Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: June 11, 2010

X. Marjou
A. Sollaud
France Telecom

December 8, 2009

Application Mechanism for maintaining alive the Network Address Translator (NAT) mappings associated to RTP flows. draft-ietf-avt-app-rtp-keepalive-07

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.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and 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/1id-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 June 11, 2010.

Copyright Notice

Table of Contents

1. Introduction

Documents [RFC4787] and [RFC5382] describe NAT behaviors and point out that two key aspects of NAT are mappings (a.k.a. bindings) and keeping them refreshed. 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.

When applied to applications using RTP [RFC3550], the RTP media stream packets themselves normally fulfill this requirement. However there exist some cases where RTP does not generate the minimum required 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 [RFC3264]. 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 [RFC4103], 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 important to note that the above examples also require the agents to use symmetric RTP [RFC4961] 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. Section 5 finally states the recommended solution for RTP keepalive.

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 [DRAFT-ICE].

The scope of the draft is also 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 [RFC3550] recommends a minimum interval of 5 seconds and that "on hold" procedures of [RFC3264] 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 additional keepalive mechanism is required at the RTP level.

As mentionned in Section 3.5 of [RFC5405] "It is important to note that keep-alive messages are NOT RECOMMENDED for general use -- they are unnecessary for many applications and can consume significant amounts of system and network resources."

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 [RFC2119].

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 Impacts on existing software SHOULD be minimized.

REQ-7 Remote peer SHOULD NOT be impacted.

REQ-8 The support for RTP keepalive SHOULD be described in the SDP.

REQ-9 More than one mechanism MAY exist.

4. List of Alternatives for Performing RTP Keepalive

This section lists, in no particular order, some alternatives that can be used to perform a keepalive message within RTP media streams.

4.1. Transport Packet of 0-byte

The application sends an empty transport packet (e.g. UDP packet, DCCP packet).

Cons:
- This alternative is specific to each transport protocol.

4.2. RTP Packet with Comfort Noise Payload

The application sends an RTP packet with a comfort-noise payload [RFC3389].

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.

4.3. 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) [DRAFT-RTP-RTCP]. 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.

4.4. STUN Indication Packet

The application sends a STUN [RFC5389] Binding Indication packet as specified in ICE [DRAFT-ICE].

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

Cons:
- The sending agent needs to support STUN.

4.5. 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 [RFC3550], 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] and [RFC3264] mandate not to send media with inactive and recvonly attributes, however this is mitigated as no real media is sent with this mechanism.

4.6. 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 [RFC3550] states that "a receiver MUST ignore packets with payload types that it does not understand".

Cons:
- [RFC4566] and [RFC3264] mandate not to send media with inactive

and recvonly attributes, however this is mitigated as no real media is sent with this mechanism.

5. Recommended Solution for Keepalive Mechanism

Some mechanisms do not meet the requirements as they are either specific to the transport (Section 4.1), or specific to a media type (Section 4.2), or waste one RTP version number (Section 4.5). These mechanisms are thus NOT RECOMMENDED.

Other mechanisms are dependent on the capabilities of the peer (Section 4.3, Section 4.4). Among these mechanisms, RTCP packets multiplexed with RTP packets (Section 4.3) is desirable because it reduces the number of ports used.

The RECOMMENDED solution is thus the "RTCP packets multiplexed with RTP packets" (Section 4.3). However, when this mechanism cannot be negotiated, it is RECOMMENDED to use the fallback "RTP Packet with Unknown Payload Type" mechanism (Section 4.6) as it will always work.

When using SDP, an agent supporting the fallback solution MUST indicate its support by adding an a=rtp-keepalive SDP attribute. This attribute is declarative only and can not be negotiated. It indicates that the fallback solution will be used if the recommended solution can not be used.

When using the SDP offer-answer [RFC3264], the agent SHOULD offer both the "a=rtcp-mux" and "a=rtp-keepalive" attributes. If "a=rtcp-mux" attribute is present in the answer, the agent uses RTCP packets being multiplexed on the RTP port as a keepalive. Otherwise, the agent uses RTP packets with an invalid payload type as a keepalive.

