             Real Time Control Protocol (RTCP) attribute in
                 Session Description Protocol (SDP)

Status of this Memo

   This document specifies an Internet standards track protocol for the
   Internet community, and requests discussion and suggestions for
   improvements. Please refer to the current edition of the "Internet
   Official Protocol Standards" (STD 1) for the standardization state
   and status of this protocol. Distribution of this memo is unlimited.

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 ports 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 ends 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 by the various
   media. Audio and video are typically sent using RTP [RFC3550], 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." RTCP port numbers were necessarily derived from the base
media port in older versions of RTP (such as [RFC1889]), but now that
this restriction has been lifted, there is a need to specify RTCP
ports explicitly in SDP. Note, however, that implementations of RTP
adhering to the earlier [RFC1889] specification may not be able to
make use of the SDP attributes specified in this document.

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.

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.

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 [RFC2234] 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. Though the examples below
relate to a method that will return only unicast addresses, both
unicast and multicast values are valid.

3. Discussion of the Solution

The implementation of the solution is fairly straightforward. The
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 why should not change the media
definition instead of adding a new attribute.

3.1. How do we Discover Port Numbers?

The proposed solution is only useful if the host can discover the
"translated port numbers", i.e., the value of the ports as they
appear on the "external side" of the NAT. One possibility is to ask
the cooperation of a well connected third party that will act as a
server according to STUN [RFC3489]. We thus obtain a four step
process:

1 - The host allocates 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 ports.

   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. There is no guarantee that
   all NAT boxes deployed on the Internet have this characteristic.
   Implementers are referred to the STUN specification [RFC3489] for an
   extensive discussion of the various types of NAT.

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.

4. UNSAF Considerations

   The RTCP attribute in SDP is used to enable establishment of RTP/RTCP
   flows through NAT. This mechanism can be used in conjunction with an
   address discovery mechanism such as STUN [RFC3489]. STUN is a short
   term fix to the NAT traversal problem, which requires thus
   consideration of the general issues linked to "Unilateral self-
   address fixing" [RFC3424].

   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 requires 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] has been registered by IANA. This
   attribute field is designed for use at media level only.

7. Intellectual Property Statement

   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.

8. Acknowledgements

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

9. References

9.1. Normative References

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

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

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

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

   [RFC3489] Rosenberg, J., Weinberger, J., Huitema, C. and R. Mahy.
              "STUN - Simple Traversal of User Datagram Protocol (UDP)
              Through Network Address Translators (NATs)", RFC 3489,
              March 2003.

   [RFC3550] Schulzrinne, H., Casner, S., Frederick, R. and V.
              Jacobson. "RTP: A Transport Protocol for Real-Time
              Applications", RFC 3550, July 2003.

9.2. Informative References

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

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

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

   [RFC3424] Daigle, L., "IAB considerations for UNilateral Self-
              Address Fixing (UNSAF) across network address
              translation", RFC 3424, November 2002.

10. Author's Address

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

   EMail: huitema@microsoft.com

11. Full Copyright Statement

Acknowledgement