Internet Engineering Task Force (IETF)
Request for Comments: 5761
Updates: 3550, 3551
Category: Standards Track
ISSN: 2070-1721
April 2010

C. Perkins
University of Glasgow
M. Westerlund
Ericsson

Multiplexing RTP Data and Control Packets on a Single Port

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 and RFC 3551 to describe when such multiplexing is and is not appropriate, and it explains how the Session Description Protocol (SDP) can be used to signal multiplexed sessions.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc5761.

Copyright Notice

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 Port Translation (NAPT) [14], this has become problematic, since maintaining multiple NAT bindings can be costly. It also complicates firewall administration, since multiple ports must be opened to allow RTP traffic. 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 simplify firewall administration, 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 that 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 and how to distinguish multiplexed packets in Section 4; we then specify when and how RTP and RTCP should be multiplexed, and how to signal multiplexed sessions, in Section 5. Quality of service and bandwidth issues are discussed in Section 6. We conclude with security considerations in Section 7 and IANA considerations in Section 8.

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 [15], 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 and 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 [16] and very large-scale multicast streaming media applications (such as the so-called triple-play IP television (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 uses of RTP, they are problematic in some cases. There are many RTP deployments that 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, the RTCP packet type field occupies the same position in the packet as the combination of the RTP marker (M) bit and the RTP payload type (PT). This field can be used to distinguish RTP and RTCP packets when two restrictions are observed: 1) the RTP payload type values used are distinct from the RTCP packet types used; and 2) for each

RTP payload type (PT), 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 the marker bit set and an RTCP packet.

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

o RTP payload types 64-65 conflict with the (obsolete) RTCP FIR and NACK packets defined in the original "RTP Payload Format for H.261 Video Streams" [3] (which was obsoleted by [17]).

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 "RTCP Extensions for Single-Source Multicast Sessions with Unicast Feedback" [6].

New RTCP packet types may be registered in the 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. RTCP packet types in the ranges 1-191 and 224-254 SHOULD only be used when other values have been exhausted.

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 Section 6.1 of [1] specifies that all RTCP packets MUST be sent as compound packets beginning with a Sender Report (SR) or a Receiver Report (RR) packet, one might wonder why RTP payload types other than 72 and 73 are prohibited when multiplexing RTP and RTCP. This is done to support [18], which allows the use of non-compound RTCP packets in some circumstances.

