

Internet Engineering Task Force  
Internet Draft  
[draft-podolsky-avt-rtprx-01.txt](#)  
March 10, 2000  
Expires: September 10, 2000

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/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.

## **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 profile, RTP/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. This section addresses the items which are modified by this profile.

RTCP packet types:

A new RTCP packet types, 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](#).



#### **4. Control packets for unicast session**

This note proposes extending the RTCP specification to accommodate retransmission requests. RTCP is a reasonable place for putting control packets into 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 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](#).

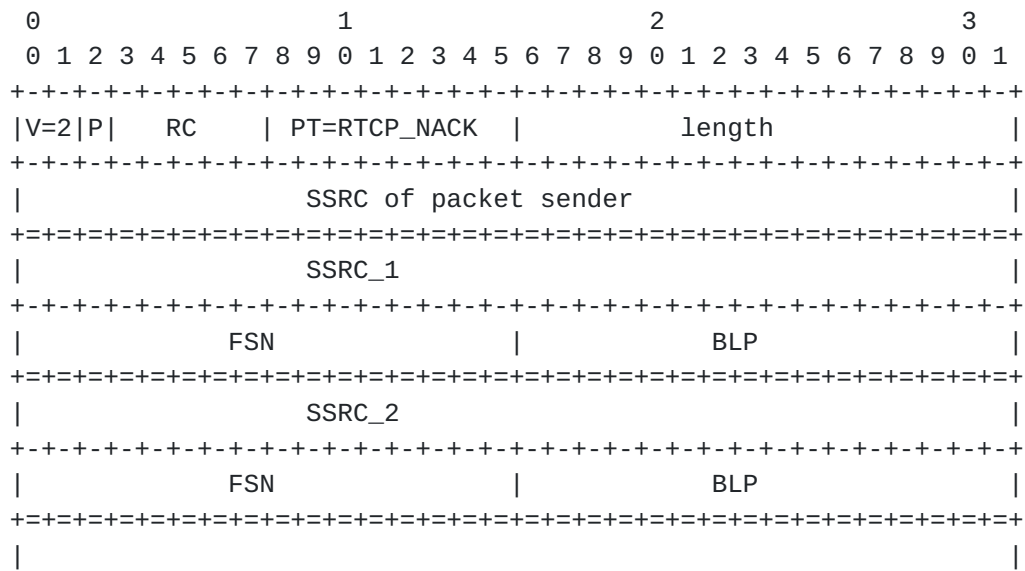
If the size of RTP payload packets is 1 KBytes and the size of RTCP packets is 40 bytes, the 3.75% limitation is large enough for a receiver to send an RTCP packet at every other data packet interval. The interval is frequent enough to allow timely retransmission requests.

The original RTCP packets (SR/RR) also SHOULD be sent based on [RFC 1889](#). 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. Packet format for requesting retransmission

This section defines a packet format for a negative acknowledgment (NACK) packet, which is sent from the receiver to the sender and which asks for the retransmission of one or more RTP packets. The format of NACK packets is as follows:



The various fields V, P, RC, SSRC and length are defined in the RTP specification [1]. The FSN and BLP fields have similar definitions to those given for the RTCP\_NACK packet specified in [3]. 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.

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 not received RTP data packet N, and that the receiver is requesting the retransmission of packet N.

bitmask of following lost packets (BLP): 16 bits

The BLP allows for the reporting of losses of the 16 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 16, then bit i of the bitmask is set to 1 if the sender has not received RTP packet number FSN+i (modulo  $2^{16}$ ) and the receiver is requesting its retransmission; 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 BLP is set to 0x00001 if the packet corresponding to the FSN and the following packet have been lost. However, the sender cannot infer that packets FSN+2 through FSN+16 have been received simply because bits 2 through 16 of the BLP are 0; all the sender knows is that their retransmission has not been requested.

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.

## 5. Examples

### 5.1 A simple retransmission request system

The simplest way to realize loss recovery using retransmission is that the receiver decides whether a NACK packet should be sent or not and the sender replies all retransmission requests. The receiver sends NACKs when it determines that one or more data packets have been lost and that their playback time has not yet passed.



This scheme can be realized without an extra control packet between the sender and the receiver. However it could results in many late packets that arrive after play out time because it does not consider transmission delay.

## **5.2. 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 the receiver assumes the retransmission will arrive in time, and it sends the request for packet  $N$ . If false the receiver does not send a retransmission request. The above check can be made more general to account for receiver decoding delays and the sender's request response time. RTT can be measured elapsed time between each NACK and the retransmission packet.  $eRTT$  is calculated from the RTT measurements through Karn's algorithm[7].

## **5. 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] Tureletti, 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

## AUTHORS' ADDRESSES

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

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

Steven McCanne  
FastForward Networks  
Email: [mccanne@cs.berkeley.edu](mailto:mccanne@cs.berkeley.edu)

This draft was created in March 10, 2000.  
It expires September 10, 2000.

