

Internet Draft
Document:
[draft-ietf-avt-rtp-retransmission-02.txt](#)
Expires: December 2002

David Leon
Viktor Varso
Nokia
June 2002

RTP retransmission framework

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

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

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

Main changes

*since [draft-ietf-avt-rtp-retransmission-01.txt](#)

IANA considerations section added

New appendix on RTP retransmission and multiplexing. It results from a discussion on mailing list.

*since [draft-ietf-avt-rtp-retransmission-00.txt](#):

An applicability statement was added.
The security considerations section was expanded.

Leon, Varsa IETF draft - Expires September 2002
 RTP retransmission framework

1
June 2002

*since [draft-leon-rtp-retransmission-02.txt](#):

The previous version of the draft described the use of the redundancy payload format ([RFC 2198](#)) in order to send retransmission data and original data in the same stream. At IETF #53, it was concluded that [RFC 2198](#) was not intended to such a use. Piggybacking retransmitted packets was thus removed from the draft.

Table of Contents

Abstract.....	1
Main changes.....	1
1 . Introduction.....	2
2 . Terminology.....	3
3 . Applicability statement.....	3
4 . Retransmission framework basic principles.....	4
5 . Retransmission payload format.....	5
6 . Use with the Extended RTP profile for RTCP-based feedback.....	6
7 . Congestion control.....	8
8 . Example scenario of unicast streaming.....	9
9 . SDP usage.....	9
10 . IANA considerations.....	9
11 . Security consideration.....	11
Appendix A : Retransmission and SSRC multiplexing.....	12
Appendix B : FEC for retransmission.....	13
References.....	14
Author's Addresses.....	15

[1](#). Introduction

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

When choosing a technique for a particular system, the tolerable latency of the application has to be taken into account. In the case of multimedia conferencing, the end-to-end delay has to be at most a

few hundred milliseconds in order to guarantee interactivity, which usually excludes the use of retransmission. On the other hand, in the case of multimedia streaming, the user can tolerate an initial latency as part of the session setup and thus an end-to-end delay of several seconds is possible. Retransmission may thus be considered for such applications.

Leon, Varsa

- Expires December 2002
RTP retransmission framework

2
June 2002

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

2. Terminology

The following terms are used in this document:

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

Original stream: refers to the stream of original packets.

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

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

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

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC-2119](#) [4].

3. Applicability statement

There are two proposals for RTP retransmissions (this proposal [draft-ietf-avt-rtp-retransmission-01.txt](#) and [draft-ietf-avt-rtp-selret-05.txt](#)). Both proposals may be optimum under a different set of constraints.

This draft enables the receiver to perform reliable loss detection of user data, i.e. the receiver can differentiate between lost packets sent for first time and lost retransmissions.

Reliable user data loss detection is required for example in RTP conversational text ([RFC 2793](#)) in order to indicate missing text to the user. For some applications, reliable loss detection of user data may not be strictly required but may enhance a receiver performance.

Leon, Varsa

- Expires December 2002
RTP retransmission framework

3
June 2002

This retransmission algorithm allows receivers to trade-off the playout delay versus the number of retransmissions for a given packet. This delay does not need to be signalled to the sender and can be changed dynamically during the session in order to adapt to varying network conditions. Receivers should choose whether to request a missing packet based on an estimation of its timestamp which is usually obtained from the observed correlation between the RTP sequence number and timestamp. Implementers should carefully design their decision retransmission request algorithm in order to limit the risk of unnecessary retransmission.

This retransmission scheme requires a separate RTP session to send retransmitted packets. As a consequence, two additional ports are needed: one port for the RTP retransmission stream and one port for the associated RTCP. While this is generally not a problem, the implementers should assess the implications in the targeted environment.

This scheme may be used in a multicast RTP session in order to perform unicast retransmission to each participant.

If a separate session is used, mixers, translators and packet caches may be able to separate retransmission packets from original packets at an RTP session level based only on the port being used and process them differently if necessary.

4. Retransmission framework basic principles

Retransmission packets MUST NOT share the RTP Sequence Number (SN) space with the original stream. The retransmission stream SHOULD use a different RTP session (as defined in RTP [5]) from that of the original stream. Since a separate session is used, the original and

retransmission streams are sent to different multicast group/unicast addresses and/or port numbers.

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

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

RTP Timestamp (TS) estimation of missing original packets is necessary at the receiver in order to decide whether a retransmission is useful or not. A retransmission is useful if the retransmission packet sent as a response to the retransmission request may still be used when it arrives at the receiver. The missing original packet timestamp can be estimated from timestamp of

Leon, Varsa	- Expires December 2002	4
	RTP retransmission framework	June 2002

packets preceding and/or following the sequence number gap caused by the missing packet in the original stream.

