

Network Working Group  
Internet-Draft  
Intended status: Best Current  
Practice  
Expires: August 11, 2008

X. Marjou  
A. Sollaud  
France Telecom  
February 8, 2008

Application Mechanism for maintaining alive the Network Address  
Translator (NAT) mappings associated to RTP flows.  
draft-ietf-avt-app-rtp-keepalive-02

## Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with [Section 6 of BCP 79](#).

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/lid-abstracts.txt>.

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

This Internet-Draft will expire on August 11, 2008.

## Copyright Notice

Copyright (C) The IETF Trust (2008).

## Abstract

This document lists the different mechanisms that enable applications using Real-time Transport Protocol (RTP) to maintain their RTP Network Address Translator (NAT) mappings alive. It also makes a recommendation for a preferred mechanism. This document is not applicable to Interactive Connectivity Establishment (ICE) agents.

Internet-Draft

RTP keepalive

February 2008

## Table of Contents

<a href="#">1.</a>	Introduction . . . . .	<a href="#">3</a>
<a href="#">2.</a>	Terminology . . . . .	<a href="#">4</a>
<a href="#">3.</a>	Requirements . . . . .	<a href="#">4</a>
<a href="#">4.</a>	List of Alternatives for Performing RTP Keepalive . . . . .	<a href="#">5</a>
<a href="#">4.1.</a>	UDP Packet of 0-byte . . . . .	<a href="#">5</a>
<a href="#">4.2.</a>	DCCP Packet of 0-byte . . . . .	<a href="#">5</a>
<a href="#">4.3.</a>	RTP Packet with Comfort Noise Payload . . . . .	<a href="#">5</a>
<a href="#">4.4.</a>	RTCP Packets Multiplexed with RTP Packets . . . . .	<a href="#">6</a>
<a href="#">4.5.</a>	STUN Indication Packet . . . . .	<a href="#">6</a>
<a href="#">4.6.</a>	RTP Packet with No-OP Payload . . . . .	<a href="#">6</a>
<a href="#">4.7.</a>	RTP Packet with Incorrect Version Number . . . . .	<a href="#">7</a>
<a href="#">4.8.</a>	RTP Packet with Unknown Payload Type . . . . .	<a href="#">7</a>
<a href="#">5.</a>	Recommended solution for keepalive mechanism . . . . .	<a href="#">8</a>
<a href="#">6.</a>	Additional considerations . . . . .	<a href="#">8</a>
<a href="#">7.</a>	Security Considerations . . . . .	<a href="#">8</a>
<a href="#">8.</a>	IANA Considerations . . . . .	<a href="#">9</a>
<a href="#">9.</a>	Acknowledgements . . . . .	<a href="#">9</a>
<a href="#">10.</a>	References . . . . .	<a href="#">9</a>
<a href="#">10.1.</a>	Normative references . . . . .	<a href="#">9</a>
<a href="#">10.2.</a>	Informative references . . . . .	<a href="#">9</a>
	Authors' Addresses . . . . .	<a href="#">10</a>
	Intellectual Property and Copyright Statements . . . . .	<a href="#">11</a>

Internet-Draft

RTP keepalive

February 2008

## 1. Introduction

Documents [\[4\]](#) and [\[5\]](#) describe NAT behaviors and point out that two key aspects of NAT are mappings (a.k.a. bindings) and their refreshment. This introduces a derived requirement for applications engaged in a multimedia session involving NAT traversal: they need to generate a minimum of flow activity in order to create NAT mappings and maintain them alive.

When applied to applications using RTP [\[2\]](#), the RTP media stream packets themselves normally fulfill this requirement. However there exist some cases where RTP do not generate a minimum flow activity.

