

Internet Engineering Task Force
Internet Draft
[draft-ietf-avt-rtp-rx-00.txt](#)
July 14, 2000
Expires: January 14, 2001

AVT WG
Koichi Yano
Matthew Podolsky
Steven McCanne

**RTP Profile for RTCP-based Retransmission Request
for Unicast session.**

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 specifies a new RTP profile for retransmission of lost packets of unicast multimedia sessions. We refer to this profile as "RTP/AVP-RX". This profile follows [RFC 1889](#) as it is for data exchange. Specifically for unicast session, it changes the RTCP interval and defines a new RTCP packet type for retransmission requests. This document also describes how retransmission requests and retransmission data may be sent within RTP.

1. Conventions and Acronyms

The keywords MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, MAY, and OPTIONAL, when they appear in this document, are to be interpreted as described in [\[2\]](#).

2. Introduction

RTP was designed as a flexible protocol capable of transporting real-time data over multicast or unicast. It has been widely deployed and used extensively for transmitting real-time (or "near" real-time) multimedia signals, commonly called media streams. And although considerable effort went into the design of RTP so that it could support multi-party interactive conferencing over multicast, a common use of it today is for one-way non-interactive transmission of media streams. Because the communication is non-interactive, extra playback buffering and delay may be tolerable, which in turn enables receivers to request and receive retransmissions of lost RTP packets before their scheduled play-out times. For example, popular commercial products like RealNetworks' G2 Player and Microsoft's NetShow use RTP as the underlying transport protocol for their media streams, and use retransmissions in unicast delivery sessions.

However, as noted in [\[4\]](#), no standard exists for requesting the retransmission of streaming media. As a result, various independent and incompatible research and commercial schemes have been developed for retransmission of RTP streams [\[3, 5\]](#). This purpose of this note is to define a standard retransmission request framework for use with unicast RTP streams. Additionally, potential deployment of RTCP packets for congestion control is described.

3. RTCP Packet Forms and Protocol Behavior

The section "RTP Profiles and Payload Format Specification" of [RFC 1889](#) enumerates a number of items that can be specified or modified in a profile. This new profile is referred to RTP/AVP-RX or AVP-RX in this note. The profile, RTP/AVP-RX, follows all of the default and/or recommended aspects of the Audio and Video Profile, RTP/AVP[6], except for the items which are described below.

RTCP packet types:

A new RTCP packet type, RTCP_NACK, is defined for requesting a retransmission. The packet format is specified in [section 4.2](#).

RTCP report interval:

In order to allow immediate RTCP response to packet loss, the RTCP report interval restriction in [RFC 1889](#) SHOULD be relaxed. However, in order to prevent RTCP packets from flooding the network, the bandwidth constraint of [RFC 1889](#) should be preserved. Recommendations about the RTCP transmission frequency are specified in [section 4.1](#).

This profile SHOULD be referred to as "RTP/AVP-RX" by applications such as session directories. A payload type which is transmitted on

"RTP/AVP-RX" MAY be designated through high level control protocols such as SDP[8], as described in Audio/Video Profile [6] or a succeeding draft. A payload type which is defined for Audio/Video profile may be used as it is by deploying a default NACK format shown in [section 4.3](#). The following is an example of SDP media description:

```
m=video 49170 RTP/AVP-RX 31
```

In this example, the media format 31 indicates that video uses H.261. The payload format in [RFC2032](#) [9] is used for RTP data transmission and NACK packets are sent with the packet format described in [section 4.3](#). RTP/AVP-RX uses the same RTCP port for NACKs as for original RTCP packets, so a separate connection does not need to be specified. A new payload type may be defined in a succeeding draft specifically for this profile. For any such payload type, a dynamic payload type number MUST be used.

The decision of whether and when to send retransmission packets is left to a sender application and is out of the scope of this document. If a sender decides to retransmit packets, the sender SHOULD send them as soon as possible after the reception of the NACK packet requesting the retransmission. In addition, the total transmission rate SHOULD be regulated within a session bandwidth limit that includes the retransmitted packets. The deployment of congestion control is RECOMMENDED as described in [section 5](#).

