

Audio/Video Transport Working Group  
Internet Draft  
[draft-ietf-avt-rtp-selret-05.txt](#)  
Expires: December 2002

A.Miyazaki  
H.Fukushima  
K.Hata  
T.Wiebke  
R.Hakenberg  
C.Burmeister  
Matsushita

June 2002

## **RTP Payload Formats to Enable Multiple Selective Retransmissions**

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of [Section 10 of RFC2026](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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

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

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

### **Abstract**

This document describes new RTP payload formats, i.e. RTX and SEL, to enable multiple and optional selective retransmissions in RTP. These are especially applicable to environments where enhanced RTCP-based feedback, as per the RTP/AVPF profile [4], is available. Using these payload formats it is possible to classify the media stream according to prioritisation of packets or according to the transmissions status of the packet ( i.e. transmission or retransmission). These payload formats can be tailored to fit different scenarios, with different requirements and where different solutions are needed. Typical scenarios are unicast transmission and small multicast groups. In this document we give examples how these payload formats can be used to accommodate different retransmission schemes.



Changes from [draft-ietf-avt-rtp-selret-02.txt](#)

The main changes from version 02 to 03 were a new section about how to indicate the payload format usage in SDP, many clarifications regarding how to use the headers of the payload formats, especially regarding combination of headers and two new examples retransmission schemes including examples how to describe those with SDP. The following list describes the detailed changes:

All sections:

- editorial changes

[Section 3](#), Requirements for Retransmissions in RTP

- new section, highlighting identified requirements

[Section 4](#), General Use of the Payload Formats

- clarification about having several headers in one packet
- list of possible header combinations
- general rules of how to use the payload formats

[Section 5](#), Retransmission Payload Format

- changes in the header format to include the possibility to signal the complete original sequence number or the delta sequence number

[Section 6](#), Selective Payload Format

- changes in the header format to include the possibility to signal the complete referenced sequence number

[Section 9](#), Relation to SDP

- new section, describing how to use SDP to indicate the usage of the payload formats

[Section 10](#), Example Usages of the Payload Formats

- adding SDP descriptions to all examples
- two new example usages:
  - retransmissions on a different stream
  - retransmissions in layered transmissions

Changes from [draft-ietf-avt-rtp-selret-03.txt](#)

The changes from version 03 are mainly editorial. Minor errors were corrected and some clarifications added.

Changes from [draft-ietf-avt-rtp-selret-04.txt](#)

All sections

- minor editorial changes and wording.

#### [Section 4](#) Applicability Statement

- This section is fully new. In the IETF#53 it was decided to proceed with both this [draft-ietf-avt-rtp-selret-05.txt](#) and [draft-ietf-avt-rtp-retransmission-00.txt](#) for Last Call, as it was understood by the chairs, that they solve different problems. It was then decided both drafts must include an applicability statement.

#### [Section 7](#) Selective Payload Format

- Rewritten: the importance of timely feedback is emphasized.

#### [Section 8](#) Feedback

- minor corrections
- the Generic ACK message format is included.

#### [Section 11](#) Example usages

- Whole Section was rearranged:
  - [Section 11.2](#): example of a more intelligent client using the ACK scheme is new. SDP description included
  - [Section 11.3](#) Selective Payload Format example usage: re-written.
  - Section Sending Retransmissions on a Different Stream (was [Section 10.3](#)) removed: the two session approach example was removed from the draft.

## Table of Contents

<a href="#">1. Conventions used in this document.....</a>	<a href="#">5</a>
<a href="#">2. Introduction to the problem.....</a>	<a href="#">5</a>
<a href="#">3. Requirements for Retransmissions in RTP.....</a>	<a href="#">6</a>
<a href="#">4. Applicability Statement.....</a>	<a href="#">8</a>
<a href="#">5. General Usage of the Payload Formats.....</a>	<a href="#">10</a>
<a href="#">5.1 General Usage Rules.....</a>	<a href="#">10</a>
<a href="#">5.2 Header Usage and Encapsulation.....</a>	<a href="#">11</a>
<a href="#">5.3 Header Examples.....</a>	<a href="#">11</a>
<a href="#">6. Retransmission Payload Format.....</a>	<a href="#">14</a>
<a href="#">7. Selective Payload Format.....</a>	<a href="#">15</a>
<a href="#">8. Feedback.....</a>	<a href="#">17</a>
<a href="#">9. Congestion Control.....</a>	<a href="#">18</a>
<a href="#">10. Relation to SDP.....</a>	<a href="#">19</a>
<a href="#">10.1 Indicating RTX Usage in SDP.....</a>	<a href="#">19</a>
<a href="#">10.2 Indicating SEL Usage in SDP.....</a>	<a href="#">20</a>
<a href="#">10.3 Indicating RTX and SEL Usage in SDP.....</a>	<a href="#">21</a>
<a href="#">11. Example Usages of the Payload Formats.....</a>	<a href="#">21</a>
<a href="#">11.1 A Simple Multiple Retransmission Algorithm.....</a>	<a href="#">22</a>
<a href="#">11.2 A Client-oriented Retransmission Algorithm.....</a>	<a href="#">24</a>
<a href="#">11.3 Selective Retransmission Algorithm.....</a>	<a href="#">25</a>
<a href="#">11.4 Using Retransmissions in Layered Transmissions.....</a>	<a href="#">26</a>
Security Considerations.....	<a href="#">28</a>
Acknowledgements.....	<a href="#">28</a>
Intellectual property considerations.....	<a href="#">28</a>
References.....	<a href="#">28</a>
Author's Addresses.....	<a href="#">29</a>

## **1. Conventions used in this document**

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](#) [1].

## **2. Introduction to the problem**

The Real-time Transport Protocol RTP [3] was designed as an Internet protocol to transmit real-time (or near real-time) data over large multicast sessions. Even though the multicast capability is a strong feature of RTP, a common use of it is non-interactive unicast or small multicast transmission of media streams. These applications, further referred to as media streaming, have relaxed delay bounds. Thus, a constant offset delay, used to store received packets at the client, is tolerable and used for different purposes, such as delay jitter compensation.