The examples are:

- o In some RTP usages, such as SIP, agents can negotiate a unidirectional media stream by using the SDP "recvonly" attribute on one agent and "sendonly" on the peer, as defined in [RFC 3264](#) [\[7\]](#). [RFC 3264](#) directs implementations not to transmit media on the receiving agent. In case the agent receiving the media is located in the private side of a NAT, it will never receive RTP packets from the public peer if the NAT mapping has not been created.
- o Similarly, a bidirectional media stream can be "put on hold". This is accomplished by using the SDP "sendonly" or "inactive" attributes. Again [RFC 3264](#) directs implementations to cease transmission of media in these cases. However, doing so may cause NAT bindings to timeout, and media won't be able to come off hold.
- o In case of audio media, if silence suppression is in use, long periods of silence may cause media transmission to cease sufficiently long for NAT bindings to time out.
- o Some RTP payload formats, such as the payload format for text conversation [\[12\]](#), may send packets so infrequently that the

interval exceeds the NAT binding timeouts.

To solve these problems, an agent therefore needs to periodically send keepalive data within the outgoing RTP session of an RTP media stream regardless of whether the media stream is currently inactive, sendonly, recvonly or sendrecv, and regardless of the presence or value of the bandwidth attribute.

It is also important to note that the above examples also require the agents to use symmetric RTP [[13](#)] in addition to RTP keepalive.

This document first states the requirements that must be supported to

perform RTP keepalives ([Section 3](#)). In a second step, the document reports the different mechanisms to overcome this problem ([Section 4](#)) and makes recommendations about their use.

The scope of the draft is limited to non-ICE agents. Indeed, ICE agents need to follow the RTP keepalive mechanism specified in the ICE specification [[6](#)].

The scope of the draft is also limited to RTP flows. In particular, this document does not address keepalive activity related to:

- o Session signaling flows, such as the Session Initiation Protocol (SIP).
- o RTCP flows.
  - \* Recall that [[2](#)] recommends a minimum interval of 5 seconds and that "on hold" procedures of [[7](#)] do not impact RTCP transmissions. Therefore, when in use, there is always some RTCP flow activity.

Note that if a given media uses a codec that already integrates a keepalive mechanism, no keepalive mechanism is required at the RTP level.

## [2](#). Terminology

In this document, the key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY",

and "OPTIONAL" are to be interpreted as described in [RFC 2119](#) [1].

### [3.](#) Requirements

This section outlines the key requirements that need to be satisfied in order to provide RTP media keepalive.

REQ-1 Some data is sent periodically within the outgoing RTP session for the whole duration of the RTP media stream.

REQ-2 Any type of transport (e.g. UDP, TCP) MUST be supported.

REQ-3 Any media type (e.g. audio, video, text) MUST be supported.

REQ-4 Any media format (e.g. G.711, H.263) MUST be supported.

REQ-5 Session signaling protocols SHOULD not be impacted.

REQ-6 Session description protocols SHOULD not be impacted.

REQ-7 Impacts on existing software SHOULD be minimized.

REQ-8 Remote peer SHOULD not be impacted.

REQ-9 More than one mechanism MAY exist.

### [4.](#) List of Alternatives for Performing RTP Keepalive

This section lists some alternatives that can be used in order to perform a keepalive message within RTP media streams.

#### [4.1.](#) UDP Packet of 0-byte

The application sends an empty UDP packet.

Cons:

- o This alternative is specific to UDP.

Recommendation:

- o This method should not be used for RTP keepalive.

#### [4.2.](#) DCCP Packet of 0-byte

The application sends an empty DCCP packet.

Cons:

- o This alternative is specific to DCCP.

Recommendation:

- o This method should not be used for RTP keepalive.

#### [4.3.](#) RTP Packet with Comfort Noise Payload

The application sends an RTP packet with a comfort-noise payload [\[10\]](#).

Cons:

- o This alternative is limited to audio formats only.
- o Comfort Noise needs to be supported by the remote peer.

- o Comfort Noise needs to be signalled in SDP offer/answer.
- o The peer is likely to render comfort noise at the other side, so the content of the payload (the noise level) needs to be carefully chosen.

Recommendation:

- o This method may be used when the media allows for it.

#### [4.4.](#) RTCP Packets Multiplexed with RTP Packets

The application sends RTCP packets in the RTP media path itself (i.e. same tuples for both RTP and RTCP packets) [\[3\]](#). RTCP packets therefore maintain the NAT mappings open.

Cons:

- o Multiplexing RTP and RTCP must be supported by the remote peer.
- o Multiplexing RTP and RTCP must be signalled in SDP offer/answer.
- o Some RTCP monitoring tools expect that RTCP are not multiplexed.

Recommendation:

- o This method must only be used for RTP keepalive when negotiated between agents.

#### [4.5.](#) STUN Indication Packet

The application sends a STUN [\[8\]](#) Binding Indication packet as specified in ICE [\[6\]](#).

Thanks to the RTP validity check, STUN packets will be ignored by the RTP stack.

Cons:

- o The sending agent needs support STUN.

Recommendation:

- o This method must only be used for sessions between ICE agents, as specified in [\[6\]](#).

#### [4.6.](#) RTP Packet with No-OP Payload

The application sends an RTP No-OP payload [\[11\]](#) .

Cons:

- o This payload type needs to be supported by the remote peer.
- o This payload type needs to be signalled in the SDP offer/answer.
- o [RFC4566](#) [\[9\]](#) and [RFC3264](#) [\[7\]](#) mandate not to send media with inactive and recvonly attributes.

Recommendation:

- o This method may be used for RTP keepalive when negotiated between agents.

[Authors note: this draft is now expired, do we keep it here?]

#### [4.7.](#) RTP Packet with Incorrect Version Number

The application sends an RTP packet with an incorrect version number, which value is zero.

Based on RTP specification [2], the peer should perform a header validity check, and therefore ignore these types of packet.

Cons:

- o Only four version numbers are possible. Using one of them for RTP keepalive would be wasteful.
- o [RFC4566](#) [9] and [RFC3264](#) [7] mandate not to send media with inactive and recvonly attributes.

Recommendation:

- o This method should not be used for RTP keepalive.

#### [4.8.](#) RTP Packet with Unknown Payload Type

The application sends an RTP packet of 0 length with a dynamic payload type that has not been negotiated by the peers (e.g. not negotiated within the SDP offer/answer, and thus not mapped to any media format).

The sequence number is incremented by one for each packet, as it is sent within the same RTP session as the actual media. The timestamp contains the same value a media packet would have at this time. The marker bit is not significant for the keepalive packets and is thus set to zero.

Normally the peer will ignore this packet, as RTP [2] states that "a receiver MUST ignore packets with payload types that it does not understand".

Cons:

- o [RFC4566](#) [9] and [RFC3264](#) [7] mandate not to send media with inactive and recvonly attributes, however this is mitigated as no real media is sent with this mechanism.

Recommendation:

- o This method should be used for RTP keepalive.



## 5. Recommended solution for keepalive mechanism

Some mechanisms do not meet the requirements as they are either specific to the transport ([Section 4.1](#), [Section 4.2](#)), specific to a media type ([Section 4.3](#)). These mechanisms are thus NOT RECOMMENDED.

Other mechanisms are dependent on the capabilities of the peer ([Section 4.4](#), [Section 4.5](#), [Section 4.6](#)). Among these mechanisms, RTCP packets multiplexed with RTP packets ([Section 4.4](#)) is desirable because it reduces the number of ports used.

The RECOMMENDED solution is thus the "RTCP packets multiplexed with RTP packets" ([Section 4.4](#)). However, when this mechanism cannot be negotiated, it is RECOMMENDED to use the fallback "RTP Packet with Unknown Payload Type" mechanism of [Section 4.8](#) as it will always work.

The exception to the above recommendation is for real-time text as [\[12\]](#) does not allow to use different payloads within a same RTP session. For real-time text, the RECOMMENDED solution is thus the "RTCP packets multiplexed with RTP packets". When this mechanism cannot be negotiated, it is RECOMMENDED to use an empty T140block containing no data in the same manner as for the idle procedure defined in [\[12\]](#).

