CLUE WG M. Duckworth, Ed.
Internet Draft Polycom
Intended status: Standards Track A. Pepperell
Expires: July 8, 2016 Acano
S. Wenger
Vidyo
January 8, 2016

Framework for Telepresence Multi-Streams
draft-ietf-clue-framework-25.txt

Abstract

This document defines a framework for a protocol to enable devices in a telepresence conference to interoperate. The protocol enables communication of information about multiple media streams so a sending system and receiving system can make reasonable decisions about transmitting, selecting and rendering the media streams. This protocol is used in addition to SIP signaling and SDP negotiation for setting up a telepresence session.

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 July 8, 2016.

Copyright Notice

Copyright (c) 2016 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

Current telepresence systems, though based on open standards such as RTP [RFC3550] and SIP [RFC3261], cannot easily interoperate with each other. A major factor limiting the interoperability of telepresence systems is the lack of a standardized way to describe and negotiate the use of multiple audio and video streams comprising the media flows. This document provides a framework for protocols to enable interoperability by handling multiple streams in a standardized way. The framework is intended to support the use cases described in Use Cases for Telepresence Multistreams [RFC7205] and to meet the requirements in Requirements for Telepresence Multistreams [RFC7262]. This includes cases using multiple media streams that are not necessarily telepresence.

This document occasionally refers to the term "CLUE", in capital letters. CLUE is an acronym for "ControLling mUltiple streams for tElepresence", which is the name of the IETF working group in which this document and certain companion documents have been developed. Often, CLUE-something refers to something that has been designed by the CLUE working group; for example, this document may be called the CLUE-framework.

The basic session setup for the use cases is based on SIP [RFC3261] and SDP offer/answer [RFC3264]. In addition to basic SIP & SDP offer/answer, CLUE specific signaling is required to exchange the information describing the multiple media streams. The motivation for this framework, an overview of the signaling, and information required to be exchanged is described in subsequent sections of this document. Companion documents describe the signaling details [I-D.ietf-clue-signaling] and the data model [I-D.ietf-clue-data-model-schema] and protocol [I-D.ietf-clue-protocol].

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].

3. Definitions

The terms defined below are used throughout this document and companion documents. In order to easily identify the use of a defined term, those terms are capitalized.

Advertisement: a CLUE message a Media Provider sends to a Media Consumer describing specific aspects of the content of the media, and any restrictions it has in terms of being able to provide certain Streams simultaneously.

Audio Capture: Media Capture for audio. Denoted as ACn in the examples in this document.

Capture: Same as Media Capture.

Capture Device: A device that converts physical input, such as audio, video or text, into an electrical signal, in most cases to be fed into a media encoder.

Capture Encoding: A specific encoding of a Media Capture, to be sent by a Media Provider to a Media Consumer via RTP.

Capture Scene: a structure representing a spatial region captured by one or more Capture Devices, each capturing media representing a portion of the region. The spatial region represented by a Capture Scene may correspond to a real region in physical space, such as a room. A Capture Scene includes attributes and one or more Capture Scene Views, with each view including one or more Media Captures.

Capture Scene View (CSV): a list of Media Captures of the same media type that together form one way to represent the entire Capture Scene.

CLUE-capable device: A device that supports the CLUE data channel [I-D.ietf-clue-datachannel], the CLUE protocol [I-D.ietf-clue-protocol] and the principles of CLUE negotiation, and seeks CLUE-enabled calls.

CLUE-enabled call: A call in which two CLUE-capable devices have successfully negotiated support for a CLUE data channel in SDP [RFC4566]. A CLUE-enabled call is not necessarily immediately able to send CLUE-controlled media; negotiation of the data channel and of the CLUE protocol must complete first. Calls between two CLUE-capable devices which have not yet successfully completed negotiation of support for the CLUE data channel in SDP are not considered CLUE- enabled.

Conference: used as defined in [RFC4353], A Framework for Conferencing within the Session Initiation Protocol (SIP).

Configure Message: A CLUE message a Media Consumer sends to a Media Provider specifying which content and Media Streams it wants to receive, based on the information in a corresponding Advertisement message.

Consumer: short for Media Consumer.

Encoding: short for Individual Encoding.

Encoding Group: A set of encoding parameters representing a total media encoding capability to be sub-divided across potentially multiple Individual Encodings.

Endpoint: A CLUE-capable device which is the logical point of final termination through receiving, decoding and rendering, and/or initiation through capturing, encoding, and sending of media streams. An endpoint consists of one or more physical devices which source and sink media streams, and exactly one [RFC4353] Participant (which, in turn, includes exactly one SIP User Agent). Endpoints can be anything from multiscreen/multicamera rooms to handheld devices.

Global View: A set of references to one or more Capture Scene Views of the same media type that are defined within Scenes of the same advertisement. A Global View is a suggestion from the Provider to the Consumer for one set of CSVs that provide a useful representation of all the scenes in the advertisement.

Global View List: A list of Global Views included in an Advertisement. A Global View List may include Global Views of different media types.

Individual Encoding: a set of parameters representing a way to encode a Media Capture to become a Capture Encoding.

Multipoint Control Unit (MCU): a CLUE-capable device that connects two or more endpoints together into one single multimedia conference [RFC5117]. An MCU includes an [RFC4353]-like Mixer, without the [RFC4353] requirement to send media to each participant.

Media: Any data that, after suitable encoding, can be conveyed over RTP, including audio, video or timed text.

Media Capture: a source of Media, such as from one or more Capture Devices or constructed from other Media streams.

Media Consumer: a CLUE-capable device that intends to receive Capture Encodings.

Media Provider: a CLUE-capable device that intends to send Capture Encodings.

Multiple Content Capture (MCC): A Capture that mixes and/or switches other Captures of a single type. (E.g. all audio or all video.) Particular Media Captures may or may not be present in the resultant Capture Encoding depending on time or space. Denoted as MCCn in the example cases in this document.

Plane of Interest: The spatial plane within a scene containing the most relevant subject matter.

Provider: Same as Media Provider.

Render: the process of generating a representation from media, such as displayed motion video or sound emitted from loudspeakers.

Scene: Same as Capture Scene

Simultaneous Transmission Set: a set of Media Captures that can be transmitted simultaneously from a Media Provider.

Single Media Capture: A capture which contains media from a single source capture device, e.g. an audio capture from a single microphone, a video capture from a single camera.

Spatial Relation: The arrangement in space of two objects, in contrast to relation in time or other relationships.

Stream: a Capture Encoding sent from a Media Provider to a Media Consumer via RTP [RFC3550].

Stream Characteristics: the media stream attributes commonly used in non-CLUE SIP/SDP environments (such as: media codec, bit rate, resolution, profile/level etc.) as well as CLUE specific attributes, such as the Capture ID or a spatial location.

Video Capture: Media Capture for video. Denoted as VCn in the example cases in this document.

Video Composite: A single image that is formed, normally by an RTP mixer inside an MCU, by combining visual elements from separate sources.

4. Overview and Motivation

This section provides an overview of the functional elements defined in this document to represent a telepresence or multistream system. The motivations for the framework described in this document are also provided.

Two key concepts introduced in this document are the terms "Media Provider" and "Media Consumer". A Media Provider represents the entity that sends the media and a Media Consumer represents the entity that receives the media. A Media Provider provides Media in the form of RTP packets, a Media Consumer consumes those RTP packets. Media Providers and Media Consumers can reside in

Endpoints or in Multipoint Control Units (MCUs). A Media Provider in an Endpoint is usually associated with the generation of media for Media Captures; these Media Captures are typically sourced from cameras, microphones, and the like. Similarly, the Media Consumer in an Endpoint is usually associated with renderers, such as screens and loudspeakers. In MCUs, Media Providers and Consumers can have the form of outputs and inputs, respectively, of RTP mixers, RTP translators, and similar devices. Typically, telepresence devices such as Endpoints and MCUs would perform as both Media Providers and Media Consumers, the former being concerned with those devices' transmitted media and the latter with those devices' received media. In a few circumstances, a CLUE-capable device includes only Consumer or Provider functionality, such as recorder-type Consumers or webcam-type Providers.

The motivations for the framework outlined in this document include the following:

(1) Endpoints in telepresence systems typically have multiple Media Capture and Media Render devices, e.g., multiple cameras and screens. While previous system designs were able to set up calls that would capture media using all cameras and display media on all screens, for example, there was no mechanism that could associate these Media Captures with each other in space and time, in a cross-vendor interoperable way.

(2) The mere fact that there are multiple capturing and rendering devices, each of which may be configurable in aspects such as zoom, leads to the difficulty that a variable number of such devices can be used to capture different aspects of a region. The Capture Scene concept allows for the description of multiple setups for those multiple capture devices that could represent sensible operation points of the physical capture devices in a room, chosen by the operator. A Consumer can pick and choose from those configurations based on its rendering abilities and inform the Provider about its choices. Details are provided in section 7.

(3) In some cases, physical limitations or other reasons disallow the concurrent use of a device in more than one setup. For example, the center camera in a typical three-camera conference room can set its zoom objective either to capture only the middle few seats, or all seats of a room, but not both concurrently. The Simultaneous Transmission Set concept allows a Provider to signal

such limitations. Simultaneous Transmission Sets are part of the Capture Scene description, and are discussed in section 8.

(4) Often, the devices in a room do not have the computational complexity or connectivity to deal with multiple encoding options simultaneously, even if each of these options is sensible in certain scenarios, and even if the simultaneous transmission is also sensible (i.e. in case of multicast media distribution to multiple endpoints).   Such constraints can be expressed by the Provider using the Encoding Group concept, described in section 9.

(5) Due to the potentially large number of RTP streams required for a Multimedia Conference involving potentially many Endpoints, each of which can have many Media Captures and media renderers, it has become common to multiplex multiple RTP streams onto the same transport address, so to avoid using the port number as a multiplexing point and the associated shortcomings such as NAT/firewall traversal. The large number of possible permutations of sensible options a Media Provider can make available to a Media Consumer makes a mechanism desirable that allows it to narrow down the number of possible options that a SIP offer/answer exchange has to consider. Such information is made available using protocol mechanisms specified in this document and companion documents. The Media Provider and Media Consumer may use information in CLUE messages to reduce the complexity of SIP offer/answer messages. Also, there are aspects of the control of both Endpoints and MCUs that dynamically change during the progress of a call, such as audio-level based screen switching, layout changes, and so on, which need to be conveyed. Note that these control aspects are complementary to those specified in traditional SIP based conference management such as BFCP. An exemplary call flow can be found in section 5.

Finally, all this information needs to be conveyed, and the notion of support for it needs to be established. This is done by the negotiation of a "CLUE channel", a data channel negotiated early during the initiation of a call. An Endpoint or MCU that rejects the establishment of this data channel, by definition, does not support CLUE based mechanisms, whereas an Endpoint or MCU that accepts it is indicating support for CLUE as specified in this document and its companion documents.

5. Description of the Framework/Model

The CLUE framework specifies how multiple media streams are to be handled in a telepresence conference.

A Media Provider (transmitting Endpoint or MCU) describes specific aspects of the content of the media and the media stream encodings it can send in an Advertisement; and the Media Consumer responds to the Media Provider by specifying which content and media streams it wants to receive in a Configure message. The Provider then transmits the asked-for content in the specified streams.

This Advertisement and Configure typically occur during call initiation, after CLUE has been enabled in a call, but MAY also happen at any time throughout the call, whenever there is a change in what the Consumer wants to receive or (perhaps less common) the Provider can send.

An Endpoint or MCU typically act as both Provider and Consumer at the same time, sending Advertisements and sending Configurations in response to receiving Advertisements. (It is possible to be just one or the other.)

The data model [I-D.ietf-clue-data-model-schema]is based around two main concepts: a Capture and an Encoding. A Media Capture (MC), such as of type audio or video, has attributes to describe the content a Provider can send. Media Captures are described in terms of CLUE-defined attributes, such as spatial relationships and purpose of the capture. Providers tell Consumers which Media Captures they can provide, described in terms of the Media Capture attributes.

A Provider organizes its Media Captures into one or more Capture Scenes, each representing a spatial region, such as a room. A Consumer chooses which Media Captures it wants to receive from the Capture Scenes.

In addition, the Provider can send the Consumer a description of the Individual Encodings it can send in terms of identifiers which relate to items in SDP [RFC4566].

The Provider can also specify constraints on its ability to provide Media, and a sensible design choice for a Consumer is to take these into account when choosing the content and Capture Encodings it requests in the later offer/answer exchange. Some constraints are due to the physical limitations of devices--for example, a camera may not be able to provide zoom and non-zoom views simultaneously. Other constraints are system based, such as maximum bandwidth.

The following diagram illustrates the information contained in an Advertisement.

```
   ...................................................................
   .  Provider Advertisement             +--------------------+      .
   .                                     | Simultaneous Sets  |      .
   .        +------------------------+   +--------------------+      .
   .        |       Capture Scene N  |   +--------------------+      .
   .      +-+----------------------+ |   | Global View List   |      .
   .      |       Capture Scene 2  | |   +--------------------+      .
   .    +-+----------------------+ | |      +----------------------+ .
   .    |  Capture Scene 1       | | |      |  Encoding Group N    | .
   .    |    +---------------+   | | |    +-+--------------------+ | .
   .    |    | Attributes    |   | | |    |   Encoding Group 2   | | .
   .    |    +---------------+   | | |  +-+--------------------+ | | .
   .    |                        | | |  |   Encoding Group 1   | | | .
   .    |    +----------------+  | | |  |     parameters       | | | .
   .    |    |  V i e w s     |  | | |  |      bandwidth       | | | .
   .    |    |  +---------+   |  | | |  | +-------------------+| | | .
   .    |    |  |Attribute|   |  | | |  | | V i d e o         || | | .
   .    |    |  +---------+   |  | | |  | | E n c o d i n g s || | | .
   .    |    |                |  | | |  | | Encoding 1        || | | .
   .    |    | View 1         |  | | |  | |                   || | | .
   .    |    |  (list of MCs) |  | |-+  | +-------------------+| | | .
   .    |    +----|-|--|------+  |-+    |                      | | | .
   .    +---------|-|--|---------+      | +-------------------+| | | .
   .              | |  |                | | A u d i o         || | | .
   .              | |  |                | | E n c o d i n g s || | | .
   .              v |  |                | | Encoding 1        || | | .
   .      +---------|--|--------+       | |                   || | | .
   .      | Media Capture N     |------>| +-------------------+| | | .
   .    +-+---------v--|------+ |       |                      | | | .
   .    | Media Capture 2     | |       |                      | |-+ .
   .  +-+--------------v----+ |-------->|                      | |   .
   .  | Media Capture  1    | | |       |                      |-+   .
   .  |  +----------------+ |---------->|                      |     .
   .  |  | Attributes     | | |_+       +----------------------+     .
   .  |  +----------------+ |_+                                      .
   .  +--------------------+                                        .
   .                                                                 .
   ...................................................................
```

Figure 1:   Advertisement Structure

A very brief outline of the call flow used by a simple system (two Endpoints) in compliance with this document can be described as follows, and as shown in the following figure.

```
         +-----------+                     +-----------+
         | Endpoint1 |                     | Endpoint2 |
         +----+------+                     +-----+-----+
              | INVITE (BASIC SDP+CLUECHANNEL)   |
              |--------------------------------->|
              |    200 0K (BASIC SDP+CLUECHANNEL)|
              |<---------------------------------|
              | ACK                              |
              |--------------------------------->|
              |                                  |
              |<################################>|
              |       BASIC MEDIA SESSION        |
              |<################################>|
              |                                  |
              |    CONNECT (CLUE CTRL CHANNEL)   |
              |================================>|
              |            ...                   |
              |<================================>|
              |   CLUE CTRL CHANNEL ESTABLISHED  |
              |<================================>|
              |                                  |
              | ADVERTISEMENT 1                  |
              |*********************************>|
              |                  ADVERTISEMENT 2 |
              |<*********************************|
              |                                  |
              |                      CONFIGURE 1 |
              |<*********************************|
              | CONFIGURE 2                      |
              |*********************************>|
              |                                  |
              | REINVITE (UPDATED SDP)           |
              |--------------------------------->|
              |              200 0K (UPDATED SDP)|
              |<---------------------------------|
              | ACK                              |
              |--------------------------------->|
              |                                  |
              |<################################>|
              |     UPDATED MEDIA SESSION        |
              |<################################>|
              |                                  |
              v                                  v
```

Figure 2:   Basic Information Flow

An initial offer/answer exchange establishes a basic media session, for example audio-only, and a CLUE channel between two Endpoints. With the establishment of that channel, the endpoints have consented to use the CLUE protocol mechanisms and, therefore, MUST adhere to the CLUE protocol suite as outlined herein.

Over this CLUE channel, the Provider in each Endpoint conveys its characteristics and capabilities by sending an Advertisement as specified herein. The Advertisement is typically not sufficient to set up all media. The Consumer in the Endpoint receives the information provided by the Provider, and can use it for several purposes. It uses it, along with information from an offer/answer exchange, to construct a CLUE Configure message to tell the Provider what the Consumer wishes to receive. Also, the Consumer may use the information provided to tailor the SDP it is going to send during any following SIP offer/answer exchange, and its reaction to SDP it receives in that step. It is often a sensible implementation choice to do so. Spatial relationships associated with the Media can be included in the Advertisement, and it is often sensible for the Media Consumer to take those spatial relationships into account when tailoring the SDP. The Consumer can also limit the number of encodings it must set up resources to receive, and not waste resources on unwanted encodings, because it has the Provider’s Advertisement information ahead of time to determine what it really wants to receive. The Consumer can also use the Advertisement information for local rendering decisions.

This initial CLUE exchange is followed by an SDP offer/answer exchange that not only establishes those aspects of the media that have not been "negotiated" over CLUE, but has also the effect of setting up the media transmission itself, involving potentially security exchanges, ICE, and whatnot. This step is plain vanilla SIP.

During the lifetime of a call, further exchanges MAY occur over the CLUE channel. In some cases, those further exchanges lead to a modified system behavior of Provider or Consumer (or both) without any other protocol activity such as further offer/answer exchanges. For example, a Configure Message requesting the Provider to place a different Capture source into a Capture Encoding, signaled over the CLUE channel, ought not to lead to heavy-handed mechanisms like SIP re-invites. However, in other cases, after the CLUE negotiation an additional offer/answer exchange becomes necessary. For example, if both sides decide to upgrade the call from a single screen to a multi-screen call and more bandwidth is required for the additional video channels compared to what was previously negotiated using offer/answer, a new O/A exchange is required.

One aspect of the protocol outlined herein and specified in more detail in companion documents is that it makes available, to the Consumer, information regarding the Provider’s capabilities to deliver Media, and attributes related to that Media such as their spatial relationship. The operation of the renderer inside the Consumer is unspecified in that it can choose to ignore some information provided by the Provider, and/or not render media streams available from the Provider (although the Consumer follows the CLUE protocol and, therefore, gracefully receives and responds to the Provider’s information using a Configure operation).

A CLUE-capable device interoperates with a device that does not support CLUE. The CLUE-capable device can determine, by the result of the initial offer/answer exchange, if the other device supports and wishes to use CLUE. The specific mechanism for this is described in [I-D.ietf-clue-signaling]. If the other device does not use CLUE, then the CLUE-capable device falls back to behavior that does not require CLUE.

As for the media, Provider and Consumer have an end-to-end communication relationship with respect to (RTP transported) media; and the mechanisms described herein and in companion documents do not change the aspects of setting up those RTP flows and sessions. In other words, the RTP media sessions conform to the negotiated SDP whether or not CLUE is used.

