Abstract

This document describes mechanisms and recommended practice for transmitting the media streams of telepresence sessions using the Real-Time Transport Protocol (RTP).

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 May 3, 2012.

Copyright Notice

Copyright (c) 2011 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. Source multiplexing - overview .............................. 3  
4. Use Cases .................................................. 4  
5. Demultiplexing .............................................. 6  
   5.1. Using the SSRC for demultiplexing ..................... 7  
   5.2. Multiplex ID ........................................... 8  
   5.3. Combined approach .................................... 9  
6. Transmission of presentation sources ....................... 9  
7. Other considerations ....................................... 10  
8. Security Considerations ..................................... 10  
9. IANA Considerations ....................................... 10  
10. References ................................................. 10  
    10.1. Normative References ................................ 10  
    10.2. Informative References .............................. 10  
Authors’ Addresses ............................................. 11
1. Introduction

Telepresence systems, of the architecture described by [I-D.ietf-clue-telepresence-use-cases] and [I-D.ietf-clue-telepresence-requirements], will send and receive multiple media streams, where the number of streams in use is potentially large and asymmetric between endpoints, and streams can come and go dynamically. These characteristics lead to a number of architectural design choices which, while still in the scope of potential architectures envisioned by the Real-Time Transport Protocol [RFC3550], must be fairly different than those typically implemented by the current generation of voice or video conferencing systems. This document makes recommendations about how streams should be encoded and transmitted in RTP for this telepresence architecture.

2. 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 [RFC2119] and indicate requirement levels for compliant implementations.

3. Source multiplexing - overview

Telepresence sessions have lots of media streams: easily dozens at a time (given, e.g., a continuous presence screen in a multi-point conference), potentially out of a possible pool of hundreds. Furthermore, endpoints will have an asymmetric number of media streams.

In such an environment the usual model of existing SIP endpoints -- sending zero or one source (in each direction) per RTP session -- doesn’t scale, and mapping asymmetric numbers of sources to sessions is needlessly complex.

Therefore, telepresence systems SHOULD use a single RTP session per media type, except where there’s a need to give sessions different transport treatment. All sources of the same media type are sent over this single RTP session. This architecture (known as "source multiplexing") was defined by [RFC3550], but was used rarely until more recently by some Telepresence systems.

Multiplexing multiple media streams in this way has additional advantages. It makes going through middle boxes considerably easier, as it allows Telepresence devices to work through SIP B2BUAs that do
not support multiple media lines of the same media type. It also simplifies NAT and firewall traversal by allowing endpoint to deal with only a single address/port mapping per media type rather than multiple mappings.

During call setup, a single RTP session is negotiated for each media type. In SDP, only one media line is negotiated per media and multiple media streams are sent over the same UDP channel negotiated using the SDP media line.

A number of protocol issues involved in multiplexing RTP streams into session are discussed in [I-D.westerlund-avtcore-multiplex-architecture] and [I-D.lennox-rtcweb-rtp-media-type-mux]. In this draft we concentrate on examining the demultiplexing of RTP streams, in the specific context of telepresence systems.

A key issue to work out is how a receiver interprets the multiple streams it receives, and correlates them with the captures it has requested. In some cases, the CLUE Framework [I-D.ietf-clue-framework]’s concept of the "capture" maps cleanly to the RTP concept of an SSRC, but in some cases it does not.

First we will consider the cases that need to be considered. We will then examine the two most obvious approaches to demultiplexing, showing their pros and cons. We then describe a third possible alternative.

4. Use Cases

There are three distinct use cases relevant for telepresence systems:

Static stream choice:

In this case, the streams sent over the multiplex are constant over the complete session. An example is a triple-camera system to MCU in which left, center and right streams are sent for the duration of the session.

This describes an endpoint to endpoint, endpoint to multipoint device, and equivalently a transcoding multipoint device to endpoint.

This is illustrated in Figure 1.
Dynamic streams from a finite set:

In this case, the receiver has requested a smaller number of streams than the number of media sources that are available, and expects the sender to switch the sources being sent based on criteria chosen by the sender. (This is called auto-switched in the CLUE Framework [I-D.ietf-clue-framework].)

An example is a triple-camera system to two-screen system, in which the sender needs to switch either LC -> LR, or CR -> LR.

This describes an endpoint to endpoint, endpoint to multipoint device, and a transcoding device to endpoint.

This is illustrated in Figure 2.