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 multicast sessions, the procedures also depend on whether Any Source Multicast (ASM) or Source-Specific Multicast (SSM) 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, and provided that multiplexing is signalled in advance. 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 following the offer/answer model [9], the "a=rtcp-mux" attribute (see Section 8) indicates the desire to multiplex RTP and RTCP onto a single port. The initial SDP offer MUST include this attribute at the media level to request multiplexing of RTP and RTCP on a single port. 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-mux
```

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 wishes to multiplex RTP and RTCP onto a single port, it MUST include a media-level "a=rtcp-mux" attribute in 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-mux" attribute, the offerer MUST NOT multiplex RTP and RTCP packets on a single port. Instead, it should send and receive RTCP on a port allocated according to the usual port-selection rules (either the port pair, or a signalled port if the "a=rtcp:" attribute [10] is also included). This will occur when talking to a peer that does not understand the "a=rtcp-mux" attribute.

When SDP is used in a declarative manner, the presence of an "a=rtcp-mux" attribute signals that the sender will multiplex RTP and RTCP on the same port. The receiver MUST be prepared to receive RTCP packets on the RTP port, and any resource reservation needs to be made including the RTCP bandwidth.

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 to 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) [19] methodology to establish RTP sessions in the presence of Network Address Translation (NAT) devices or other middleboxes. If RTP and RTCP are sent on separate ports, the RTP media stream comprises two components in ICE (one for RTP and one for RTCP), with connectivity checks being performed for each component. If RTP and RTCP are to be multiplexed on the same port some of these connectivity checks can be avoided, reducing the overhead of ICE.

If it is desired to use both ICE and multiplexed RTP and RTCP, the initial offer MUST contain an "a=rtcp-mux" attribute to indicate that RTP and RTCP multiplexing is desired and MUST contain "a=candidate:" lines for both RTP and RTCP along with an "a=rtcp:" line indicating a fallback port for RTCP in the case that the answerer does not support RTP and RTCP multiplexing. This MUST be done for each media where RTP and RTCP multiplexing is desired.

If the answerer wishes to multiplex RTP and RTCP on a single port, it MUST generate an answer containing an "a=rtcp-mux" attribute and a single "a=candidate:" line corresponding to the RTP port (i.e., there is no candidate for RTCP) for each media where it is desired to use

RTP and RTCP multiplexing. The answerer then performs connectivity checks for that media as if the offer had contained only a single candidate for RTP. If the answerer does not want to multiplex RTP and RTCP on a single port, it MUST NOT include the "a=rtcp-mux" attribute in its answer and MUST perform connectivity checks for all offered candidates in the usual manner.

On receipt of the answer, the offerer looks for the presence of the "a=rtcp-mux" line for each media where multiplexing was offered. If this is present, then connectivity checks proceed as if only a single candidate (for RTP) were offered, and multiplexing is used once the session is established. If the "a=rtcp-mux" line is not present, the session proceeds with connectivity checks using both RTP and RTCP candidates, eventually leading to a session being established with RTP and RTCP on separate ports (as signalled by the "a=rtcp:" attribute).

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) [20] and RObust Header Compression (ROHC) [21] [22]. 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 can disrupt the patterns of change between headers and has the potential to significantly reduce header compression efficiency. The extent of this disruption depends on the header compression algorithm used and on the way flows are classified. A well-designed classifier should be able to separate RTP and RTCP multiplexed on the same port into different compression contexts, using the payload type field, such that the effect on the compression ratio is small. A classifier that assigns compression contexts based only on the IP addresses and UDP ports will not perform well. It is expected that implementations of header compression will need to be updated to efficiently support RTP and RTCP multiplexed on the same port.

This effect of multiplexing RTP and RTCP on header compression may be especially significant in those environments, such as some wireless telephony systems, that 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 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 co-located, 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 (using "a=rtcp-mux" 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, although doing so may hinder third-party monitoring devices if the session uses the simple feedback model. When using SDP, the multiplexing SHOULD be signalled using the "a=rtcp-mux" attribute.

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

Multiplexing RTP and RTCP packets on a single port in an SSM session has the potential for interactions with header compression described in Section 5.1.4.

6. Multiplexing, Bandwidth, and Quality of Service

Multiplexing RTP and RTCP has implications on the use of the 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:" (or "b=TIAS:" [11]) 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 [12], 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 usage of multiplexing RTP and RTCP is not believed to introduce any new security considerations. Known major issues are the integrity and authentication of the signalling used to set up the multiplexing as well as the integrity, authentication, and confidentiality of the actual RTP and RTCP traffic. 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) [13] 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 synchronization source (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

Following the guidelines in [8], the IANA has registered one new SDP attribute:

- o Contact name/email: authors of RFC 5761
- o Attribute name: rtcp-mux
- o Long-form attribute name: RTP and RTCP multiplexed on one port
- o Type of attribute: media level
- o Subject to charset: no

This attribute is used to signal that RTP and RTCP traffic should be multiplexed on a single port (see Section 5 of this memo). It is a property attribute, which does not take a value.

The rules for assignment of RTP RTCP Control Packet Types in the RTP Parameters registry are updated as follows. When assigning RTP RTCP Control Packet types, IANA is requested to assign unused values from the range 200-223 where possible. If that range is fully occupied, values from the range 194-199 may be used, and then values from the ranges 1-191 and 224-254. This improves header validity checking of RTCP packets compared to RTP packets or other unrelated packets. The values 0 and 255 are avoided for improved validity checking relative to random packets since all-zeros and all-ones are common values.

9. Acknowledgements

We wish to thank Steve Casner, Joerg Ott, Christer Holmberg, Gunnar Hellstrom, Randell Jesup, Hadriel Kaplan, Harikishan Desineni, Stephan Wenger, Jonathan Rosenberg, Roni Even, Ingemar Johansson, Dave Singer, Kevin Johns, and David Black for their comments on this memo. This work was 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] Ott, J., Chesterfield, J., and E. Schooler, "RTP Control Protocol (RTCP) Extensions for Single-Source Multicast Sessions with Unicast Feedback", RFC 5760, February 2010.

[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] Westerlund, M., "A Transport Independent Bandwidth Modifier for the Session Description Protocol (SDP)", RFC 3890, September 2004.

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

[13] 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

[14] Srisuresh, P. and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)", RFC 3022, January 2001.

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

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

[17] Even, R., "RTP Payload Format for H.261 Video Streams", RFC 4587, August 2006.

[18] Johansson, I. and M. Westerlund, "Support for Reduced-Size Real-Time Transport Control Protocol (RTCP): Opportunities and Consequences", RFC 5506, April 2009.

[19] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", RFC 5245, April 2010.

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

[21] 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.

[22] Sandlund, K., Pelletier, G., and L-E. Jonsson, "The RObust Header Compression (ROHC) Framework", RFC 5795, March 2010.

Authors' Addresses

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

EMail: csp@csperkins.org

Magnus Westerlund
Ericsson
Farogatan 6
Stockholm SE-164 80
Sweden

EMail: magnus.westerlund@ericsson.com