The fact that RTP streams for original and retransmission packets do not share the same SN space guarantees that the RTP timestamp estimation method is reliable. Reliability would be sacrificed if original and retransmission packets were sent in the same RTP stream as the timestamp estimate for a lost retransmission packet would then be incorrect. This is because the retransmission packet usually has a smaller "out-of-order" timestamp than the timestamp of the consecutive original packets.

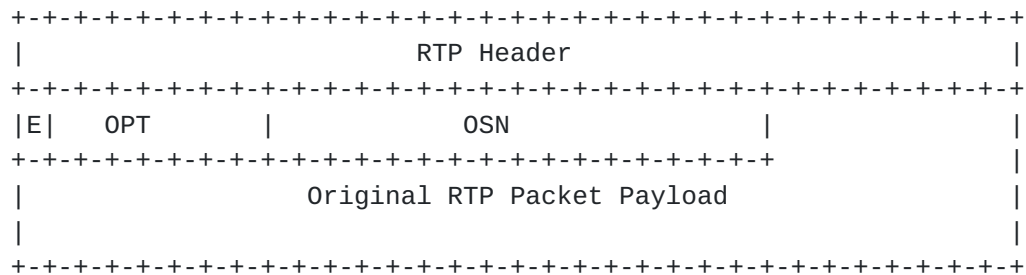
When the retransmission stream is sent to a multicast RTP session, receivers may choose whether to subscribe or not to the RTP session carrying the retransmission stream. Therefore, a multicast streaming application can use retransmission and still be backwards compatible as receivers which do not implement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

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

As a consequence of having separate RTP sessions for original and retransmission streams, there are also separate RTCP streams and statistics for these two sessions. There is thus no corruption of the original stream RTCP statistics. The RTP sender is able to know the packet loss and jitter of the original stream. It can thus estimate what the quality of the received signal would be without the use of retransmission.

5. Retransmission payload format

The payload format of a retransmission packet is shown below.



Leon, Varsa

- Expires December 2002
RTP retransmission framework

5
June 2002

RTP header usage:

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

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

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

6.1 Sending rules for RTCP-based feedback

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

The NACK message format defined in the profile should be used. Early RTCP packets should not be used and the NACK message should always be appended to a regularly scheduled compound RTCP packet that is sent at every RTCP report interval.

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

6.2 Receiver algorithm for generating retransmission requests

This section gives some general guidelines on how a receiver should decide whether or not to request a packet retransmission. An actual receiver implementation should take into account such factors as the network environment and the media type.

Leon, Varsa

- Expires December 2002
RTP retransmission framework

6
June 2002

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

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

It can be seen that the needed receiver buffering delay is dependent on the amount of RTCP traffic allowed in the session. It is illustrated in [Section 8](#) that moderate RTCP feedback traffic is enough to perform retransmission with reasonable receiver buffering delay.

A receiver should maintain a list of missing original packet sequence numbers. A receiver needs also to store for each missing original packet an estimated RTP timestamp as described in [Section 4](#). At the next scheduled RTCP packet sending time, the receiver estimates which of the missing packets should be requested in the NACK message (see usage of the PID and BLP fields of the NACK message format in [4]) of the RTCP compound packet. A missing packet should be requested if it is estimated the associated retransmission packet could still be used at the time it arrives at the receiver. The receiver should remove from its list of missing packets, the packets which were deemed too old to be requested.

If the retransmission stream is sent to a multicast session, the receiver should listen to NACK messages from other receivers. If a NACK message for the sequence number of a missing packet has been sent by another receiver, the receiver should ignore that sequence number in its list of missing packets and refrain from sending a retransmission request for that sequence number.

The same retransmission request may be resent in the original RTP session if the requested packet was not received after an estimated retransmission reception time. This increases the robustness to the loss of a NACK message or of a retransmission packet. The number of retransmission requests that may be sent for a given missing original packet sequence number depends on the receiver buffering delay.

The receiver should upon reception of a retransmission packet remove the corresponding original packet sequence number(OSN in the retransmission payload format) from the list of missing sequence numbers.

[6.3](#) RTCP sending rules in the retransmission RTP session

Leon, Varsa

- Expires December 2002
RTP retransmission framework

7
June 2002

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

the retransmission RTP session would be 0.1 kbps.

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

7. Congestion control

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

The RTP profile under which the retransmission scheme is used defines an appropriate congestion control mechanism in different environments. Following the rules under the profile, an RTP application can determine its acceptable bitrate and packet rate in order to be fair to other applications such as TCP flows and other RTP flows.