6. Spatial Relationships

In order for a Consumer to perform a proper rendering, it is often necessary or at least helpful for the Consumer to have received spatial information about the streams it is receiving. CLUE defines a coordinate system that allows Media Providers to describe the spatial relationships of their Media Captures to enable proper scaling and spatially sensible rendering of their streams. The coordinate system is based on a few principles:

- Each Capture Scene has a distinct coordinate system, unrelated to the coordinate systems of other scenes.

- Simple systems which do not have multiple Media Captures to associate spatially need not use the coordinate model, although it can still be useful to provide an Area of Capture.

- Coordinates can be either in real, physical units (millimeters), have an unknown scale or have no physical scale. Systems which know their physical dimensions (for example professionally installed Telepresence room systems) MUST provide those real-world measurements to enable the best user experience for advanced receiving systems that can utilize this information. Systems which don't know specific physical dimensions but still know relative distances MUST use 'unknown scale'. 'No scale' is intended to be used only where Media Captures from different devices (with potentially different scales) will be forwarded alongside one another (e.g. in the case of an MCU).

  * "Millimeters" means the scale is in millimeters.

  * "Unknown" means the scale is not necessarily millimeters, but the scale is the same for every Capture in the Capture Scene.

  * "No Scale" means the scale could be different for each capture- an MCU Provider that advertises two adjacent captures and picks sources (which can change quickly) from different endpoints might use this value; the scale could be different and changing for each capture. But the areas of capture still represent a spatial relation between captures.

- The coordinate system is right-handed Cartesian X, Y, Z with the origin at a spatial location of the Provider's choosing. The Provider MUST use the same coordinate system with the same scale and origin for all coordinates within the same Capture Scene.

The direction of increasing coordinate values is:
X increases from left to right, from the point of view of an observer at the front of the room looking toward the back
Y increases from the front of the room to the back of the room
Z increases from low to high (i.e. floor to ceiling)

Cameras in a scene typically point in the direction of increasing Y, from front to back. But there could be multiple cameras pointing in different directions. If the physical space does not have a well-defined front and back, the provider chooses any direction for X and Y and Z consistent with right-handed coordinates.

7. Media Captures and Capture Scenes

This section describes how Providers can describe the content of media to Consumers.

7.1. Media Captures

Media Captures are the fundamental representations of streams that a device can transmit. What a Media Capture actually represents is flexible:

- It can represent the immediate output of a physical source (e.g. camera, microphone) or 'synthetic' source (e.g. laptop computer, DVD player)

- It can represent the output of an audio mixer or video composer

- It can represent a concept such as 'the loudest speaker'

- It can represent a conceptual position such as 'the leftmost stream'

To identify and distinguish between multiple Capture instances Captures have a unique identity. For instance: VC1, VC2 and AC1, AC2, where VC1 and VC2 refer to two different video captures and AC1 and AC2 refer to two different audio captures.

Some key points about Media Captures:

- A Media Capture is of a single media type (e.g. audio or video)
- A Media Capture is defined in a Capture Scene and is given an Advertisement unique identity. The identity may be referenced outside the Capture Scene that defines it through a Multiple Content Capture (MCC)
- A Media Capture may be associated with one or more Capture Scene Views
- A Media Capture has exactly one set of spatial information
- A Media Capture can be the source of at most one Capture Encoding

Each Media Capture can be associated with attributes to describe what it represents.

7.1.1. Media Capture Attributes

Media Capture Attributes describe information about the Captures. A Provider can use the Media Capture Attributes to describe the Captures for the benefit of the Consumer of the Advertisement message. All these attributes are optional. Media Capture Attributes include:

- Spatial information, such as point of capture, point on line of capture, and area of capture, all of which, in combination define the capture field of, for example, a camera
- Other descriptive information to help the Consumer choose between captures (e.g. description, presentation, view, priority, language, person information and type)

The sub-sections below define the Capture attributes.

7.1.1.1. Point of Capture

The Point of Capture attribute is a field with a single Cartesian (X, Y, Z) point value which describes the spatial location of the capturing device (such as camera). For an Audio Capture with multiple microphones, the Point of Capture defines the nominal mid-point of the microphones.

7.1.1.2. Point on Line of Capture

The Point on Line of Capture attribute is a field with a single Cartesian (X, Y, Z) point value which describes a position in space of a second point on the axis of the capturing device, toward the direction it is pointing; the first point being the Point of Capture (see above).

Together, the Point of Capture and Point on Line of Capture define the direction and axis of the capturing device, for example the optical axis of a camera or the axis of a microphone. The Media Consumer can use this information to adjust how it renders the received media if it so chooses.

For an Audio Capture, the Media Consumer can use this information along with the Audio Capture Sensitivity Pattern to define a 3-dimensional volume of capture where sounds can be expected to be picked up by the microphone providing this specific audio capture. If the Consumer wants to associate an Audio Capture with a Video Capture, it can compare this volume with the area of capture for video media to provide a check on whether the audio capture is indeed spatially associated with the video capture. For example, a video area of capture that fails to intersect at all with the audio volume of capture, or is at such a long radial distance from the microphone point of capture that the audio level would be very low, would be inappropriate.

7.1.1.3. Area of Capture

The Area of Capture is a field with a set of four (X, Y, Z) points as a value which describes the spatial location of what is being "captured". This attribute applies only to video captures, not other types of media. By comparing the Area of Capture for different Video Captures within the same Capture Scene a Consumer can determine the spatial relationships between them and render them correctly.

The four points MUST be co-planar, forming a quadrilateral, which defines the Plane of Interest for the particular Media Capture.

If the Area of Capture is not specified, it means the Video Capture might be spatially related to other Captures in the same Scene, but there is no detailed information on the relationship.For a switched Capture that switches between different sections within a larger area, the area of capture MUST use coordinates for the larger potential area.

7.1.1.4. Mobility of Capture

The Mobility of Capture attribute indicates whether or not the point of capture, line on point of capture, and area of capture values stay the same over time, or are expected to change (potentially frequently). Possible values are static, dynamic, and highly dynamic.

An example for "dynamic" is a camera mounted on a stand which is occasionally hand-carried and placed at different positions in order to provide the best angle to capture a work task. A camera worn by a person who moves around the room is an example for "highly dynamic". In either case, the effect is that the capture point, capture axis and area of capture change with time.

The capture point of a static Capture MUST NOT move for the life of the CLUE session. The capture point of dynamic Captures is categorized by a change in position followed by a reasonable period of stability--in the order of magnitude of minutes. Highly dynamic captures are categorized by a capture point that is constantly moving. If the "area of capture", "capture point" and "line of capture" attributes are included with dynamic or highly dynamic Captures they indicate spatial information at the time of the Advertisement.

7.1.1.5. Audio Capture Sensitivity Pattern

The Audio Capture Sensitivity Pattern attribute applies only to audio captures. This attribute gives information about the nominal sensitivity pattern of the microphone which is the source of the Capture. Possible values include patterns such as omni, shotgun, cardioid, hyper-cardioid.

7.1.1.6. Description

The Description attribute is a human-readable description (which could be in multiple languages) of the Capture.

7.1.1.7. Presentation

The Presentation attribute indicates that the capture originates from a presentation device, that is one that provides supplementary information to a conference through slides, video, still images, data etc. Where more information is known about the capture it MAY be expanded hierarchically to indicate the different types of presentation media, e.g. presentation.slides, presentation.image etc.

Note: It is expected that a number of keywords will be defined that provide more detail on the type of presentation. Refer to [I-D.ietf-clue-data-model-schema] for how to extend the model.

7.1.1.8. View

The View attribute is a field with enumerated values, indicating what type of view the Capture relates to. The Consumer can use this information to help choose which Media Captures it wishes to receive. Possible values are:

Room - Captures the entire scene

Table - Captures the conference table with seated people

Individual - Captures an individual person

Lectern - Captures the region of the lectern including the presenter, for example in a classroom style conference room

Audience - Captures a region showing the audience in a classroom style conference room

7.1.1.9. Language

The Language attribute indicates one or more languages used in the content of the Media Capture. Captures MAY be offered in different languages in case of multilingual and/or accessible conferences. A Consumer can use this attribute to differentiate between them and pick the appropriate one.

Note that the Language attribute is defined and meaningful both for audio and video captures. In case of audio captures, the meaning is obvious. For a video capture, "Language" could, for example, be sign interpretation or text.

The Language attribute is coded per [RFC5646].

7.1.1.10. Person Information

The Person Information attribute allows a Provider to provide specific information regarding the people in a Capture (regardless of whether or not the capture has a Presentation attribute). The Provider may gather the information automatically or manually from a variety of sources however the xCard [RFC6351] format is used to convey the information. This allows various information such as Identification information (section 6.2/[RFC6350]), Communication Information (section 6.4/[RFC6350]) and Organizational information (section 6.6/[RFC6350]) to be communicated. A Consumer may then automatically (i.e. via a policy) or manually select Captures based on information about who is in a Capture. It also allows a Consumer to render information regarding the people participating in the conference or to use it for further processing.

The Provider may supply a minimal set of information or a larger set of information. However it MUST be compliant to [RFC6350] and supply a "VERSION" and "FN" property. A Provider may supply multiple xCards per Capture of any KIND (section 6.1.4/[RFC6350]).

In order to keep CLUE messages compact the Provider SHOULD use a URI to point to any LOGO, PHOTO or SOUND contained in the xCARD rather than transmitting the LOGO, PHOTO or SOUND data in a CLUE message.

7.1.1.11. Person Type

The Person Type attribute indicates the type of people contained in the capture with respect to the meeting agenda (regardless of whether or not the capture has a Presentation attribute). As a capture may include multiple people the attribute may contain multiple values. However values MUST NOT be repeated within the attribute.

An Advertiser associates the person type with an individual capture when it knows that a particular type is in the capture. If an Advertiser cannot link a particular type with some certainty to a capture then it is not included. A Consumer on reception of a capture with a person type attribute knows with some certainly that the capture contains that person type. The capture may contain other person types but the Advertiser has not been able to determine that this is the case.

The types of Captured people include:

- Chair - the person responsible for running the meeting according to the agenda.
- Vice-Chair - the person responsible for assisting the chair in running the meeting.
- Minute Taker - the person responsible for recording the minutes of the meeting.
- Attendee - the person has no particular responsibilities with respect to running the meeting.
- Observer - an Attendee without the right to influence the discussion.
- Presenter - the person is scheduled on the agenda to make a presentation in the meeting. Note: This is not related to any "active speaker" functionality.
- Translator - the person is providing some form of translation or commentary in the meeting.
- Timekeeper - the person is responsible for maintaining the meeting schedule.

Furthermore the person type attribute may contain one or more strings allowing the Provider to indicate custom meeting specific types.

7.1.1.12. Priority

The Priority attribute indicates a relative priority between different Media Captures. The Provider sets this priority, and the Consumer MAY use the priority to help decide which Captures it wishes to receive.

The "priority" attribute is an integer which indicates a relative priority between Captures. For example it is possible to assign a priority between two presentation Captures that would allow a remote Endpoint to determine which presentation is more important. Priority is assigned at the individual Capture level. It represents the Provider’s view of the relative priority between Captures with a priority. The same priority number MAY be used across multiple Captures. It indicates they are equally important. If no priority is assigned no assumptions regarding relative importance of the Capture can be assumed.

7.1.1.13. Embedded Text

The Embedded Text attribute indicates that a Capture provides embedded textual information. For example the video Capture may contain speech to text information composed with the video image.

7.1.1.14. Related To

The Related To attribute indicates the Capture contains additional complementary information related to another Capture. The value indicates the identity of the other Capture to which this Capture is providing additional information.

For example, a conference can utilize translators or facilitators that provide an additional audio stream (i.e. a translation or description or commentary of the conference). Where multiple captures are available, it may be advantageous for a Consumer to select a complementary Capture instead of or in addition to a Capture it relates to.

7.2. Multiple Content Capture

The MCC indicates that one or more Single Media Captures are multiplexed (temporally and/or spatially) or mixed in one Media Capture. Only one Capture type (i.e. audio, video, etc.) is allowed in each MCC instance. The MCC may contain a reference to the Single Media Captures (which may have their own attributes) as well as attributes associated with the MCC itself. A MCC may also contain other MCCs. The MCC MAY reference Captures from within the Capture Scene that defines it or from other Capture Scenes. No ordering is implied by the order that Captures appear within a MCC. A MCC MAY contain no references to other Captures to indicate that the MCC contains content from multiple sources but no information regarding those sources is given. MCCs either contain the referenced Captures and no others, or have no referenced captures and therefore may contain any Capture.

One or more MCCs may also be specified in a CSV. This allows an Advertiser to indicate that several MCC captures are used to represent a capture scene. Table 14 provides an example of this case.

As outlined in section 7.1. each instance of the MCC has its own Capture identity i.e. MCC1. It allows all the individual captures contained in the MCC to be referenced by a single MCC identity.

The example below shows the use of a Multiple Content Capture:

Capture Scene #1	
VC1	{MC attributes}
VC2	{MC attributes}
VC3	{MC attributes}
MCC1(VC1,VC2,VC3)	{MC and MCC attributes}
CSV(MCC1)	

Table 1: Multiple Content Capture concept

This indicates that MCC1 is a single capture that contains the Captures VC1, VC2 and VC3 according to any MCC1 attributes.

7.2.1. MCC Attributes

Media Capture Attributes may be associated with the MCC instance and the Single Media Captures that the MCC references. A Provider should avoid providing conflicting attribute values between the MCC and Single Media Captures. Where there is conflict the attributes of the MCC override any that may be present in the individual Captures.

A Provider MAY include as much or as little of the original source Capture information as it requires.

There are MCC specific attributes that MUST only be used with Multiple Content Captures. These are described in the sections below. The attributes described in section 7.1.1. MAY also be used with MCCs.

The spatial related attributes of an MCC indicate its area of capture and point of capture within the scene, just like any other media capture. The spatial information does not imply anything about how other captures are composed within an MCC.

For example: A virtual scene could be constructed for the MCC capture with two Video Captures with a "MaxCaptures" attribute set to 2 and an "Area of Capture" attribute provided with an overall area. Each of the individual Captures could then also include an "Area of Capture" attribute with a sub-set of the overall area. The Consumer would then know how each capture is related to others within the scene, but not the relative position of the individual captures within the composed capture.

Capture Scene #1	
VC1	AreaofCapture=(0,0,0)(9,0,0) (0,0,9)(9,0,9)
VC2	AreaofCapture=(10,0,0)(19,0,0) (10,0,9)(19,0,9)

MCC1(VC1,VC2)	MaxCaptures=2
	AreaofCapture=(0,0,0)(19,0,0) (0,0,9)(19,0,9)

CSV(MCC1)	

Table 2: Example of MCC and Single Media Capture attributes

The sub-sections below describe the MCC only attributes.

7.2.1.1. Maximum Number of Captures within a MCC

The Maximum Number of Captures MCC attribute indicates the maximum number of individual Captures that may appear in a Capture Encoding at a time. The actual number at any given time can be less than or equal to this maximum. It may be used to derive how the Single Media Captures within the MCC are composed / switched with regards to space and time.

A Provider can indicate that the number of Captures in a MCC Capture Encoding is equal "=" to the MaxCaptures value or that there may be any number of Captures up to and including "<=" the MaxCaptures value. This allows a Provider to distinguish between a MCC that purely represents a composition of sources versus a MCC that represents switched or switched and composed sources.

MaxCaptures may be set to one so that only content related to one of the sources are shown in the MCC Capture Encoding at a time or it may be set to any value up to the total number of Source Media Captures in the MCC.

The bullets below describe how the setting of MaxCapture versus the number of Captures in the MCC affects how sources appear in a Capture Encoding:

- When MaxCaptures is set to <= 1 and the number of Captures in the MCC is greater than 1 (or not specified) in the MCC this is a switched case. Zero or 1 Captures may be switched into the Capture Encoding. Note: zero is allowed because of the "<=".
- When MaxCaptures is set to = 1 and the number of Captures in the MCC is greater than 1 (or not specified) in the MCC this is a switched case. Only one Capture source is contained in a Capture Encoding at a time.
- When MaxCaptures is set to <= N (with N > 1) and the number of Captures in the MCC is greater than N (or not specified) this is a switched and composed case. The Capture Encoding may contain purely switched sources (i.e. <=2 allows for 1 source on its own), or may contain composed and switched sources (i.e. a composition of 2 sources switched between the sources).
- When MaxCaptures is set to = N (with N > 1) and the number of Captures in the MCC is greater than N (or not specified) this

is a switched and composed case. The Capture Encoding contains composed and switched sources (i.e. a composition of N sources switched between the sources). It is not possible to have a single source.

. When MaxCaptures is set to <= to the number of Captures in the MCC this is a switched and composed case. The Capture Encoding may contain media switched between any number (up to the MaxCaptures) of composed sources.

. When MaxCaptures is set to = to the number of Captures in the MCC this is a composed case. All the sources are composed into a single Capture Encoding.

If this attribute is not set then as default it is assumed that all source media capture content can appear concurrently in the Capture Encoding associated with the MCC.

For example: The use of MaxCaptures equal to 1 on a MCC with three Video Captures VC1, VC2 and VC3 would indicate that the Advertiser in the Capture Encoding would switch between VC1, VC2 or VC3 as there may be only a maximum of one Capture at a time.

7.2.1.2. Policy

The Policy MCC Attribute indicates the criteria that the Provider uses to determine when and/or where media content appears in the Capture Encoding related to the MCC.

The attribute is in the form of a token that indicates the policy and an index representing an instance of the policy. The same index value can be used for multiple MCCs.

The tokens are:

SoundLevel - This indicates that the content of the MCC is determined by a sound level detection algorithm. The loudest (active) speaker (or a previous speaker, depending on the index value) is contained in the MCC.

RoundRobin - This indicates that the content of the MCC is determined by a time based algorithm. For example: the Provider provides content from a particular source for a period of time and then provides content from another source and so on.

An index is used to represent an instance in the policy setting. An index of 0 represents the most current instance of the policy, i.e.

the active speaker, 1 represents the previous instance, i.e. the previous active speaker and so on.

The following example shows a case where the Provider provides two media streams, one showing the active speaker and a second stream showing the previous speaker.

Capture Scene #1	
VC1	
VC2	

MCC1(VC1,VC2)	Policy=SoundLevel:0
	MaxCaptures=1

MCC2(VC1,VC2)	Policy=SoundLevel:1
	MaxCaptures=1

CSV(MCC1,MCC2)	

Table 3: Example Policy MCC attribute usage

7.2.1.3. Synchronisation Identity

The Synchronisation Identity MCC attribute indicates how the individual Captures in multiple MCC Captures are synchronised. To indicate that the Capture Encodings associated with MCCs contain Captures from the same source at the same time a Provider should set the same Synchronisation Identity on each of the concerned MCCs. It is the Provider that determines what the source for the Captures is, so a Provider can choose how to group together Single Media Captures into a combined "source" for the purpose of switching them together to keep them synchronized according to the SynchronisationID attribute. For example when the Provider is in an MCU it may determine that each separate CLUE Endpoint is a remote source of media. The Synchronisation Identity may be used across media types, i.e. to synchronize audio and video related MCCs.

Without this attribute it is assumed that multiple MCCs may provide content from different sources at any particular point in time.

