

RTP Interoperability Statement

draft-ietf-avt-rtp-interop-04.txt

Status of this memo

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

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/lid-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

Distribution of this document is unlimited.

Comments are solicited and should be addressed to the author and/or the IETF Audio/Video Transport working group's mailing list at rem-conf@es.net.

Abstract

It is required to demonstrate interoperability of RTP implementations in order to move the RTP specification to draft standard. This memo outlines those features to be tested, as the first stage of an interoperability statement.

1 Introduction

The Internet standards process [1] places a number of requirements on a standards track protocol specification. In particular, when advancing a protocol from proposed standard to draft standard it is necessary to demonstrate at least two independent and interoperable implementations, from different code bases, of all options and features of that protocol. Further, in cases where one or more options or features have not been demonstrated in at least two interoperable

implementations, the specification may advance to the draft standard level only if those options or features are removed. The Real-time Transport Protocol, RTP, was originally specified in RFC1889 as a proposed standard [2]. The revision of this specification for draft standard status is now well underway, so it has become necessary to conduct such an interoperability demonstration.

This memo describes the set of features and options of the RTP specification which need to be tested as a basis for this demonstration. Due to the nature of RTP there are necessarily two types of test described: those which directly affect the interoperability of implementations at a ``bits on the wire level'' and those which affect scalability and safety of the protocol but do not directly affect interoperability. A related memo [4] describes a testing framework which may aid with interoperability testing.

This memo is for information only and does not specify a standard of any kind.

2 Features and options required to demonstrate interoperability

In order to demonstrate interoperability it is required to produce a statement of interoperability for each feature noted below. Such a statement should note the pair of implementations tested, including version numbers, and a pass/fail statement for each feature. It is not expected that every implementation will implement every feature, but each feature needs to be demonstrated by some pair of applications.

Note that some of these tests depend on the particular profile used, or upon options in that profile. For example, it will be necessary to test audio and video applications operating under [3] separately.

1. Interoperable exchange of data packets using the basic RTP header with no header extension, padding or CSRC list.
 - o PASS: rat vs vat
 - o PASS: IP/TV vs vat/vic
2. Interoperable exchange of data packets which use padding.
 - o FAIL: need to test IP/TV against Quicktime
3. Interoperable exchange of data packets which use a header extension. There are three possibilities here: a) if both implementations use a header extension in the same manner, it should be verified that the receiver correctly receives the information contained in the extension header; b) If the sender uses a header extension and the receiver does not, it should be verified that the receiver ignores the extension; c) If neither implementation implements an extended header, this test is considered a failure.
 - o FAIL: IP/TV, rat, rtptools, rtplib should all support this, but have not been tested.

4. Interoperable exchange of data packets using the marker bit as specified in the profile.
 - o PASS: rat vs vat
 - o PASS: IP/TV vs vic
5. Interoperable exchange of data packets using the payload type field to differentiate multiple payload formats according to a profile definition.
 - o PASS: rat vs vat
 - o PASS: IP/TV vs vat/vic
6. Interoperable exchange of data packets containing a CSRC list.
 - o PASS: rat vs vat
 - o PASS: IP/TV vs vat/vic
7. Interoperable exchange of RTCP packets, which must be compound packets containing at least an initial SR or RR packet and an SDES CNAME packet. Other RTCP packet types may be included, but this is not required for this test.
 - o PASS: rat vs vat
 - o PASS: IP/TV vs vat/vic
8. Interoperable exchange of sender report packets when the receiver of the sender reports is not also a sender (ie: sender reports which only contain sender info, with no report blocks).
 - o PASS: rat vs vat
 - o PASS: IP/TV vs vat/vic
9. Interoperable exchange of sender report packets when the receiver of the sender reports is also a sender (ie: sender reports which contain one or more report blocks).
 - o PASS: rat vs vat
 - o PASS: IP/TV vs vat/vic (IP/TV never sends SR with report blocks, but does successfully receive them from vic/vat).
10. Interoperable exchange of receiver report packets.
 - o PASS: rat vs vat
 - o PASS: IP/TV vs vat/vic
11. Interoperable exchange of receiver report packets when not receiving data (ie: the empty receiver report which has to be sent first in each compound RTCP packet when no-participants are transmitting data).
 - o PASS: rat vs vat
 - o PASS: IP/TV vs vat/vic

12. Interoperable and correct choice of CNAME, according to the rules in the RTP specification and profile (applications using the audio/video profile [3] under IPv4 should typically generate a CNAME of the form 'example@10.0.0.1', or '10.0.0.1' if they are on a machine which no concept of usernames).
 - o PASS: rat vs vat
 - o PASS: IP/TV vs vat/vic
13. Interoperable exchange of source description packets containing a CNAME item.
 - o PASS: rat vs vat
 - o PASS: IP/TV vs vat/vic
14. Interoperable exchange of source description packets containing a NAME item.
 - o PASS: rat vs vat
 - o PASS: IP/TV vs vat/vic
15. Interoperable exchange of source description packets containing an EMAIL item.
 - o PASS: rat vs vat
 - o PASS: IP/TV vs vat/vic
16. Interoperable exchange of source description packets containing a PHONE item.
 - o PASS: rat vs vat
 - o PASS: IP/TV vs vat/vic
17. Interoperable exchange of source description packets containing a LOC item.
 - o PASS: rat vs vat
 - o PASS: IP/TV vs vat/vic
18. Interoperable exchange of source description packets containing a TOOL item.
 - o PASS: rat vs vat
 - o PASS: IP/TV vs vat/vic
19. Interoperable exchange of source description packets containing a NOTE item.
 - o PASS: rat vs vat
 - o PASS: IP/TV vs vat/vic