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

If enhanced service is used, it should 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 retransmit only the packets that it deems important and ignore NACK messages for other packets in order to limit the bitrate

These congestion control mechanisms should keep the congestion level under an acceptable limit. This would in turn guarantee that the packet loss should be moderate. Too high a packet loss would mean

Leon, Varša

- Expires December 2002
RTP retransmission framework

8
June 2002

that congestion is not kept under control and retransmission should then not be used.

8. Example scenario of unicast streaming

Let us assume that the RTP session bandwidth is 64 kbps and the receiver buffering delay is 3 seconds. The network round trip time is 500 ms and Receiver RTCP packets (including NACK messages) are sent at 2-second intervals. With these time limitations, when the receiver algorithm for generating retransmission requests described in [Section 6.2](#) is followed, only one request per packet loss can be sent. Let us assume an original packet rate of 50 packets per second (1 packet every 20 ms). At this packet rate, with the packet loss distribution worst case scenario where every 17th original packet is lost (i.e. 5.88% uniform packet loss), a maximum of 6 NACK messages need to be appended to the RTCP compound packet that is sent every 2 seconds. The 6 NACK messages can report any packet losses in an SN range of $6 \times 17 = 102$, thus the number of required NACK messages would not increase with higher packet loss rates. The overhead required per RTCP compound packet for the 6 NACK messages is 36 bytes which carries 6 PID and 6 BLP fields in addition to the feedback message headers.

The total compound RTCP packets size is thus: IPv4 (20) + UDP (8) + RR(header 8 + report 24) + CNAME(12) + NACK (36) = 108 bytes. The needed receiver RTCP bandwidth would then be 0.432 kbps. This RTCP bandwidth is well below the recommended RTCP bandwidth. The receiver RTCP recommended bandwidth in an RTP session with 64 kbps is about $0.25 \times 64 = 1.6$ kbps.

9. SDP usage

The binding to the payload type number is indicated by an rtpmap attribute. The name used in the binding is "rtx".

An SDP example is shown below:

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

10. IANA considerations

10.1 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.

Optional parameters: none

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

Security considerations: see [Section 11](#)

Interoperability considerations: none

Published specification: RFC xxx

Applications which use this media type: multimedia streaming applications

Additional information: none

Person & email address to contact for further information:
david.leon@nokia.com

Intended usage: COMMON

Author/Change controller: David Leon

[10.2](#) 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.

Optional parameters: none

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

Security considerations: see [Section 11](#)

Interoperability considerations: none

Published specification: RFC xxx

Applications which use this media type: multimedia streaming applications

Additional information: none

Leon, Varsa	- Expires December 2002	10
	RTP retransmission framework	June 2002

Person & email address to contact for further information:
David.leon@nokia.com

Intended usage: COMMON

Author/Change controller: David Leon

10.3 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.

Optional parameters: none

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

Security considerations: see [Section 11](#)

Interoperability considerations: none

Published specification: RFC xxx

Applications which use this media type: multimedia streaming applications

Additional information: none

Person & email address to contact for further information:
David.leon@nokia.com

Intended usage: COMMON

Author/Change controller: David Leon

11. Security consideration

Retransmission and FEC [7] have similar security conditions as far as encryption and congestion control are concerned.

Applications utilizing encryption SHOULD encrypt both the original and 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.

Congestion control considerations with the use of retransmission are dealt with in [Section 7](#) of this document.

Leon, Varsa

- Expires December 2002
RTP retransmission framework

11
June 2002

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

Appendix A: Retransmission and SSRC multiplexing

In this document, it is required that two separate RTP streams with their own sequence number space be used for the original stream and the retransmission stream. This should be achieved by sending the RTP stream and its associated retransmission stream to different ports.

Another way the original and the retransmission stream could be multiplexed is through the use of different SSRCs if these streams are sent to the same port.

In general, SSRC multiplexing is not feasible as in a multicast group it would not be possible to associate an original stream and its retransmission stream since the SSRC is a random number chosen by the RTP sender.

However, in unicast this should not be an issue if exactly one RTP stream and its retransmission stream are multiplexed based on their SSRC. Since the receiver knows the payload types used by the original stream and the retransmission stream, it would be able to derive which SSRC maps to the original data and which SSRC maps to retransmissions.

SSRC multiplexing should generally be avoided as discussed in [Section 5.2](#) of RTP [5]. One of the advantage of multiplexing at the transport layer (i.e. based on the IP address or port number) is the use of different network paths or resources for different streams. However, in the case of unicast retransmission, it is the same media sent in both the original and the retransmission.