For example:

Capture Scene #1	
VC1	Description=Left
VC2	Description=Centre
VC3	Description=Right
AC1	Description=Room
CSV(VC1,VC2,VC3)	
CSV(AC1)	
Capture Scene #2	
VC4	Description=Left
VC5	Description=Centre
VC6	Description=Right
AC2	Description=Room
CSV(VC4,VC5,VC6)	
CSV(AC2)	
Capture Scene #3	
VC7	
AC3	
Capture Scene #4	
VC8	
AC4	
Capture Scene #5	
MCC1(VC1,VC4,VC7)	SynchronisationID=1
	MaxCaptures=1
MCC2(VC2,VC5,VC8)	SynchronisationID=1
	MaxCaptures=1
MCC3(VC3,VC6)	MaxCaptures=1
MCC4(AC1,AC2,AC3,AC4)	SynchronisationID=1
	MaxCaptures=1
CSV(MCC1,MCC2,MCC3)	
CSV(MCC4)	

Table 4: Example Synchronisation Identity MCC attribute usage

The above Advertisement would indicate that MCC1, MCC2, MCC3 and MCC4 make up a Capture Scene. There would be four Capture Encodings (one for each MCC). Because MCC1 and MCC2 have the same SynchronisationID, each Encoding from MCC1 and MCC2 respectively would together have content from only Capture Scene 1 or only Capture Scene 2 or the combination of VC7 and VC8 at a particular point in time. In this case the Provider has decided the sources to be synchronized are Scene #1, Scene #2, and Scene #3 and #4 together. The Encoding from MCC3 would not be synchronised with MCC1 or MCC2. As MCC4 also has the same Synchronisation Identity as MCC1 and MCC2 the content of the audio Encoding will be synchronised with the video content.

7.2.1.4. Allow Subset Choice

The Allow Subset Choice MCC attribute is a boolean value, indicating whether or not the Provider allows the Consumer to choose a specific subset of the Captures referenced by the MCC. If this attribute is true, and the MCC references other Captures, then the Consumer MAY select (in a Configure message) a specific subset of those Captures to be included in the MCC, and the Provider MUST then include only that subset. If this attribute is false, or the MCC does not reference other Captures, then the Consumer MUST NOT select a subset.

7.3. Capture Scene

In order for a Provider’s individual Captures to be used effectively by a Consumer, the Provider organizes the Captures into one or more Capture Scenes, with the structure and contents of these Capture Scenes being sent from the Provider to the Consumer in the Advertisement.

A Capture Scene is a structure representing a spatial region containing one or more Capture Devices, each capturing media representing a portion of the region. A Capture Scene includes one or more Capture Scene Views (CSV), with each CSV including one or more Media Captures of the same media type. There can also be Media Captures that are not included in a Capture Scene View. A Capture Scene represents, for example, the video image of a group of people seated next to each other, along with the sound of their voices, which could be represented by some number of VCs and ACs in the Capture Scene Views. An MCU can also describe in Capture Scenes what it constructs from media Streams it receives.

A Provider MAY advertise one or more Capture Scenes. What constitutes an entire Capture Scene is up to the Provider. A simple Provider might typically use one Capture Scene for participant media (live video from the room cameras) and another Capture Scene for a computer generated presentation. In more complex systems, the use of additional Capture Scenes is also sensible. For example, a classroom may advertise two Capture Scenes involving live video, one including only the camera capturing the instructor (and associated audio), the other including camera(s) capturing students (and associated audio).

A Capture Scene MAY (and typically will) include more than one type of media. For example, a Capture Scene can include several Capture Scene Views for Video Captures, and several Capture Scene Views for Audio Captures. A particular Capture MAY be included in more than one Capture Scene View.

A Provider MAY express spatial relationships between Captures that are included in the same Capture Scene. However, there is no spatial relationship between Media Captures from different Capture Scenes. In other words, Capture Scenes each use their own spatial measurement system as outlined above in section 6.

A Provider arranges Captures in a Capture Scene to help the Consumer choose which captures it wants to render. The Capture Scene Views in a Capture Scene are different alternatives the Provider is suggesting for representing the Capture Scene. Each Capture Scene View is given an advertisement unique identity. The order of Capture Scene Views within a Capture Scene has no significance. The Media Consumer can choose to receive all Media Captures from one Capture Scene View for each media type (e.g. audio and video), or it can pick and choose Media Captures regardless of how the Provider arranges them in Capture Scene Views. Different Capture Scene Views of the same media type are not necessarily mutually exclusive alternatives. Also note that the presence of multiple Capture Scene Views (with potentially multiple encoding options in each view) in a given Capture Scene does not necessarily imply that a Provider is able to serve all the associated media simultaneously (although the construction of such an over-rich Capture Scene is probably not sensible in many cases). What a Provider can send simultaneously is determined through the Simultaneous Transmission Set mechanism, described in section 8.

Captures within the same Capture Scene View MUST be of the same media type - it is not possible to mix audio and video captures in the same Capture Scene View, for instance. The Provider MUST be capable of encoding and sending all Captures (that have an encoding group) in a single Capture Scene View simultaneously. The order of Captures within a Capture Scene View has no significance. A Consumer can decide to receive all the Captures in a single Capture Scene View, but a Consumer could also decide to receive just a subset of those captures. A Consumer can also decide to receive Captures from different Capture Scene Views, all subject to the constraints set by Simultaneous Transmission Sets, as discussed in section 8.

When a Provider advertises a Capture Scene with multiple CSVs, it is essentially signaling that there are multiple representations of the same Capture Scene available. In some cases, these multiple views would be used simultaneously (for instance a "video view" and an "audio view"). In some cases the views would conceptually be alternatives (for instance a view consisting of three Video Captures covering the whole room versus a view consisting of just a single Video Capture covering only the center of a room). In this latter example, one sensible choice for a Consumer would be to indicate (through its Configure and possibly through an additional offer/answer exchange) the Captures of that Capture Scene View that most closely matched the Consumer’s number of display devices or screen layout.

The following is an example of 4 potential Capture Scene Views for an endpoint-style Provider:

1. (VC0, VC1, VC2) - left, center and right camera Video Captures

2. (MCC3) - Video Capture associated with loudest room segment

3. (VC4) - Video Capture zoomed out view of all people in the room

4. (AC0) - main audio

The first view in this Capture Scene example is a list of Video Captures which have a spatial relationship to each other. Determination of the order of these captures (VC0, VC1 and VC2) for rendering purposes is accomplished through use of their Area of Capture attributes. The second view (MCC3) and the third view (VC4) are alternative representations of the same room’s video, which might be better suited to some Consumers’ rendering capabilities. The inclusion of the Audio Capture in the same Capture Scene indicates that AC0 is associated with all of those

Video Captures, meaning it comes from the same spatial region. Therefore, if audio were to be rendered at all, this audio would be the correct choice irrespective of which Video Captures were chosen.

7.3.1. Capture Scene attributes

Capture Scene Attributes can be applied to Capture Scenes as well as to individual media captures. Attributes specified at this level apply to all constituent Captures. Capture Scene attributes include

. Human-readable description of the Capture Scene, which could be in multiple languages;
. xCard scene information
. Scale information (millimeters, unknown, no scale), as described in Section 6.

7.3.1.1. Scene Information

The Scene information attribute provides information regarding the Capture Scene rather than individual participants. The Provider may gather the information automatically or manually from a variety of sources. The scene information attribute allows a Provider to indicate information such as: organizational or geographic information allowing a Consumer to determine which Capture Scenes are of interest in order to then perform Capture selection. It also allows a Consumer to render information regarding the Scene or to use it for further processing.

As per 7.1.1.10. the xCard format is used to convey this information and the Provider may supply a minimal set of information or a larger set of information.

In order to keep CLUE messages compact the Provider SHOULD use a URI to point to any LOGO, PHOTO or SOUND contained in the xCARD rather than transmitting the LOGO, PHOTO or SOUND data in a CLUE message.

7.3.2. Capture Scene View attributes

A Capture Scene can include one or more Capture Scene Views in addition to the Capture Scene wide attributes described above. Capture Scene View attributes apply to the Capture Scene View as a whole, i.e. to all Captures that are part of the Capture Scene View.

Capture Scene View attributes include:

. Human-readable description (which could be in multiple languages) of the Capture Scene View

7.4. Global View List

An Advertisement can include an optional Global View list. Each item in this list is a Global View. The Provider can include multiple Global Views, to allow a Consumer to choose sets of captures appropriate to its capabilities or application. The choice of how to make these suggestions in the Global View list for what represents all the scenes for which the Provider can send media is up to the Provider. This is very similar to how each CSV represents a particular scene.

As an example, suppose an advertisement has three scenes, and each scene has three CSVs, ranging from one to three video captures in each CSV. The Provider is advertising a total of nine video Captures across three scenes. The Provider can use the Global View list to suggest alternatives for Consumers that can’t receive all nine video Captures as separate media streams. For accommodating a Consumer that wants to receive three video Captures, a Provider might suggest a Global View containing just a single CSV with three Captures and nothing from the other two scenes. Or a Provider might suggest a Global View containing three different CSVs, one from each scene, with a single video Capture in each.

Some additional rules:

. The ordering of Global Views in the Global View list is insignificant.
. The ordering of CSVs within each Global View is insignificant.
. A particular CSV may be used in multiple Global Views.
. The Provider must be capable of encoding and sending all Captures within the CSVs of a given Global View simultaneously.

The following figure shows an example of the structure of Global Views in a Global View List.

```
      ........................................................
      . Advertisement                                        .
      .                                                      .
      . +--------------+         +-------------------------+ .
      . |Scene 1       |         |Global View List         | .
      . |              |         |                         | .
      . | CSV1 (v)<----------------- Global View (CSV 1)   | .
      . |         <-------.      |                         | .
      . |              |  *--------- Global View (CSV 1,5) | .
      . | CSV2 (v)     |  |      |                         | .
      . |              |  |      |                         | .
      . | CSV3 (v)<---------*------- Global View (CSV 3,5) | .
      . |              |  | |    |                         | .
      . | CSV4 (a)<----------------- Global View (CSV 4)   | .
      . |         <-----------.  |                         | .
      . +--------------+  | | *----- Global View (CSV 4,6) | .
      .                   | | |  |                         | .
      . +--------------+  | | |  +-------------------------+ .
      . |Scene 2       |  | | |                              .
      . |              |  | | |                              .
      . | CSV5 (v)<-------' | |                              .
      . |         <---------' |                              .
      . |              |      |       (v) = video            .
      . | CSV6 (a)<-----------'       (a) = audio            .
      . |              |                                     .
      . +--------------+                                     .
      '......................................................'
```

Figure 3:   Global View List Structure

8. Simultaneous Transmission Set Constraints

In many practical cases, a Provider has constraints or limitations on its ability to send Captures simultaneously. One type of limitation is caused by the physical limitations of capture mechanisms; these constraints are represented by a Simultaneous Transmission Set. The second type of limitation reflects the encoding resources available, such as bandwidth or video encoding throughput (macroblocks/second). This type of constraint is captured by Individual Encodings and Encoding Groups, discussed below.

Some Endpoints or MCUs can send multiple Captures simultaneously; however sometimes there are constraints that limit which Captures can be sent simultaneously with other Captures. A device may not be able to be used in different ways at the same time. Provider Advertisements are made so that the Consumer can choose one of several possible mutually exclusive usages of the device. This type of constraint is expressed in a Simultaneous Transmission Set, which lists all the Captures of a particular media type (e.g. audio, video, text) that can be sent at the same time. There are different Simultaneous Transmission Sets for each media type in the Advertisement. This is easier to show in an example.

Consider the example of a room system where there are three cameras each of which can send a separate Capture covering two persons each- VC0, VC1, VC2. The middle camera can also zoom out (using an optical zoom lens) and show all six persons, VC3. But the middle camera cannot be used in both modes at the same time - it has to either show the space where two participants sit or the whole six seats, but not both at the same time. As a result, VC1 and VC3 cannot be sent simultaneously.

Simultaneous Transmission Sets are expressed as sets of the Media Captures that the Provider could transmit at the same time (though, in some cases, it is not intuitive to do so). If a Multiple Content Capture is included in a Simultaneous Transmission Set it indicates that the Capture Encoding associated with it could be transmitted as the same time as the other Captures within the Simultaneous Transmission Set. It does not imply that the Single Media Captures contained in the Multiple Content Capture could all be transmitted at the same time.

In this example the two Simultaneous Transmission Sets are shown in Table 5. If a Provider advertises one or more mutually exclusive Simultaneous Transmission Sets, then for each media type the Consumer MUST ensure that it chooses Media Captures that lie wholly within one of those Simultaneous Transmission Sets.

Simultaneous Sets
{VC0, VC1, VC2}
{VC0, VC3, VC2}

Table 5: Two Simultaneous Transmission Sets

A Provider OPTIONALLY can include the Simultaneous Transmission Sets in its Advertisement. These constraints apply across all the

Capture Scenes in the Advertisement. It is a syntax conformance requirement that the Simultaneous Transmission Sets MUST allow all the media Captures in any particular Capture Scene View to be used simultaneously. Similarly, the Simultaneous Transmission Sets MUST reflect the simultaneity expressed by any Global View.

For shorthand convenience, a Provider MAY describe a Simultaneous Transmission Set in terms of Capture Scene Views and Capture Scenes. If a Capture Scene View is included in a Simultaneous Transmission Set, then all Media Captures in the Capture Scene View are included in the Simultaneous Transmission Set. If a Capture Scene is included in a Simultaneous Transmission Set, then all its Capture Scene Views (of the corresponding media type) are included in the Simultaneous Transmission Set. The end result reduces to a set of Media Captures, of a particular media type, in either case.

If an Advertisement does not include Simultaneous Transmission Sets, then the Provider MUST be able to simultaneously provide all the Captures from any one CSV of each media type from each Capture Scene. Likewise, if there are no Simultaneous Transmission Sets and there is a Global View list, then the Provider MUST be able to simultaneously provide all the Captures from any particular Global View (of each media type) from the Global View list.

If an Advertisement includes multiple Capture Scene Views in a Capture Scene then the Consumer MAY choose one Capture Scene View for each media type, or MAY choose individual Captures based on the Simultaneous Transmission Sets.

9. Encodings

Individual encodings and encoding groups are CLUE’s mechanisms allowing a Provider to signal its limitations for sending Captures, or combinations of Captures, to a Consumer. Consumers can map the Captures they want to receive onto the Encodings, with the encoding parameters they want. As for the relationship between the CLUE-specified mechanisms based on Encodings and the SIP offer/answer exchange, please refer to section 5.

9.1. Individual Encodings

An Individual Encoding represents a way to encode a Media Capture as a Capture Encoding, to be sent as an encoded media stream from the Provider to the Consumer. An Individual Encoding has a set of parameters characterizing how the media is encoded.

Different media types have different parameters, and different encoding algorithms may have different parameters. An Individual Encoding can be assigned to at most one Capture Encoding at any given time.

Individual Encoding parameters are represented in SDP [RFC4566], not in CLUE messages. For example, for a video encoding using H.26x compression technologies, this can include parameters such as:

- . Maximum bandwidth;
- . Maximum picture size in pixels;
- . Maximum number of pixels to be processed per second;

The bandwidth parameter is the only one that specifically relates to a CLUE Advertisement, as it can be further constrained by the maximum group bandwidth in an Encoding Group.

9.2. Encoding Group

An Encoding Group includes a set of one or more Individual Encodings, and parameters that apply to the group as a whole. By grouping multiple individual Encodings together, an Encoding Group describes additional constraints on bandwidth for the group. A single Encoding Group MAY refer to Encodings for different media types.

The Encoding Group data structure contains:

- . Maximum bitrate for all encodings in the group combined;
- . A list of identifiers for the Individual Encodings belonging to the group.

When the Individual Encodings in a group are instantiated into Capture Encodings, each Capture Encoding has a bitrate that MUST be less than or equal to the max bitrate for the particular Individual Encoding. The "maximum bitrate for all encodings in the group" parameter gives the additional restriction that the sum of all the individual Capture Encoding bitrates MUST be less than or equal to this group value.

The following diagram illustrates one example of the structure of a media Provider’s Encoding Groups and their contents.

```
   ,-------------------------------------------------.
   |             Media Provider                      |
   |                                                 |
   |  ,--------------------------------------.       |
   |  | ,--------------------------------------.     |
   |  | | ,--------------------------------------.   |
   |  | | |          Encoding Group              |   |
   |  | | | ,-----------.                        |   |
   |  | | | |           | ,---------.            |   |
   |  | | | |           | |         | ,---------.|   |
   |  | | | | Encoding1 | |Encoding2| |Encoding3||   |
   |  `.| | |           | |         | `---------'|   |
   |    `.| `-----------' `---------'            |   |
   |      `--------------------------------------'   |
   `-------------------------------------------------'
```

Figure 4:   Encoding Group Structure

A Provider advertises one or more Encoding Groups. Each Encoding Group includes one or more Individual Encodings. Each Individual Encoding can represent a different way of encoding media. For example one Individual Encoding may be 1080p60 video, another could be 720p30, with a third being CIF, all in, for example, H.264 format.
While a typical three codec/display system might have one Encoding Group per "codec box" (physical codec, connected to one camera and one screen), there are many possibilities for the number of Encoding Groups a Provider may be able to offer and for the encoding values in each Encoding Group.

There is no requirement for all Encodings within an Encoding Group to be instantiated at the same time.

9.3. Associating Captures with Encoding Groups

Each Media Capture, including MCCs, MAY be associated with one Encoding Group. To be eligible for configuration, a Media Capture MUST be associated with one Encoding Group, which is used to instantiate that Capture into a Capture Encoding. When an MCC is configured all the Media Captures referenced by the MCC will appear in the Capture Encoding according to the attributes of the chosen encoding of the MCC. This allows an Advertiser to specify encoding attributes associated with the Media Captures without the need to provide an individual Capture Encoding for each of the inputs.

If an Encoding Group is assigned to a Media Capture referenced by the MCC it indicates that this Capture may also have an individual Capture Encoding.

For example:

Capture Scene #1	
VC1	EncodeGroupID=1
VC2	
MCC1(VC1,VC2)	EncodeGroupID=2
CSV(VC1)	
CSV(MCC1)	

Table 6: Example usage of Encoding with MCC and source Captures

This would indicate that VC1 may be sent as its own Capture Encoding from EncodeGroupID=1 or that it may be sent as part of a Capture Encoding from EncodeGroupID=2 along with VC2.

More than one Capture MAY use the same Encoding Group.

The maximum number of Capture Encodings that can result from a particular Encoding Group constraint is equal to the number of individual Encodings in the group. The actual number of Capture Encodings used at any time MAY be less than this maximum. Any of the Captures that use a particular Encoding Group can be encoded according to any of the Individual Encodings in the group.

It is a protocol conformance requirement that the Encoding Groups MUST allow all the Captures in a particular Capture Scene View to be used simultaneously.

10. Consumer’s Choice of Streams to Receive from the Provider

After receiving the Provider’s Advertisement message (that includes media captures and associated constraints), the Consumer composes its reply to the Provider in the form of a Configure message. The Consumer is free to use the information in the Advertisement as it chooses, but there are a few obviously sensible design choices, which are outlined below.

If multiple Providers connect to the same Consumer (i.e. in an MCU-less multiparty call), it is the responsibility of the Consumer to compose Configures for each Provider that both fulfill each Provider’s constraints as expressed in the Advertisement, as well as its own capabilities.

In an MCU-based multiparty call, the MCU can logically terminate the Advertisement/Configure negotiation in that it can hide the characteristics of the receiving endpoint and rely on its own capabilities (transcoding/transrating/...) to create Media Streams that can be decoded at the Endpoint Consumers. The timing of an MCU’s sending of Advertisements (for its outgoing ports) and Configures (for its incoming ports, in response to Advertisements received there) is up to the MCU and implementation dependent.

As a general outline, a Consumer can choose, based on the Advertisement it has received, which Captures it wishes to receive, and which Individual Encodings it wants the Provider to use to encode the Captures.

On receipt of an Advertisement with an MCC the Consumer treats the MCC as per other non-MCC Captures with the following differences:

- The Consumer would understand that the MCC is a Capture that includes the referenced individual Captures (or any Captures, if none are referenced) and that these individual Captures are delivered as part of the MCC’s Capture Encoding.

