INTERNET DRAFT
<draft-ietf-mmusic-sdp4nat-03.txt>
Expires March 20, 2002

C. Huitema
Microsoft

September 20, 2002

RTCP attribute in SDP

Status of this memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026.

This document is an Internet-Draft. 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

The session description protocol (SDP) is used to describe the parameters of media streams used in multimedia sessions. When a session requires multiple ports, SDP assumes that these port have consecutive numbers. However, when the session crosses a network address translation device that also uses port mapping, the ordering of ports can be destroyed by the translation. To handle this, we propose an extension attribute to SDP.

1 Introduction

The session invitation protocol (SIP, [RFC3261]) is often used to establish multi-media sessions on the Internet. There are often cases today in which one or both end of the connection are hidden behind a network address translation device [RFC2766]. In this case, the SDP text must document the IP addresses and UDP ports as they appear on the "public Internet" side of the NAT; in this memo, we will suppose that the host located behind a NAT has a way to obtain these numbers; a possible way to learn these numbers is briefly outlined in section 3. However, just learning the numbers is not enough.

The SIP messages use the encoding defined in SDP [RFC2327] to describe the IP addresses and TCP or UDP ports used my the various media. Audio and video are typically sent using RTP [RFC1889], which requires two UDP ports, one for the media and one for the control

protocol (RTCP). SDP carries only one port number per media, and states that "other ports used by the media application (such as the RTCP port) should be derived algorithmically from the base media port." When the media is transmitted using RTP [RFC1889], the choice of the port number is very specific: "for UDP and similar protocols, RTP uses an even port number and the corresponding RTCP stream uses the next higher (odd) port number; if an application is supplied with an odd number for use as the RTP port, it should replace this number with the next lower (even) number."

When the NAT device performs port mapping, there is no guarantee that the mappings of two separate ports reflects the sequencing and the parity of the original port numbers; in fact, when the NAT manages a pool of IP addresses, it is even possible that the RTP and the RTCP ports may be mapped to different addresses. In order to successfully establish connections despite the misordering of the port numbers and the possible parity switches caused by the NAT, we propose to use a specific SDP attribute to document the RTCP port and optionally the RTCP address, and we also propose to make the behavior of RTP implementations more conforming to the robustness principle.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

2 Description of the solution

The main part of our solution is the declaration of an SDP attribute for documenting the port used by RTCP. In order for the solution to be useful, the RTP implementation must be made more tolerant than specified in [RFC1889].

2.1 The RTCP attribute

The RTCP attribute is used to document the RTCP port used for media stream, when that port is not the next higher (odd) port number following the RTP port described in the media line. The RTCP attribute is a "value" attribute, and follows the general syntax specified page 18 of [RFC2327]: "a=<attribute>:<value>". For the

RTCP attribute:

* the name is the ascii string "rtcp" (lower case),

* the value is the RTCP port number and optional address.

The formal description of the attribute is defined by the following ABNF syntax:

```
rtcp-attribute =  "a=rtcp:" port  [nettype space addrtype space
                        connection-address] CRLF
```


In this description, the "port", "nettype", "addrtype" and "connection-address" tokens are defined as specified in "Appendix A: SDP Grammar" of [RFC2327].

Example encodings could be:

```
    m=audio 49170 RTP/AVP 0
    a=rtcp:53020

    m=audio 49170 RTP/AVP 0
    a=rtcp:53020 IN IP4 126.16.64.4

    m=audio 49170 RTP/AVP 0
    a=rtcp:53020 IN IP6 2001:2345:6789:ABCD:EF01:2345:6789:ABCD
```

The RTCP attribute MAY be used as a media level attribute; it MUST NOT be used as a session level attribute.

2.2 Oddity tolerant RTP

In order to successfully exchange RTP packets with a host located behind a NAT, a corresponding RTP implementation should be more tolerant than specified in [RFC1889]. If it receives an SDP text specifying the use of a specific port number for RTP, and another specific port number for RTCP, the implementation SHOULD send packets to exactly these port numbers, regardless of whether the numbers are odd or even, in sequence or separate.

For compatibility with existing implementations, the modified RTP behavior MUST NOT be used if the RTCP port is not explicitly specified. An implementation that wishes to receive RTP packets on an odd port number MUST document both the RTP and the RTCP ports in

the SDP description, even if the RTCP port is immediately consecutive to the RTP port.

3 Discussion of the solution

The implementation of the solution is fairly straightforward. The three questions that have been most often asked regarding this solution are whether this is useful, i.e. whether a host can actually discover port numbers in an unmodified NAT, whether it is sufficient, i.e. whether or not there is a need to document more than one ancillary port per media type, and whether relaxing the RTP requirements is legitimate.

3.1 How do we discover port numbers?

The proposed solution assumes that we can discover the "translated port numbers", i.e. the value of the ports as they appear on the "external side" of the NAT. There are multiple ways to achieve this result. One possibility is to ask the cooperation of a well connected third party that will act as a server according to [STUN].

We thus obtain a three step process:

1) The host allocate two UDP ports numbers for an RTP/RTCP pair,

2) The host sends a UDP message from each port to the STUN server,

3) The STUN server reads the source address and port of the packet, and copies them in the text of a reply,

4) The host parses the reply according to the STUN protocol and learns the external address and port corresponding to each of the two UDP port.

This algorithm supposes that the NAT will use the same translation for packets sent to the third party and to the "SDP peer" with which the host wants to establish a connection. The experience shows that this is the case for a large fraction of NATs.

