Network Working Group
Internet-Draft
Updates: 3550 (if approved)
Intended status: Standards Track
Expires: April 23, 2007
October 20, 2006

C. Perkins
University of Glasgow
M. Westerlund
Ericsson

Multiplexing RTP Data and Control Packets on a Single Port
draft-ietf-avt-rtp-and-rtcp-mux-01.txt

Status of this Memo

Copyright Notice

Abstract

This memo discusses issues that arise when multiplexing RTP data packets and RTP control protocol (RTCP) packets on a single UDP port. It updates RFC 3550 to describe when such multiplexing is, and is not, appropriate, and explains how the Session Description Protocol (SDP) can be used to signal multiplexed sessions.

Table of Contents

1. Introduction

The Real-time Transport Protocol (RTP) [1] comprises two components: a data transfer protocol, and an associated control protocol (RTCP). Historically, RTP and RTCP have been run on separate UDP ports. With increased use of Network Address Translation (NAT) this has become problematic, since opening multiple NAT pinholes can be costly. This memo discusses how the RTP and RTCP flows for a single media type can be run on a single port, to ease NAT traversal, and considers when such multiplexing is appropriate. The multiplexing of several types of media (e.g. audio and video) onto a single port is not considered here (but see Section 5.2 of [1]).

This memo is structured as follows: in Section 2 we discuss the design choices which led to the use of separate ports, and comment on the applicability of those choices to current network environments. We discuss terminology in Section 3, how to distinguish multiplexed packets in Section 4, and then specify when and how RTP and RTCP should be multiplexed in Section 5. Quality of service and bandwidth issues are discussion in Section 6. We conclude with security considerations in Section 7.

This memo updates Section 11 of [1].

2. Background

An RTP session comprises data packets and periodic control (RTCP) packets. RTCP packets are assumed to use "the same distribution mechanism as the data packets" and the "underlying protocol MUST provide multiplexing of the data and control packets, for example using separate port numbers with UDP" [1]. Multiplexing was deferred to the underlying transport protocol, rather than being provided within RTP, for the following reasons:

1. Simplicity: an RTP implementation is simplified by moving the RTP and RTCP demultiplexing to the transport layer, since it need not concern itself with the separation of data and control packets. This allows the implementation to be structured in a very natural fashion, with a clean separation of data and control planes.

2. Efficiency: following the principle of integrated layer processing [13] an implementation will be more efficient when demultiplexing happens in a single place (e.g. according to UDP port) than when split across multiple layers of the stack (e.g. according to UDP port then according to packet type).

3. To enable third party monitors: while unicast voice-over-IP has always been considered, RTP was also designed to support loosely coupled multicast conferences [14] and very large-scale multicast streaming media applications (such as the so-called "triple-play" IPTV service). Accordingly, the design of RTP allows the RTCP packets to be multicast using a separate IP multicast group and UDP port to the data packets. This not only allows participants in a session to get reception quality feedback, but also enables deployment of third party monitors which listen to reception quality without access to the data packets. This was intended to provide manageability of multicast sessions, without compromising privacy.

While these design choices are appropriate for many use of RTP, they are problematic in some cases. There are many RTP deployments which don't use IP multicast, and with the increased use of Network Address Translation (NAT) the simplicity of multiplexing at the transport layer has become a liability, since it requires complex signalling to open multiple NAT pinholes. In environments such as these, it is desirable to provide an alternative to demultiplexing RTP and RTCP using separate UDP ports, instead using only a single UDP port and demultiplexing within the application.

This memo provides such an alternative by multiplexing RTP and RTCP packets on a single UDP port, distinguished by the RTP payload type and RTCP packet type values. This pushes some additional work onto the RTP implementation, in exchange for simplified NAT traversal.

3. Terminology

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 RFC 2119 [2].

4. Distinguishable RTP and RTCP Packets

When RTP and RTCP packets are multiplexed onto a single port, they can be distinguished provided: 1) the RTP payload type (PT) values used are distinct from the RTCP packet types used; and 2) for each RTP payload type, PT+128 is distinct from the RTCP packet types used. The first constraint precludes a direct conflict between RTP payload type and RTCP packet type, the second constraint precludes a conflict between an RTP data packet with marker bit set and an RTCP packet. This demultiplexing method works because the RTP payload type and RTCP packet type occupy the same position within the packet.