- The Consumer may utilise any of the attributes associated with the referenced individual Captures and any Capture Scene attributes from where the individual Captures were defined to choose Captures and for rendering decisions.

- If the MCC attribute Allow Subset Choice is true, then the Consumer may or may not choose to receive all the indicated Captures. It can choose to receive a sub-set of Captures indicated by the MCC.

For example if the Consumer receives:

```
        MCC1(VC1,VC2,VC3){attributes}
```

A Consumer could choose all the Captures within a MCC however if the Consumer determines that it doesn’t want VC3 it can return MCC1(VC1,VC2). If it wants all the individual Captures then it returns only the MCC identity (i.e. MCC1). If the MCC in the advertisement does not reference any individual captures, or the Allow Subset Choice attribute is false, then the Consumer cannot choose what is included in the MCC, it is up to the Provider to decide.

A Configure Message includes a list of Capture Encodings. These are the Capture Encodings the Consumer wishes to receive from the Provider. Each Capture Encoding refers to one Media Capture and one Individual Encoding.

For each Capture the Consumer wants to receive, it configures one of the Encodings in that Capture’s Encoding Group. The Consumer does this by telling the Provider, in its Configure Message, which Encoding to use for each chosen Capture. Upon receipt of this Configure from the Consumer, common knowledge is established between Provider and Consumer regarding sensible choices for the media streams. The setup of the actual media channels, at least in the simplest case, is left to a following offer/answer exchange. Optimized implementations may speed up the reaction to the offer/answer exchange by reserving the resources at the time of finalization of the CLUE handshake.

CLUE advertisements and configure messages don’t necessarily require a new SDP offer/answer for every CLUE message exchange. But the resulting encodings sent via RTP must conform to the most recent SDP offer/answer result.

In order to meaningfully create and send an initial Configure, the Consumer needs to have received at least one Advertisement, and an SDP offer defining the Individual Encodings, from the Provider.

In addition, the Consumer can send a Configure at any time during the call. The Configure MUST be valid according to the most recently received Advertisement. The Consumer can send a Configure either in response to a new Advertisement from the Provider or on its own, for example because of a local change in conditions (people leaving the room, connectivity changes, multipoint related considerations).

When choosing which Media Streams to receive from the Provider, and the encoding characteristics of those Media Streams, the Consumer advantageously takes several things into account: its local preference, simultaneity restrictions, and encoding limits.

10.1. Local preference

A variety of local factors influence the Consumer’s choice of Media Streams to be received from the Provider:

- if the Consumer is an Endpoint, it is likely that it would choose, where possible, to receive video and audio Captures that match the number of display devices and audio system it has

- if the Consumer is an MCU, it may choose to receive loudest speaker streams (in order to perform its own media composition) and avoid pre-composed video Captures

- user choice (for instance, selection of a new layout) may result in a different set of Captures, or different encoding characteristics, being required by the Consumer

10.2. Physical simultaneity restrictions

Often there are physical simultaneity constraints of the Provider that affect the Provider’s ability to simultaneously send all of the captures the Consumer would wish to receive. For instance, an MCU, when connected to a multi-camera room system, might prefer to receive both individual video streams of the people present in the room and an overall view of the room from a single camera. Some Endpoint systems might be able to provide both of these sets of streams simultaneously, whereas others might not (if the overall room view were produced by changing the optical zoom level on the center camera, for instance).

10.3. Encoding and encoding group limits

Each of the Provider’s encoding groups has limits on bandwidth, and the constituent potential encodings have limits on the bandwidth, computational complexity, video frame rate, and resolution that can be provided. When choosing the Captures to be received from a Provider, a Consumer device MUST ensure that the encoding characteristics requested for each individual Capture fits within the capability of the encoding it is being configured to use, as well as ensuring that the combined encoding characteristics for Captures fit within the capabilities of their associated encoding groups. In some cases, this could cause an otherwise "preferred" choice of capture encodings to be passed over in favor of different Capture Encodings--for instance, if a set of three Captures could only be provided at a low resolution then a three screen device could switch to favoring a single, higher quality, Capture Encoding.

11. Extensibility

One important characteristics of the Framework is its extensibility. The standard for interoperability and handling multiple streams must be future-proof. The framework itself is inherently extensible through expanding the data model types. For example:

- Adding more types of media, such as telemetry, can done by defining additional types of Captures in addition to audio and video.

- Adding new functionalities, such as 3-D video Captures, say, may require additional attributes describing the Captures.

The infrastructure is designed to be extended rather than requiring new infrastructure elements. Extension comes through adding to defined types.

12. Examples - Using the Framework (Informative)

This section gives some examples, first from the point of view of the Provider, then the Consumer, then some multipoint scenarios

12.1. Provider Behavior

This section shows some examples in more detail of how a Provider can use the framework to represent a typical case for telepresence rooms. First an endpoint is illustrated, then an MCU case is shown.

12.1.1. Three screen Endpoint Provider

Consider an Endpoint with the following description:

3 cameras, 3 displays, a 6 person table

- Each camera can provide one Capture for each 1/3 section of the table

- A single Capture representing the active speaker can be provided (voice activity based camera selection to a given encoder input port implemented locally in the Endpoint)

- A single Capture representing the active speaker with the other 2 Captures shown picture in picture (PiP) within the stream can be provided (again, implemented inside the endpoint)

- A Capture showing a zoomed out view of all 6 seats in the room can be provided

The video and audio Captures for this Endpoint can be described as follows.

Video Captures:

- VC0- (the left camera stream), encoding group=EG0, view=table

- VC1- (the center camera stream), encoding group=EG1, view=table

- VC2- (the right camera stream), encoding group=EG2, view=table

- MCC3- (the loudest panel stream), encoding group=EG1, view=table, MaxCaptures=1, policy=SoundLevel

- MCC4- (the loudest panel stream with PiPs), encoding group=EG1, view=room, MaxCaptures=3, policy=SoundLevel

- VC5- (the zoomed out view of all people in the room), encoding group=EG1, view=room

- VC6- (presentation stream), encoding group=EG1, presentation

The following diagram is a top view of the room with 3 cameras, 3 displays, and 6 seats. Each camera captures 2 people. The six seats are not all in a straight line.

```
      ,-. d
     (   )`--.__        +---+
      `-' /     `--.__  |   |
    ,-.  |            `-.._ |_-+Camera 2 (VC2)
   (   ).'     <--(AC1)-+-''`+-+
    `-' |_...---''      |   |
    ,-.c+-..__          +---+
   (   )|     ``--..__  |   |
    `-' |             ``+-..|_-+Camera 1 (VC1)
    ,-. |      <--(AC2)..--'|+-+                          ^
   (   )|     __..--'   |   |                             |
    `-'b|..--'          +---+                             |X
    ,-. |``---..___     |   |                             |
   (   )\          ```--..._|_-+Camera 0 (VC0)            |
    `-'  \     <--(AC0) ..-'''-+                          |
     ,-. \      __.--'' |   |                  <----------+
    (   ) |..-''        +---+                      Y
     `-' a                          (0,0,0) origin is under Camera 1
```

Figure 5:   Room Layout Top View

The two points labeled b and c are intended to be at the midpoint between the seating positions, and where the fields of view of the cameras intersect.

The plane of interest for VC0 is a vertical plane that intersects points ’a’ and ’b’.

The plane of interest for VC1 intersects points ’b’ and ’c’. The plane of interest for VC2 intersects points ’c’ and ’d’.

This example uses an area scale of millimeters.

Areas of capture:

	bottom left	bottom right	top left	top right
VC0	(-2011,2850,0)	(-673,3000,0)	(-2011,2850,757)	(-673,3000,757)
VC1	(-673,3000,0)	(673,3000,0)	(-673,3000,757)	(673,3000,757)
VC2	(673,3000,0)	(2011,2850,0)	(673,3000,757)	(2011,3000,757)
MCC3	(-2011,2850,0)	(2011,2850,0)	(-2011,2850,757)	(2011,3000,757)
MCC4	(-2011,2850,0)	(2011,2850,0)	(-2011,2850,757)	(2011,3000,757)
VC5	(-2011,2850,0)	(2011,2850,0)	(-2011,2850,757)	(2011,3000,757)
VC6	none			

Points of capture:

VC0	(-1678,0,800)
VC1	(0,0,800)
VC2	(1678,0,800)
MCC3	none
MCC4	none
VC5	(0,0,800)
VC6	none

In this example, the right edge of the VC0 area lines up with the left edge of the VC1 area. It doesn’t have to be this way. There could be a gap or an overlap. One additional thing to note for this example is the distance from a to b is equal to the distance from b to c and the distance from c to d. All these distances are 1346 mm. This is the planar width of each area of capture for VC0, VC1, and VC2.

Note the text in parentheses (e.g. "the left camera stream") is not explicitly part of the model, it is just explanatory text for this example, and is not included in the model with the media captures and attributes. Also, MCC4 doesn’t say anything about how a capture is composed, so the media consumer can’t tell based on this capture that MCC4 is composed of a "loudest panel with PiPs".

Audio Captures:

Three ceiling microphones are located between the cameras and the table, at the same height as the cameras. The microphones point down at an angle toward the seating positions.

- o AC0 (left), encoding group=EG3

- o AC1 (right), encoding group=EG3

- o AC2 (center) encoding group=EG3

- o AC3 being a simple pre-mixed audio stream from the room (mono), encoding group=EG3

- o AC4 audio stream associated with the presentation video (mono) encoding group=EG3, presentation

	Point of capture:	Point on Line of Capture:
AC0	(-1342,2000,800)	(-1342,2925,379)
AC1	(1342,2000,800)	(1342,2925,379)
AC2	(0,2000,800)	(0,3000,379)
AC3	(0,2000,800)	(0,3000,379)
AC4	none	

The physical simultaneity information is:

Simultaneous transmission set #1 {VC0, VC1, VC2, MCC3, MCC4, VC6}

Simultaneous transmission set #2 {VC0, VC2, VC5, VC6}

This constraint indicates it is not possible to use all the VCs at the same time. VC5 cannot be used at the same time as VC1 or MCC3 or MCC4. Also, using every member in the set simultaneously may not make sense - for example MCC3(loudest) and MCC4 (loudest with PiP). In addition, there are encoding constraints that make choosing all of the VCs in a set impossible. VC1, MCC3, MCC4, VC5, VC6 all use EG1 and EG1 has only 3 ENCs. This constraint shows up in the encoding groups, not in the simultaneous transmission sets.

In this example there are no restrictions on which Audio Captures can be sent simultaneously.

Encoding Groups:

This example has three encoding groups associated with the video captures. Each group can have 3 encodings, but with each potential encoding having a progressively lower specification. In this example, 1080p60 transmission is possible (as ENC0 has a maxPps value compatible with that). Significantly, as up to 3 encodings are available per group, it is possible to transmit some video Captures simultaneously that are not in the same view in the Capture Scene. For example VC1 and MCC3 at the same time. The information below about Encodings is a summary of what would be conveyed in SDP, not directly in the CLUE Advertisement.

```
encodeGroupID=EG0, maxGroupBandwidth=6000000
    encodeID=ENC0, maxWidth=1920, maxHeight=1088, maxFrameRate=60,
                   maxPps=124416000, maxBandwidth=4000000
    encodeID=ENC1, maxWidth=1280, maxHeight=720, maxFrameRate=30,
                   maxPps=27648000, maxBandwidth=4000000
    encodeID=ENC2, maxWidth=960, maxHeight=544, maxFrameRate=30,
                   maxPps=15552000, maxBandwidth=4000000
encodeGroupID=EG1  maxGroupBandwidth=6000000
    encodeID=ENC3, maxWidth=1920, maxHeight=1088, maxFrameRate=60,
                   maxPps=124416000, maxBandwidth=4000000
    encodeID=ENC4, maxWidth=1280, maxHeight=720, maxFrameRate=30,
                   maxPps=27648000, maxBandwidth=4000000
    encodeID=ENC5, maxWidth=960, maxHeight=544, maxFrameRate=30,
                   maxPps=15552000, maxBandwidth=4000000
encodeGroupID=EG2  maxGroupBandwidth=6000000
    encodeID=ENC6, maxWidth=1920, maxHeight=1088, maxFrameRate=60,
                   maxPps=124416000, maxBandwidth=4000000
    encodeID=ENC7, maxWidth=1280, maxHeight=720, maxFrameRate=30,
                   maxPps=27648000, maxBandwidth=4000000
    encodeID=ENC8, maxWidth=960, maxHeight=544, maxFrameRate=30,
                   maxPps=15552000, maxBandwidth=4000000
```

Figure 6:   Example Encoding Groups for Video

For audio, there are five potential encodings available, so all five Audio Captures can be encoded at the same time.

```
encodeGroupID=EG3, maxGroupBandwidth=320000
    encodeID=ENC9, maxBandwidth=64000
    encodeID=ENC10, maxBandwidth=64000
    encodeID=ENC11, maxBandwidth=64000
    encodeID=ENC12, maxBandwidth=64000
    encodeID=ENC13, maxBandwidth=64000
```

Figure 7:   Example Encoding Group for Audio

Capture Scenes:

The following table represents the Capture Scenes for this Provider. Recall that a Capture Scene is composed of alternative Capture Scene Views covering the same spatial region. Capture Scene #1 is for the main people captures, and Capture Scene #2 is for presentation.

Each row in the table is a separate Capture Scene View

Capture Scene #1
VC0, VC1, VC2
MCC3
MCC4
VC5
AC0, AC1, AC2
AC3

Capture Scene #2
VC6
AC4

Table 7: Example Capture Scene Views

Different Capture Scenes are distinct from each other, and are non-overlapping. A Consumer can choose a view from each Capture Scene. In this case the three Captures VC0, VC1, and VC2 are one way of representing the video from the Endpoint. These three Captures should appear adjacent next to each other. Alternatively, another way of representing the Capture Scene is with the capture MCC3, which automatically shows the person who is talking. Similarly for the MCC4 and VC5 alternatives.

As in the video case, the different views of audio in Capture Scene #1 represent the "same thing", in that one way to receive the audio is with the 3 Audio Captures (AC0, AC1, AC2), and another way is with the mixed AC3. The Media Consumer can choose an audio CSV it is capable of receiving.

The spatial ordering is understood by the Media Capture attributes Area of Capture, Point of Capture and Point on Line of Capture.

A Media Consumer would likely want to choose a Capture Scene View to receive based in part on how many streams it can simultaneously receive. A consumer that can receive three video streams would probably prefer to receive the first view of Capture Scene #1 (VC0, VC1, VC2) and not receive the other views. A consumer that can receive only one video stream would probably choose one of the other views.

If the consumer can receive a presentation stream too, it would also choose to receive the only view from Capture Scene #2 (VC6).

12.1.2. Encoding Group Example

This is an example of an Encoding Group to illustrate how it can express dependencies between Encodings. The information below about Encodings is a summary of what would be conveyed in SDP, not directly in the CLUE Advertisement.

```
encodeGroupID=EG0 maxGroupBandwidth=6000000
    encodeID=VIDENC0, maxWidth=1920, maxHeight=1088,
      maxFrameRate=60, maxPps=62208000, maxBandwidth=4000000
    encodeID=VIDENC1, maxWidth=1920, maxHeight=1088,
      maxFrameRate=60, maxPps=62208000, maxBandwidth=4000000
    encodeID=AUDENC0, maxBandwidth=96000
    encodeID=AUDENC1, maxBandwidth=96000
    encodeID=AUDENC2, maxBandwidth=96000
```

Here, the Encoding Group is EG0. Although the Encoding Group is capable of transmitting up to 6Mbit/s, no individual video Encoding can exceed 4Mbit/s.

This encoding group also allows up to 3 audio encodings, AUDENC<0-2>. It is not required that audio and video encodings reside within the same encoding group, but if so then the group’s overall maxBandwidth value is a limit on the sum of all audio and video encodings configured by the consumer. A system that does not wish or need to combine bandwidth limitations in this way should instead use separate encoding groups for audio and video in order for the bandwidth limitations on audio and video to not interact.

Audio and video can be expressed in separate encoding groups, as in this illustration.

```
encodeGroupID=EG0 maxGroupBandwidth=6000000
    encodeID=VIDENC0, maxWidth=1920, maxHeight=1088,
      maxFrameRate=60, maxPps=62208000, maxBandwidth=4000000
    encodeID=VIDENC1, maxWidth=1920, maxHeight=1088,
      maxFrameRate=60, maxPps=62208000, maxBandwidth=4000000
encodeGroupID=EG1 maxGroupBandwidth=500000
    encodeID=AUDENC0, maxBandwidth=96000
    encodeID=AUDENC1, maxBandwidth=96000
    encodeID=AUDENC2, maxBandwidth=96000
```

12.1.3. The MCU Case

This section shows how an MCU might express its Capture Scenes, intending to offer different choices for consumers that can handle different numbers of streams. Each MCC is for video. A single Audio Capture is provided for all single and multi-screen configurations that can be associated (e.g. lip-synced) with any combination of Video Captures (the MCCs) at the consumer.

Capture Scene #1	
MCC	for a single screen consumer
MCC1, MCC2	for a two screen consumer
MCC3, MCC4, MCC5	for a three screen consumer
MCC6, MCC7, MCC8, MCC9	for a four screen consumer
AC0	AC representing all participants
CSV(MCC0)	
CSV(MCC1,MCC2)	
CSV(MCC3,MCC4,MCC5)	
CSV(MCC6,MCC7, MCC8,MCC9)	
CSV(AC0)	

Table 8: MCU main Capture Scenes

If / when a presentation stream becomes active within the conference the MCU might re-advertise the available media as:

Capture Scene #2	note
VC10	video capture for presentation
AC1	presentation audio to accompany VC10
CSV(VC10)	
CSV(AC1)	

Table 9: MCU presentation Capture Scene

12.2. Media Consumer Behavior

This section gives an example of how a Media Consumer might behave when deciding how to request streams from the three screen endpoint described in the previous section.

The receive side of a call needs to balance its requirements, based on number of screens and speakers, its decoding capabilities and available bandwidth, and the provider’s capabilities in order to optimally configure the provider’s streams. Typically it would want to receive and decode media from each Capture Scene advertised by the Provider.

A sane, basic, algorithm might be for the consumer to go through each Capture Scene View in turn and find the collection of Video Captures that best matches the number of screens it has (this might include consideration of screens dedicated to presentation video display rather than "people" video) and then decide between alternative views in the video Capture Scenes based either on hard-coded preferences or user choice. Once this choice has been made, the consumer would then decide how to configure the provider’s encoding groups in order to make best use of the available network bandwidth and its own decoding capabilities.

12.2.1. One screen Media Consumer

MCC3, MCC4 and VC5 are all different views by themselves, not grouped together in a single view, so the receiving device should choose between one of those. The choice would come down to whether to see the greatest number of participants simultaneously at roughly equal precedence (VC5), a switched view of just the loudest region (MCC3) or a switched view with PiPs (MCC4). An endpoint device with a small amount of knowledge of these differences could offer a dynamic choice of these options, in-call, to the user.

12.2.2. Two screen Media Consumer configuring the example

Mixing systems with an even number of screens, "2n", and those with "2n+1" cameras (and vice versa) is always likely to be the problematic case. In this instance, the behavior is likely to be determined by whether a "2 screen" system is really a "2 decoder" system, i.e., whether only one received stream can be displayed per screen or whether more than 2 streams can be received and spread across the available screen area. To enumerate 3 possible behaviors here for the 2 screen system when it learns that the far end is "ideally" expressed via 3 capture streams:

1. Fall back to receiving just a single stream (MCC3, MCC4 or VC5 as per the 1 screen consumer case above) and either leave one screen blank or use it for presentation if / when a presentation becomes active.