4. Control packets for unicast session

This note proposes extending the RTCP specification to accommodate retransmission requests. We define a new RTCP type, RTCP_NACK, for notifying the sender of lost packets. RTCP is a reasonable place for inserting these requests because it is a well-established control protocol and no extra connections need to be established. However, the RTCP interval specified in [1] is too large for timely retransmission requests. Thus, the current RTCP minimum transmission interval should be relaxed.

[4.1. RTCP Transmission Bandwidth](#)

The RTCP transmission interval specified in [1] is not frequent enough to allow efficient control of a unicast session needing to request retransmissions of lost packets. [RFC 1889](#) recommends a minimum interval of 5 seconds between control packets. In order to make RTCP applicable for unicast retransmission, a packet-granularity RTCP interval is necessary.

The minimum interval restriction SHOULD be removed for unicast RTP

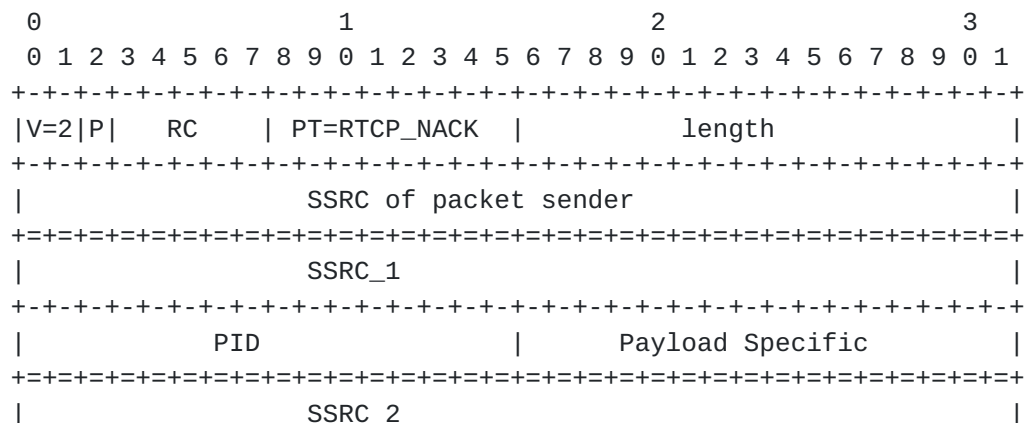
sessions so that retransmission requests are allowed to be sent immediately after loss detection. In order to prevent RTCP packets from causing network congestion, it is RECOMMENDED that the average bandwidth allocated to RTCP be fixed at 5% of the overall session bandwidth. The recommended default sender-receiver allocation values are 1/4 of the RTCP bandwidth (1.25%) for the data sender and the remainder of the bandwidth (3.75%) for the receiver, following the conventions of [RFC 1889](#).

If the size of RTP payload packets is 1 KBytes and the size of RTCP packets is 40 bytes, the 3.75% limitation is large enough for a receiver to send an RTCP packet at every other data packet interval. The interval is frequent enough to allow timely retransmission requests. In the case of consecutive lost packets, a NACK packet can aggregate loss information of up to 16 packets when the default NACK format is used.

In order to regulate the sending rate of RTCP packets within the RTCP bandwidth, the deployment of a traffic shaping scheme such as a token bucket is RECOMMENDED. The token rate SHOULD be set at $(rtcp_bw / rtcp_size)$, where $rtcp_bw$ is the RTCP bandwidth and $rtcp_size$ is the size of RTCP packets. The bucket size SHOULD be determined according to the playback buffer size. As long as a token is in the bucket, the RTCP packet is sent immediately, and the long-term RTCP rate is regulated within the RTCP bandwidth limit.

