7] is an application-level protocol for control over the delivery of data with real-time properties. This section looks at the issues involved in controlling RTP sessions that use retransmissions.

9.1 RTSP control with SSRC-multiplexing

In the case of SSRC-multiplexing, the "m" line includes both

original and retransmission payload types and has a single RTSP "control" attribute. The receiver uses the "m" line to request SETUP and TEARDOWN of the whole media session. The RTP profile contained in the Transport header MUST be the AVPF profile or

another suitable profile allowing extended feedback. If the SSRC value is included in the SETUP response's Transport header, it MUST be that of the original stream.

In order to control the sending of the session original media stream, the receiver sends as usual PLAY and PAUSE requests to the sender for the session. The RTP-info header that is used to set RTP-specific parameters in the PLAY response MUST be set according to the RTP information of the original stream.

When the receiver starts receiving the original stream, it can then request retransmission through RTCP NACKs without additional RTSP signalling.

9.2 RTSP control with session-multiplexing

In the case of session-multiplexing, each SDP "m" line has an RTSP "control" attribute. Hence, when retransmission is used, both the original session and the retransmission have their own "control" attributes. The receiver can associate the original session and the retransmission session through the FID semantics as specified in Section 8.

The original and the retransmission streams are set up and torn down separately through their respective media "control" attribute. The RTP profile contained in the Transport header MUST be the AVPF profile or another suitable profile allowing extended feedback for both the original and the retransmission session.

The RTSP presentation SHOULD support aggregate control and SHOULD contain a session level RTSP URL. The receiver SHOULD use aggregate control for an original session and its associated retransmission session. Otherwise, there would need to be two different 'session-id' values, i.e. different values for the original and retransmission sessions, and the sender would not know how to associate them.

The session-level "control" attribute is then used as usual to control the playing of the original stream. When the receiver starts receiving the original stream, it can then request retransmissions through RTCP without additional RTSP signalling.

9.3 RTSP control of the retransmission stream

Because of the nature of retransmissions, the sending of retransmission packets SHOULD NOT be controlled through RTSP PLAY and PAUSE requests. The PLAY and PAUSE requests SHOULD NOT affect the retransmission stream. Retransmission packets are sent upon receiver requests in the original RTCP stream, regardless of the

state.

9.4 Cache control

Retransmission streams SHOULD NOT be cached.

In the case of session-multiplexing, the "Cache-Control" header SHOULD be set to "no-cache" for the retransmission stream.

In the case of SSRC-multiplexing, RTSP cannot specify independent caching for the retransmission stream, because there is a single "m" line in SDP. Therefore, the implementer should take this fact into account when deciding whether to cache an SSRC-multiplexed session or not.

10. Implementation examples

This document mandates only the sender and receiver behaviours that are necessary for interoperability. In addition, certain algorithms, such as rate control or buffer management when targeted at specific environments, may enhance the retransmission efficiency.

This section gives an overview of different implementation options allowed within this specification.

The first example describes a minimal receiver implementation. With this implementation, it is possible to retransmit lost RTP packets, detect efficiently the loss of retransmissions and perform multiple retransmissions, if needed. Most of the necessary processing is done at the server.

The second example shows how a receiver may implement additional enhancements that might help reduce sender buffer requirements and optimise the retransmission efficiency

The third example shows how retransmissions may be used in (small) multicast groups in conjunction with layered encoding. It illustrates that retransmissions and layered encoding may be complementary techniques.

10.1 A minimal receiver implementation example

This section gives an example of an implementation supporting multiple retransmissions. The sender transmits the original data in RTP packets using the MPEG-4 video RTP payload format. It is assumed that NACK feedback messages are used, as per [1]. An SDP description example with SSRC-multiplexing is given below:

```
v=0
o=mascha 2980675221 2980675778 IN IP4 host.example.net
```

```
c=IN IP4 192.0.2.0
m=video 49170 RTP/AVPF 96 97
a=rtpmap:96 MP4V-ES/90000
a=rtcp-fb:96 nack
```

```
a=rtpmap:97 rtx/90000
a=fmtp:97 apt=96;rtx-time=3000
```

The format-specific parameter "rtx-time" indicates that the server will buffer the sent packets in a retransmission buffer for 3.0 seconds, after which the packets are deleted from the retransmission buffer and will never be sent again.