2. Receive 3 streams (VC0, VC1 and VC2) and display across 2 screens (either with each capture being scaled to 2/3 of a screen and the center capture being split across 2 screens) or, as would be necessary if there were large bezels on the screens, with each stream being scaled to 1/2 the screen width and height and there being a 4th "blank" panel. This 4th panel could potentially be used for any presentation that became active during the call.

3. Receive 3 streams, decode all 3, and use control information indicating which was the most active to switch between showing the left and center streams (one per screen) and the center and right streams.

For an endpoint capable of all 3 methods of working described above, again it might be appropriate to offer the user the choice of display mode.

12.2.3. Three screen Media Consumer configuring the example

This is the most straightforward case - the Media Consumer would look to identify a set of streams to receive that best matched its available screens and so the VC0 plus VC1 plus VC2 should match optimally. The spatial ordering would give sufficient information for the correct Video Capture to be shown on the correct screen, and the consumer would either need to divide a single encoding group’s capability by 3 to determine what resolution and frame rate to configure the provider with or to configure the individual Video Captures’ Encoding Groups with what makes most sense (taking into account the receive side decode capabilities, overall call bandwidth, the resolution of the screens plus any user preferences such as motion vs. sharpness).

12.3. Multipoint Conference utilizing Multiple Content Captures

The use of MCCs allows the MCU to construct outgoing Advertisements describing complex media switching and composition scenarios. The following sections provide several examples.

Note: In the examples the identities of the CLUE elements (e.g. Captures, Capture Scene) in the incoming Advertisements overlap. This is because there is no co-ordination between the endpoints. The MCU is responsible for making these unique in the outgoing advertisement.

12.3.1. Single Media Captures and MCC in the same Advertisement

Four endpoints are involved in a Conference where CLUE is used. An MCU acts as a middlebox between the endpoints with a CLUE channel between each endpoint and the MCU. The MCU receives the following Advertisements.

Capture Scene #1	Description=AustralianConfRoom

VC1	Description=Audience
	EncodeGroupID=1

CSV(VC1)	

Table 10: Advertisement received from Endpoint A

Capture Scene #1	Description=ChinaConfRoom

VC1	Description=Speaker
	EncodeGroupID=1

VC2	Description=Audience
	EncodeGroupID=1

CSV(VC1, VC2)	

Table 11: Advertisement received from Endpoint B

Capture Scene #1	Description=USAConfRoom

VC1	Description=Audience
	EncodeGroupID=1

CSV(VC1)	

Table 12: Advertisement received from Endpoint C

Note: Endpoint B above indicates that it sends two streams.

If the MCU wanted to provide a Multiple Content Capture containing a round robin switched view of the audience from the 3 endpoints and the speaker it could construct the following advertisement:

Advertisement sent to Endpoint F

Capture Scene #1	Description=AustralianConfRoom
VC1 CSV(VC1)	Description=Audience
Capture Scene #2	Description=ChinaConfRoom

VC2
VC3
CSV(VC2, VC3)
	Description=Speaker
	Description=Audience

Capture Scene #3	Description=USAConfRoom
VC4 CSV(VC4)	Description=Audience
Capture Scene #4	

MCC1(VC1,VC2,VC3,VC4)
CSV(MCC1)
	Policy=RoundRobin:1
	MaxCaptures=1
	EncodingGroup=1

Table 13: Advertisement sent to Endpoint F - One Encoding

Alternatively if the MCU wanted to provide the speaker as one media stream and the audiences as another it could assign an encoding group to VC2 in Capture Scene 2 and provide a CSV in Capture Scene #4 as per the example below.

Advertisement sent to Endpoint F

Capture Scene #1	Description=AustralianConfRoom
VC1	Description=Audience
CSV(VC1)	
Capture Scene #2	Description=ChinaConfRoom
VC2	Description=Speaker
	EncodingGroup=1
VC3	Description=Audience
CSV(VC2, VC3)	
Capture Scene #3	Description=USAConfRoom
VC4	Description=Audience
CSV(VC4)	
Capture Scene #4	
MCC1(VC1,VC3,VC4)	Policy=RoundRobin:1
	MaxCaptures=1
	EncodingGroup=1
	AllowSubset=True
MCC2(VC2)	MaxCaptures=1
	EncodingGroup=1
CSV2(MCC1,MCC2)	

Table 14: Advertisement sent to Endpoint F - Two Encodings

Therefore a Consumer could choose whether or not to have a separate speaker related stream and could choose which endpoints to see. If it wanted the second stream but not the Australian conference room it could indicate the following captures in the Configure message:

MCC1(VC3,VC4)	Encoding
VC2	Encoding

Table 15: MCU case: Consumer Response

12.3.2. Several MCCs in the same Advertisement

Multiple MCCs can be used where multiple streams are used to carry media from multiple endpoints. For example:

A conference has three endpoints D, E and F. Each end point has three video captures covering the left, middle and right regions of each conference room. The MCU receives the following advertisements from D and E.

Capture Scene #1	Description=AustralianConfRoom

VC1	CaptureArea=Left
	EncodingGroup=1

VC2	CaptureArea=Centre
	EncodingGroup=1

VC3	CaptureArea=Right
	EncodingGroup=1

CSV(VC1,VC2,VC3)	

Table 16: Advertisement received from Endpoint D

Capture Scene #1	Description=ChinaConfRoom

VC1	CaptureArea=Left
	EncodingGroup=1

VC2	CaptureArea=Centre
	EncodingGroup=1

VC3	CaptureArea=Right
	EncodingGroup=1

CSV(VC1,VC2,VC3)	

Table 17: Advertisement received from Endpoint E

The MCU wants to offer Endpoint F three Capture Encodings. Each Capture Encoding would contain all the Captures from either Endpoint D or Endpoint E depending based on the active speaker. The MCU sends the following Advertisement:

Capture Scene #1	Description=AustralianConfRoom
VC1	
VC2	
VC3	
CSV(VC1,VC2,VC3)	
Capture Scene #2	Description=ChinaConfRoom
VC4	
VC5	
VC6	
CSV(VC4,VC5,VC6)	
Capture Scene #3	
MCC1(VC1,VC4)	CaptureArea=Left
	MaxCaptures=1
	SynchronisationID=1
	EncodingGroup=1
MCC2(VC2,VC5)	CaptureArea=Centre
	MaxCaptures=1
	SynchronisationID=1
	EncodingGroup=1
MCC3(VC3,VC6)	CaptureArea=Right
	MaxCaptures=1
	SynchronisationID=1
	EncodingGroup=1
CSV(MCC1,MCC2,MCC3)	

Table 18: Advertisement sent to Endpoint F

12.3.3. Heterogeneous conference with switching and composition

Consider a conference between endpoints with the following characteristics:

Endpoint A - 4 screens, 3 cameras

Endpoint B - 3 screens, 3 cameras

Endpoint C - 3 screens, 3 cameras

Endpoint D - 3 screens, 3 cameras

Endpoint E - 1 screen, 1 camera

Endpoint F - 2 screens, 1 camera

Endpoint G - 1 screen, 1 camera

This example focuses on what the user in one of the 3-camera multi-screen endpoints sees. Call this person User A, at Endpoint A. There are 4 large display screens at Endpoint A. Whenever somebody at another site is speaking, all the video captures from that endpoint are shown on the large screens. If the talker is at a 3-camera site, then the video from those 3 cameras fills 3 of the screens. If the talker is at a single-camera site, then video from that camera fills one of the screens, while the other screens show video from other single-camera endpoints.

User A hears audio from the 4 loudest talkers.

User A can also see video from other endpoints, in addition to the current talker, although much smaller in size. Endpoint A has 4 screens, so one of those screens shows up to 9 other Media Captures in a tiled fashion. When video from a 3 camera endpoint appears in the tiled area, video from all 3 cameras appears together across the screen with correct spatial relationship among those 3 images.

```
      +---+---+---+ +-------------+ +-------------+ +-------------+
      |   |   |   | |             | |             | |             |
      +---+---+---+ |             | |             | |             |
      |   |   |   | |             | |             | |             |
      +---+---+---+ |             | |             | |             |
      |   |   |   | |             | |             | |             |
      +---+---+---+ +-------------+ +-------------+ +-------------+
```

Figure 8:   Endpoint A - 4 Screen Display

User B at Endpoint B sees a similar arrangement, except there are only 3 screens, so the 9 other Media Captures are spread out across the bottom of the 3 displays, in a picture-in-picture (PiP) format. When video from a 3 camera endpoint appears in the PiP area, video from all 3 cameras appears together across a single screen with correct spatial relationship.

```
              +-------------+ +-------------+ +-------------+
              |             | |             | |             |
              |             | |             | |             |
              |             | |             | |             |
              | +-+ +-+ +-+ | | +-+ +-+ +-+ | | +-+ +-+ +-+ |
              | +-+ +-+ +-+ | | +-+ +-+ +-+ | | +-+ +-+ +-+ |
              +-------------+ +-------------+ +-------------+
```

Figure 9:   Endpoint B - 3 Screen Display with PiPs

When somebody at a different endpoint becomes the current talker, then User A and User B both see the video from the new talker appear on their large screen area, while the previous talker takes one of the smaller tiled or PiP areas. The person who is the current talker doesn't see themselves; they see the previous talker in their large screen area.

One of the points of this example is that endpoints A and B each want to receive 3 capture encodings for their large display areas, and 9 encodings for their smaller areas. A and B are be able to each send the same Configure message to the MCU, and each receive the same conceptual Media Captures from the MCU. The differences are in how they are rendered and are purely a local matter at A and B.

The Advertisements for such a scenario are described below.

Capture Scene #1	Description=Endpoint x
VC1	EncodingGroup=1
VC2	EncodingGroup=1
VC3	EncodingGroup=1
AC1	EncodingGroup=2
CSV1(VC1, VC2, VC3)	
CSV2(AC1)	

Table 19: Advertisement received at the MCU from Endpoints A to D

Capture Scene #1	Description=Endpoint y
VC1	EncodingGroup=1
AC1	EncodingGroup=2
CSV1(VC1)	
CSV2(AC1)	

Table 20: Advertisement received at the MCU from Endpoints E to G

Rather than considering what is displayed CLUE concentrates more on what the MCU sends. The MCU doesn’t know anything about the number of screens an endpoint has.

As Endpoints A to D each advertise that three Captures make up a Capture Scene, the MCU offers these in a "site" switching mode. That is that there are three Multiple Content Captures (and Capture Encodings) each switching between Endpoints. The MCU switches in the applicable media into the stream based on voice activity. Endpoint A will not see a capture from itself.

Using the MCC concept the MCU would send the following Advertisement to endpoint A:

Capture Scene #1	Description=Endpoint B
VC4	CaptureArea=Left
VC5	CaptureArea=Center
VC6	CaptureArea=Right
AC1	
CSV(VC4,VC5,VC6)	
CSV(AC1)	
Capture Scene #2	Description=Endpoint C
VC7	CaptureArea=Left
VC8	CaptureArea=Center
VC9	CaptureArea=Right
AC2	
CSV(VC7,VC8,VC9)	
CSV(AC2)	
Capture Scene #3	Description=Endpoint D

VC10
VC11
VC12
AC3
CSV(VC10,VC11,VC12)
CSV(AC3)
	CaptureArea=Left
	CaptureArea=Center
	CaptureArea=Right

Capture Scene #4	Description=Endpoint E

VC13
AC4
CSV(VC13)
CSV(AC4)	

Capture Scene #5	Description=Endpoint F

VC14
AC5
CSV(VC14)
CSV(AC5)	

Capture Scene #6	Description=Endpoint G

VC15
AC6
CSV(VC15)
CSV(AC6)	

Table 21: Advertisement sent to endpoint A - Source Part

The above part of the Advertisement presents information about the sources to the MCC. The information is effectively the same as the received Advertisements except that there are no Capture Encodings associated with them and the identities have been re-numbered.

In addition to the source Capture information the MCU advertises "site" switching of Endpoints B to G in three streams.

Capture Scene #7	Description=Output3streammix

MCC1(VC4,VC7,VC10, VC13)	CaptureArea=Left
	MaxCaptures=1

	SynchronisationID=1
	Policy=SoundLevel:0
	EncodingGroup=1

MCC2(VC5,VC8,VC11, VC14)	CaptureArea=Center
	MaxCaptures=1
	SynchronisationID=1
	Policy=SoundLevel:0
	EncodingGroup=1

MCC3(VC6,VC9,VC12, VC15)	CaptureArea=Right
	MaxCaptures=1
	SynchronisationID=1
	Policy=SoundLevel:0
	EncodingGroup=1

MCC4() (for audio)	CaptureArea=whole scene
	MaxCaptures=1
	Policy=SoundLevel:0
	EncodingGroup=2

MCC5() (for audio)	CaptureArea=whole scene
	MaxCaptures=1
	Policy=SoundLevel:1
	EncodingGroup=2

MCC6() (for audio)	CaptureArea=whole scene
	MaxCaptures=1
	Policy=SoundLevel:2
	EncodingGroup=2

MCC7() (for audio)	CaptureArea=whole scene
	MaxCaptures=1
	Policy=SoundLevel:3
	EncodingGroup=2

CSV(MCC1,MCC2,MCC3)
CSV(MCC4,MCC5,MCC6, MCC7)	

Table 22: Advertisement send to endpoint A - switching part

The above part describes the switched 3 main streams that relate to site switching. MaxCaptures=1 indicates that only one Capture from the MCC is sent at a particular time. SynchronisationID=1 indicates that the source sending is synchronised. The provider can choose to group together VC13, VC14, and VC15 for the purpose of switching according to the SynchronisationID. Therefore when the provider switches one of them into an MCC, it can also switch the others even though they are not part of the same Capture Scene.

All the audio for the conference is included in this Scene #7. There isn’t necessarily a one to one relation between any audio capture and video capture in this scene. Typically a change in loudest talker will cause the MCU to switch the audio streams more quickly than switching video streams.

The MCU can also supply nine media streams showing the active and previous eight speakers. It includes the following in the Advertisement:

Capture Scene #8	Description=Output9stream

MCC8(VC4,VC5,VC6,VC7,
VC8,VC9,VC10,VC11,
VC12,VC13,VC14,VC15)
	MaxCaptures=1
	Policy=SoundLevel:0
	EncodingGroup=1

MCC9(VC4,VC5,VC6,VC7,
VC8,VC9,VC10,VC11,
VC12,VC13,VC14,VC15)
	MaxCaptures=1
	Policy=SoundLevel:1
	EncodingGroup=1

to	to

MCC16(VC4,VC5,VC6,VC7,
VC8,VC9,VC10,VC11,
VC12,VC13,VC14,VC15)
	MaxCaptures=1
	Policy=SoundLevel:8
	EncodingGroup=1

CSV(MCC8,MCC9,MCC10, MCC11,MCC12,MCC13, MCC14,MCC15,MCC16)	

Table 23: Advertisement sent to endpoint A - 9 switched part

The above part indicates that there are 9 capture encodings. Each of the Capture Encodings may contain any captures from any source site with a maximum of one Capture at a time. Which Capture is present is determined by the policy. The MCCs in this scene do not have any spatial attributes.

Note: The Provider alternatively could provide each of the MCCs above in its own Capture Scene.

If the MCU wanted to provide a composed Capture Encoding containing all of the 9 captures it could advertise in addition:

Capture Scene #9	Description=NineTiles

MCC13(MCC8,MCC9,MCC10,
MCC11,MCC12,MCC13,
MCC14,MCC15,MCC16)
	MaxCaptures=9
	EncodingGroup=1

CSV(MCC13)	

Table 24: Advertisement sent to endpoint A - 9 composed part

As MaxCaptures is 9 it indicates that the capture encoding contains information from 9 sources at a time.

The Advertisement to Endpoint B is identical to the above other than the captures from Endpoint A would be added and the captures from Endpoint B would be removed. Whether the Captures are rendered on a four screen display or a three screen display is up to the Consumer to determine. The Consumer wants to place video captures from the same original source endpoint together, in the correct spatial order, but the MCCs do not have spatial attributes. So the Consumer needs to associate incoming media packets with the original individual captures in the advertisement (such as VC4, VC5, and VC6) in order to know the spatial information it needs for correct placement on the screens. The Provider can use the RTCP CaptureId SDES item and associated RTP header extension, as described in [I-D.ietf-clue-rtp-mapping], to convey this information to the Consumer.

12.3.4. Heterogeneous conference with voice activated switching

This example illustrates how multipoint "voice activated switching" behavior can be realized, with an endpoint making its own decision about which of its outgoing video streams is considered the "active talker" from that endpoint. Then an MCU can decide which is the active talker among the whole conference.

Consider a conference between endpoints with the following characteristics:

Endpoint A - 3 screens, 3 cameras

Endpoint B - 3 screens, 3 cameras

Endpoint C - 1 screen, 1 camera

This example focuses on what the user at endpoint C sees. The user would like to see the video capture of the current talker, without composing it with any other video capture. In this example endpoint C is capable of receiving only a single video stream. The following tables describe advertisements from A and B to the MCU, and from the MCU to C, that can be used to accomplish this.

Capture Scene #1	Description=Endpoint x

VC1	CaptureArea=Left
	EncodingGroup=1

VC2	CaptureArea=Center
	EncodingGroup=1

VC3	CaptureArea=Right
	EncodingGroup=1

MCC1(VC1,VC2,VC3)	MaxCaptures=1
	CaptureArea=whole scene
	Policy=SoundLevel:0
	EncodingGroup=1

AC1	CaptureArea=whole scene
	EncodingGroup=2

CSV1(VC1, VC2, VC3)	
CSV2(MCC1)	
CSV3(AC1)	

Table 25: Advertisement received at the MCU from Endpoints A and B

Endpoints A and B are advertising each individual video capture, and also a switched capture MCC1 which switches between the other three based on who is the active talker. These endpoints do not advertise distinct audio captures associated with each individual video capture, so it would be impossible for the MCU (as a media consumer) to make its own determination of which video capture is the active talker based just on information in the audio streams.

Capture Scene #1	Description=conference

MCC1()	CaptureArea=Left
	MaxCaptures=1
	SynchronisationID=1
	Policy=SoundLevel:0
	EncodingGroup=1

MCC2()	CaptureArea=Center
	MaxCaptures=1
	SynchronisationID=1
	Policy=SoundLevel:0
	EncodingGroup=1

MCC3()	CaptureArea=Right
	MaxCaptures=1
	SynchronisationID=1
	Policy=SoundLevel:0
	EncodingGroup=1

MCC4()	CaptureArea=whole scene
	MaxCaptures=1
	Policy=SoundLevel:0
	EncodingGroup=1

MCC5() (for audio)	CaptureArea=whole scene
	MaxCaptures=1
	Policy=SoundLevel:0
	EncodingGroup=2

MCC6() (for audio)	CaptureArea=whole scene
	MaxCaptures=1
	Policy=SoundLevel:1
	EncodingGroup=2

CSV1(MCC1,MCC2,MCC3	
CSV2(MCC4)	
CSV3(MCC5,MCC6)	

Table 26: Advertisement sent from the MCU to C

The MCU advertises one scene, with four video MCCs. Three of them in CSV1 give a left, center, right view of the conference, with "site switching". MCC4 provides a single video capture representing a view of the whole conference. The MCU intends for MCC4 to be switched between all the other original source captures. In this example advertisement the MCU is not giving all the information about all the other endpoints' scenes and which of those captures is included in the MCCs. The MCU could include all that information if it wants to give the consumers more information, but it is not necessary for this example scenario.

The Provider advertises MCC5 and MCC6 for audio. Both are switched captures, with different SoundLevel policies indicating they are the top two dominant talkers. The Provider advertises CSV3 with both MCCs, suggesting the Consumer should use both if it can.