[4.2. Generic Packet format for requesting retransmission](#)

This section defines a generic packet format for a negative acknowledgment (NACK) packet, which is sent from the receiver to the sender and which notifies the packet loss of one or more RTP packets. This generic packet format provides a way of identifying a packet and a 16 bit field for payload specific use. When a payload format does not specify the use of the field, the default format in 4.3 MUST be used. The generic format of NACK packets is as follows:




```

+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|           PID           |           Payload Specific           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                           |

```

The various fields V, P, RC, SSRC and length are defined in the RTP specification [1]. We summarize the meaning of all of the fields below:

version (V): 2 bits

This field identifies the RTP version. The current version is 2.

padding (P): 1 bit

If set, the padding bit indicates that the RTCP NACK packet contains additional padding octets at the end which are not part of the retransmission control information but are included in the length field.

report count (RC): 5 bits

The number of NACK blocks for different SSRC.

packet type (PT): 8 bits

This is the RTCP packet type which identifies the packet as being an RTCP NACK.

length: 16 bits

The length of this RTCP NACK packet in 32-bit words minus one, including the header and any padding. This conforms with the definition of the length field used in RTCP sender and receiver reports [1].

SSRC: 16 bits

The synchronization source identifier for the originator of this NACK packet.

SSRC_n: 16 bits

The SSRC identifier of the source to which the information in this NACK report block pertains.

Packet ID (PID): 16 bits

The PID field is used to specify a lost packet. Payload definition can decide how to identify a packet. Typically RTP sequence number is used for PID as the default format.

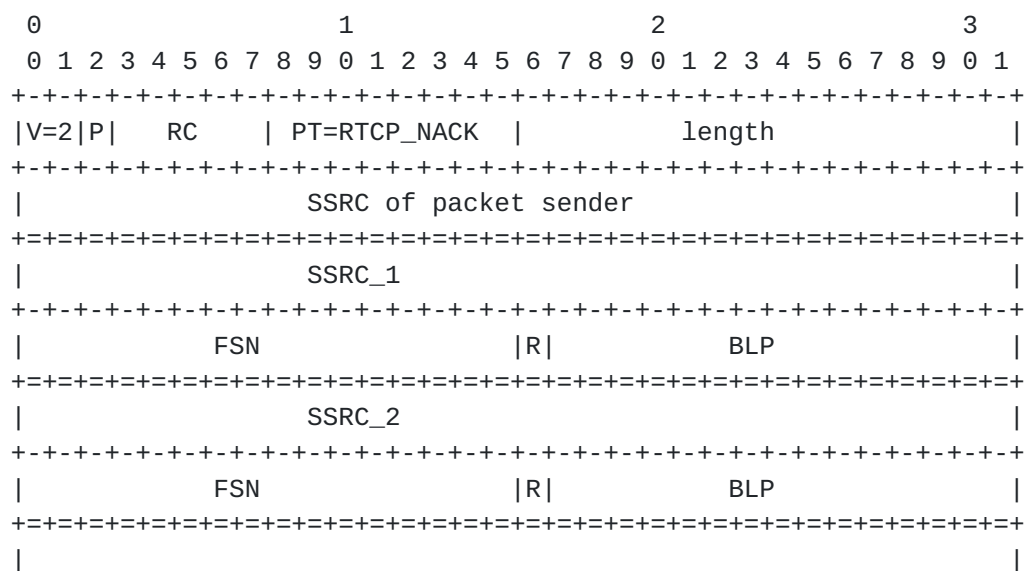
Payload Specific: 16 bits

This 16 bit field can be used for payload specific purposes. If a payload type does not specify the use of this field, the default packet format in 4.3 must be used.

Note that the receiver may detect RTP packet loss via sequence number gaps, timer expirations, or other methods; however, details of the loss detection scheme are outside of the scope of this document.

4.3. Default NACK format

This section shows the default use of the packet id field and the payload specific field. RTP/AVP-RX profile MAY be deployed for transmitting a payload which is defined for RTP/AVP profile. Unless any specific packet format for AVP-RX profile is defined in a payload type, this default packet format MUST be used for NACKs.