In this implementation example, the required RTP receiver processing to handle retransmission is kept to a minimum. The receiver detects packet loss from the gaps observed in the received sequence numbers. It signals lost packets to the sender through NACKs as defined in the AVPF profile [1]. The receiver should take into account the signalled sender retransmission buffer length in order to dimension its own reception buffer. It should also derive from the buffer length the maximum number of times the retransmission of a packet can be requested.

The sender should retransmit the packets selectively, i.e. it should choose whether to retransmit a requested packet depending on the packet importance, the observed QoS and congestion state of the network connection to the receiver. Obviously, the sender processing increases with the number of receivers as state information and processing load must be allocated to each receiver.

10.2 An enhanced receiver implementation example

The receiver may have more accurate information than the sender about the current network QoS such as available bandwidth, packet loss rate, delay and jitter. In addition, other receiver-specific parameters such as buffer level, estimated importance of the lost packet and application level QoS may be used by the receiver to make a more efficient use of RTP retransmission by selectively sending NACKs for important lost packets and not for others. For example, a receiver may decide to suppress a request for a packet loss that could be concealed locally, or for a retransmission that would arrive late.

Furthermore, a receiver may acknowledge the received packets. This can be done by sending ACKs, as per [1]. Upon receiving an ACK, the sender may delete all the acknowledged packets from its retransmission buffer. Note that this would also require only limited increase in the required RTCP bandwidth as long as ACK packets are sent seldom enough.

This implementation may help reduce buffer requirements at the sender and optimise the performance of the implementation by using

selective requests.

Note that these receiver enhancements do not need to be negotiated as they do not affect the sender implementation. However, in order to allow the receiver to acknowledge packets, it is needed

to allow the use of ACKs in the SDP description, by means of an additional SDP "a=rtcp-fb" line, as follows:

```
v=0
o=mascha 2980675221 2980675778 IN IP4 host.example.net
c=IN IP4 192.0.2.0
m=video 49170 RTP/AVPF 96 97
a=rtpmap:96 MP4V-ES/90000
a=rtcp-fb:96 nack
a=rtcp-fb:96 ack
a=rtpmap:97 rtx/90000
a=fmtp:97 apt=96;rtx-time=3000
```

10.3 Retransmission of Layered Encoded Media in Multicast

This section shows how to combine retransmissions with layered encoding in multicast sessions. Note that the retransmission framework is not intended as a complete solution to reliable multicast. Refer to RFC 2887 [10], for an overview of the problems related with reliable multicast transmission.

Packets of different importance are sent in different RTP sessions. The retransmission streams corresponding to the different layers can themselves be seen as different retransmission layers. The relative importance of the different retransmission streams should reflect the relative importance of the different original streams.

In multicast, SSRC-multiplexing of the original and retransmission streams is not allowed as per Section 5.3 of this document. For this reason, the retransmission stream(s) MUST be sent in different RTP session(s) using session-multiplexing.

An SDP description example of multicast retransmissions for layered encoded media is given below:

```
m=video 8000 RTP/AVPF 98
c=IN IP4 192.0.2.0/127/3
a=rtpmap:98 MP4V-ES/90000
a=rtcp-fb:98 nack
m=video 8000 RTP/AVPF 99
c=IN IP4 192.0.2.4/127/3
a=rtpmap:99 rtx/90000
a=fmtp:99 apt=98;rtx-time=3000
```

The server and the receiver may implement the retransmission methods illustrated in the previous examples. In addition, they may choose to request and retransmit a lost packet depending on the layer it belongs to.

11. IANA considerations

A new MIME subtype name, "rtx", has been registered for four different media types, as follows: "video", "audio", "text" and "application". An additional REQUIRED parameter, "apt", and an OPTIONAL parameter, "rtx-time", are defined. See Section 8 for details.