Endpoint C, in its configure message to the MCU, requests to receive MCC4 for video, and MCC5 and MCC6 for audio. In order for the MCU to get the information it needs to construct MCC4, it has to send configure messages to A and B asking to receive MCC1 from each of them, along with their AC1 audio. Now the MCU can use audio energy information from the two incoming audio streams from A and B to determine which of those alternatives is the current talker. Based on that, the MCU uses either MCC1 from A or MCC1 from B as the source of MCC4 to send to C.

13. Acknowledgements

Allyn Romanow and Brian Baldino were authors of early versions. Mark Gorzynski also contributed much to the initial approach. Many others also contributed, including Christian Groves, Jonathan Lennox, Paul Kyzivat, Rob Hansen, Roni Even, Christer Holmberg, Stephen Botzko, Mary Barnes, John Leslie, Paul Coverdale.

14. IANA Considerations

None.

15. Security Considerations

There are several potential attacks related to telepresence, and specifically the protocols used by CLUE, in the case of conferencing sessions, due to the natural involvement of multiple endpoints and the many, often user-invoked, capabilities provided by the systems.

An MCU involved in a CLUE session can experience many of the same attacks as that of a conferencing system such as that enabled by the XCON framework [RFC5239]. Examples of attacks include the following: an endpoint attempting to listen to sessions in which it is not authorized to participate, an endpoint attempting to disconnect or mute other users, and theft of service by an endpoint in attempting to create telepresence sessions it is not allowed to create. Thus, it is RECOMMENDED that an MCU implementing the protocols necessary to support CLUE, follow the security recommendations specified in the conference control protocol documents. In the case of CLUE, SIP is the conferencing protocol, thus the security considerations in [RFC4579] MUST be followed. Other security issues related to MCUs are discussed in the XCON framework [RFC5239]. The use of xCard with potentially sensitive information provides another reason to implement recommendations of section 11/[RFC5239].

One primary security concern, surrounding the CLUE framework introduced in this document, involves securing the actual protocols and the associated authorization mechanisms. These concerns apply to endpoint to endpoint sessions, as well as sessions involving multiple endpoints and MCUs. Figure 2 in section 5 provides a basic flow of information exchange for CLUE and the protocols involved.

As described in section 5, CLUE uses SIP/SDP to establish the session prior to exchanging any CLUE specific information. Thus the security mechanisms recommended for SIP [RFC3261], including user authentication and authorization, MUST be supported. In addition, the media MUST be secured. DTLS/SRTP MUST be supported and SHOULD be used unless the media, which is based on RTP, is secured by other means (see [RFC7201] [RFC7202]). Media security is also discussed in [I-D.ietf-clue-signaling] and [I-D.ietf-clue-rtp-mapping]. Note that SIP call setup is done before any CLUE specific information is available so the authentication and authorization are based on the SIP mechanisms. The entity that will be authenticated may use the Endpoint identity or the endpoint user identity; this is an application issue and not a CLUE specific issue.

A separate data channel is established to transport the CLUE protocol messages. The contents of the CLUE protocol messages are based on information introduced in this document. The CLUE data model [I-D.ietf-clue-data-model-schema] defines through an XML schema the syntax to be used. Some of the information which could possibly introduce privacy concerns is the xCard information as described in section 7.1.1.10. The decision about which xCard information to send in the CLUE channel is an application policy for point to point and multipoint calls based on the authenticated identity that can be the endpoint identity or the user of the endpoint. For example the telepresence multipoint application can authenticate a user before starting a CLUE exchange with the telepresence system and have a policy per user.

In addition, the (text) description field in the Media Capture attribute (section 7.1.1.6) could possibly reveal sensitive information or specific identities. The same would be true for the descriptions in the Capture Scene (section 7.3.1) and Capture Scene View (7.3.2) attributes. An implementation SHOULD give users control over what sensitive information is sent in an Advertisement. One other important consideration for the information in the xCard as well as the description field in the Media Capture and Capture Scene View attributes is that while the endpoints involved in the session have been authenticated, there is no assurance that the information in the xCard or description fields is authentic. Thus, this information MUST NOT be used to make any authorization decisions.

While other information in the CLUE protocol messages does not reveal specific identities, it can reveal characteristics and capabilities of the endpoints. That information could possibly uniquely identify specific endpoints. It might also be possible for an attacker to manipulate the information and disrupt the CLUE sessions. It would also be possible to mount a DoS attack on the CLUE endpoints if a malicious agent has access to the data channel. Thus, it MUST be possible for the endpoints to establish a channel which is secure against both message recovery and message modification. Further details on this are provided in the CLUE data channel solution document [I-D.ietf-clue-datachannel].

There are also security issues associated with the authorization to perform actions at the CLUE endpoints to invoke specific capabilities (e.g., re-arranging screens, sharing content, etc.). However, the policies and security associated with these actions are outside the scope of this document and the overall CLUE solution.

16. Changes Since Last Version

NOTE TO THE RFC-Editor: Please remove this section prior to publication as an RFC.

Changes from 24 to 25:

Updates from IESG review.

1. A few clarifications in various places.
2. Change references to RFC5239 and RFC5646 from informative to normative.

Changes from 23 to 24:

1. Updates to Security Considerations section.
2. Update version number of references to other CLUE documents in progress.

Changes from 22 to 23:

1. Updates to Security Considerations section.
2. Update version number of references to other CLUE documents in progress.
3. Change some "MAY" to "may".
4. Fix a few grammatical errors.

Changes from 21 to 22:

1. Add missing references.
2. Update version number of referenced working group drafts.
3. Minor updates for idnits issues.

Changes from 20 to 21:

1. Clarify CLUE can be useful for multi-stream non-telepresence cases.
2. Remove unnecessary ambiguous sentence about optional use of CLUE protocol.

3. Clarify meaning if Area of Capture is not specified.
4. Remove use of "conference" where it didn’t fit according to the definition. Use "CLUE session" or "meeting" instead.
5. Embedded Text Attribute: Remove restriction it is for video only.
6. Minor cleanup in section 12 examples.
7. Minor editorial corrections suggested by Christian Groves.

Changes from 19 to 20:

1. Define term "CLUE" in introduction.
2. Add MCC attribute Allow Subset Choice.
3. Remove phrase about reducing SDP size, replace with potentially saving consumer resources.
4. Change example of a CLUE exchange that does not require SDP exchange.
5. Language attribute uses RFC5646.
6. Change Member person type to Attendee. Add Observer type.
7. Clarify DTLS/SRTP MUST be supported.
8. Change SHOULD NOT to MUST NOT regarding using xCard or description information for authorization decisions.
9. Clarify definition of Global View.
10. Refer to signaling doc regarding interoperating with a device that does not support CLUE.
11. Various minor editorial changes from working group last call feedback.
12. Capitalize defined terms.

Changes from 18 to 19:

1. Remove the Max Capture Encodings media capture attribute.
2. Refer to RTP mapping document in the MCC example section.
3. Update references to current versions of drafts in progress.

Changes from 17 to 18:

1. Add separate definition of Global View List.
2. Add diagram for Global View List structure.
3. Tweak definitions of Media Consumer and Provider.

Changes from 16 to 17:

1. Ticket #59 - rename Capture Scene Entry (CSE) to Capture Scene View (CSV)

2. Ticket #60 - rename Global CSE List to Global View List

3. Ticket #61 - Proposal for describing the coordinate system. Describe it better, without conflicts if cameras point in different directions.

4. Minor clarifications and improved wording for Synchronisation Identity, MCC, Simultaneous Transmission Set.

5. Add definitions for CLUE-capable device and CLUE-enabled call, taken from the signaling draft.

6. Update definitions of Capture Device, Media Consumer, Media Provider, Endpoint, MCU, MCC.

7. Replace "middle box" with "MCU".

8. Explicitly state there can also be Media Captures that are not included in a Capture Scene View.

9. Explicitly state "A single Encoding Group MAY refer to encodings for different media types."

10. In example 12.1.1 add axes and audio captures to the diagram, and describe placement of microphones.

11. Add references to data model and signaling drafts.

12. Split references into Normative and Informative sections. Add heading number for references section.

Changes from 15 to 16:

1. Remove Audio Channel Format attribute

2. Add Audio Capture Sensitivity Pattern attribute

3. Clarify audio spatial information regarding point of capture and point on line of capture. Area of capture does not apply to audio.

4. Update section 12 example for new treatment of audio spatial information.

5. Clean up wording of some definitions, and various places in sections 5 and 10.

6. Remove individual encoding parameter paragraph from section 9.

7. Update Advertisement diagram.

8. Update Acknowledgements.

9. References to use cases and requirements now refer to RFCs.

10. Minor editorial changes.

Changes from 14 to 15:

1. Add "=" and "<=" qualifiers to MaxCaptures attribute, and clarify the meaning regarding switched and composed MCC.

2. Add section 7.3.3 Global Capture Scene Entry List, and a few other sentences elsewhere that refer to global CSE sets.

3. Clarify: The Provider MUST be capable of encoding and sending all Captures (*that have an encoding group*) in a single Capture Scene Entry simultaneously.

4. Add voice activated switching example in section 12.

5. Change name of attributes Participant Info/Type to Person Info/Type.

6. Clarify the Person Info/Type attributes have the same meaning regardless of whether or not the capture has a Presentation attribute.

7. Update example section 12.1 to be consistent with the rest of the document, regarding MCC and capture attributes.

8. State explicitly each CSE has a unique ID.

Changes from 13 to 14:

1. Fill in section for Security Considerations.

2. Replace Role placeholder with Participant Information, Participant Type, and Scene Information attributes.

3. Spatial information implies nothing about how constituent media captures are combined into a composed MCC.

4. Clean up MCC example in Section 12.3.3. Clarify behavior of tiled and PIP display windows. Add audio. Add new open issue about associating incoming packets to original source capture.

5. Remove editor’s note and associated statement about RTP multiplexing at end of section 5.

6. Remove editor’s note and associated paragraph about overloading media channel with both CLUE and non-CLUE usage, in section 5.

7. In section 10, clarify intent of media encodings conforming to SDP, even with multiple CLUE message exchanges. Remove associated editor’s note.

Changes from 12 to 13:

1. Added the MCC concept including updates to existing sections to incorporate the MCC concept. New MCC attributes: MaxCaptures, SynchronisationID and Policy.

2. Removed the "composed" and "switched" Capture attributes due to overlap with the MCC concept.

3. Removed the "Scene-switch-policy" CSE attribute, replaced by MCC and SynchronisationID.

4. Editorial enhancements including numbering of the Capture attribute sections, tables, figures etc.

Changes from 11 to 12:

1. Ticket #44. Remove note questioning about requiring a Consumer to send a Configure after receiving Advertisement.

2. Ticket #43. Remove ability for consumer to choose value of attribute for scene-switch-policy.

3. Ticket #36. Remove computational complexity parameter, MaxGroupPps, from Encoding Groups.

4. Reword the Abstract and parts of sections 1 and 4 (now 5) based on Mary’s suggestions as discussed on the list. Move part of the Introduction into a new section Overview & Motivation.

5. Add diagram of an Advertisement, in the Overview of the Framework/Model section.

6. Change Intended Status to Standards Track.

7. Clean up RFC2119 keyword language.

Changes from 10 to 11:

1. Add description attribute to Media Capture and Capture Scene Entry.

2. Remove contradiction and change the note about open issue regarding always responding to Advertisement with a Configure message.

3. Update example section, to cleanup formatting and make the media capture attributes and encoding parameters consistent with the rest of the document.

Changes from 09 to 10:

1. Several minor clarifications such as about SDP usage, Media Captures, Configure message.

2. Simultaneous Set can be expressed in terms of Capture Scene and Capture Scene Entry.

3. Removed Area of Scene attribute.

4. Add attributes from draft-groves-clue-capture-attr-01.

5. Move some of the Media Capture attribute descriptions back into this document, but try to leave detailed syntax to the data model. Remove the OUTSOURCE sections, which are already incorporated into the data model document.

Changes from 08 to 09:

1. Use "document" instead of "memo".

2. Add basic call flow sequence diagram to introduction.

3. Add definitions for Advertisement and Configure messages.

4. Add definitions for Capture and Provider.

5. Update definition of Capture Scene.

6. Update definition of Individual Encoding.

7. Shorten definition of Media Capture and add key points in the Media Captures section.

8. Reword a bit about capture scenes in overview.

9. Reword about labeling Media Captures.

10. Remove the Consumer Capability message.

11. New example section heading for media provider behavior

12. Clarifications in the Capture Scene section.

13. Clarifications in the Simultaneous Transmission Set section.

14. Capitalize defined terms.

15. Move call flow example from introduction to overview section

16. General editorial cleanup

17. Add some editors’ notes requesting input on issues

18. Summarize some sections, and propose details be outsourced to other documents.

Changes from 06 to 07:

1. Ticket #9. Rename Axis of Capture Point attribute to Point on Line of Capture. Clarify the description of this attribute.

2. Ticket #17. Add "capture encoding" definition. Use this new term throughout document as appropriate, replacing some usage of the terms "stream" and "encoding".

3. Ticket #18. Add Max Capture Encodings media capture attribute.

4. Add clarification that different capture scene entries are not necessarily mutually exclusive.

Changes from 05 to 06:

1. Capture scene description attribute is a list of text strings, each in a different language, rather than just a single string.

2. Add new Axis of Capture Point attribute.

3. Remove appendices A.1 through A.6.

4. Clarify that the provider must use the same coordinate system with same scale and origin for all coordinates within the same capture scene.

Changes from 04 to 05:

1. Clarify limitations of "composed" attribute.

2. Add new section "capture scene entry attributes" and add the attribute "scene-switch-policy".

3. Add capture scene description attribute and description language attribute.

4. Editorial changes to examples section for consistency with the rest of the document.

Changes from 03 to 04:

1. Remove sentence from overview - "This constitutes a significant change ..."

2. Clarify a consumer can choose a subset of captures from a capture scene entry or a simultaneous set (in section "capture scene" and "consumer’s choice...").

3. Reword first paragraph of Media Capture Attributes section.

4. Clarify a stereo audio capture is different from two mono audio captures (description of audio channel format attribute).

5. Clarify what it means when coordinate information is not specified for area of capture, point of capture, area of scene.

6. Change the term "producer" to "provider" to be consistent (it was just in two places).

7. Change name of "purpose" attribute to "content" and refer to RFC4796 for values.

8. Clarify simultaneous sets are part of a provider advertisement, and apply across all capture scenes in the advertisement.

9. Remove sentence about lip-sync between all media captures in a capture scene.

10. Combine the concepts of "capture scene" and "capture set" into a single concept, using the term "capture scene" to replace the previous term "capture set", and eliminating the original separate capture scene concept.

17. Normative References

[I-D.ietf-clue-datachannel]
Holmberg, C., "CLUE Protocol Data Channel", draft-ietf-clue-datachannel-11 (work in progress), November 2015.

[I-D.ietf-clue-data-model-schema]
Presta, R., Romano, S P., "An XML Schema for the CLUE data model", draft-ietf-clue-data-model-schema-11 (work in progress), October 2015.

[I-D.ietf-clue-protocol]
Presta, R. and S. Romano, "CLUE protocol", draft-ietf-clue-protocol-06 (work in progress), October 2015.

[I-D.ietf-clue-signaling]
Kyzivat, P., Xiao, L., Groves, C., Hansen, R., "CLUE Signaling", draft-ietf-clue-signaling-06 (work in progress), August 2015.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[RFC3264] Rosenberg, J., Schulzrinne, H., "An Offer/Answer Model with the 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., Jacobsen, V., Perkins, C., "SDP: Session Description Protocol", RFC 4566, July 2006.

[RFC4579] Johnston, A., Levin, O., "SIP Call Control - Conferencing for User Agents", RFC 4579, August 2006

[RFC5239] Barnes, M., Boulton, C., Levin, O., "A Framework for Centralized Conferencing", RFC 5239, June 2008.

[RFC5646] Phillips, A., Davis, M., "Tags for Identifying Languages", RFC 5646, September 2009.

[RFC6350] Perreault, S., "vCard Format Specification", RFC 6350, August 2011.

[RFC6351] Perreault, S., "xCard: vCard XML Representation", RFC 6351, August 2011.

18. Informative References

[I-D.ietf-clue-rtp-mapping]
Even, R., Lennox, J., "Mapping RP streams to CLUE media captures", draft-ietf-clue-rtp-mapping-05 (work in progress), October 2015.

[RFC4353] Rosenberg, J., "A Framework for Conferencing with the Session Initiation Protocol (SIP)", RFC 4353, February 2006.

[RFC5117] Westerlund, M. and S. Wenger, "RTP Topologies", RFC 5117, January 2008.

[RFC7201] Westerlund, M., Perkins, C., "Options for Securing RTP Sessions", RFC 7201, April 2014.

[RFC7202] Perkins, C., Westerlund, M., "Why RTP Does Not Mandate a Single Media Security Solution ", RFC 7202, April 2014.

[RFC7205] Romanow, A., Botzko, S., Duckworth, M., Even, R., "Use Cases for Telepresence Multistreams", RFC 7205, April 2014.

[RFC7262] Romanow, A., Botzko, S., Barnes, M., "Requirements for Telepresence Multistreams", RFC 7262, June 2014.

19. Authors’ Addresses

Mark Duckworth (editor)
Polycom
Andover, MA 01810
USA

Email: mark.duckworth@polycom.com

Andrew Pepperell
Acano
Uxbridge, England
UK

Email: apeppere@gmail.com

Stephan Wenger
Vidyo, Inc.
433 Hackensack Ave.
Hackensack, N.J. 07601
USA

Email: stewe@stewe.org

CLUE WG A. Romanow
Internet-Draft Cisco Systems
Intended status: Informational S. Botzko
Expires: June 15, 2014 M. Barnes
Polycom
December 12, 2013

Requirements for Telepresence Multi-Streams
draft-ietf-clue-telepresence-requirements-07.txt

Abstract

This memo discusses the requirements for specifications, that enable telepresence interoperability by describing behaviors and protocols for Controlling Multiple Streams for Telepresence (CLUE). In addition, the problem statement and related definitions are also covered herein.

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 June 15, 2014.

Copyright Notice

Copyright (c) 2013 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

Telepresence systems greatly improve collaboration. In a telepresence conference (as used herein), the goal is to create an environment that gives the users a feeling of (co-located) presence - the feeling that a local user is in the same room with other local users and the remote parties. Currently, systems from different vendors often do not interoperate because they do the same tasks differently, as discussed in the Problem Statement section below.

The approach taken in this memo is to set requirements for a future specification(s) that, when fulfilled by an implementation of the specification(s), provide for interoperability between IETF protocol based telepresence systems. It is anticipated that a solution for the requirements set out in this memo likely involves the exchange of adequate information about participating sites; information that is currently not standardized by the IETF.

The purpose of this document is to describe the requirements for a specification that enables interworking between different SIP-based [RFC3261] telepresence systems, by exchanging and negotiating appropriate information. In the context of the requirements in this document and related solution documents, this includes both point to point SIP sessions as well as SIP based conferences as described in the SIP conferencing framework [RFC4353] and the SIP based conference control [RFC4579] specifications. Non IETF protocol based systems, such as those based on ITU-T Rec. H.323, are out of scope. These requirements are for the specification, they are not requirements on the telepresence systems implementing the solution/protocol that will be specified.

Telepresence systems of different vendors, today, can follow radically different architectural approaches while offering a similar user experience. CLUE will not dictate telepresence architectural and implementation choices; however it will describe a protocol architecture for CLUE and how it relates to other protocols. CLUE enables interoperability between telepresence systems by exchanging information about the systems’ characteristics. Systems can use this information to control their behavior to allow for interoperability between those systems.

A telepresence session requires at least one sending and one receiving endpoint. Multiparty telepresence sessions include more than two endpoints, and centralized infrastructure such as Multipoint Control Units (MCUs) or equivalent. CLUE specifies the syntax, semantics, and control flow of information to enable the best possible user experience at those endpoints.