Typically RTP is run over the User Datagram Protocol (UDP). While the unreliable UDP performs good for reliable channels, the quality of the received media stream is bad, when transmitted over error-prone links. Especially compressed streams, such as MPEG video-data, are highly susceptible to packet loss, which will lead to a bad video quality at the clients. E.g. in a wireless environment with an unstable transmission quality (bit error rate (BER) about  $10E-3$  ..  $10E-6$ ) a sufficient quality of video and voice transmitted by RTP cannot be achieved in most of the cases.

The scenario for media streaming over an error-prone link might be the one in Figure 1. Media files are stored on a media server which is connected via an Internet link to a mobile network. A client using a mobile terminal is connected to the mobile network via a wireless link. Both, the Internet link and the wireless link

are generally unreliable and subject to packet losses, either due to congestion or to bit errors.

This scenario is depicted below.

Miyazaki, et. al. Expires December 2002 5  
Multiple Selective RTP Retransmissions June 2002

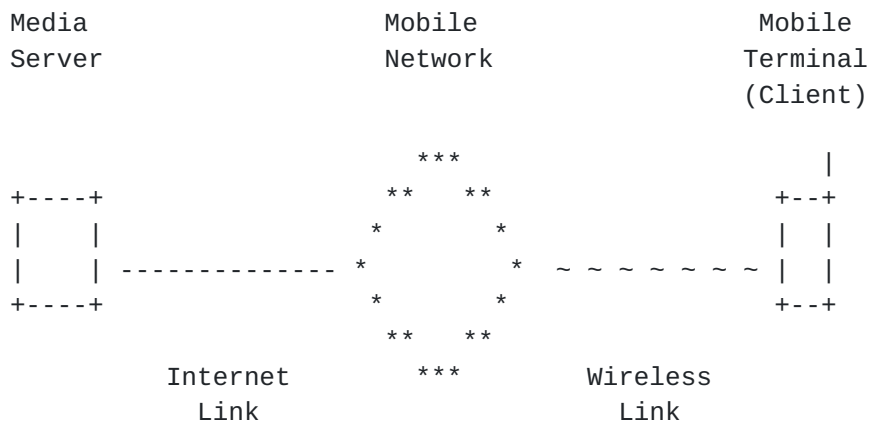


Figure 1 : Scenario for media streaming

To achieve a better quality of the received media stream, especially over error-prone links, retransmissions of lost packets of the stream is a possible solution.

### 3. Requirements for Retransmissions in RTP

There are several issues that have to be considered for retransmissions in RTP. We describe in this section some of those, which lead to a set of requirements for the retransmission scheme.

As the RTP sender SHOULD employ a congestion control algorithm if the session is transmitted over a best effort environment (see RTP

[3]), the sending rate is typically limited. Packet losses might also decrease the allowed sending rate, if these are interpreted as congestion signal. What is more, in case of transmission over a mobile channel the bandwidth is a very scarce resource and has to be used with great care. Thus, RTP senders (especially when transmitting data over wireless channels) are often confronted with strictly limited available bit rates.

Hence, it is in most cases not useful or even not allowed to send retransmissions of all lost packets additionally to the original stream (note however that there are scenarios of Quality of Service (QoS) controlled networks, where it can be possible to reserve enough bandwidth in advance to transmit retransmission of all lost packets in addition to the original stream).

On the other hand, the media data is typically coded for compression before transmission. This reduction in redundancy leads to the streams being more susceptible to errors. Another property of some encoded media streams (e.g. video) is that they may consist of data of different importance. Some parts of the data might be absolutely vital for displaying the media, while other parts may only be required to enhance the quality of the displayed stream.

A common mechanism to achieve different QoS for different parts of

Miyazaki, et. al.	Expires December 2002	6
Multiple Selective RTP Retransmissions		June 2002

the data is to split the data in layers and transmit these layers in different RTP sessions, i.e. layered encoding and transmission. The layers are typically hierarchically ordered, which means that one layer (i.e. the base layer, 1st layer) is independently decodable; the next layer (2nd, i.e. the first enhancement layer) enhances the quality of the received stream, but depends on the base layer to be decoded; the next layer (3rd) depends on the first enhancement layer and so on. As the layers are transmitted in different sessions, the client can choose how many layers it would like to receive, according to available bandwidth or other criteria. Layered encoding and transmission is an accepted technique, especially for multicast transmissions, to achieve different QoS levels within one media stream or to perform congestion control in multicast.

For such a scheme to be feasible with retransmissions in RTP, it should be possible to keep the service differentiation between the layers, i.e. to differentiate between retransmissions of different streams.

Thus, when transmitting compressed media (e.g. MPEG video) over strictly limited bandwidth links (as described above), it has to be taken care that important data is retransmitted. Thereby, it might be possible to skip the transmission of non-important data, for the



benefit of transmitting important data. With a strategy of this kind, the bandwidth requirements may be kept, while the quality of the received stream can be increased.

For all these reasons, a retransmission scheme that enables the retransmission of all or selected packets and attempts to retransmit these packets one or several times without introducing avoidable delay is needed. Additionally, layered encoding has to be supported.

We see already that there is the need for retransmission schemes with sophisticated functionalities. It might not be possible to design one scheme that fits all requirements in an optimum way. Therefore, we define in this document two new RTP payload formats. With these formats it is possible to design retransmission schemes (where the intelligence is at the server or at the client side) that fulfill the requirements written above. We will give examples of how to use the payload formats to enable different retransmission schemes at the end of this document.

Miyazaki, et. al.	Expires December 2002	7
Multiple Selective RTP Retransmissions		June 2002

#### **4. Applicability Statement**

There are two proposals for adding retransmissions to RTP (this [draft-ietf-avt-rtp-selret-05.txt](#) and [draft-ietf-avt-rtp-retransmission-01.txt](#)). Both proposals may be optimum under a different set of constraints, as they solve different problems. This section describes the applicability of the "RTP Payload Formats to Enable Multiple Selective Retransmissions". Herewith, we present applications that benefit from these retransmission payload formats, i.e. RTX and SEL, and mention those for which it is not applicable.