The following conflicts between RTP and RTCP packet types are known:

- o RTP payload types 64-65 conflict with the RTCP FIR and NACK packets defined in the RTP Payload Format for H.261 [3].
- o RTP payload types 72-76 conflict with the RTCP SR, RR, SDES, BYE and APP packets defined in the RTP specification [1].
- o RTP payload types 77-78 conflict with the RTCP RTPFB and PSFB packets defined in the RTP/AVPF profile [4].
- o RTP payload type 79 conflicts with RTCP Extended Report (XR) [5] packets.
- o RTP payload type 80 conflicts with Receiver Summary Information (RSI) packets defined in the RTCP Extensions for Single-Source Multicast Sessions with Unicast Feedback [6].

New RTCP packet types may be registered in future, and will further reduce the RTP payload types that are available when multiplexing RTP and RTCP onto a single port. To allow this multiplexing, future RTCP packet type assignments SHOULD be made after the current assignments in the range 209-223, then in the range 194-199, so that only the RTP payload types in the range 64-95 are blocked.

Given these constraints, it is RECOMMENDED to follow the guidelines in the RTP/AVP profile [7] for the choice of RTP payload type values, with the additional restriction that payload type values in the range 64-95 MUST NOT be used. Specifically, dynamic RTP payload types SHOULD be chosen in the range 96-127 where possible. Values below 64 MAY be used if that is insufficient, in which case it is RECOMMENDED that payload type numbers that are not statically assigned by [7] be used first.

> Note: since all RTCP packets MUST be sent as compound packets beginning with an SR or an RR packet ([1] Section 6.1), one might wonder why RTP payload types other than 72 and 73 are prohibited when multiplexing RTP and RTCP. This is done to ensure robustness against broken nodes which send non-compliant RTCP packets, which might otherwise be confused with multiplexed RTP packets.

5. Multiplexing RTP and RTCP on a Single Port

The procedures for multiplexing RTP and RTCP on a single port depend on whether the session is a unicast session or a multicast session. For a multicast sessions, also depends whether ASM or SSM multicast is to be used.

5.1. Unicast Sessions

It is acceptable to multiplex RTP and RTCP packets on a single UDP port to ease NAT traversal for unicast sessions, provided the RTP payload types used in the session are chosen according to the rules in Section 4. The following sections describe how such multiplexed sessions can be signalled using the Session Initiation Protocol (SIP) with the offer/answer model.

5.1.1. SDP Signalling

When the Session Description Protocol (SDP) [8] is used to negotiate RTP sessions according to the offer/answer model [9], the "a=rtcp:" attribute [10] is used to indicate the port chosen for RTCP traffic if the default of using an odd/even port pair is not desirable. If RTP and RTCP are to be multiplexed on a single port, this attribute MUST be included in the initial SDP offer, and MUST indicate the same port as included in the "m=" line. For example:

```
v=0
o=csp 1153134164 1153134164 IN IP6 2001:DB8::211:24ff:fea3:7a2e
s=-
c=IN IP6 2001:DB8::211:24ff:fea3:7a2e
t=1153134164 1153137764
m=audio 49170 RTP/AVP 97
a=rtpmap:97 iLBC/8000
a=rtcp:49170
```

This offer denotes a unicast voice-over-IP session using the RTP/AVP profile with iLBC coding. The answerer is requested to send both RTP and RTCP to port 49170 on IPv6 address 2001:DB8::211:24ff:fea3:7a2e.

If the answerer supports multiplexing of RTP and RTCP onto a single port it MUST include an "a=rtcp:" attribute in the answer, and the port specified in that attribute MUST be the same as that used for RTP in the "m=" line of the answer. The RTP payload types used in the answer MUST conform to the rules in Section 4.

If the answer does not contain an "a=rtcp:" attribute, the offerer MUST NOT multiplex RTP and RTCP packets on a single port. Instead, it must send and receive RTCP on a port allocated according to the usual port pair rules. This will occur when talking to a peer that does not understand the "a=rtcp:" attribute or this specification.

Answerers which support the "a=rtcp:" attribute but do not understand this memo should return an answer which does not contain an "a=rtcp:" attribute (since Section 11 of [1] prohibits such sessions unless the mechanisms described in this memo are used). It is likely that this is a poorly tested feature of older implementations, however, and implementations should be robust to unexpected behaviour.

5.1.2. Interactions with SIP forking

When using SIP with a forking proxy, it is possible that an INVITE request may result in multiple 200 (OK) responses. If RTP and RTCP multiplexing is offered in that INVITE, it is important to be aware that some answerers may support multiplexed RTP and RTCP, some not. This will require the offerer listen for RTCP on both the RTP port and the usual RTCP port, and to send RTCP on both ports, unless branches of the call that support multiplexing are re-negotiated to use separate RTP and RTCP ports.