Sending endpoints, or MCUs, are not mandated to use any of the CLUE specifications that describe their capabilities, attributes, or behavior. Similarly, it is not envisioned that endpoints or MCUs must ever take into account information received. However, by making available as much information as possible, and by taking into account as much information as has been received or exchanged, MCUs and endpoints are expected to select operation modes that enable the best possible user experience under their constraints.

The document structure is as follows: Definitions are set out, followed by a description of the problem of telepresence interoperability that led to this work. Then the requirements to a specification addressing the current shortcomings are enumerated and discussed.

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].

3. Definitions

The following terms are used throughout this document and serve as reference for other documents.

Audio Mixing: refers to the accumulation of scaled audio signals to produce a single audio stream. See RTP Topologies, [RFC5117].

Conference: used as defined in [RFC4353], A Framework for Conferencing within the Session Initiation Protocol (SIP).

Endpoint: The logical point of final termination through receiving, decoding and rendering, and/or initiation through capturing, encoding, and sending of media streams. An endpoint consists of one or more physical devices which source and sink media streams, and exactly one [RFC4353] Participant (which, in turn, includes exactly one SIP User Agent). In contrast to an endpoint, an MCU may also send and receive media streams, but it is not the initiator nor the final terminator in the sense that Media is Captured or Rendered. Endpoints can be anything from multiscreen/multicamera rooms to handheld devices.

Endpoint Characteristics: include placement of Capture and Rendering Devices, capture/render angle, resolution of cameras and screens, spatial location and mixing parameters of microphones. Endpoint characteristics are not specific to individual media streams sent by the endpoint.

Layout: How rendered media streams are spatially arranged with respect to each other on a single screen/mono audio telepresence endpoint, and how rendered media streams are arranged with respect to each other on a multiple screen/speaker telepresence endpoint. Note that audio as well as video is encompassed by the term layout--in other words, included is the placement of audio streams on speakers as well as video streams on video screens.

Local: Sender and/or receiver physically co-located ("local") in the context of the discussion.

MCU: Multipoint Control Unit (MCU) - a device that connects two or more endpoints together into one single multimedia conference [RFC5117]. An MCU may include a Mixer [RFC4353].

Media: Any data that, after suitable encoding, can be conveyed over RTP, including audio, video or timed text.

Model: a set of assumptions a telepresence system of a given vendor adheres to and expects the remote telepresence system(s) also to adhere to.

Remote: Sender and/or receiver on the other side of the communication channel (depending on context); not Local. A remote can be an Endpoint or an MCU.

Render: the process of generating a representation from a media, such as displayed motion video or sound emitted from loudspeakers.

Telepresence: an environment that gives non co-located users or user groups a feeling of (co-located) presence - the feeling that a Local user is in the same room with other Local users and the Remote parties. The inclusion of Remote parties is achieved through multimedia communication including at least audio and video signals of high fidelity.

4. Problem Statement

In order to create a "being there" experience characteristic of telepresence, media inputs need to be transported, received, and coordinated between participating systems. Different telepresence systems take diverse approaches in crafting a solution, or, they implement similar solutions quite differently.

They use disparate techniques, and they describe, control and negotiate media in dissimilar fashions. Such diversity creates an interoperability problem. The same issues are solved in different ways by different systems, so that they are not directly interoperable. This makes interworking difficult at best and sometimes impossible.

Worse, many telepresence systems use proprietary protocol extensions to solve telepresence-related problems, even if those extensions are based on common standards such as SIP.

Some degree of interworking between systems from different vendors is possible through transcoding and translation. This requires additional devices, which are expensive, often not entirely automatic, and they sometimes introduce unwelcome side effects, such as additional delay or degraded performance. Specialized knowledge is currently required to operate a telepresence conference with endpoints from different vendors, for example to configure transcoding and translating devices. Often such conferences do not start as planned, or are interrupted by difficulties that arise.

The general problem that needs to be solved can be described as follows. Today, each endpoint sends audio and video captures based upon an implicitly assumed model for rendering a realistic depiction based on this information. If all endpoints are manufactured by the same vendor, they work with the same model and render the information according to the model implicitly assumed by the vendor. However, if the devices are from different vendors, the models they each use for rendering presence can and usually do differ. The result can be that the telepresence systems actually connect, but the user experience suffers, for example because one system assumes that the first video stream is captured from the right camera, whereas the other assumes the first video stream is captured from the left camera.

If Alice and Bob are at different sites, Alice needs to tell Bob about the camera and sound equipment arrangement at her site so that Bob’s receiver can create an accurate rendering of her site. Alice and Bob need to agree on what the salient characteristics are as well as how to represent and communicate them. Characteristics may include number, placement, capture/render angle, resolution of cameras and screens, spatial location and audio mixing parameters of microphones.

The telepresence multi-stream work seeks to describe the sender situation in a way that allows the receiver to render it realistically even though it may have a different rendering model than the sender.

5. Requirements

Although some aspects of these requirements can be met by existing technology, such as SDP, they are stated here to have a complete record of what the requirements for CLUE are, whether new work is needed or they can be met by existing technology. Figuring this out will be part of the solution development, rather than part of the requirements. Note, the term "solution" is used in these requirements to mean the protocol specifications, including extensions to existing protocols as well as any new protocols, developed to support the use cases. The solution can introduce additional functionality that isn’t mapped directly to these requirements - e.g., the detailed information carried in the signaling protocol(s). In cases where the requirements are directly related to a specific use case, a reference to the use case is provided.

REQMT-1: The solution MUST support a description of the spatial arrangement of source video images sent in video streams which enables a satisfactory reproduction at the receiver of the original scene. This applies to each site in a point to point or a multipoint meeting and refers to the spatial ordering within a site, not to the ordering of images between sites.

Use case point to point symmetric, and all other use cases.

REQMT-1a: The solution MUST support a means of allowing the preservation of the order of images in the captured scene. For example, if John is to Susan’s right in the image capture, John is also to Susan’s right in the rendered image.

REQMT-1b: The solution MUST support a means of allowing the preservation of order of images in the scene in two dimensions - horizontal and vertical.

REQMT-1c: The solution MUST support a means to identify the point of capture of individual video captures in three dimensions.

REQMT-1d: The solution MUST support a means to identify the area of coverage of individual video captures in three dimensions.

REQMT-2: The solution MUST support a description of the spatial arrangement of captured source audio sent in audio streams which enables a satisfactory reproduction at the receiver in a spatially correct manner. This applies to each site in a point to point or a multipoint meeting and refers to the spatial ordering within a site, not the ordering of channels between sites.

Use case point to point symmetric, and all use cases, especially heterogeneous.

REQMT-2a: The solution MUST support a means of preserving the spatial order of audio in the captured scene. For example, if John sounds as if he is at Susan’s right in the captured audio, John voice is also placed at Susan’s right in the rendered image.

REQMT-2b: The solution MUST support a means to identify the number and spatial arrangement of audio channels including monaural, stereophonic (2.0), and 3.0 (left, center, right) audio channels.

REQMT-2c: The solution MUST support a means to identify the point of capture of individual audio captures in three dimensions.

REQMT-2d: The solution MUST support a means to identify the area of coverage of individual audio captures in three dimensions.

REQMT-3: The solution MUST enable individual audio streams to be associated with one or more video image captures, and individual video image captures to be associated with one or more audio captures, for the purpose of rendering proper position.

Use case is point to point symmetric, and all use cases.

REQMT-4: The solution MUST enable interoperability between endpoints that have a different number of similar devices. For example, one endpoint may have 1 screen, 1 speaker, 1 camera, 1 mic, and another endpoint may have 3 screens, 2 speakers, 3 cameras and 2 microphones. Or, in a multi-point conference, one endpoint may have one screen, another may have 2 screens and a third may have 3 screens. This includes endpoints where the number of devices of a given type is zero.

Use case is asymmetric point to point and multipoint.

REQMT-5: The solution MUST support means of enabling interoperability between telepresence endpoints where cameras are of different picture aspect ratios.

REQMT-6: The solution MUST provide scaling information which enables rendering of a video image at the actual size of the captured scene.

REQMT-7: The solution MUST support means of enabling interoperability between telepresence endpoints where displays are of different resolutions.

REQMT-8: The solution MUST support methods for handling different bit rates in the same conference.

REQMT-9: The solution MUST support means of enabling interoperability between endpoints that send and receive different numbers of media streams.

Use case heterogeneous and multipoint.

REQMT-10: The solution MUST ensure that endpoints that support telepresence extensions can establish a session with a SIP endpoint that does not support the telepresence extensions. For example, in the case of a SIP endpoint that supports a single audio and a single video stream, an endpoint that supports the telepresence extensions would setup a session with a single audio and single video stream using existing SIP and SDP mechanisms.

REQMT-11: The solution MUST support a mechanism for determining whether or not an endpoint or MCU is capable of telepresence extensions.

REQMT-12: The solution MUST support a means to enable more than two endpoints to participate in a teleconference.

Use case multipoint.

REQMT-13: The solution MUST support both transcoding and switching approaches to providing multipoint conferences.

REQMT-14: The solution MUST support mechanisms to allow media from one source endpoint or/and multiple source endpoints to be sent to a remote endpoint at a particular point in time. Which media is sent at a point in time may be based on local policy.

REQMT-15: The solution MUST provide mechanisms to support the following:

* Presentations with different media sources

* Presentations for which the media streams are visible to all endpoints

* Multiple, simultaneous presentation media streams, including presentation media streams that are spatially related to each other.

Use case is presentation.

REQMT-16: The specification of any new protocols for the solution MUST provide extensibility mechanisms.

REQMT-17: The solution MUST support a mechanism for allowing information about media captures to change during a conference.

REQMT-18: The solution MUST provide a mechanism for the secure exchange of information about the media captures.

6. Acknowledgements

This draft has benefitted from all the comments on the mailing list and a number of discussions. So many people contributed that it is not possible to list them all. However, the comments provided by Roberta Presta, Christian Groves and Paul Coverdale during WGLC were particularly helpful in completing the WG document.

7. IANA Considerations

There are no IANA considerations associated with this specification.

8. Security Considerations

Requirement REQMT-18 identifies the need to securely transport the information about media captures. It is important to note that session setup for a telepresence session will use SIP for basic session setup and either SIP or CCMP for a multi-party telepresence session. Information carried in the SIP signaling can be secured by the SIP security mechanisms as defined in [RFC3261]. In the case of conference control using CCMP, the security model and mechanisms as defined in the XCON Framework [RFC5239] and CCMP [RFC6503] documents would meet the requirement. Any additional signaling mechanism used to transport the information about media captures would need to define the mechanisms by the which the information is secure. The details for the mechanisms needs to be defined and described in the CLUE framework document and related solution document(s).

9. Informative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[RFC4353] Rosenberg, J., "A Framework for Conferencing with the Session Initiation Protocol (SIP)", RFC 4353, February 2006.

[RFC4579] Johnston, A. and O. Levin, "Session Initiation Protocol (SIP) Call Control - Conferencing for User Agents", BCP 119, RFC 4579, August 2006.

[RFC5117] Westerlund, M. and S. Wenger, "RTP Topologies", RFC 5117, January 2008.

[RFC5239] Barnes, M., Boulton, C., and O. Levin, "A Framework for Centralized Conferencing", RFC 5239, June 2008.

[RFC6503] Barnes, M., Boulton, C., Romano, S., and H. Schulzrinne, "Centralized Conferencing Manipulation Protocol", RFC 6503, March 2012.

Appendix A. Changes From Earlier Versions

Note to the RFC-Editor: please remove this section prior to publication as an RFC.

A.1. Changes from draft -06

Addressing IETF LC comments/editorial nits resulting in the following changes:

- Included expansion of CLUE in the abstract.

- Deleted definitions for "Left" and "Right".

- Section 5 - clarified that solution = protocol specifications to support requirements.

- REQMT-1d, REQMT-2d: Changed term "extent" to "area of coverage"

- REQMT-10 - clarified requirement with regards to interworking with non-CLUE endpoints

- REQMT-15 - reworded to be more specific and normative

- REQMT-16 - expanded on what is meant by "extensibility"

A.2. Changes from draft -05

Addressing WGLC comments resulting in the following changes:

- REQMT-12: Changed term "site" to "endpoint"

- Intro: clarified that SIP based conferencing also is relevant to CLUE.

- Intro: clarified that while CLUE doesn’t dictate implementation choices, it does describe a framework for the protocol solution.

- Clarified that mapping to use cases isn’t comprehensive (i.e., only done when there is a direct correlation).

- Added text that the requirements do not reflect all those required for the solution - i.e., the solution can provide more functionality as needed.

- Editorial nits and clarifications - changed lc "must" to UC (REQMT-17).

A.3. Changes from draft -04

- Removed REQMT-2c, related to issue #37 in the tracker.

- Deleted REQMT-3b. Condensed REQMT-3 to subsume REQMT-3a. This is related to Issue #38 in the tracker.

- Updated REQMT-14 based on (mailing list) resolution of Issue #39.

- Deleted OPEN issue section as those were transferred to the ID tracker and have been resolved either by changes to this document or to earlier versions of the document

A.4. Changes from draft -03

- Added a tad more text to the security section Paragraph 18.

A.5. Changes from draft -02

- Updated IANA section - i.e., no IANA registrations required.

- Added security requirement Paragraph 18.

- Added some initial text to the security section.

A.6. Changes from draft -01

- Cleaned up the Problem Statement section, re-worded.

- Added Requirement Paragraph 17 in response to WG Issue #4 to make a requirement for dynamically changing information. Approved by WG

- Added requirements #1.c and #1.d. Approved by WG

- Added requirements #2.d and #2.e. Approved by WG

A.7. Changes From Draft -00

- Requirement #2, The solution MUST support a means to identify monaural, stereophonic (2.0), and 3.0 (left, center, right) audio channels.

  changed to

  The solution MUST support a means to identify the number and spatial arrangement of audio channels including monaural, stereophonic (2.0), and 3.0 (left, center, right) audio channels.

- Added back references to the Use case document.
  - Requirement #1 Use case point to point symmetric, and all other use cases.
  - Requirement #2 Use case point to point symmetric, and all use cases, especially heterogeneous.
  - Requirement #3 Use case point to point symmetric, and all use cases.
  - Requirement #4 Use case is asymmetric point to point, and multipoint.
  - Requirement #9 Use case heterogeneous and multipoint.
  - Requirement #12 Use case multipoint.

Authors' Addresses

Allyn Romanow
Cisco Systems
San Jose, CA 95134
USA

Email: allyn@cisco.com

Stephen Botzko
Polycom
Andover, MA 01810
US

Email: stephen.botzko@polycom.com

Mary Barnes
Polycom

Email: mary.ietf.barnes@gmail.com

CLUE WG
Internet-Draft
Intended status: Informational
Expires: August 9, 2014

A. Romanow
Cisco
S. Botzko

M. Duckworth
Polycom
R. Even, Ed.
Huawei Technologies

February 5, 2014

Use Cases for Telepresence Multi-streams
draft-ietf-clue-telepresence-use-cases-09.txt

Abstract

Telepresence conferencing systems seek to create an environment that gives non co-located users or user groups a feeling of co-located presence through multimedia communication including at least audio and video signals of high fidelity. A number of techniques for handling audio and video streams are used to create this experience. When these techniques are not similar, interoperability between different systems is difficult at best, and often not possible. Conveying information about the relationships between multiple streams of media would allow senders and receivers to make choices to allow telepresence systems to interwork. This memo describes the most typical and important use cases for sending multiple streams in a telepresence conference.

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 August 9, 2014.

Copyright Notice

Copyright (c) 2014 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

Telepresence applications try to provide a "being there" experience for conversational video conferencing. Often this telepresence application is described as "immersive telepresence" in order to distinguish it from traditional video conferencing, and from other forms of remote presence not related to conversational video conferencing, such as avatars and robots. The salient characteristics of telepresence are often described as: actual sized, immersive video, preserving interpersonal interaction and allowing non-verbal communication.

Although telepresence systems are based on open standards such as RTP [RFC3550], SIP [RFC3261], H.264 [H.264], and the H.323[ITU.H323]suite of protocols, they cannot easily interoperate with each other without operator assistance and expensive additional equipment which translates from one vendor’s protocol to another.

The basic features that give telepresence its distinctive characteristics are implemented in disparate ways in different systems. Currently Telepresence systems from diverse vendors interoperate to some extent, but this is not supported in a standards based fashion. Interworking requires that translation and transcoding devices be included in the architecture. Such devices increase latency, reducing the quality of interpersonal interaction. Use of these devices is often not automatic; it frequently requires substantial manual configuration and a detailed understanding of the nature of underlying audio and video streams. This state of affairs is not acceptable for the continued growth of telepresence - these systems should have the same ease of interoperability as do telephones. Thus, a standard way of describing the multiple streams constituting the media flows and the fundamental aspects of their behavior, would allow telepresence systems to interwork.

This document presents a set of use cases describing typical scenarios. Requirements will be derived from these use cases in a separate document. The use cases are described from the viewpoint of the users. They are illustrative of the user experience that needs to be supported. It is possible to implement these use cases in a variety of different ways.

Many different scenarios need to be supported. This document describes in detail the most common and basic use cases. These will cover most of the requirements. There may be additional scenarios that bring new features and requirements which can be used to extend the initial work.

Point-to-point and Multipoint telepresence conferences are considered. In some use cases, the number of screens is the same at all sites, in others, the number of screens differs at different sites. Both use cases are considered. Also included is a use case describing display of presentation material or content.

The multipoint use cases may include a variety of systems from conference room systems to handheld devices and such a use case is described in the document.

The document structure is as follows: Section 2 gives an overview of scenarios, and Section 3 describes use cases.

2. Telepresence Scenarios Overview

This section describes the general characteristics of the use cases and what the scenarios are intended to show. The typical setting is a business conference, which was the initial focus of telepresence. Recently consumer products are also being developed. We specifically do not include in our scenarios the physical infrastructure aspects of telepresence, such as room construction, layout and decoration. Furthermore, these use cases do not describe all the aspects needed to create the best user experience (for example the human factors).

We also specifically do not attempt to precisely define the boundaries between telepresence systems and other systems, nor do we attempt to identify the "best" solution for each presented scenario.

Telepresence systems are typically composed of one or more video cameras and encoders and one or more display screens of large size (diagonal around 60"). Microphones pick up sound and audio codec(s) and produce one or more audio streams. The cameras used to capture the telepresence users are referred to as participant cameras (and likewise for screens). There may also be other cameras, such as for document display. These will be referred to as presentation or content cameras, which generally have different formats, aspect ratios, and frame rates from the participant cameras. The presentation streams may be shown on participant screen, or on auxiliary display screens. A user’s computer may also serve as a virtual content camera, generating an animation or playing a video for display to the remote participants.

We describe such a telepresence system as sending one or more video streams, audio streams, and presentation streams to the remote system(s).

The fundamental parameters describing today’s typical telepresence scenarios include:

1. The number of participating sites

2. The number of visible seats at a site

3. The number of cameras

4. The number and type of microphones

5. The number of audio channels

6. The screen size

7. The screen capabilities - such as resolution, frame rate, aspect ratio

8. The arrangement of the screens in relation to each other

9. The number of primary screens at each sites

10. Type and number of presentation screens

11. Multipoint conference display strategies - for example, the camera-to-screen mappings may be static or dynamic

12. The camera point of capture.

13. The cameras fields of view and how they spatially relate to each other.

As discussed in the introduction, the basic features that give telepresence its distinctive characteristics are implemented in disparate ways in different systems.

There is no agreed upon way to adequately describe the semantics of how streams of various media types relate to each other. Without a standard for stream semantics to describe the particular roles and activities of each stream in the conference, interoperability is cumbersome at best.