3.2 Do we need to support multiple ports?

Most media streams are transmitted using a single pair of RTP and RTCP ports. It is possible, however, to transmit a single media over several RTP flows, for example using hierarchical encoding. In this case, SDP will encode the port number used by RTP on the first flow,

and the number of flows, as in:

```
       m=video 49170/2 RTP/AVP 31
```

In this example, the media is sent over 2 consecutive pairs of ports, corresponding respectively to RTP for the first flow (even number, 49170), RTCP for the first flow (odd number, 49171), RTP for the second flow (even number, 49172), and RTCP for the second flow (odd number, 49173).

In theory, it would be possible to modify SDP and document the many ports corresponding to the separate encoding layers. However, layered encoding is not much used in practice, and when used is mostly used in conjunction with multicast transmission. The translation issues documented in this memo apply uniquely to unicast transmission, and thus there is no short term need for the support of multiple port descriptions. It is more convenient and more robust to focus on the simple case in which a media is sent over exactly one RTP/RTCP stream.

3.3 Why not expand the media definition?

The RTP ports are documented in the media description line, and it would seem convenient to document the RTCP port at the same place, rather than create an RTCP attribute. We considered this design alternative and rejected it for two reasons: adding an extra port number and an option address in the media description would be awkward, and more importantly it would create problems with existing

applications, which would have to reject the entire media description if they did not understand the extension. On the contrary, adding an attribute has a well defined failure mode: implementations that don't understand the "a=rtcp" attribute will simply ignore it; they will fail to send RTCP packets to the specified address, but they will at least be able to receive the media in the RTP packets.

3.4 Is a tolerant RTP legitimate?

Our solution explicitly asks implementers to disregard a part of the RTP specification that mandates use of even port numbers for RTP and the consecutive odd port number for RTCP. We believe that this is very much in the spirit of the robustness principle attributed to Jon Postel, i.e. "Be conservative in what you do, be liberal in what you accept from others."

This approach has been validated with the AVT working group of the IETF, which is in charge of maintaining the RTP standard. We expect that the revised version of the RTP standard will lift the restrictions on port numbers imposed in [RFC1889], e.g. specify that for applications in which the RTP and RTCP destination port numbers are specified via explicit, separate parameters (using a signaling protocol or other means), the application MAY disregard the restrictions that the port numbers be even/odd and consecutive although the use of an even/odd port pair is still encouraged.

4 UNSAF considerations

The RTCP attribute in SDP is used to enable establishment of RTP/RTCP flows through NAT, in conjunction with an address discovery mechanism such as STUN. This mechanism is a short term fix to the NAT traversal problem, which requires thus consideration of the general issues linked to "Unilateral self-address fixing" [UNSAF].

The RTCP attribute addresses a very specific problem, the documentation of port numbers as they appear after address translation by a port-mapping NAT. The RTCP attribute SHOULD NOT be used for other applications.

We expect that, with time, one of two exit strategies can be developed. The IETF may develop an explicit "middlebox control" protocol, that will enable applications to obtain a pair of port numbers appropriate for RTP and RTCP. Another possibility is the deployment of IPv6, which will enable use of "end to end" addressing, and guarantee that the two hosts will be able to use appropriate ports. In both cases, there will be no need for documenting a "non standard" RTCP port with the RTCP attribute.

5 Security Considerations

This SDP extension is not believed to introduce any significant

security risk to multi-media applications. One could conceive that a malevolent third party would use the extension to redirect the RTCP fraction of an RTP exchange, but this require intercepting and rewriting the signaling packet carrying the SDP text; if an interceptor can do that, many more attacks are available, including a wholesale change of the addresses and port numbers at which the media will be sent.

In order to avoid attacks of this sort, when SDP is used in a signaling packet where it is of the form application/sdp, end-to-end integrity using S/MIME [RFC3369] is the technical method to be implemented and applied. This is compatible with SIP [RFC3261].

6 IANA Considerations

This document defines a new SDP parameter, the attribute field "rtcp", which per [RFC2327] should be registered by IANA. This attribute field is designed for use at media level only.

7 Copyright

The following copyright notice is copied from RFC 2026 [Bradner, 1996], Section 10.4, and describes the applicable copyright for this document.

Copyright (C) The Internet Society March 21, 2001. All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assignees.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS 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.

8 Intellectual Property

The following notice is copied from RFC 2026 [Bradner, 1996], Section 10.4, and describes the position of the IETF concerning intellectual property claims made against this document.

The IETF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use other technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on the IETF's procedures with respect to rights in standards-track and standards-related documentation can be found in BCP-11. Copies of claims of rights made available for publication 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 Secretariat.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director.

9 Acknowledgements

The original idea for using the "rtcp" attribute was developed by Ann Demirtjis. The draft was reviewed by the MMUSIC and AVT working groups of the IETF.

10 References

[RFC3261] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler. SIP: Session Initiation Protocol. RFC 3261, June 2002.

[RFC2327] M. Handley, V. Jacobson, "SDP: Session Description Protocol", RFC 2327, April 1998.

[RFC3369] R. Housley. Cryptographic Message Syntax (CMS). RFC 3369, August 2002.

[RFC1889] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson. "RTP: A Transport Protocol for Real-Time Applications", RFC 1889, January 1996.

[RFC2766] G. Tsirtsis, P. Srisuresh. "Network Address Translation - Protocol Translation (NAT-PT)", RFC 2766, February 2000.

Requirement Levels", RFC 2119, March 1997.

[RFC2234] D. Crocker, P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, November 1997.

[UNSAF] L. Daigle, "IAB considerations for UNilateral self-address fixing (UNSAF) across network address translation," Internet Draft, Internet Engineering Task Force, Approved Sep 2002.
draft-iab-unsaf-considerations-02.txt

11 Author's Addresses

Christian Huitema
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052-6399

Email: huitema@microsoft.com

Table of Contents: