Network Working Group
Internet-Draft
Intended status: Best Current Practice
Expires: December 24, 2007

X. Marjou
A. Sollaud
France Telecom
June 22, 2007

Application Mechanism for maintaining alive the Network Address Translator (NAT) mappings associated to RTP flows. draft-ietf-avt-app-rtp-keepalive-00

Status of this Memo

Copyright Notice

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.

Table of Contents

1. Introduction

Documents [2] and [3] 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 [4], 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:

- 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 [6]. 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.
- 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.
- 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.
- 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, several mechanisms are laid-out to overcome this problem (Section 4).

The scope of the draft is limited to RTP flows. In particular, this

document does not address keepalive activity related to:

- Session signaling flows, such as the Session Initiation Protocol (SIP).
- RTCP flows.
  - Recall that [4] recommends a minimum interval of 5 seconds and that "on hold" procedures of [6] do not impact RTCP transmissions. Therefore, when in use, there is always some RTCP flow activity.

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:
- This alternative is specific to UDP.

4.2. RTCP Packets Multiplexed with RTP Packets

The application sends RTCP packets in the RTP media stream itself (i.e. same tuples for both RTP and RTCP packets) [7]. RTCP packets therefore maintain the NAT mappings open.

Cons:
- Multiplexing RTP and RTCP must be supported by the remote peer.
- Multiplexing RTP and RTCP must be signalled in SDP offer/answer.
- Some RTCP monitoring tools expect that RTCP are not multiplexed. [[OPEN-POINT: is this argument strong enough to keep it?]]

4.3. STUN Indication Packet

The application sends a STUN [8] Binding Indication packet as

specified in ICE [5].

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

Cons:
- The agent needs to support STUN.

4.4. 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 [4], the peer should perform a header validity check, and therefore ignore these types of packet.

Cons:
- Only four version numbers are possible. Using one of them for RTP keepalive would be wasteful.
- RFC4566 [9] and RFC3264 [6] mandate not to send media with inactive and recvonly attributes.

4.5. RTP Packet with Comfort Noise Payload

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

Cons:
- This alternative is limited to audio formats only.
- Comfort Noise needs to be supported by the remote peer.
- Comfort Noise needs to be signalled in SDP offer/answer.
- 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.

4.6. RTP Packet with No-Op Payload

The application sends an RTP No-OP payload [11] .

Cons:
- This payload type needs to be supported by the remote peer.
- This payload type needs to be signalled in the SDP offer/answer.

- o RFC4566 [9] and RFC3264 [6] mandate not to send media with inactive and recvonly attributes.

4.7. RTP Packet with Unknown Payload Type

The application sends an RTP packet 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).

Normally the peer will ignore it, as RTP [4] states that "a receiver MUST ignore packets with payload types that it does not understand".

Cons:
- o RFC4566 [9] and RFC3264 [6] mandate not to send media with inactive and recvonly attributes.
- o [[OPEN-POINT: should we say something about the content of the payload?]]

5. Additional considerations

An application supporting this specification must transmit keepalive packets during the whole duration of the media session.

The application can send keepalive packets under the form of any of the above mechanisms.

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.

Keepalive packets MUST be sent every Tr seconds. Tr SHOULD be configurable, and otherwise MUST default to 15 seconds. [Note: same

value as in [5].]

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

6. Security Considerations

T.B.D.

7. Acknowledgements

Jonathan Rosenberg provided the major inputs for this draft via the ICE specification. In addition, thanks to Dan Wing for his useful inputs and comments.

8. References

8.1. Normative references

[1] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, March 1997.

[2] Audet, F. and C. Jennings, "Network Address Translation (NAT) Behavioral Requirements for Unicast UDP", RFC 4787, January 2007.

[3] 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.

[4] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", RFC 3550, July 2003.

[5] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Methodology for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", draft-ietf-mmusic-ice-16 (work in progress), June 2007.

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

[7] Perkins, C. and M. Magnus, "Multiplexing RTP Data and Control Packets on a Single Port", draft-ietf-avt-rtp-and-rtcp-mux-05 (work in progress), May 2007.

[8] Rosenberg, J., Huitema, C., Mahy, R., and D. Wing, "Simple

Traversal Underneath Network Address Translators (NAT) (STUN)", draft-ietf-behave-rfc3489bis-06 (work in progress), March 2007.

[9] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006.

[10] Robert, 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.

8.2. Informative references

[12] Hellstrom, G. and P. Jones, "RTP Payload for Text Conversation", RFC 4103, June 2005.

[13] Wing, D., "Symmetric RTP/RTCP", draft-wing-behave-symmetric-rtprtcp-03 (work in progress), April 2007.

Authors' Addresses

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

Email: xavier.marjou@orange-ftgroup.com

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

Email: aurelien.sollaud@orange-ftgroup.com

Full Copyright Statement

Intellectual Property

Acknowledgment

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