5.1.3. Interactions with ICE

It is common to use the Interactive Connectivity Establishment (ICE) [15] methodology to establish RTP sessions in the presence of Network Address Translation (NAT) devices or other middleboxes. An RTP media stream usually comprises two components in ICE (one for RTP and one for RTCP), and connectivity checks are performed for each component. The RTCP port must always be explicitly signalled using the "a=rtcp:" attribute when using ICE.

When RTP and RTCP are to be multiplexed on a single UDP port, there is no need to perform ICE checks for the RTCP traffic. Accordingly, each RTP stream MUST be treated as comprising only a single component when multiplexing is in use (this can be detected by the port on the "a=rtcp:" line matching that on the "m=" line). Multiplexing RTP and RTCP therefore cuts the number of ICE checks that must be performed in half.

5.1.4. Interactions with Header Compression

Multiplexing RTP and RTCP packets onto a single port may negatively impact header compression schemes, for example Compressed RTP (CRTP) [16] and RObust Header Compression (ROHC) [17]. Header compression exploits patterns of change in the RTP headers of consecutive packets to send an indication that the packet changed in the expected way, rather than sending the complete header each time. This can lead to significant bandwidth savings if flows have uniform behaviour. The presence of RTCP packets multiplexed with RTP data packets disrupts these patterns, however, and can significantly reduce the gains available due to header compression.

This effect may be especially significant in those environments, such as some wireless telephony systems, which rely on the efficiency of header compression to match the media to a limited capacity channel.

The implications of multiplexing RTP and RTCP should be carefully considered before use in such environments.

5.2. Any Source Multicast Sessions

The problem of NAT traversal is less severe for any source multicast (ASM) RTP sessions than for unicast RTP sessions, and the benefit of using separate ports for RTP and RTCP is greater, due to the ability to support third party RTCP only monitors. Accordingly, RTP and RTCP packets SHOULD NOT be multiplexed onto a single port when using ASM multicast RTP sessions, and SHOULD instead use separate ports and multicast groups.

5.3. Source Specific Multicast Sessions

RTP sessions running over Source Specific Multicast (SSM) send RTCP packets from the source to receivers via the multicast channel, but use a separate unicast feedback mechanism [6] to send RTCP from the receivers back to the source, with the source either reflecting the RTCP packets to the group, or sending aggregate summary reports.

Following the terminology of [6], we identify three RTP/RTCP flows in an SSM session:

1. RTP and RTCP flows between media sender and distribution source. In many scenarios, the media sender and distribution source are collocated, so multiplexing is not a concern. If the media sender and distribution source are connected by a unicast connection, the rules in Section 5.1 of this memo apply to that connection. If the media sender and distribution source are connected by an Any Source Multicast connection, the rules in Section 5.2 apply to that connection. If the media sender and distribution source are connected by a Source Specific Multicast connection, the RTP and RTCP packets MAY be multiplexed on a single port, provided this is signalled (for example, using "a=rtcp:" with the same port number as specified for RTP on the "m=" line, if using SDP).

2. RTP and RTCP sent from the distribution source to the receivers. The distribution source MAY multiplex RTP and RTCP onto a single port to ease NAT traversal issues on the forward SSM path, since this does not hinder third party monitoring. When using SDP, the multiplexing MUST be signalled using the "a=rtcp:" attribute [10] with the same port number as specified for RTP on the "m=" line.

3. RTCP sent from receivers to distribution source. This is an RTCP only path, so multiplexing is not a concern.

Multiplexing RTP and RTCP onto a single port is more acceptable for an SSM session than for an ASM session, since SSM sessions cannot readily make use of third party reception quality monitoring devices that listen to the multicast RTCP traffic but not the data traffic (since the RTCP traffic is unicast to the distribution source, rather than multicast, and since one cannot subscribe to only the RTCP packets on the SSM channel, even if sent on a separate port).

6. Multiplexing, Bandwidth, and Quality of Service

Multiplexing RTP and RTCP has implications on the use of Quality of Service (QoS) mechanism that handles flow that are determined by a three or five tuple (protocol, port and address for source and/or destination). In these cases the RTCP flow will be merged with the RTP flow when multiplexing them together. Thus the RTCP bandwidth requirement needs to be considered when doing QoS reservations for the combined RTP and RTCP flow. However from an RTCP perspective it is beneficial to receive the same treatment of RTCP packets as for RTP as it provides more accurate statistics for the measurements performed by RTCP.