It also has to be noted that the SSRC-multiplexing approach is allowed in FEC [7]. [Section 3](#) of FEC states "This document does not prescribe the definition of "separate streams", but leaves this to applications and higher level protocols to define. For multicast, the separate stream may be implemented by separate multicast groups, different ports in the same group, or by a different SSRC within the

same group/port. For unicast, different ports or different SSRC may be used. Each of these approaches has drawbacks and benefits which depend on the application".

This retransmission draft recommends the following rules:

Separate addresses and/or ports must be used in the multicast case and should be used in the unicast case to multiplex the original stream and the retransmission stream. In the unicast case, exactly one RTP stream and its associated retransmission stream may be sent to the same port and multiplexed based on their SSRCs

The motivation not to prevent SSRC multiplexing is to minimise the number of ports usage when it may be a performance issue for high-scale RTP servers and/or middle-ware proxies while allowing the

Leon, Varsa	- Expires December 2002	12
	RTP retransmission framework	June 2002

original and the retransmitted data to be sent into separate RTP streams with their own sequence number spaces.

Appendix B: FEC for retransmission

It is possible to retransmit the payload of an original packet by sending a FEC packet as defined in [7] instead of using the retransmission payload format. The base sequence number of the FEC header is the sequence number of the original packet that carried the same payload and the mask is set to zero. There are some advantages in using the FEC payload format in particular in multicast sessions as a single FEC packet may repair the loss of different packets at different receivers. In a multicast session, FEC may be combined with retransmission requests as described in [8] in order to achieve scalable reliable multicast transmission.

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

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

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

*Assuming that no retransmission payload format were defined, the

FEC payload format would then be used in different ways: sending computed FEC packets proactively for correction of lost packets at the receiver without requiring NACK messages (referred hereafter as proactive FEC) or sending retransmitted data upon receipt of a NACK message from the receiver.

Although they could use the same payload format, these two repair mechanisms are different. For a given amount of overhead, they offer a different residual packet loss vs. latency trade-off.

Retransmission provides higher packet loss recovery at the expense of higher delay. If a sender uses proactive FEC and none of the packets protected by a given FEC packet is lost, there is then no use of the FEC packet at the receiver. On the other hand, data which are retransmitted are known to have been lost.

Therefore, a receiver may wish to use only retransmission and not receive any proactive FEC from the sender in order to trade-off itself the buffering delay, the data-loss rate and the overhead. There is thus a motivation to let the receiver decide at session setup whether the sender may send proactive FEC or retransmission only.

Leon, Varsha

- Expires December 2002
RTP retransmission framework

13
June 2002

If the same payload format were used for these different purposes, the receiver would not know at session establishment which repair mechanism is used by the sender (without defining out-of-band signalling). The sender would decide by itself whether FEC or retransmission (or both) is used and the receiver would not know until receiving the FEC packets whether proactive FEC or retransmission is used.

On the other hand, having different payload formats or at least different out of band signalling for these two repair mechanisms (e.g. through defining the binding name 'RTX' SDP rtpmap attribute) would allow the receiver to choose which mechanism to use (among those supported by the sender). The receiver would then have an explicit way to tell the sender not to send proactive FEC.

Furthermore, if the receiver did not know at session establishment whether it will receive FEC packets or retransmission, it would have to be prepared to receive FEC and thus be able to perform FEC packet recovery operations. Implementers would thus not be able choose to implement only a FEC or retransmission scheme.

References

- 1 Perkins, C., Hodson, O., "Options for Repair of Streaming Media",

[RFC 2354](#), June 1998.

- 2 Schulzrinne, H and Casner, S. " RTP Profile for Audio and VideoConferences with Minimal Control," Internet Draft [draft-ietf-avt-profile-new-12.txt](#), November 2001.
- 3 J Ott, S Wenger, S Fukunaga, N Sato, K Yano, M Akihiro, H Koichi, R Hakenberg, C. Burmeister "Extended RTP profile for RTCP-based feedback", November 2001
- 4 [RFC 2119](#) S Bradner ., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997
- 5 Schulzrinne, H., Casner, S., Frederick, R. and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", Internet Draft [draft-ietf-avt-rtp-new-11.txt](#), November 2001.
- 6 Hellstrom, J, "RTP for conversational text", [RFC 2793](#), May 2000
- 7 Rosenberg, J. and Schulzrinne, H., " An RTP Payload Format for Generic Forward Error Correction", [RFC 2733](#), December 1999.
- 8 Nonnenmacher, Biersack, E, Towsley, D,. "Parity-based loss recovery for reliable multicast transmission", ACM SIGCOMM'97, Cannes, France, September 1997

Leon, Varsa	- Expires December 2002	14
	RTP retransmission framework	June 2002

Author's Addresses

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

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

Leon, Varsa

- Expires December 2002

15