12. Security considerations

If cryptography is used to provide security services on the original stream, then the same services, with equivalent cryptographic strength, MUST be provided on the retransmission stream. Old keys will likely need to be cached so that when the keys change for the original stream, the old key is used until it is determined that the key has changed on the retransmission packets as well.

The use of the same key for the retransmitted stream and the original stream may lead to security problems, e.g. two-time pads. This sharing has to be evaluated towards the chosen security protocol and security algorithms.

Furthermore, it is RECOMMENDED that the cryptography mechanisms used for this payload format provide protection against known plaintext attacks. RTP recommends that the initial RTP timestamp SHOULD be random to secure the stream against known plaintext attacks. This payload format does not follow this recommendation as the initial timestamp will be the media timestamp of the first retransmitted packet. However, since the initial timestamp of the original stream is itself random, if the original stream is encrypted, the first retransmitted packet timestamp would also be random to an attacker. Therefore, confidentiality would not be compromised.

Congestion control considerations with the use of retransmission are dealt with in Section 7 of this document.

Any other security considerations of the profile under which the retransmission scheme is used should be applied. The retransmission payload format MUST NOT be used under the SAVP profile defined by the Secure Real-Time Transport Protocol (SRTP)[12] but instead an extension of SRTP should be defined to secure the AVPF profile. The definition of such a profile is out of the scope of this document.

13. Acknowledgements

We would like to express our gratitude to Carsten Burmeister for

his participation in the development of this document. Our thanks also go to Koichi Hata, Colin Perkins, Stephen Casner, Magnus Westerlund, Go Hori and Rahul Agarwal for their helpful comments.

14. References

14.1 Normative References

1 J. Ott, S. Wenger, N. Sato, C. Burmeister, J. Rey, "Extended RTP profile for RTCP-based feedback", draft-ietf-avt-rtcp-feedback-04.txt, September 2002.

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

3 H. Schulzrinne, S. Casner, R. Frederick and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", draft-ietf-avt-rtp-new-12.txt, March 2003.

4 S. Casner, "SDP bandwidth modifiers for RTCP bandwidth", draft-ietf-avt-rtcp-bw-05.txt, May 2002.

5 M. Handley, V. Jacobson, "SDP: Session Description Protocol", RFC 2327, April 1998.

6 G. Camarillo, J. Holler, G. AP. Eriksson, "Grouping of media lines in the Session Description Protocol (SDP)", RFC 3388, December 2002.

7 H. Schulzrinne, A. Rao, R. Lanphier, "Real Time Streaming Protocol (RTSP)", RFC 2326, April 1998.

14.2 Informative References

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

9 G. Hellstrom, "RTP for conversational text", RFC 2793, May 2000

10 M. Handley, et al., "The Reliable Multicast Design Space for Bulk Data Transfer", RFC 2887, August 2000.

11 Friedman, et. al., "RTP Extended Reports", Work in Progress.

12 M. Baugher, D. A. McGrew, D. Oran, R. Blom, E. Carrara, M. Naslund, K. Norrman, "The Secure Real-Time Transport Protocol", draft-ietf-avt-srtp-05.txt, June 2002.

13 R. Hovey and S. Bradner, "The Organizations Involved in the IETF Standards Process," BCP 11, RFC 2028, IETF, October 1996.

15. Author's Addresses

Jose Rey rey@panasonic.de
Panasonic European Laboratories GmbH

Monzastr. 4c
D-63225 Langen, Germany
Phone: +49-6103-766-134
Fax: +49-6103-766-166

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

Akihiro Miyazaki akihiro@isl.mei.co.jp
Core Software Development Center
Corporate Software Development Division
Matsushita Electric Industrial Co., Ltd.
1006 Kadoma, Kadoma City, Osaka 571-8501, Japan
Phone: +81-6-6900-9192
Fax: +81-6-6900-9193

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

Rolf Hakenberg hakenberg@panasonic.de
Panasonic European Laboratories GmbH
Monzastr. 4c
D-63225 Langen, Germany
Phone: +49-6103-766-162
Fax: +49-6103-766-166

IPR Notices

Secretariat.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required

to practice this standard. Please address the information to the IETF Executive Director.

Full Copyright Statement