6. Media Format Exceptions

When a given media format does not allow the keepalive solution recommended in Section 5, an alternative mechanism SHOULD be defined in the payload format specification for this media format.

Real-time text payload format [RFC4103] is an example of such a media

format.

6.1. Real-time Text Payload Format Keepalive Mechanism

Real-time text payload format [RFC4103] does not allow different payloads within a same RTP session, so the fallback mechanism does not work.

For real-time text, the RECOMMENDED solution is 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 [RFC4103].

7. Timing and Transport Considerations

An application supporting this specification MUST transmit either keepalive packets or media packets at least once every Tr seconds during the whole duration of the media session. The minimum RECOMMENDED Tr value is 15 seconds, and Tr SHOULD be configurable to larger values.

When using the "RTCP packets multiplexed with RTP packets" solution for keepalive, Tr MUST comply with the RTCP timing rules of [RFC3550]. The fallback "RTP Packet with Unknown Payload Type" solution uses RTP, and thus does not have these RTCP constraints.

Keepalive 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.

8. Security Considerations

The RTP keepalive packets are sent on the same path as regular RTP media packets and may be perceived as an attack by a peer. However, [RFC3550] mandates a peer to "ignore packets with payload types that it does not understand". A peer that does not understand the keepalive message will thus appropriately drop the received packets.

9. IANA Considerations

9.1. Registration of the SDP 'rtp-keepalive' Attribute

This section instructs the IANA to register the following SDP att-field under the Session Description Protocol (SDP) Parameters registry:

Contact name: xavier.marjou@orange-ftgroup.com

Attribute name: rtp-keepalive

Long-form attribute name: RTP keepalive

Type of attribute Media level

Subject to charset: No

Purpose of attribute: The 'rtp-keepalive' attribute declares that the agent supports the fallback RTP keepalive mechanism (Section 4.6).

Allowed attribute values: None

10. Acknowledgements

Jonathan Rosenberg provided the major inputs for this draft via the ICE specification. In addition, thanks to Alfred E. Heggestad, Colin Perkins, Dan Wing, Gunnar Hellstrom, Hadriel Kaplan, Randell Jesup, Remi Denis-Courmont, and Steve Casner for their useful inputs and comments.

11. References

11.1. Normative references

[DRAFT-RTP-RTCP] 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.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time

Applications", STD 64, RFC 3550, July 2003.

[RFC4961] Wing, D., "Symmetric RTP / RTP Control Protocol (RTCP)", BCP 131, RFC 4961, July 2007.

[RFC5405] Eggert, L. and G. Fairhurst, "Unicast UDP Usage Guidelines for Application Designers", BCP 145, RFC 5405, November 2008.

11.2. Informative references

[DRAFT-ICE] 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.

[RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, June 2002.

[RFC3389] Zopf, R., "Real-time Transport Protocol (RTP) Payload for Comfort Noise (CN)", RFC 3389, September 2002.

[RFC4103] Hellstrom, G. and P. Jones, "RTP Payload for Text Conversation", RFC 4103, June 2005.

[RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006.

[RFC4787] Audet, F. and C. Jennings, "Network Address Translation (NAT) Behavioral Requirements for Unicast UDP", BCP 127, RFC 4787, January 2007.

[RFC5382] Guha, S., Biswas, K., Ford, B., Sivakumar, S., and P. Srisuresh, "NAT Behavioral Requirements for TCP", BCP 142, RFC 5382, October 2008.

[RFC5389] Rosenberg, J., Mahy, R., Matthews, P., and D. Wing,

"Session Traversal Utilities for NAT (STUN)", RFC 5389, October 2008.

Authors' Addresses

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

Email: xavier.marjou@orange-ftgroup.com

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

Email: aurelien.sollaud@orange-ftgroup.com