The bandwidth required for a multiplexed stream comprises the session bandwidth of the RTP stream, plus the bandwidth used by RTCP. In the usual case, the RTP session bandwidth is signalled in the SDP "b=AS:" line, and the RTCP traffic is limited to 5% of this value. Any QoS reservation SHOULD therefore be made for 105% of the "b=AS:" value. If a non-standard RTCP bandwidth fraction is used, signalled by the SDP "b=RR:" and/or "b=RS:" lines [11], then any QoS reservation SHOULD be made for bandwidth equal to (AS + RS + RR), taking the RS and RR values from the SDP answer.

7. Security Considerations

The security considerations in the RTP specification [1] and any applicable RTP profile (e.g. [7]) and payload format(s) apply.

If the Secure Real-time Transport Protocol (SRTP) [12] is to be used in conjunction with multiplexed RTP and RTCP, then multiplexing MUST be done below the SRTP layer. The sender generates SRTP and SRTCP packets in the usual manner, based on their separate cryptographic contexts, and multiplexes them onto a single port immediately before transmission. At the receiver, the cryptographic context is derived from the SSRC, destination network address and destination transport port number in the usual manner, augmented using the RTP payload type and RTCP packet type to demultiplex SRTP and SRTCP according to the rules in Section 4 of this memo. After the SRTP and SRTCP packets have been demultiplexed, cryptographic processing happens in the usual manner.

8. IANA Considerations

No IANA actions are required.

9. Acknowledgements

We wish to thank Steve Casner, Joerg Ott, Christer Holmberg, Gunnar Hellstrom, Randell Jesup, Hadriel Kaplan, Harikishan Desineni, and Stephan Wenger for their comments on this memo. This work is supported in part by the UK Engineering and Physical Sciences Research Council.

10. References

10.1. Normative References

[1] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.

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

[3] Turletti, T., "RTP Payload Format for H.261 Video Streams", RFC 2032, October 1996.

[4] Ott, J., Wenger, S., Sato, N., Burmeister, C., and J. Rey, "Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/AVPF)", RFC 4585, July 2006.

[5] Friedman, T., Caceres, R., and A. Clark, "RTP Control Protocol Extended Reports (RTCP XR)", RFC 3611, November 2003.

[6] Chesterfield, J., "RTCP Extensions for Single-Source Multicast Sessions with Unicast Feedback", draft-ietf-avt-rtcpssm-11 (work in progress), March 2006.

[7] Schulzrinne, H. and S. Casner, "RTP Profile for Audio and Video Conferences with Minimal Control", STD 65, RFC 3551, July 2003.

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

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

[10] Huitema, C., "Real Time Control Protocol (RTCP) attribute in Session Description Protocol (SDP)", RFC 3605, October 2003.

[11] Casner, S., "Session Description Protocol (SDP) Bandwidth Modifiers for RTP Control Protocol (RTCP) Bandwidth", RFC 3556, July 2003.

[12] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, March 2004.

10.2. Informative References

[13] Clark, D. and D. Tennenhouse, "Architectural Considerations for a New Generation of Protocols", Proceedings of ACM SIGCOMM 1990, September 1990.

[14] Casner, S. and S. Deering, "First IETF Internet Audiocast", ACM SIGCOMM Computer Communication Review, Volume 22, Number 3, July 1992.

[15] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Methodology for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", draft-ietf-mmusic-ice-10 (work in progress), August 2006.

[16] Casner, S. and V. Jacobson, "Compressing IP/UDP/RTP Headers for Low-Speed Serial Links", RFC 2508, February 1999.

[17] Bormann, C., Burmeister, C., Degermark, M., Fukushima, H., Hannu, H., Jonsson, L-E., Hakenberg, R., Koren, T., Le, K., Liu, Z., Martensson, A., Miyazaki, A., Svanbro, K., Wiebke, T., Yoshimura, T., and H. Zheng, "RObust Header Compression (ROHC): Framework and four profiles: RTP, UDP, ESP, and uncompressed", RFC 3095, July 2001.

Authors' Addresses

Colin Perkins
University of Glasgow
Department of Computing Science
17 Lilybank Gardens
Glasgow G12 8QQ
UK

Email: csp@csperkins.org

Magnus Westerlund
Ericsson
Torshamgatan 23
Stockholm SE-164 80
Sweden

Email: magnus.westerlund@ericsson.com

Acknowledgment

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