

AVTCORE
Internet-Draft
Intended status: Informational
Expires: September 6, 2012

J. Lennox
Vidyo
March 5, 2012

Real-Time Transport Protocol (RTP) Topology Considerations for Offer/
Answer-Initiated Sessions.
draft-lennox-avtcore-rtp-topo-offer-answer-00

Abstract

This document discusses a number of considerations related to the topologies of Real-Time Transport Protocol (RTP) sessions initiated using the Session Description (SDP) unicast Offer/Answer Model, especially as applied to source-multiplexed sessions.

The primary observation is that certain topologies cannot be created by unicast SDP Offer/Answer. Notably, it is not possible to negotiate the topology that [RFC 5117](#) calls Topo-Transport-Translator (or "relay").

As a consequence of this limitation, certain topological assumptions can safely be made for RTP sessions initiated using unicast SDP Offer/Answer; and therefore, certain optimizations to RTP are possible in such sessions. This document also describes the optimizations that these assumptions make possible.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on September 6, 2012.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the

document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	3
2.	Terminology	3
3.	RTP Topologies	3
4.	RTP Topologies and SDP Offer/Answer	5
5.	Advantages for Assuming RTCP rewriting	6
5.1.	Independent RTCP bandwidth	6
5.2.	Optimization of receiver reports	7
6.	Normative recommendations	8
7.	Limitations of media and RTCP modifying middleboxes	9
8.	Security Considerations	10
9.	IANA Considerations	10
10.	References	10
10.1.	Normative References	10
10.2.	Informative References	11
	Author's Address	11

1. Introduction

For interactive conferencing, by the far most common method of negotiating Real-Time Transport Protocol (RTP) [[RFC3550](#)] sessions is to use the Session Description Protocol [[RFC4566](#)] Offer/Answer Model [[RFC3264](#)]. In particular, conferences are typically negotiated using offer/answer unicast streams.

As discussed in [[RFC5117](#)], however, RTP sessions can be arranged in a fairly large number of topologies, more complexly than the simple dichotomy of "unicast" or "multicast" used by the Offer/Answer model. Most of the unicast-based topologies in [RFC 5117](#) can be negotiated as SDP Offer/Answer unicast streams. However, for reasons that are explained in [Section 3](#), one topology in particular cannot be negotiated using SDP Offer/Answer: Topo-Transport-Translator.

While this might initially seem to be a limitation of SDP Offer/Answer, it actually turns out that if an endpoint can assume that its RTP topologies are limited to those that can be negotiated using offer/answer, a number of RTP optimizations become possible. These are discussed in [Section 5](#), with specific recommendations in [Section 6](#).

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC 2119](#) [[RFC2119](#)] and indicate requirement levels for compliant implementations.

3. RTP Topologies

[RFC5117] discusses multi-endpoint topologies of RTP sessions in

detail. For the purposes of this document, a few topologies in particular are of interest, and will be described in detail.

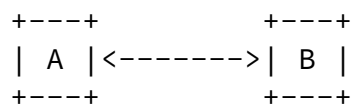


Figure 1: Point to Point

The simplest topology, shown in Figure 1, is Point to Point (Topo-

Point-to-Point). A sends to B, and only B, while B sends to A, and only A. An endpoint can still use multiple synchronization sources (SSRCs) in a session.

This topology is straightforwardly negotiated by SDP Offer-Answer unicast streams.

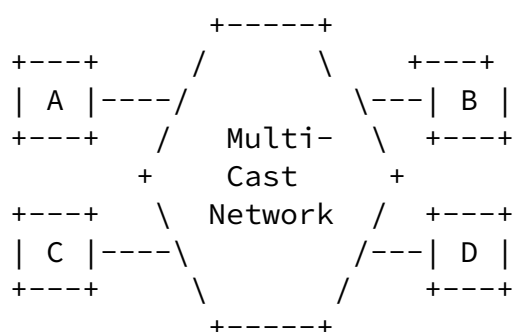


Figure 2: Point to Multipoint Using Multicast

Secondly, in the Topo-Multicast topology, shown in Figure 2, traffic from each endpoint in an RTP session is received by all other session participants, transported by the network level by sending to a special IP address. These sessions can be negotiated by the Offer/Answer model as multicast streams.

Multicast sessions are often supported within individual domains, but are not typically supported across the public Internet.

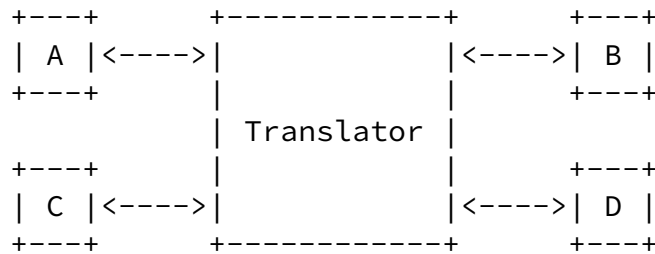


Figure 3: RTP Translator (Relay) with Only Unicast Paths

Finally, for the purposes of this discussion, one other topology described in [\[RFC5117\]](#) is specifically relevant; the others can all be generalized. The specific topology is the topology Topo-Transport-Translator, illustrated in Figure 3, which simply forwards unicast traffic (both media and Real-Time Transport Control Protocol (RTCP) traffic) among all the unicast connections to a central

translator.

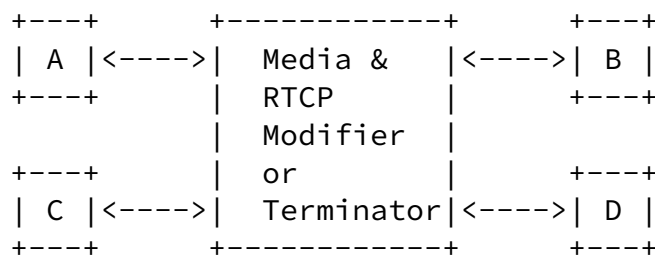


Figure 4: RTCP-Modifying Central Box

By contrast, the topologies Topo-Media-Translator, Topo-Mixer, Topo-Video-Switch-Mixer, and Topo-RTCP-Terminating-MCU can all be summarized by the diagram shown in Figure 4. These topologies all have the common property that they have a central box (a media translator, mixer, or MCU) which terminates media and RTCP traffic, and forwards it, modified to a greater or lesser extent, to the topology's other endpoints.

4. RTP Topologies and SDP Offer/Answer

The SDP Offer/Answer Model [[RFC3264](#)] specifies different behavior depending on whether the address of the transport connection being negotiated is a unicast and multicast address. Effectively, offer/answer assumes that the stream being negotiated corresponds either to Topo-Point-to-Point or Topo-Multicast.

Most of the [RFC 5117](#) unicast topologies described in [Section 3](#) can be negotiated by the centralized point negotiating with each endpoint using the Offer/Answer unicast mode. However, the Topo-Transport-Translator cannot be.

The difficulty is that SDP Offer/Answer unicast exchange is designed to negotiate each end of the exchange's separate view of the session, and each endpoint has a fair bit of control over what its view of the session should look like. However, because the translator at the center of the Topo-Transport-Translator topology forwards media and RTCP unmodified, it is necessary that all participants have a common view of all non-transport aspects of the session.

Thus, the freedom that the Offer/Answer model gives each endpoint to control its view of the session prevents the central box from enforcing a single, uniform view of it. Among the session aspects that can be different among the session participants are:

Media Types: Unicast Offer/Answer participants have the freedom to remove media types from the session description (in an answer or an updated offer). They also have the freedom to change some fmtp media type parameters. Moreover, though [RFC 3264](#) indicates that it is NOT RECOMMENDED, they have the freedom to change the mapping between media types and RTP payload type numbers.

Bandwidth: An answerer can specify a bandwidth attribute (SDP b= value) for any media stream, indicating the bandwidth that the answerer would like the offerer to use when sending media. This does bear any relationship to bandwidth values in use in the other direction. (This is somewhat problematic as SDP bandwidth parameters are used to calculate RTCP bandwidth, and thus RTP session membership timeout intervals.)

Packetization Time: An SDP offer or answer can specify theptime attribute (packetization time interval) with which it wants to

receive media. This value is independent in offers and answers.

In addition to these mismatches for the attributes specified by the core SDP Offer/Answer specification, there are of course many extensions to SDP which specify Offer/Answer behavior. These are not discussed here, but many of them would have similar issues with the Topo-Transport-Translator topology.

Any of the media and RTCP terminating topologies described in [Section 3](#) as modifying media and RTCP will be able to repair these mismatches, or else reject an endpoint that asks for a configuration beyond its capacity to repair. The mismatch difficulties arise only for the Topo-Transport-Translator.

[5.](#) Advantages for Assuming RTCP rewriting

If we assume that we always have a central box that can rewrite, or generates its own, media and RTCP, a number of optimizations and protocol clarifications become possible.

[5.1.](#) Independent RTCP bandwidth

SDP Unicast Offer/answer allows RTP session bandwidth to be specified independently in each direction of the offer/answer exchange. The assumption is that bandwidth in each direction is (over the relevant bottleneck links) non-rival, and that the available bitrates can in some circumstances be dramatically asymmetric.

It has always been somewhat unclear how offer/answer asymmetric bandwidths interact with the RTCP bandwidth fraction (5%, or the SDP bandwidth modifiers).

If we assume that RTCP is never passively relayed, but rather will always either be consumed locally or will actively be rewritten before being forwarded, this problem largely goes away. Each side of the unicast RTP session domain gets the appropriate fraction of its (sending) RTP bandwidth to send RTCP. It can divide this fraction among its sources as it wishes, subject to the constraint that a regular report is sent for each source with appropriate frequency to prevent timeouts. Group size estimation is only needed for timeout

calculation. It can be done independently for sending and receiving media.

Since RTCP bandwidth can be shared among all the sources, a sender can then also send feedback from multiple of its sources in a single compound RTCP packet, up to transport MTU issues, reducing transport overhead.

[5.2.](#) Optimization of receiver reports

For the benefit of Topo-Multicast and Topo-Transport-Translator, in an RTCP session, all session participants send RTCP reception reports (in SR or RR RTCP packets) for every active RTP source from which they have received packets in the previous reporting interval.

This means that the number of reception reports is quadratic in the number of sources in a conference. (Specifically, the number equals the number of conference participants, times the number of active senders whos sent during a report interval. However, because the report interval itself scales with the number of sources, this will in a many-to-many conference converge to being quadratic in the number of sources.)

In cases where there is an media-and-RTCP-modifying middlebox, this quadratic behavior is useless. The relevant reception report information is that between and endpoint and the middlebox, since the middlebox can often perform reliability and repair mechanisms on its own. These excess reception reports then increase the size of RTCP packets, which by the formulas for calculating RTCP packet transmission schedules reduces the RTCP timing interval. Thus, these excess reception reports consume bandwidth which could instead be used for timely RTCP feedback of relevant data.

These quadratic reception reports are particularly useless in scenarios where a given session participant is sending multiple sources of its own (rather than forwarding multiple remote sources) in the same RTP session. Examples of such use cases are the CLUE Telepresence model [[I-D.lennox-clue-rtp-usage](#)], bundling of multiple media types onto a single RTP session [[I-D.ietf-mmusic-sdp-bundle-negotiation](#)], and single-session RTP

scenarios in which SDP source descriptions [[RFC5576](#)] are used.

The most useless data is reception reports by one local source about another, since these will always (by definition) be "received" perfectly (with zero loss and jitter) by their sender.

Nearly as useless redundant feedback from multiple co-located sources about the same remote source. Since RTP traffic is in fact received by an endpoint, not a source, this information will either be identical (if an endpoint chooses to synchronize its RTCP feedback messages) or multiple, non-commensurate transmissions of the same information (if it does not).

Also often useless is feedback by one remote source about another one -- while there are some conceivable use cases where this could be relevant information (for instance, a monitoring application), in most conferencing models, this is uninteresting and unimportant.

[6.](#) Normative recommendations

Based on the analysis in [Section 5](#), this section makes some normative recommendations for the behavior of RTP endpoints in sessions negotiated using unicast SDP Offer/Answer.

(Open issue: it is possible that these recommendations might need to be a normative update to [[RFC3550](#)]; alternatively, they may just be implementation guidance.)

When an RTP [[RFC3550](#)] session is negotiated using unicast SDP offer/answer [[RFC3264](#)], RTCP bandwidth, and thus RTCP packet intervals and RTP group membership timeout rules, MUST be calculated separately for the receiving and sending direction, using the rules specified in [[RFC3550](#)] as modified by any SDP attributes or the RTP profile in use. An endpoint MAY send RTCP up to its available bandwidth, independent of the bandwidth consumed in the reverse direction, again subject to the SDP modifiers and profiles in use.

An endpoint MAY choose to send multiple sources' RTCP messages in a single compound RTCP packet (though such compound packets SHOULD NOT exceed the path MTU, if avoidable and if it is known). This will reduce the average compound RTCP packet size, and thus increase the frequency with which RTCP messages can be sent. Regular (non-feedback) RTCP compound packets MUST still begin with an SR or RR packet, but otherwise MAY contain RTCP packets in any order. Receivers MUST be prepared to receive such compound packets.

An endpoint SHOULD NOT send reception reports from one of its own sources about another one ("cross-reports"). Endpoints receiving reception reports MUST be prepared that their peers might not be sending reception reports about their own sources.

Similarly, an endpoint sending multiple sources SHOULD NOT send reception reports about a remote source from more than one of its local sources. Instead, it SHOULD create or pick one local source as the "reporting" source for each remote source, which sends full report blocks; all its other sources SHOULD be treated as if they were disconnected, and never saw that remote source. This reporting source MAY be one of the sending sources in the session, or MAY be a receive-only source created simply for the purpose of sending feedback. An endpoint MAY choose different local sources as the reporting source for different remote sources (for example, if it is using bundle [[I-D.ietf-mmusic-sdp-bundle-negotiation](#)], it could choose to send reports about remote audio sources from its local audio source, and reports about remote video sources from its local video source), or it MAY choose a single local source for all its reports. If the reporting source leaves the session (sends BYE), another reporting source MUST be chosen. If the AVPF [[RFC4585](#)] RTP profile, or one of its secure equivalents, is in use, this "reporting" source SHOULD also be the source for any AVPF feedback messages about its remote sources, as well. Endpoints interpreting reception reports MUST be prepared to receive RTCP SR or RR messages where only one remote source is reporting about its sources.

[7.](#) Limitations of media and RTCP modifying middleboxes

There are a few limitations of media and RTCP modifying middleboxes, compared to what can be done by the Topo-Transport-Translator topology.

A media and RTCP modifying middlebox will, necessarily, be more complex (and thus be more expensive, or have lower capacity), than a pure transport forwarder.

It is not possible to deploy new RTCP extensions across an unmodified RTCP-modifying central box, as that box will not know how to re-write these extensions so they are correctly forwarded.

If SRTP is in use, these central middleboxes must be trusted with the SRTP keying material. (Since SRTP keying material is usually negotiated hop-by-hop, they may be doing a complete SRTP decryption and re-encryption, with unrelated keys, and possibly even translating

between different ciphers or cipher strengths.)

It is possible, if the recommendations of [Section 6](#) are in use, that a naive RTCP monitor might think that an RTP flow should actually be interpreted as Topo-Transport-Translator. In this case, it might think that there is a network disconnection between the non-reporting sources and the sources on which they are not reporting. However, architecturally it is very unclear if such monitors actually exist, for conferencing applications, or would care about a disconnection of this sort.

[8.](#) Security Considerations

See the security considerations of [\[RFC5117\]](#). Notably, as discussed in [Section 7](#), a centralized media and RTCP modifying box will need to terminate SRTP and SRTCP, and so must be a trusted entity.

[9.](#) IANA Considerations

This document makes no requests of IANA.

Note to the RFC Editor: please remove this section before publication.

[10.](#) References

[10.1.](#) Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", [RFC 3264](#), June 2002.
- [RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, [RFC 3550](#), July 2003.

- [RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", [RFC 4566](#), July 2006.
- [RFC4585] 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.

Lennox

Expires September 6, 2012

[Page 10]

Internet-Draft RTP Offer/Answer Topology Considerations

March 2012

[10.2.](#) Informative References

- [I-D.ietf-mmusic-sdp-bundle-negotiation]
Holmberg, C. and H. Alvestrand, "Multiplexing Negotiation Using Session Description Protocol (SDP) Port Numbers", [draft-ietf-mmusic-sdp-bundle-negotiation-00](#) (work in progress), February 2012.
- [I-D.lennox-clue-rtp-usage]
Romanow, A., Lennox, J., and P. Witty, "Real-Time Transport Protocol (RTP) Usage for Telepresence Sessions", [draft-lennox-clue-rtp-usage-02](#) (work in progress), February 2012.
- [RFC4588] Rey, J., Leon, D., Miyazaki, A., Varsa, V., and R. Hakenberg, "RTP Retransmission Payload Format", [RFC 4588](#), July 2006.
- [RFC5117] Westerlund, M. and S. Wenger, "RTP Topologies", [RFC 5117](#), January 2008.
- [RFC5576] Lennox, J., Ott, J., and T. Schierl, "Source-Specific Media Attributes in the Session Description Protocol (SDP)", [RFC 5576](#), June 2009.

Author's Address

Jonathan Lennox
Vidyo, Inc.
433 Hackensack Avenue
Seventh Floor

Hackensack, NJ 07601
US

Email: jonathan@vidyo.com

Lennox

Expires September 6, 2012

[Page 11]