RTCP attribute in SDP

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, [RFC2543]) 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 is hidden behind a network address translation device [RFC2766]. In this cases, 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 [RFC2237] 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 port, 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 order to successfully establish connections despites 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 we also propose to make the behavior of RTP implementations more conform to the robustness principle.

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, encoded as 1 to 5 digits.

An example encoding could be:

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

The RTCP attribute may only 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.

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 port 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, using a three step process:

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

2) The host send a UDP message from each port to the third party,

3) The third party reads the source address and port of the packet, and copy them in the text of a reply,

4) The host parses the reply 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 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."

4 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.

5 IANA Considerations

This document does not call for an IANA action, unless the IANA is registering attributes types for SDP - and there is no documentation to that effect on the IANA web site.

6 Copyright

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

7 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.

8 Acknowledgements

The original idea for using the "rtcp" attribute was developed by Ann Demirtjis.

9 References

[RFC2543] M. Handley, H. Schulzrinne, E. Schooler, J. Rosenberg "SIP: Session Initiation Protocol", RFC 2543, March 1999.

[RFC2237] M. Handley, V. Jacobson, "SDP: Session Description Protocol", RFC 2327, April 1998.

[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.

10 Author's Addresses

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

Email: huitema@exchange.microsoft.com