The main difference between both retransmission proposals is the way the retransmissions are sent, i.e. in the same RTP session for this proposal and in an additional session for the "RTP Retransmission Framework" proposal. We put our emphasis on the effects of this

difference.

Using the RTX payload format to send the retransmissions in the same RTP session as the original media data, it is always possible to reliably detect the loss of RTP packets. As soon as the receiver detects a gap in the sequence number space it knows an RTP packet was lost. However, it is not immediately possible to detect the nature of the lost data, as it might be a retransmission or new user data. Using the payload format to multiplex the retransmissions with the RTP sequence number can result in gaps in the final sequence number space. More exactly: there will be one gap for each retransmission. While this is generally not a problem, the implementer should check carefully whether the application can cope with this. The authors are aware of the fact that the "RTP Conversational Text" ([RFC2793](#)) requires RTP to indicate lost data packets to the application. For this application the sequence number multiplexing should not be used.

In the following, some implementation examples of this proposal are presented. The examples are described in detail in [section 11](#) and are shortly mentioned here to help in understanding the applicability of the proposed payload formats.

A simple retransmission mechanism that can run on thin clients can be built with the RTX payload format. It needs no additional timers. The client will detect lost RTP packets (transmissions or retransmissions) immediately by a gap in the received sequence numbers. After detecting lost RTP data the client can request a retransmission as soon as it is allowed to send an RTCP feedback message.

For this simple solution, the server should take the final decision regarding retransmissions according to a judgment of the time after which packets become invalid at the client. Additional timers, e.g. for RTT estimation or for retransmissions, are therefore not

Miyazaki, et. al.	Expires December 2002	8
Multiple Selective RTP Retransmissions		June 2002

required at the client. However, when using this simple scheme it is not possible to detect the loss of a retransmission request. A solution for this case is presented in the next paragraph.

In some environments, feedback is limited due to back-channel constraints or by the application running over RTP, as per [4]. For such environments, it is important to limit the quantity of feedback messages and to schedule them appropriately so that they are still useful to the server. In order to meet these two requirements the

SEL payload format, i.e. Selective Payload Format, is presented in this document in [section 7](#). The SEL payload format enables the server to signal the priority of a packet in-band. The SEL payload format also enables the client to resolve the time stamp of lost high priority RTP packets, thus enabling the detection of lost retransmission requests for high priority packets. For details see [section 7](#) of this document.

Until now, the server was in charge of taking the final decision regarding retransmissions. For those environments in which the server load is an issue, a more client-oriented retransmission scheme can be built using the payload formats described in this draft. Thereby, this scheme requires slightly more intelligence at the client, but less complexity at the server: the client issues periodically (maybe additionally to the retransmission requests) cumulative acknowledgements. By doing this, the server is informed about the outdated packets, complexity is removed from the server and the scheme is made scalable. See [section 11.2](#) for details.

Retransmissions imply overhead. Every retransmission that is sent using this payload format includes at least two additional bytes overhead. A retransmission of a retransmission increases this overhead again by two bytes, e.g. the second retransmission of one packet will carry four bytes overhead. This is needed to enable the receiver to 1) resolve the RTP sequence number used for the transmission and possible retransmissions and to 2) simplify the encapsulation at the server.

Retransmissions lead to a highly increased RTCP jitter calculation. As the retransmission usually is received highly out of order, the RTCP jitter will increase. While one could argue whether the jitter calculation is usable this way or not, the implementer should take this fact into consideration.

If RTP mixers or translators wanted to separate the stream in original data and retransmissions they could do that according to the Payload Type field in the RTP header.

## **5. General Usage of the Payload Formats**

The payload formats presented in this document, i.e. the RTX retransmission payload format and the SEL selective retransmission

payload format, are meant as a solution to enable retransmissions in RTP. Based on this scheme we present examples how the payload formats could be used to realize different retransmission schemes. Thereby, the focus may lay, for example, on optimizing bandwidth efficiency within the media stream (or on the back-channel), on reducing the server( or client) load or on the need for supporting layered encoding.

The payload formats are not meant to replace a media RTP payload format, but to be used additionally. This means that an RTP packet using one or both of the formats defined below, can have several encapsulated headers, where the outer header always indicates the payload type of the next inner header (see examples at the end of this section).

### **5.1 General Usage Rules**

If the RTX payload format is used, e.g. a payload type number is dynamically assigned, all retransmissions **SHOULD** be sent using this RTX payload type. Thereby the header encapsulation scheme, which is described in [section 5.2](#) **SHOULD** be used.

The sender of RTP packets, using the payload formats specified in This document has different possibilities for transmitting RTP packets. The sender **MAY** use the Selective (SEL) payload format to signal the packet's priority and the sequence number of the last high priority packet in-band. This can be used at the receiver of the packets to detect if lost packets had high or low priority. Additionally, information on the time stamp of the lost packet might also be conveyed by the SEL payload format. This can help the receiver when judging whether or not a retransmission request should be issued or not and to detect the loss of a retransmission request.

If the SEL payload format is specified for an RTP session, e.g. a payload type number is dynamically assigned, the client **SHOULD** report the loss of high priority packets in the first place. After that and, as long as RTCP feedback bandwidth allows, the client **MAY** additionally report the loss of low priority packets. Furthermore, the client **MAY** send a retransmission request for several packets, including low priority packets (using the BLP field of the generic NACK packet, see [section 8](#)), as long as there is, either at least one high priority packet lost or the client cannot judge if there might be a high priority packet lost within the lost packets.

## **5.2 Header Usage and Encapsulation**

For the encapsulation of retransmissions in the RTX packet the following rules apply:

1. The sender of the retransmission SHOULD strip off the RTP header of the original RTP packet, that is to be retransmitted, but store the values of the header fields temporarily for use in this packet's header fields (see step 2 and 4). It is important to note that only the RTP header is stripped off, payload format specific headers are kept as they are. Hence, particularly RTX and SEL specific headers SHOULD NOT be stripped off, but SHOULD be transmitted together with the headers that are generated in the next steps.
2. An RTX header is generated and placed before the packet. If the difference between the RTP sequence number used for this RTP packet and the sequence number used for the original transmission fits into 8-bits, then the 8-bit delta SN field MAY be used instead of the full sequence number. If it exceeds the 8 bits, the complete sequence number of the original RTP transmissions MUST be included in the header. Receivers MUST be able to handle either RTX format.
3. A SEL header MAY be placed before the packet, if the sender wishes to signal the priority of the packet in-band.
4. The RTP header is generated. Thereby the fields of the header are filled as usual, e.g. as described in the payload format that is used for the media to be transported. However, the RTP sequence number MUST be increased by one to the previous RTP packet and the timestamp and marker bit MUST be set to the values of the original transmission's RTP header.

## **5.3 Header Examples**

In the following, we give examples of RTP packets using the RTX and SEL payload format:

1. Transmission of an RTP packet, using the SEL format to signal the priority of the packet in-band:

```

+-----+
|          RTP header with PT=SEL          |
|                                          |
+-----+
|          SEL Payload Type specific header          |
|          with PT=media_pt                    |
+-----+
:          Media Payload Type specific header          :
:                                                  :
+-----+
|          Media Payload                      |
|                                          |
:                                                  :
:..... . . . . . :

```

2. Retransmission of an RTP packet, using the RTX format for the retransmission:

```

+-----+
|          RTP header with PT=RTX          |
|                                          |
+-----+
|          RTX Payload Type specific header          |
|          with PT=media_pt                    |
+-----+
:          Media Payload Type specific header          :
:                                                  :
+-----+
|          Media Payload                      |
|                                          |
:                                                  :
:..... . . . . . :

```

3. Retransmission of an RTP packet, which was originally transmitted using the SEL format to signal the priority of the packet in-band. The RTX format for the retransmission is used as an additional header, as well as a new SEL header to signal the new priority of the packet:

```

+-----+
|               RTP header with PT=SEL               |
|                                                     |
+-----+
|               SEL Payload Type specific header      |
|               with PT=RTX                          |
+-----+
|               RTX Payload Type specific header      |
|               with PT=SEL                          |
+-----+
|               SEL Payload Type specific header      |
|               with PT=media_pt                     |
+-----+
|: Media Payload Type specific header :|
|:                                   :|
+-----+
|:                                   :|
|               Media Payload                       |
|:                                   :|
+:                                   :+
+-----+

```

4. Retransmission of a retransmitted RTP packet, using the RTX format twice, conform to the rules given above:

```

+-----+
|               RTP header with PT=RTX               |
|-----+
|               RTX Payload Type specific header      |
|               with PT=RTX                          |
+-----+
|               RTX Payload Type specific header      |
|               with PT=media_pt                     |
+-----+
|               Media Payload Type specific header    |
|-----+
|

```

```

|           Media Payload           |
:                                   :
:..... . . . . :

```

The previous examples show that there might be quite some headers involved in sending a single RTP packet. There are some observations to be done on this; first, the header encapsulation is straight forward regarding server side implementation of retransmissions and buffering of transmitted packets. Second, the overhead because of the retransmission headers is quite small. Usually only one RTX header is used for retransmissions and thus the additional overhead is two or four bytes. Retransmitted retransmissions have two RTX headers, which sums up to 4 to 8 bytes of additional overhead.

Miyazaki, et. al.	Expires December 2002	13
Multiple Selective RTP Retransmissions		June 2002

Multiple retransmissions can happen and enhance the performance, but still these are quite rare cases. Considering that RTP is designed for real-time or near-real-time services, several retransmission attempts will soon run out of time.

## 6. Retransmission Payload Format

Retransmissions in RTP SHOULD be sent using RTX payload format. This enables the receiver to reconstruct exactly the original RTP header and all its containing information. Additionally, multiple retransmissions are enabled. The syntax of the payload type specific header is as follows:

```

  0   1   2   3   4   5   6   7
+---+---+---+---+---+---+---+
| X | PT of inner packet header |
+---+---+---+---+---+---+---+
| (Delta) RTP Sequence Number |
+---+---+---+---+---+---+---+
: original RTP Sequence Number : if X=1
+.....+

```

Extended Sequence Number (X): 1 bit

This bit indicates whether a Delta RTP Sequence Number is used or the complete original RTP sequence number. For X=0 the Delta RTP Sequence Number are transmitted, for X=1 the original RTP sequence number is inserted in the header.

Payload Type of inner packet header: 7 bit

This field contains the original payload type information, i.e. the payload type of the payload that is transported in this



packet. This might be the payload type of the media itself or of another encapsulating payload type, e.g. the SEL payload type.

(Delta) RTP Sequence Number: 8 bit

The value of this field is dependent on the setting of the X-bit (see above). If the X-bit is set to zero, this field contains the difference of this packet's RTP sequence number to the RTP sequence number of the original transmission of this packet's payload, i.e.  $\Delta SN = (SN(this) - SN(original)) \bmod 2^{16}$ . If the number of bits that are needed to encode the difference exceeds 8, X=1 MUST be set and the complete original RTP sequence number is transmitted in the following two bytes (in network byte order), i.e. covering this field and the following

original RTP Sequence Number: 8 bit

For X=1 this field is used together with the previous eight bits to carry the complete original RTP sequence number (see also "(Delta) RTP Sequence Number").

Miyazaki, et. al.	Expires December 2002	14
	Multiple Selective RTP Retransmissions	June 2002

## **7. Selective Payload Format**

In certain scenarios it is necessary to either 1) limit the amount of feedback sent to the server or 2) be able to schedule feedback appropriately so that it can still be used to repair the stream or 3) both.

In such cases it is required to provide the client with means of deciding which packets are more important than others. The selective payload format (SEL) allows to indicate the priority of a packet in-band.

The low priority packets are needed to detect the loss of high priority packets, as they contain the sequence number of the last important packet sent. If the receiver gets a low priority packet, it has only to check whether it has received the indicated last high priority packet or not. If a high priority packet is lost the receiver SHOULD report the loss of this packet. If packets of low priority are lost the receiver MAY report this.

Additionally, the SEL payload type provides resilience against the feedback loss. The SEL payload format enables the client to detect the loss of retransmission requests of high priority packets. By using the timestamp information of lost high priority packets, which may be conveyed in the SEL header, the client is able to estimate if a retransmission request for these high priority packets would be

useful. If it is, the client reports these losses via RTCP feedback. If after an estimated RTT afterwards the requested packets didn't arrive at the client, it may conclude that the retransmission request was lost and a new one should be issued. Hereby, an RTT estimation at the client would help in estimating when a retransmission should have arrived at the client.

The syntax of the SEL header is as follows:

```

0   1   2   3   4   5   6   7
+---+---+---+---+---+---+---+
|DTI| PT of inner packet header |
+---+---+---+---+---+---+---+
|PRI| X |      (Delta) SN      |
+---+---+---+---+---+---+---+
:                               :
+      complete SN      +  if X==1
:                               :
+.....+
:                               :
+      Delta Time      +  if DTI==1
:                               :
+.....+

```

Miyazaki, et. al.	Expires December 2002	15
	Multiple Selective RTP Retransmissions	June 2002

DTI (Delta Time Indicator): 1 bit

DTI=1 indicates that the delta time field is present. DTI=0 means that the Delta Time field is not used.

Payload Type of inner packet header: 7 bit

This field contains the original payload type information, i.e. the payload type of the payload that is transported in this packet.

PRI (Priority): 1 bit

This field indicates the priority of the packet. PRI=1 means a high priority packet, where PRI=0 is sent for low priority packets.

X (Extended RTP Sequence Number): 1 bit

This field indicates whether a short "Delta SN" is used to indicate the last high priority packet (X=0) or the complete RTP sequence number of the last high priority packet is transmitted (X=1).

Delta SN (Delta Sequence Number): 6 bit

This field is only present in case of X=0, where it contains the offset in sequence number space to the previous important packet. It is calculated as  $DSN = SN_{this} - SN_{important}$ . If X=1 this field MUST be ignored.

Complete SN (complete referenced RTP Sequence Number): 16 bit

This field is present if X=1. In this case it contains the complete RTP sequence number of the last high priority packet.

Delta Time: 16 bit

This field indicates the difference between the time stamp of the last important RTP packet (that should be retransmitted if lost) and that of the current RTP packet. It can be used by the receiver to compute the timestamp of a lost important packet. The default unit of this field is [ms] and by default the following table is used to encode the Delta Time values:

Miyazaki, et. al.	Expires December 2002	16
Multiple Selective RTP Retransmissions		June 2002

Delta Time [ms]	bits
more than 32766	0111 1111 1111 1111
32766	0111 1111 1111 1110
32765	0111 1111 1111 1101
:	:
2	0000 0000 0000 0010
1	0000 0000 0000 0001
0	0000 0000 0000 0000
-1	1111 1111 1111 1111
-2	1111 1111 1111 1110
:	:
-32766	1000 0000 0000 0010
-32767	1000 0000 0000 0001
less than -32767	1000 0000 0000 0000

It is possible to define other units or coding tables, however, how to specify and negotiate this is outside the scope of this document.

## 8. Feedback

Feedback is an important part of any retransmission scheme. To be efficient, also under near real-time conditions, it might be needed to receive this feedback with the lowest delay that is possible. The basic RTP sets constraints about feedback timing rules, which might not be suitable for a retransmission scheme under real-time or near real-time conditions.

To enable low delay feedback a new extended RTP profile for RTCP-based feedback was defined, i.e. RTP/AVPF [4]. This profile redefines RTCP feedback timing rules according to the group size of the multicast session. In unicast sessions it is possible to send feedback virtually immediately. In the RTP/AVPF profile [4], two major changes are introduced: 1) the 5 second rule from RTP [3] is not enforced anymore and 2) the application MAY introduce an application-defined minimum interval between Receiver Reports. See [4] for details.

In [4] RTCP feedback messages are defined, which are suited for retransmission requests and acknowledgements. Generally, a simple feedback message, called Generic NACK is suitable to be used as a retransmission request. Similarly, a Generic ACK is suitable to acknowledge packets, whereas a range of packets or, selectively, a subset from it.

The packet format for a Generic NACK Feedback message is defined in [4] as follows:

Miyazaki, et. al.	Expires December 2002	17
Multiple Selective RTP Retransmissions		June 2002

```

0                               1                               2                               3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|V=2|P|0| FMT=1 |      PT=RTPFB      |      length      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|
|      SSRC of packet sender      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|
|      SSRC of media source      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Therefore, even though retransmission algorithms normally contradict congestion control, they could enhance the quality of service, in

environments where losses occur not only due to congestion but also (or mainly) due to bit errors on the link. Taking this information into account the following rule MUST be followed, if the retransmissions are used in a best effort environment (e.g. the Internet).

At reception of a retransmission request or detecting packet loss by other means, the sender MUST NOT increase the sending bit rate on mid-term timescales. Retransmissions MAY be sent within the current bit rate, e.g. by skipping or delaying the transmission of other scheduled packets.

The rules for congestion control of RTP [3] and RTP/AVPF [4] MUST be followed, where applies.

## **10. Relation to SDP**

The payload formats, described in this document can be indicated using SDP. This section specifies the usage of SDP in relation to the RTX and SEL payload format.

### **10.1 Indicating RTX Usage in SDP**

RTX packets are retransmissions of RTP packets. The original RTP packets get a payload type value assigned, dynamically or statically. There is no static payload type assignment for RTX, so dynamic payload type numbers MUST be used. The binding to the number is indicated by an "rtpmap" attribute. The name used in this binding is "rtx". The payload type number is indicated in the "m" line of the media as an (additional) payload type.

We define an OPTIONAL payload format-specific parameter in SDP for use with "a=fmtp", namely "rtx-time". "rtx-time" is used to indicate the maximum time a server will try to retransmit a lost packet. This is indicated by <rtx-time-value>.

The syntax is as follows:

```
a=fmtp:<number> rtx-time=<rtx-time-val>
```

where,

<number> indicates the assigned payload type number (indicated in the "m" line) to the 'rtx' payload format.

<rtx-time-val> indicates the time in [ms], measured from the first sending of a packet until when the server will not try to retransmit the packet again. This is a maximum time, the client can decrease this (e.g. because of limited memory resources), by not sending retransmission requests earlier.

The "rtx-time" parameter MAY be used only with the retransmission payload format "rtx" specified in this document.

No fmtp attribute line present for a retransmission stream means that the maximum retransmission time is not defined, but MAY be negotiated by other means.

A sample indication of the RTX payload format in SDP is as follows:

```
m = video 8000 RTP/AVPF 98 99
a = rtpmap:98 MP4V-ES/90000
a = rtpmap:99 rtx/90000
```

An additional line

```
a = fmtp:99 rtx-time=3500
```

will indicate that the server will try to retransmit packets only 3.5 seconds, after which it will not try again.

## **10.2 Indicating SEL Usage in SDP**

SEL packets are RTP packets with an additional SEL specific header, which enables a priority detection (also of lost packets). It is expected that the SEL packets are typically sent either instead of all other media packets, or in combination with other media packets. E.g. all media packets might be encapsulated in SEL packets, or only one importance level is encapsulated, where the other packets are sent with their original payload type.

There is no static payload type assignment for SEL, so dynamic payload type numbers MUST be used. The binding to the number is indicated by an rtpmap attribute. The name used in this binding is "sel". The payload type number is indicated in the "m" line of the media as an additional payload type to the original one.

The priority indicating packets, using the SEL payload type, are sent to the same transport address and port as the media stream is sent or would have been sent.

A sample indication of SEL in SDP is as follows:

Miyazaki, et. al.	Expires December 2002	20
	Multiple Selective RTP Retransmissions	June 2002

```
m = video 8000 RTP/AVPF 98 99
a = rtpmap:98 sel/90000
a = rtpmap:99 MP4V-ES/90000
```

### **10.3 Indicating RTX and SEL Usage in SDP**

Both payload formats, SEL and RTX, MAY be used together. The following is an example of how to indicate the usage of both formats:

```
m = video 8000 RTP/AVPF 98 99 100
c = IN IP4 192.129.35.13
a = rtpmap:98 sel/90000
a = rtpmap:99 MP4V-ES/90000
a = rtpmap:100 rtx/90000
```

An additional line

```
a = fntp:100 rtx-time=3500
```

will indicate that the server will try to retransmit packets only 3.5 seconds, after which it will not try again.

## **11. Example Usages of the Payload Formats**

In [section 3](#) we investigated the requirements for a retransmission scheme to be used for RTP. We saw that different applications and environments have different requirements on a retransmission scheme, which are not necessarily compatible. Thus, there is a need for different retransmission schemes, depending on the environment and application that it should be used for.

In this section we show four examples how the two payload formats, RTX and SEL, can be used to build efficient retransmission schemes. Note that it is not necessarily needed that client and server have negotiated the same retransmission scheme. The rules of how to use the payload formats, see [section 5](#) for details, describe the general behavior that the client (server) can expect from the server (client). However certain algorithms and behaviors of clients or servers may enhance the retransmission scheme or make it efficient



in certain environments. These kind of schemes are described below, where no standardized coordinated behavior of client and server is needed, but both enhance the functionality rather independently from each other.

First, we consider a quite simple retransmission algorithm suitable for the most common applications and environments. With this algorithm it is possible to retransmit lost RTP packets, detect the

Miyazaki, et. al.	Expires December 2002	21
Multiple Selective RTP Retransmissions		June 2002

loss of retransmissions efficiently and retransmit lost packets several times, if needed.

A second example gives a sample implementation of a more intelligent client. In this case the client sends, in addition to the Generic NACK messages, periodic Generic ACK messages. This enables the client to influence the retransmission decisions. Also some workload is shifted from server to the client, which makes the scheme scalable.

The third example uses the SEL payload format additionally to signal the need for a retransmission request from the server to the client in-band. This leads to the client being able to judge the priority of lost packets and selectively request retransmissions. Thus, bandwidth on the feedback channel can be saved. The server, in turn, is aware of which packets should be given priority when retransmitting, thus enhancing the transmission quality.

The fourth example shows how retransmissions can be performed for a layered encoded stream. Different layers are sent in different RTP sessions and retransmissions in this case are done per RTP session. In this fashion, the retransmissions fit well into the layered principle of the coding.

### **11.1 A Simple Multiple Retransmission Algorithm**

This section gives an examples how the RTX payload format is used to enable multiple retransmissions in RTP. It is assumed that Generic NACK feedback messages are used. A sample indication in SDP could be:

```
m = video 8000 RTP/AVPF 98 99
a = rtpmap:98 MP4V-ES/90000
a = rtpmap:99 rtx/90000
a = fmtp:99 rtcp-fb nack
```

```
a = fntp:99 rtx-time=3500
```

The sender transmits the media data in normal RTP packets using an RTP payload format for that media (e.g. MPEG-4 as described in [5]). If packets get lost the receiver will detect that by the gap in the sequence number. To signal lost packets the NACK message, described in [section 8](#) is used. The PID field is used to specify a lost packet. The default method, which is to use the RTP sequence number

to indicate the lost packet, is used. The BLP field can be used to indicate several lost packets in one NACK message. The detailed semantics is described in [4].

Example of a retransmission packet:

```

0                                     1                                     2                                     3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+
|V=2|P|X| CC=0 | PT=RTX | RTP sequence number |
+-+-+-+-+
| RTP timestamp = timestamp of original transmission |
+-+-+-+-+
| SSRC of media source |
+-+-+-+-+

```