The FSN and BLP fields have similar definitions to those given for the RTCP_NACK packet specified in [3]. The R bit is used to notify the sender whether lost packets need to be retransmitted. NACK packets SHOULD be sent for all lost packets, even packets the receiver may no longer need to be retransmitted, in order to allow effective operation of congestion control. The receiver can notify the sender whether the packet needs retransmission through the R field in a NACK packet. The meanings of these fields are as follows:

first sequence number (FSN): 16 bits

The FSN corresponds to an RTP sequence number of a media stream. A NACK packet with a FSN field set to N is to be interpreted as a signal that the receiver has detected loss of RTP data packet N.

requirement of retransmission (R): 1 bit

This field indicates whether the receiver is requesting retransmission of the lost packets designated by the FSN and BLP fields. If this bit is set to 1, the lost packets of this NACK block are requested to be retransmitted. If the bit is set to 0,

the lost packets are not to be retransmitted, and this NACK packet was sent only for the purpose of reporting packet loss.

bitmask of following lost packets (BLP): 15 bits

The BLP allows for reporting losses of any of the 15 RTP packets immediately following the RTP packet indicated by the FSN. The BLP's definition is identical to that given in [3]. Denoting the BLP's least significant bit as bit 1, and its most significant bit as bit 15, then bit i of the bitmask is set to 1 if the sender has not received RTP packet number $\text{FSN}+i$ (modulo 2^{16}) and the receiver decides this packet is lost; bit i is set to 0 otherwise. Note that the sender MUST NOT assume that a receiver has received a packet because its bitmask was set to 0. For example, the least significant bit of the BLP would be set to 1 if the packet corresponding to the FSN and the following packet have been lost. However, the sender cannot infer that packets $\text{FSN}+2$ through $\text{FSN}+15$ have been received simply because bits 2 through 15 of the BLP are 0; all the sender knows is that the receiver has not reported them as lost at this time.

4.4. Sender and Receiver Report

The original RTCP packets (SR/RR) SHOULD also be sent based on [RFC 1889](#). Loss fraction and other fields in SR and RR SHOULD be generated based on original data transmissions and without taking into account requested retransmissions that did not arrive. The reason is that although the loss fraction in RR should ideally account for loss of any data in the forward transmission path, it is difficult to identify the reason of an undelivered retransmission packet. There are three possible reasons a requested retransmission may be missing: the retransmitted packet was lost in the forward transmission path, the retransmission request was lost in the reverse transmission path, or the sender chose not to respond to the retransmission request.

The Receiver Reports SHOULD be issued more frequently than specified in [RFC 1889](#) for use in congestion control ([Section 5](#)). A receiver SHOULD send a RR within the RTT intervals and can estimate the RTT through the elapsed time between transmission of a NACK packet and reception of the corresponding retransmission packet.

5. Congestion Control

Fair behavior against other types of traffic (mainly TCP) has come to be a significant issue for the stability of the Internet. The need to employ congestion control to make RTP streams share bandwidth fairly with other streams has become more important now than ever before. The deployment of congestion control is RECOMMENDED for a

stream following the AVP-RX profile. In conjunction with the frequent RTCP interval formulated in this proposal, NACKs and RRs can provide the necessary information for effective congestion control. This section describes the applicability of the RTCP (RR and NACK) information for congestion control.

For many congestion control schemes[10,11] to operate, the loss rate and RTT is required. Because RRs include the loss fraction, and because the RTT can be calculated by the sender based on the RRs as described in [RFC1889\[1\]](#), if RRs are sent frequently enough ([section 4.3](#)) then a sender can execute a congestion control scheme such as a TCP fair equation-based scheme[10].

A congestion control scheme, TFRC[11], recommends the use of the loss interval rate instead of the loss rate. By keeping track of sequence number of lost packets provided by NACK packets, the sender can calculate the loss interval and execute the congestion control described in [\[11\]](#). The RTT is calculated based on RR reception timing in this case as well.