20. Interoperable exchange of source description packets containing a PRIV item.
 - o FAIL: need to test rtplib against rtpdump?
21. Interoperable exchange of BYE packets containing a single SSRC.
 - o PASS: rat vs vat
 - o PASS: IP/TV vs vat/vic
22. Interoperable exchange of BYE packets containing multiple SSRCs.
 - o FAIL: rat can send these, but vat only accepts the first SSRC
 - o FAIL: IP/TV sends only one SSRC in BYE, but should accept multiple
 - o FAIL: need to test rat-3.0.x against rtplib
23. Interoperable exchange of BYE packets containing the optional reason for leaving text.
 - o FAIL: need to test IP/TV with rtplib (IP/TV will generate these when an SSRC collision occurs).
24. Interoperable exchange of BYE packets containing the optional reason for leaving text and multiple SSRCs.
 - o FAIL: does anyone implement both?
25. Interoperable exchange of application defined RTCP packets. As with the RTP header extension this test takes two forms: if both implementations implement the same application defined packet it should be verified that those packets can be interoperably exchanged. If only one implementation uses application defined packets, it should be verified that the other implementation can receive compound RTCP packets containing an APP packet whilst ignoring the APP packet. If neither implementation implements APP packets this test is considered a failure.
 - o FAIL: a number of libraries (rtplib, rat, IP/TV Msock) support this but have not been tested. Quicktime uses APP packets and needs to be tested against one of the others.
26. Interoperable exchange of encrypted RTP packets using DES encryption in CBC mode.
 - o PASS: rat vs vat
27. Interoperable exchange of encrypted RTCP packets using DES encryption in CBC mode.
 - o PASS (sort of): rat vs vat (vat gets the padding wrong in some cases, but mostly it works).

28. Interoperable exchange of encrypted RTCP packets using DES encryption in CBC mode, when those compound RTCP packets have been split into an encrypted packet and an unencrypted packet.

- o FAIL: not tested (rtplib supports this?)

3 Features and options relating to scalability

In addition to the basic interoperability tests, RTP includes a number of features relating to scaling of the protocol to large groups. Since these features are those which have undergone the greatest change in the update of the RTP specification, it is considered important to demonstrate their correct implementation. However, since these changes do not affect the bits-on-the-wire behaviour of the protocol, it is not possible to perform a traditional interoperability test. As an alternative to such testing we require that multiple independent implementations complete the following demonstrations.

1. Demonstrate correct implementation of basic RTCP transmission rules: periodic transmission of RTCP packets at the minimum (5 second) interval and randomisation of the transmission interval.
 - o PASS: rat, IP/TV
2. Demonstrate correct implementation of the RTCP step join backoff algorithm as a receiver.
 - o FAIL: rat and rtplib support this but have not been tested
3. Demonstrate correct implementation of the RTCP step join backoff algorithm as a sender.
 - o FAIL: rat and rtplib support this but have not been tested
4. Demonstrate correct steady state scaling of the RTCP interval according to the group size.
 - o PASS: rat, IP/TV
5. Demonstrate correct steady state scaling of the RTCP interval according to the group size with compensation for the number of senders.
 - o FAIL: IP/TV works, need results for more implementations
6. Demonstrate correct implementation of the RTCP reverse reconsideration algorithm.
 - o FAIL

7. Demonstrate correct implementation of the RTCP BYE reconsideration algorithm.
 - o FAIL
8. Demonstrate correct implementation of the RTCP member timeout algorithm.
 - o FAIL
9. Demonstrate random choice of SSRC.
 - o PASS: rat, IP/TV, LiveCaster
10. Demonstrate random selection of initial RTP sequence number.
 - o PASS: rat, LiveCaster
11. Demonstrate random selection of initial RTP timestamp.
 - o PASS: rat, LiveCaster
12. Demonstrate correct implementation of the SSRC collision/loop detection algorithm.
 - o PASS: IP/TV, vat
13. Demonstrate correct generation of reception report statistics in SR/RR packets.
 - o PASS: rat, IP/TV
14. Demonstrate correct generation of the sender info block in SR packets.
 - o PASS: rat, IP/TV

4 Author's Address

Colin Perkins
USC Information Sciences Institute
4350 North Fairfax Drive
Suite 620
Arlington, VA 22203
USA

Email: csp@isi.edu

5 Acknowledgments

Thanks to Steve Casner, Jonathan Rosenberg and Bill Fenner for their helpful feedback.

6 References

- [1] S. Bradner, ``The Internet Standards Process -- Revision 3'', RFC2026, Internet Engineering Task Force, October 1996.
- [2] H. Schulzrinne, S. Casner, R. Frederick and V. Jacobson, ``RTP: A Transport Protocol to Real-Time Applications'', RFC1889, Internet Engineering Task Force, January 1996.
- [3] H. Schulzrinne, ``RTP Profile for Audio and Video Conferences with Minimal Control'', draft-ietf-avt-profile-new-08.txt, January 2000.
- [4] C. S. Perkins, J. Rosenberg and H. Schulzrinne, ``RTP Testing Strategies'', draft-ietf-avt-rtptest-03.txt, July 2000.