```

|0| PT=media_pt |0|   DSN=12   |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
:
:                               Media Payload
:
:
:..... . . . . . . . . . . . . . . . . . . . . :

```

Note that the client is able to detect the loss of a retransmission by a gap in the sequence number. For the client this looks just like another lost packet, and thus a retransmission request will be sent. If the server, after receiving the request, decided that the packet is still to be retransmitted another time, an additional RTX header is placed in the header. The client, after receiving the packet, can then detect to which retransmission request(s) the packet belongs.

## 11.2 A Client-oriented Retransmission Algorithm

In the simple retransmission scheme described in the previous section the server judges every requested packet for retransmission before actually sending it. In the case of a large number of clients the processing load at the server might become considerable. This can be avoided by adding a little more intelligence to the client.

A simplified server would retransmit every requested packet without performing any judgment. By doing this and if retransmissions of a packet are lost several times, it might well happen that the packet becomes too old to be used by the client. However, the server keeps on sending it, unaware of the situation.

Thus, it is necessary to eliminate those packets that have been requested for retransmission but would not arrive on time. For this purpose, an intelligent client MAY make additional use of periodic Generic ACK feedback messages. These Generic ACKs, as per [4], would acknowledge cumulatively the last outdated packets and selectively acknowledge those packets following it. These Generic ACKs MAY also contain pseudo-ACKs, that is, bits in the BLP field that acknowledge packets which didn't arrive at the client but the client estimates to be outdated or doesn't wish to receive anymore. In a simple case the client might judge a packet as outdated when the display time is

older than the current time. A more complex judgment might involve an RTT estimation at the client.

A sample indication in SDP would be:

```
m = video 8000 RTP/AVPF 98 99
a = rtpmap:98 MP4V-ES/90000
a = rtpmap:99 rtx/90000
a = rtcp-fb:99 nack
a = rtcp-fb:99 ack
```

We have seen that the retransmission judgment is not performed at the server anymore; the functionality of the server is reduced to resolving which packets are not to be sent anymore as indicated by the Generic ACK messages. This reduces processing load at the server and increases scalability.

Furthermore, this scheme allows the client to dynamically adjust its buffer space available dynamically in-band, without having to previously communicate this to the server by other means, e.g. SDP. If the client decides to increase (decrease) the buffer space, it might do this by just incrementing (decrementing) the spacing between ACK messages. A client might dynamically decide to increase its buffer space if, for example, the packet loss rate or/and other factors, prevent it from receiving an acceptable quality.

Miyazaki, et. al.	Expires December 2002	24
	Multiple Selective RTP Retransmissions	June 2002

### **11.3 Selective Retransmission Algorithm**

A special case, where the above described multiple retransmission algorithm does not work well, is if the feedback channel has a very limited capacity. Possible scenarios are 1) small multicast groups, where according to the defined timing rules for RTCP it is seldom possible to send immediate feedback, 2) scenarios where the feedback channel is mainly used for other purposes, e.g. wireless control channel and 3) asymmetric channels, where the feedback channel is smaller than the forward channel, e.g. satellite links.

In this kind of scenarios it is very important to limit the feedback to the very minimum, which could mean that feedback is only sent in form of retransmission requests for really important data and the feedback on other less important packets is sent as long as there's remaining feedback bandwidth.

Another scenario is that of a client having several applications running on the same terminal, e.g. streaming and down-/uploads. To

manage the different applications, the client may decide to limit the bandwidth reserved for each application. If the application in particular runs over RTP with extended feedback capabilities (as per [4]) the application might set an application-defined maximum RTCP feedback bandwidth. This application-defined maximum RTCP feedback bandwidth should ensure the correct working of the application under the most common network conditions. Nevertheless, this maximum bandwidth may be surpassed with on-demand feedback messages (always remaining under the RTCP bandwidth share, as defined in RTP [3]). On-demand feedback can be sent to report the loss of those packets considered to be important and which need immediate feedback. By doing this, bandwidth on the feedback channel is saved in a way that ensures the correct functioning of all applications and the loss of important packets is reported as soon as allowed by the timing rules defined in RTP/AVPF [4]. Please refer to [4] for details.

We have seen that to solve the above described requirements on the retransmission scheme, it is necessary that the receiver knows the priority of the lost packets to decide if a retransmission request should be sent or not and also, when. In this case, the selective packet format is used.

Hence, in scenarios as described above, where feedback can be sent only very infrequently and/or when the feedback timeliness is critical, a solution which uses the selective payload format might be the only possible solution at all.

The mechanism signaled in SDP, assuming 3.5 seconds buffering delay and NACK-based feedback, is as follows:

Miyazaki, et. al.	Expires December 2002	25
	Multiple Selective RTP Retransmissions	June 2002