Dynamic streams from an infinite set:

This case describes a switched multipoint device to endpoint, in which the multipoint device can choose to send any streams received from any other endpoints within the conference to the endpoint.

For example, in an MCU to triple-screen system, the MCU could send e.g. LCR of a triple-camera system -> LCR, or CCC of three single-camera endpoints -> LCR.
This is illustrated in Figure 3.

Figure 3: Multipoint Infinite Streams

Within any of these cases, every stream within the multiplexed session MUST have a unique SSRC. The SSRC is chosen at random [RFC3550] to ensure uniqueness (within the conference), and contains no meaningful information.

Any source may choose to restart a stream at any time, resulting in a new SSRC. For example, a transcoding MCU might, for reasons of load balancing, transfer an encoder onto a different DSP, and throw away all context of the encoding at this state, sending an RTCP BYE message for the old SSRC, and picking a new SSRC for the stream when started on the new DSP.

Because of this possibility of changing the SSRC at any time, all our use cases can be considered to be the third and most difficult case, that of dynamic streams from an infinite set. Thus, this is the only case we will consider.

5. Demultiplexing

There are two obvious choices in order to demultiplex: the SSRC, which is guaranteed to be unique for a stream, but conveys no
intrinsic useful information, or an additional multiplex ID tagged on to media packets. There may be other choices, e.g., payload type number, which might be appropriate for multiplexing one audio with one video stream on the same RTP session, but this not relevant for the cases discussed here.

For receivers with limited decoding resources, it is particularly important to ensure that the number of streams which the receiver is expecting to receive never exceeds the maximum number it has requested. On a change of stream, the receiver can be expected to have a one-out, one-in policy, so that the decoder of the stream currently being decoded is stopped before starting the decoder for the stream replacing it. The sender should therefore indicate to the receiver which stream will be replaced upon a stream change.

5.1. Using the SSRC for demultiplexing

Using the SSRC has the advantage of being included already in each RTP packet. However, there are some disadvantages to consider. First, the SSRC needs to be linked to some metadata to associate it to the capture stream. This is because although it uniquely identifies a media stream, it does not indicate which of the requested streams each SSRC is tied to. If more than one media stream is expected, it is therefore required to send some additional metadata to indicate the link between the SSRC and the CLUE stream ID. This is simply a mapping from transmitted SSRC to stream ID, updated as new SSRCs replace old ones.

Because of the one-out, one-in codec policy, the receiver must know in advance of receiving the media stream how to allocate its decoding resources. Although it could cache incoming media received before it knows what multiplex stream it applies to, this will require an unknown amount of storage space (particularly if the metadata is lost), and could lead to significant latency, after which the receiver may not find it possible to catch up because of resource constraints, or else it would require an expensive state refresh, such as a Full Intra Request (FIR) [RFC5104].

In addition, a receiver will have to store lookup tables of SSRCs to stream IDs/decoders etc. Because of the large SSRC space (32 bits), this will have to be in the form of something like a hash map, and a lookup will have to be performed for every incoming packet, which may prove costly on the receiver side.

Consider the choices for where to put the metadata. The metadata could be sent in the CLUE messaging. The use of a reliable transport means that it can be sure that the metadata will not be lost, but if this reliability is achieved through retransmission, the time taken
for the metadata to reach all receivers (particularly in a very large scale conference, e.g., with thousands of users) could result in very poor switching times, providing a bad user experience.

A second option for sending the metadata is in RTCP, for instance as a new SDES item. This is likely to follow the same path as media, and therefore if the metadata is sent slightly in advance of the media, it can be expected to be received in advance of the media. However, because RTCP is lossy, the metadata may not be received for some time, resulting in the receiver of the media not knowing how to route the received media. A system of acks and retransmissions could mitigate this, but this results in the same high switching latency behaviour as discussed for using CLUE as a transport for the metadata.

5.2. Multiplex ID

The second option is to tag each media packet with an RTP header extension [RFC5285] carrying a multiplex ID. This means that a receiver immediately knows how to interpret received media, even when an unknown SSRC is seen. As long as the media carries a known multiplex ID, it can be assumed that this media stream will replace the stream currently being received with that multiplex ID.

This gives significant advantages to switching latency, as a switch between sources can be acheived without any form of negotiation with the receiver. There is no chance of receiving media without knowing to which switched capture it belongs.

Although multiplex IDs may be chosen by either the sender or receiver, the multiplex ID can, if chosen by the receiver, contain semantic information relevant to the receiver. For example, on a large multipoint device with many DSPs, the receiver chosen multiplex ID could identify the DSP to which the media should be sent, and possibly contain routing information to the DSP.

However, there are also significant disadvantages in using a multiplex ID. It introduces additional processing costs.

Multiplex IDs are scoped only within one hop (i.e., within a cascaded conference a multiplex ID that is used from the source to the first MCU is not meaningful between two MCUs, or between an MCU and a receiver), and so they may need to be modified at every stage.

To add or modify the multiplex ID is an expensive operation, particularly if SRTP is used to authenticate the packet. Modification to the contents of the RTP header requires a reauthentication of the complete packet, and this could prove to be a
limiting factor in the throughput of a multipoint device. However, it may be that reauthentication is required in any case due to the nature of SDP. SDP permits the receiver to choose payload types, meaning that a similar option to modify the payload type in the packet header will cause the need to reauthenticate.

5.3. Combined approach

The two major flaws of the above methods (poor switching performance of SSRC multiplexing, high computational cost on switching nodes) can be mitigated with a combined method. In this, the multiplex ID can be included in packets belonging to the first frame of media (typically an IDR/GDR), but following this only the SSRC is used to demultiplex.

Because the IDR is already required to be received before any further frames can be decoded, this does not create any further restrictions on the media stream -- existing mechanisms to ensure the reliability of an IDR frame can be used. It does introduce extra complexity on the demultiplex side, requiring a two stage process of inspecting the packet for a multiplex ID, and, if it is not present, looking for the SSRC in a table of known streams.

The solution is somewhat more complex if it is possible for a source to change which switched capture is sending it: for instance, in the second example in Section 4, when the sender switches from sending LC -> LR to sending CR -> LR, the sender’s "C" source moves from the receiver’s "R" multiplex ID to the receiver’s "L" multiplex ID. For reasons of coding efficiency, it is desirable in this case to avoid sending a new IDR frame for the "C" stream, if the receiver’s architecture allows the same decoding state to be used for its various multiplex IDs. In this case, the multiplex ID could be sent for a small number of frames after the source’s multiplex ID has changed.

6. Transmission of presentation sources

Most existing videoconferencing systems use separate RTP sessions for main and presentation video sources, distinguished by the SDP content attribute [RFC4796]. The use of [I-D.ietf-clue-framework] the CLUE telepresence framework to describe multiplexed streams can remove this need. However, it could still be useful in some cases to make the distinction between presentation and main video sources at the transport layer. In particular, if different treatment is desired at the transport layer or below (e.g. different VLANs, different QoS characteristics, etc.) for main video vs presentation, the use of multiple RTP sessions m lines with different transport addresses.
could would be necessary.

7. Other considerations

As currently defined, H.281 Far-End Camera Control [ITU.H281.1994][RFC4573] does not, in SIP-based videoconferences, support selecting among multiple remote sources (though it does in H.323 conferences controlled by an MCU, which can assign terminal IDs to sources). When RTP sessions contain multiple sources, this limitation becomes pressing. (However, this problem does not appear to be in scope of the CLUE working group.)

8. Security Considerations

The security considerations for multiplexed RTP do not seem to be different than for non-multiplexed RTP.

9. IANA Considerations

This document makes no requests of IANA.

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

10. References

10.1. Normative References


10.2. Informative References


[I-D.ietf-clue-telepresence-requirements]
Romanow, A. and S. Botzko, "Requirements for Telepresence Multi-Streams", 
draft-ietf-clue-telepresence-requirements-01 (work in progress), October 2011.

[I-D.ietf-clue-telepresence-use-cases]  

[I-D.lennox-rtcweb-rtp-media-type-mux]  

[I-D.westerlund-avtcore-multiplex-architecture]  
Westerlund, M., Burman, B., and C. Perkins, "RTP Multiplexing Architecture", draft-westerlund-avtcore-multiplex-architecture-00 (work in progress), October 2011.

[ITU.H281.1994]  

[RFC4573]  

[RFC4796]  

[RFC5104]  

[RFC5285]  
Authors’ Addresses

Jonathan Lennox
Vidyo, Inc.
433 Hackensack Avenue
Seventh Floor
Hackensack, NJ 07601
US

Email: jonathan@vidyo.com

Allyn Romanow
Cisco Systems
San Jose, CA 95134
USA

Email: allyn@cisco.com

Paul Witty
Cisco Systems
Langley, England
UK

Email: pauwitty@cisco.com