[6. Examples of retransmission decisions](#)

Retransmitted packets which arrive after their scheduled playout time waste network resources even when successfully received. This section gives examples of how the decision to retransmit a packet may be made and how the R bit in the NACK packets is used.

[6.1 A simple retransmission request system](#)

The simplest way to realize loss recovery using retransmissions is to let the receiver decide whether a NACK packet with R bit set should be sent and have the sender respond to all retransmission requests with retransmitted packets. In this case the receiver sends NACKs with the R bit set whenever it determines that one or more data packets have been lost and that their playback times have not yet passed. If the receiver detects lost packets after the playback time passed, the R bit in NACK packets is set to 0.

This scheme can be realized without an extra control packet between the sender and the receiver. However it could result in many late packets that arrive after play out time because it does not account for the delay between the time a receiver transmits a NACK and the time at which the retransmitted packet(s) corresponding to that NACK arrive (which is one RTT or greater).

6.2. Receiver-based decisions on requesting retransmissions

Before making a retransmission request at time T_r of packet N (scheduled for play-out at time $T_p[N]$), the receiver checks if

$$T_r + eRTT < T_p[N]$$

where $eRTT$ is an estimate of the round trip time (RTT). If true then the receiver assumes the retransmission will arrive in time, and it sends the request for packet N with R bit set. If false the receiver sends a NACK whose R bit is set to 0. The above check can be made more general to account for receiver decoding delays and the sender's delay in responding to the retransmission request. The RTT estimate can be based on the measured elapsed time between the transmission of each NACK and the reception of each retransmitted packet. $eRTT$ is calculated from these RTT measurements through Karn's algorithm[7].

7. References

- [1] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A transport protocol for real-time applications," [RFC 1889](#), January 1996.
- [2] S. Bradner, "Key words for use in RFCs to indicate requirement levels," Request for Comments (Best Current Practice) [2119](#), Internet Engineering Task Force, Mar. 1997.
- [3] Turletti, T. and Huitema, C., "RTP Payload Format for H.261 Video Streams," [RFC 2032](#), October 1996.
- [4] Perkins, C. and Hodson, O., "Options for Repair of Streaming Media" [RFC 2354](#), June 1998.
- [5] R. X. Xu, A. C. Myers, H. Zhang, and R. Yavatkar. Resilient multicast support for continuous media applications. In Proceedings of the 7th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV'97), Washington University in St. Louis, Missouri, May 1997.
- [6] Schulzrinne, H., "RTP Profile for Audio and Video Conferences with Minimal Control," [RFC 1890](#), January 1996.
- [7] P. Karn and C. Partridge, "Improving Round-Trip Time Estimates in Reliable Transport Protocol." ACM Transactions on Computer Systems (TOCS), Vol. 9, No. 4, pp. 364-373, November 1991
- [8] M. Handley and V. Jacobson, "SDP: Session Description Protocol."

Request for Comments [RFC 2327](#).

[9] T. Turletti and C. Huitema, "RTP Payload Format for H.261 Video Streams." Request for Comments [RFC 2032](#), October 1996.

[10] J. Mahdavi, S. Floyd, "TCP-Friendly Unicast Rate-Based Flow Control", Technical note sent to the end2end-interest mailing list, January 8, 1997.

[11] S. Floyd, M. Handley, J. Padhye, and J. Widmer, "Equation-Based Congestion Control for Unicast Applications." ACM SIGCOMM 2000.

AUTHORS' ADDRESSES

Koichi Yano
UC Berkeley Computer Science Dept.
Phone: 510-642-9513
Email: yano@cs.berkeley.edu
URL: <http://www.cs.berkeley.edu/~yano>

Matthew Podolsky
UC Berkeley Computer Science Dept.
Phone: 510-642-8905
Email: podolsky@eecs.berkeley.edu
URL: <http://www.eecs.berkeley.edu/~podolsky>

Steven McCanne
FastForward Networks
Email: mccanne@cs.berkeley.edu

This draft was created in July, 2000.
It expires Jan, 2001.