```
m = video 8000 RTP/AVPF 98 99 100
a = rtpmap:98 sel/90000
a = rtpmap:99 MP4V-ES/90000
a = rtpmap:100 rtx/90000
a = fmtp:100 rtcp-fb nack
a = fmtp:100 rtx-time=3500
```

The sender may send all (or a part of the) media data in RTP packets using the SEL payload type plus additionally the media payload type. The priority field is set according to the packet's priority. In all SEL packets the Delta SN field is used to signal the delta to the sequence number of the previous high priority packet. The receiver

can use this information to check if the last high priority packet was received correctly or not. If this packet was not received or he cannot reliably detect the priority of lost packets (if many packets were lost in a row), a retransmission request is sent. If the receiver detects that low priority packets were lost it can additionally report this fact as long as the RTCP feedback bandwidth allows.

The receiver can also reconstruct the timestamp of the lost packet if the sender used the Delta Time field. Using this time information it is possible to calculate the time after which the packet cannot be used any more (display time). In this case the receiver may maintain a round trip time (RTT) measurement and compare the calculated display time with the estimated time when the retransmission could arrive. If the retransmission would arrive already outdated, a retransmission request should not be sent. At receiving a retransmission request from the receiver, the server may make the same decision as described in the previous section, i.e. it judges according to the packets priority (which is most probably quite high, because a retransmission request for it was sent), network status etc. if the packet should be retransmitted. If so, it uses the retransmission packet type, to enable multiple retransmissions as described in the previous section. The SEL payload type is additionally used to enable the priority detection at the client.

If after an estimated time the requested high priority packet has not arrived at the receiver, the receiver may conclude that the retransmission request was lost and issue a new one, if the timing allows.

#### **11.4 Using Retransmissions in Layered Transmissions**

As described above in [section 3](#), layered encoding and transmission is a well accepted mechanism to perform rate control, especially in multicast. In this section we will show a sample mechanism how the retransmission framework interacts with layered transmissions. Note

Miyazaki, et. al.	Expires December 2002	26
	Multiple Selective RTP Retransmissions	June 2002

that retransmissions in multicast is still an open research topic. We do not intend to solve this and/or specify a retransmission scheme for multicast. However, we specify how the retransmission framework can support a layered retransmission scheme syntactically. The loss detection and decisions when to retransmit a packet to which receiver group is clearly outside the scope of this document.

As packets of different priorities or importance are sent in different RTP streams, the retransmissions should be sent in the corresponding layer. The rationale for this is that the retransmission will probably have the same or even greater importance over time.

An SDP indication would be:

```
m = video 8000 RTP/AVPF 98 99
a = rtpmap:98 MP4V-ES/90000
a = rtpmap:99 rtx/90000
a = fmp:99 rtcp-fb nack
a = fmp:99 rtx-time=3500
c = IN IP4 224.2.1.1/127/3
```

for multicast or

```
m = video 8000/3 RTP/AVPF 98 99
a = rtpmap:98 MP4V-ES/90000
a = rtpmap:99 rtx/90000
a = fmp:99 rtcp-fb nack
a = fmp:99 rtx-time=3500
```

for unicast.

In both cases NACK-based feedback and 3.5 seconds buffering delay is assumed.

Thus, three addresses/ports are allocated, each transmitting the original data of the corresponding layer. After a loss detection the receiver(s) signal the lost packet to the server. If the simple retransmission scheme is used, the server has to make a retransmission decision, which can be based, beside the arguments given in the previous examples, on the layer, where the packet was lost. In case of a positive retransmission decision, the server uses the RTX payload format to retransmit the packet on its corresponding layer and within the corresponding sequence number space. With this it is immediately possible for the receivers to detect a lost retransmission.

RTP packets transporting information with the proposed payload formats are subject to the security considerations discussed in RTP specification [3], in the RTP/AVPF profile specification [4] and in the RTP/AVP profile specification [6].

This profile does not specify any additional security services.

## Acknowledgements

For the work on this retransmission framework, we received a lot of help and advice from the AVT working group and the chairs, Colin Perkins and Stephen Casner. Especially the mechanism of sending retransmissions with a new sequence number in the same sequence number space, using a different payload type was initiated by Colin Perkins.

Furthermore, we would like to thank Norihito Takatori, Seiji Okumura and Tsugihiko Ohno from the Mitsubishi Electric Corporation, in Kanagawa (Japan) for the constant support during the generation of this document.

## Intellectual property considerations

Matsushita has filed patent applications that might possibly have technical relation to this contribution.

If part(s) of the contribution by Matsushita employee is (are) included in a standard and Matsushita has patents and/or patent application(s) that are essential to implementation of such included part(s) in said standard, Matsushita is prepared to grant - on the basis of reciprocity (grantback) - a license on such included part(s) on reasonable, non-discriminatory terms and conditions (in according with paragraph 10.3.3 of the [RFC 2026](#)).

## References

- 1 Bradner, S., "The Internet Standards Process -- Revision 3", [BCP 9](#), [RFC 2026](#), October 1996.
- 2 Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997
- 3 Schulzrinne, H., Casner, S., Frederick, R. and V. Jacobson, "RTP: A Transport Protocol for real-time applications", [RFC 1889](#), January 1996.
- 4 Ott, J. et al., "Extended RTP Profile for RTCP-based Feedback", IETF Internet Draft, [draft-ietf-avt-rtcp-feedback-03](#),



work in progress, June 2002.

- 5 Kikuchi, et al., "RTP Payload Format for MPEG-4 Audio/Visual Streams", [RFC 3016](#), November 2000.
- 6 H. Schulzrinne and S. Casner, "RTP Profile for Audio and Video Conferences with Minimal Control,"  
Internet Draft [draft-ietf-avt-profile-new-12.txt](#), November 2001.

#### Author's Addresses

Akihiro Miyazaki  
Matsushita Electric Industrial Co., Ltd  
1006, Kadoma, Kadoma City, Osaka, Japan  
Mail [akihiro@isl.mei.co.jp](mailto:akihiro@isl.mei.co.jp)

Hideaki Fukushima  
Matsushita Electric Industrial Co., Ltd  
1006, Kadoma, Kadoma City, Osaka, Japan  
Mail [fukushima@isl.mei.co.jp](mailto:fukushima@isl.mei.co.jp)

Koichi Hata  
Matsushita Electric Industrial Co., Ltd  
1006, Kadoma, Kadoma City, Osaka, Japan  
Mail [hata@isl.mei.co.jp](mailto:hata@isl.mei.co.jp)

Thomas Wiebke  
Panasonic European Laboratories GmbH  
Monzastr. 4c, 63225 Langen, Germany  
Mail [wiebke@panasonic.de](mailto:wiebke@panasonic.de)

Rolf Hakenberg  
Panasonic European Laboratories GmbH  
Monzastr. 4c, 63225 Langen, Germany  
Mail [hakenberg@panasonic.de](mailto:hakenberg@panasonic.de)

Carsten Burmeister  
Panasonic European Laboratories GmbH  
Monzastr. 4c, 63225 Langen, Germany  
Mail [burmeister@panasonic.de](mailto:burmeister@panasonic.de)

Miyazaki, et. al.

Expires December 2002

29