## 6. Additional considerations

An application supporting this specification must transmit keepalive packets every  $T_r$  seconds during the whole duration of the media session.  $T_r$  SHOULD be configurable, and otherwise MUST default to 15 seconds.

Keepalives packets within a particular RTP session MUST use the tuple (source IP address, source TCP/UDP ports, target IP address, target TCP/UDP Port) of the regular RTP packets.

The agent SHOULD only send RTP keepalive when it does not send regular RTP packets.

## 7. Security Considerations

The keepalive packets are sent on the same path as regular RTP media packets. In addition, they do not convey any valuable information.

So the mechanism described here does not imply new security issues.

## 8. IANA Considerations

This document has no actions for IANA.

## 9. Acknowledgements

Jonathan Rosenberg provided the major inputs for this draft via the ICE specification. In addition, thanks to Alfred E. Heggstad, Colin Perkins, Dan Wing, Gunnar Hellstrom, and Randell Jesup for their useful inputs and comments.

## 10. References

### 10.1. Normative references

- [1] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [2] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, [RFC 3550](#), July 2003.
- [3] Perkins, C. and M. Magnus, "Multiplexing RTP Data and Control Packets on a Single Port", [draft-ietf-avt-rtp-and-rtcp-mux-07](#) (work in progress), August 2007.

### 10.2. Informative references

- [4] Audet, F. and C. Jennings, "Network Address Translation (NAT) Behavioral Requirements for Unicast UDP", [BCP 127](#), [RFC 4787](#), January 2007.
- [5] Guha, S., Biswas, K., Ford, B., Francis, P., Sivarkumar, S., and P. Srisuresh, "NAT Behavioral Requirements for TCP", [draft-ietf-behave-tcp-07](#) (work in progress), April 2007.
- [6] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Methodology for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", [draft-ietf-mmusic-ice-19](#) (work in progress), October 2007.

- [7] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", [RFC 3264](#), June 2002.

- [8] Rosenberg, J., Huitema, C., Mahy, R., and D. Wing, "Simple Traversal Underneath Network Address Translators (NAT) (STUN)", [draft-ietf-behave-rfc3489bis-13](#) (work in progress), November 2007.
- [9] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", [RFC 4566](#), July 2006.
- [10] Zopf, R., "Real-time Transport Protocol (RTP) Payload for Comfort Noise (CN)", [RFC 3389](#), September 2002.
- [11] Andreason, F., Oran, D., and D. Wing, "A No-OP Payload Format for RTP", [draft-ietf-avt-rtp-no-op-04](#) (work in progress), May 2007.
- [12] Hellstrom, G. and P. Jones, "RTP Payload for Text Conversation", [RFC 4103](#), June 2005.
- [13] Wing, D., "Symmetric RTP / RTP Control Protocol (RTCP)", [BCP 131](#), [RFC 4961](#), July 2007.

#### Authors' Addresses

Xavier Marjou  
France Telecom  
2, avenue Pierre Marzin  
Lannion 22307  
France

Email: [xavier.marjou@orange-ftgroup.com](mailto:xavier.marjou@orange-ftgroup.com)

Aurelien Sollaud  
France Telecom  
2, avenue Pierre Marzin  
Lannion 22307  
France

Email: aurelien.sollaud@orange-ftgroup.com

Marjou & Sollaud

Expires August 11, 2008

[Page 10]

---

Internet-Draft

RTP keepalive

February 2008

#### Full Copyright Statement

Copyright (C) The IETF Trust (2008).

This document is subject to the rights, licenses and restrictions contained in [BCP 78](#), and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

#### Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in [BCP 78](#) and [BCP 79](#).

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this

specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at [ietf-ipr@ietf.org](mailto:ietf-ipr@ietf.org).

#### Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).