In a multiple screen conference, the video and audio streams sent from remote participants must be understood by receivers so that they can be presented in a coherent and life-like manner. This includes the ability to present remote participants at their actual size for their apparent distance, while maintaining correct eye contact, gesticular cues, and simultaneously providing a spatial audio sound stage that is consistent with the displayed video.

The receiving device that decides how to render incoming information needs to understand a number of variables such as the spatial position of the speaker, the field of view of the cameras, the camera zoom, which media stream is related to each of the screens, etc. It is not simply that individual streams must be adequately described, to a large extent this already exists, but rather that the semantics of the relationships between the streams must be communicated. Note that all of this is still required even if the basic aspects of the streams, such as the bit rate, frame rate, and aspect ratio, are known. Thus, this problem has aspects considerably beyond those encountered in interoperation of single camera/screen video conferencing systems.

3. Use Case Scenarios

The use case scenarios focus on typical implementations. There are a number of possible variants for these use cases, for example, the audio supported may differ at the end points (such as mono or stereo versus surround sound), etc.

Many of these systems offer a full conference room solution where local participants sit at one side of a table and remote participants are displayed as if they are sitting on the other side of the table. The cameras and screens are typically arranged to provide a panoramic (left to right from the local user view point) view of the remote room.

The sense of immersion and non-verbal communication is fostered by a number of technical features, such as:

1. Good eye contact, which is achieved by careful placement of participants, cameras and screens.

2. Camera field of view and screen sizes are matched so that the images of the remote room appear to be full size.

3. The left side of each room is presented on the right screen at the far end; similarly the right side of the room is presented on the left screen. The effect of this is that participants of each site appear to be sitting across the table from each other. If two participants on the same site glance at each other, all participants can observe it. Likewise, if a participant at one site gestures to a participant on the other site, all participants observe the gesture itself and the participants it includes.

3.1. Point to point meeting: symmetric

In this case each of the two sites has an identical number of screens, with cameras having fixed fields of view, and one camera for each screen. The sound type is the same at each end. As an example, there could be 3 cameras and 3 screens in each room, with stereo sound being sent and received at each end.

Each screen is paired with a corresponding camera. Each camera / screen pair is typically connected to a separate codec, producing a video encoded stream for transmission to the remote site, and receiving a similarly encoded stream from the remote site.

Each system has one or multiple microphones for capturing audio. In some cases, stereophonic microphones are employed. In other systems, a microphone may be placed in front of each participant (or pair of participants). In typical systems all the microphones are connected to a single codec that sends and receives the audio streams as either stereo or surround sound. The number of microphones and the number of audio channels are often not the same as the number of cameras. Also the number of microphones is often not the same as the number of loudspeakers.

The audio may be transmitted as multi-channel (stereo/surround sound) or as distinct and separate monophonic streams. Audio levels should be matched, so the sound levels at both sites are identical. Loudspeaker and microphone placements are chosen so that the sound "stage" (orientation of apparent audio sources) is coordinated with the video. That is, if a participant at one site speaks, the participants at the remote site perceive her voice as originating from her visual image. In order to accomplish this, the audio needs to be mapped at the received site in the same fashion as the video. That is, audio received from the right side of the room needs to be output from loudspeaker(s) on the left side at the remote site, and vice versa.

3.2. Point to point meeting: asymmetric

In this case, each site has a different number of screens and cameras than the other site. The important characteristic of this scenario is that the number of screens is different between the two sites. This creates challenges which are handled differently by different telepresence systems.

This use case builds on the basic scenario of 3 screens to 3 screens. Here, we use the common case of 3 screens and 3 cameras at one site, and 1 screen and 1 camera at the other site, connected by a point to point call. The screen sizes and camera fields of view at both sites are basically similar, such that each camera view is designed to show two people sitting side by side. Thus the 1 screen room has up to 2 people seated at the table, while the 3 screen room may have up to 6 people at the table.

The basic considerations of defining left and right and indicating relative placement of the multiple audio and video streams are the same as in the 3-3 use case. However, handling the mismatch between the two sites of the number of screens and cameras requires more complicated maneuvers.

For the video sent from the 1 camera room to the 3 screen room, usually what is done is to simply use 1 of the 3 screens and keep the second and third screens inactive or, for example, put up the current date. This would maintain the "full size" image of the remote side.

For the other direction, the 3 camera room sending video to the 1 screen room, there are more complicated variations to consider. Here are several possible ways in which the video streams can be handled.

1. The 1 screen system might simply show only 1 of the 3 camera images, since the receiving side has only 1 screen. Two people are seen at full size, but 4 people are not seen at all. The choice of which 1 of the 3 streams to display could be fixed, or could be selected by the users. It could also be made automatically based on who is speaking in the 3 screen room, such that the people in the 1 screen room always see the person who is speaking. If the automatic selection is done at the sender, the transmission of streams that are not displayed could be suppressed, which would avoid wasting bandwidth.

2. The 1 screen system might be capable of receiving and decoding all 3 streams from all 3 cameras. The 1 screen system could then compose the 3 streams into 1 local image for display on the single screen. All six people would be seen, but smaller than full size. This could be done in conjunction with reducing the image resolution of the streams, such that encode/decode resources and bandwidth are not wasted on streams that will be downsized for display anyway.

3. The 3 screen system might be capable of including all 6 people in a single stream to send to the 1 screen system. For example, it could use PTZ (Pan Tilt Zoom) cameras to physically adjust the cameras such that 1 camera captures the whole room of six people. Or it could recompose the 3 camera images into 1 encoded stream to send to the remote site. These variations also show all six people, but at a reduced size.

4. Or, there could be a combination of these approaches, such as simultaneously showing the speaker in full size with a composite of all the 6 participants in smaller size.

The receiving telepresence system needs to have information about the content of the streams it receives to make any of these decisions. If the systems are capable of supporting more than one strategy, there needs to be some negotiation between the two sites to figure out which of the possible variations they will use in a specific point to point call.

3.3. Multipoint meeting

In a multipoint telepresence conference, there are more than two sites participating. Additional complexity is required to enable media streams from each participant to show up on the screens of the other participants.

Clearly, there are a great number of topologies that can be used to display the streams from multiple sites participating in a conference.

One major objective for telepresence is to be able to preserve the "Being there" user experience. However, in multi-site conferences it is often (in fact usually) not possible to simultaneously provide full size video, eye contact, common perception of gestures and gaze by all participants. Several policies can be used for stream distribution and display: all provide good results but they all make different compromises.

One common policy is called site switching. Let’s say the speaker is at site A and everyone else are at various "remote" sites. When the room at site A shown, all the camera images from site A are forwarded to the remote sites. Therefore at each receiving remote site, all the screens display camera images from site A. This can be used to preserve full size image display, and also provide full visual context of the displayed far end, site A. In site switching, there is a fixed relation between the cameras in each room and the screens in remote rooms. The room or participants being shown is switched from time to time based on who is speaking or by manual control, e.g., from site A to site B.

Segment switching is another policy choice. Still using site A as where the speaker is, and "remote" to refer to all the other sites, in segment switching, rather than sending all the images from site A, only the speaker at site A is shown. The camera images of the current speaker and previous speakers (if any) are forwarded to the other sites in the conference. Therefore the screens in each site are usually displaying images from different remote sites - the current speaker at site A and the previous ones. This strategy can be used to preserve full size image display, and also capture the non-verbal communication between the speakers. In segment switching, the display depends on the activity in the remote rooms - generally, but not necessarily based on audio / speech detection).

A third possibility is to reduce the image size so that multiple camera views can be composited onto one or more screens. This does not preserve full size image display, but provides the most visual context (since more sites or segments can be seen). Typically in this case the display mapping is static, i.e., each part of each room is shown in the same location on the display screens throughout the conference.

Other policies and combinations are also possible. For example, there can be a static display of all screens from all remote rooms, with part or all of one screen being used to show the current speaker at full size.

3.4. Presentation

In addition to the video and audio streams showing the participants, additional streams are used for presentations.

In systems available today, generally only one additional video stream is available for presentations. Often this presentation stream is half-duplex in nature, with presenters taking turns. The presentation stream may be captured from a PC screen, or it may come from a multimedia source such as a document camera, camcorder or a DVD. In a multipoint meeting, the presentation streams for the currently active presentation are always distributed to all sites in the meeting, so that the presentations are viewed by all.

Some systems display the presentation streams on a screen that is mounted either above or below the three participant screens. Other systems provide screens on the conference table for observing presentations. If multiple presentation screens are used, they generally display identical content. There is considerable variation in the placement, number, and size or presentation screens.

In some systems presentation audio is pre-mixed with the room audio. In others, a separate presentation audio stream is provided (if the presentation includes audio).

In H.323[ITU.H323] systems, H.239[ITU.H239] is typically used to control the video presentation stream. In SIP systems, similar control mechanisms can be provided using BFCP [RFC4582] for presentation token. These mechanisms are suitable for managing a single presentation stream.

Although today’s systems remain limited to a single video presentation stream, there are obvious uses for multiple presentation streams:

1. Frequently the meeting convener is following a meeting agenda, and it is useful for her to be able to show that agenda to all participants during the meeting. Other participants at various remote sites are able to make presentations during the meeting, with the presenters taking turns. The presentations and the agenda are both shown, either on separate screens, or perhaps re-scaled and shown on a single screen.

2. A single multimedia presentation can itself include multiple video streams that should be shown together. For instance, a presenter may be discussing the fairness of media coverage. In addition to slides which support the presenter's conclusions, she also has video excerpts from various news programs which she shows to illustrate her findings. She uses a DVD player for the video excerpts so that she can pause and reposition the video as needed.

3. An educator who is presenting a multi-screen slide show. This show requires that the placement of the images on the multiple screens at each site be consistent.

There are many other examples where multiple presentation streams are useful.

3.5. Heterogeneous Systems

It is common in meeting scenarios for people to join the conference from a variety of environments, using different types of endpoint devices. A multi-screen immersive telepresence conference may include someone on a PC-based video conferencing system, a participant calling in by phone, and (soon) someone on a handheld device.

What experience/view will each of these devices have?

Some may be able to handle multiple streams and others can handle only a single stream. (We are not here talking about legacy systems, but rather systems built to participate in such a conference, although they are single stream only.) In a single video stream , the stream may contain one or more compositions depending on the available screen space on the device. In most cases an intermediate transcoding device will be relied upon to produce a single stream, perhaps with some kind of continuous presence.

Bit rates will vary - the handheld and phone having lower bit rates than PC and multi-screen systems.

Layout is accomplished according to different policies. For example, a handheld and PC may receive the active speaker stream. The decision can either be made explicitly by the receiver or by the sender if it can receive some kind of rendering hint. The same is true for audio -- i.e., that it receives a mixed stream or a number of the loudest speakers if mixing is not available in the network.

For the PC based conferencing participant, the user's experience depends on the application. It could be single stream, similar to a handheld but with a bigger screen. Or, it could be multiple streams, similar to an immersive telepresence system but with a smaller screen. Control for manipulation of streams can be local in the software application, or in another location and sent to the application over the network.

The handheld device is the most extreme. How will that participant be viewed and heard? It should be an equal participant, though the bandwidth will be significantly less than an immersive system. A receiver may choose to display output coming from a handheld differently based on the resolution, but that would be the case with any low resolution video stream, e.g., from a powerful PC on a bad network.

The handheld will send and receive a single video stream, which could be a composite or a subset of the conference. The handheld could say what it wants or could accept whatever the sender (conference server or sending endpoint) thinks is best. The handheld will have to signal any actions it wants to take the same way that immersive system signals actions.

3.6. Multipoint Education Usage

The importance of this example is that the multiple video streams are not used to create an immersive conferencing experience with panoramic views at all the sites. Instead the multiple streams are dynamically used to enable full participation of remote students in a university class. In some instances the same video stream is displayed on multiple screens in the room, in other instances an available stream is not displayed at all.

The main site is a university auditorium which is equipped with three cameras. One camera is focused on the professor at the podium. A second camera is mounted on the wall behind the professor and captures the class in its entirety. The third camera is co-located with the second, and is designed to capture a close up view of a questioner in the audience. It automatically zooms in on that student using sound localization.

Although the auditorium is equipped with three cameras, it is only equipped with two screens. One is a large screen located at the front so that the class can see it. The other is located at the rear so the professor can see it. When someone asks a question, the front screen shows the questioner. Otherwise it shows the professor (ensuring everyone can easily see her).

The remote sites are typical immersive telepresence room with three camera/screen pairs.

All remote sites display the professor on the center screen at full size. A second screen shows the entire classroom view when the professor is speaking. However, when a student asks a question, the second screen shows the close up view of the student at full size. Sometimes the student is in the auditorium; sometimes the speaking student is at another remote site. The remote systems never display the students that are actually in that room.

If someone at the remote site asks a question, then the screen in the auditorium will show the remote student at full size (as if they were present in the auditorium itself). The screen in the rear also shows this questioner, allowing the professor to see and respond to the student without needing to turn her back on the main class.

When no one is asking a question, the screen in the rear briefly shows a full-room view of each remote site in turn, allowing the professor to monitor the entire class (remote and local students). The professor can also use a control on the podium to see a particular site - she can choose either a full-room view or a single camera view.

Realization of this use case does not require any negotiation between the participating sites. Endpoint devices (and a Multipoint Control Unit (MCU),if present) - need to know who is speaking and what video stream includes the view of that speaker. The remote systems need some knowledge of which stream should be placed in the center. The ability of the professor to see specific sites (or for the system to show all the sites in turn) would also require the auditorium system to know what sites are available, and to be able to request a particular view of any site. Bandwidth is optimized if video that is not being shown at a particular site is not distributed to that site.

3.7. Multipoint Multiview (Virtual space)

This use case describes a virtual space multipoint meeting with good eye contact and spatial layout of participants. The use case was proposed very early in the development of video conferencing systems as described in 1983 by Allardyce and Randal [virtualspace]. The use case is illustrated in figure 2-5 of their report. The virtual space expands the point to point case by having all multipoint conference participants "seat" in a virtual room. In this case each participant has a fixed "seat" in the virtual room so each participant expects to see a different view having a different participant on his left and right side. Today, the use case is implemented in multiple telepresence type video conferencing systems on the market. The term "virtual space" was used in their report. The main difference between the result obtained with modern systems and those from 1983 are larger screen sizes.

Virtual space multipoint as defined here assumes endpoints with multiple cameras and screens. Usually there is the same number of cameras and screens at a given endpoint. A camera is positioned above each screen. A key aspect of virtual space multipoint is the details of how the cameras are aimed. The cameras are each aimed on the same area of view of the participants at the site. Thus each camera takes a picture of the same set of people but from a different angle. Each endpoint sender in the virtual space multipoint meeting therefore offers a choice of video streams to remote receivers, each stream representing a different view point. For example a camera positioned above a screen to a participant’s left may take video pictures of the participant’s left ear while at the same time, a camera positioned above a screen to the participant’s right may take video pictures of the participant’s right ear.

Since a sending endpoint has a camera associated with each screen, an association is made between the receiving stream output on a particular screen and the corresponding sending stream from the camera associated with that screen. These associations are repeated for each screen/camera pair in a meeting. The result of this system is a horizontal arrangement of video images from remote sites, one per screen. The image from each screen is paired with the camera output from the camera above that screen resulting in excellent eye contact.

3.8. Multiple presentations streams - Telemedicine

This use case describes a scenario where multiple presentation streams are used. In this use case, the local site is a surgery room connected to one or more remote sites that may have different capabilities. At the local site three main cameras capture the whole room (typical 3 camera Telepresence case). Also multiple presentation inputs are available: a surgery camera which is used to provide a zoomed view of the operation, an endoscopic monitor, an X-ray CT image output device, a B-ultrasonic apparatus, a cardiogram generator, an MRI image instrument, etc. These devices are used to provide multiple local video presentation streams to help the surgeon monitor the status of the patient and assist in the surgical process.

The local site may have three main screens and one (or more) presentation screen(s). The main screens can be used to display the remote experts. The presentation screen(s) can be used to display multiple presentation streams from local and remote sites simultaneously. The three main cameras capture different parts of the surgery room. The surgeon can decide the number, the size and the placement of the presentations displayed on the local presentation screen(s). He can also indicate which local presentation captures are provided for the remote sites. The local site can send multiple presentation captures to remote sites and it can receive multiple presentations related to the patient or the procedure from them.

One type of remote site is a single or dual screen and one camera system used by a consulting expert. In the general case the remote sites can be part of a multipoint Telepresence conference. The presentation screens at the remote sites allow the experts to see the details of the operation and related data. Like the main site, the experts can decide the number, the size and the placement of the presentations displayed on the presentation screens. The presentation screens can display presentation streams from the surgery room or from other remote sites and also local presentation streams. Thus the experts can also start sending presentation streams, which can carry medical records, pathology data, or their reference and analysis, etc.

Another type of remote site is a typical immersive Telepresence room with three camera/screen pairs allowing more experts to join the consultation. These sites can also be used for education. The teacher, who is not necessarily the surgeon, and the students are in different remote sites. Students can observe and learn the details of the whole procedure, while the teacher can explain and answer questions during the operation.

All remote education sites can display the surgery room. Another option is to display the surgery room on the center screen, and the rest of the screens can show the teacher and the student who is asking a question. For all the above sites, multiple presentation screens can be used to enhance visibility: one screen for the zoomed surgery stream and the others for medical image streams, such as MRI images, cardiogram, B-ultrasonic images and pathology data.

4. Acknowledgements

The document has benefitted from input from a number of people including Alex Eleftheriadis, Marshall Eubanks, Tommy Andre Nyquist, Mark Gorzynski, Charles Eckel, Nermeen Ismail, Mary Barnes, Pascal Buhler, Jim Cole.

Special acknowledgement to Lennard Xiao who contributed the text for the telemedicine use case and to Claudio Allocchio for his detailed review of the document.

5. IANA Considerations

This document contains no IANA considerations.

6. Security Considerations

While there are likely to be security considerations for any solution for telepresence interoperability, this document has no security considerations.

7. Informative References

[H.264] "Advanced video coding for generic audiovisual services", ITU-T Recommendation H.264, April 2013.

[ITU.H239] "Role management and additional media channels for H.300-series terminals", ITU-T Recommendation H.239, September 2005.

[ITU.H323] "Packet-based Multimedia Communications Systems", ITU-T Recommendation H.323, December 2009.

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, 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.

[RFC4582] Camarillo, G., Ott, J., and K. Drage, "The Binary Floor Control Protocol (BFCP)", RFC 4582, November 2006.

[virtualspace] Allardyce, and Randall, "Development of Teleconferencing Methodologies With Emphasis on Virtual Space Videe and Interactive Graphics", 1983.

Authors’ Addresses

Allyn Romanow
Cisco
San Jose, CA 95134
US

Email: allyn@cisco.com

Stephen Botzko
US

Email: stephen.botzko@gmail.com

Mark Duckworth
Polycom
Andover, MA 01810
US

Email: mark.duckworth@polycom.com

Roni Even (editor)
Huawei Technologies
Tel Aviv
Israel

Email: roni.even@mail01.huawei.com

CLUE
Internet-Draft
Intended status: Standards Track
Expires: December 3, 2012

J. Lennox
Vidyo
P. Witty

A. Romanow
Cisco Systems
June 1, 2012

Real-Time Transport Protocol (RTP) Usage for Telepresence Sessions
draft-lennox-clue-rtp-usage-04

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 December 3, 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

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.

Furthermore, captures, as defined by the CLUE Framework [I-D.ietf-clue-framework], are a somewhat different concept than RTP’s concept of media streams, so there is a need to communicate the associations between them.

This document makes recommendations, for this telepresence architecture, about how streams should be encoded and transmitted in RTP, and how their relation to captures should be communicated.

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. RTP requirements for CLUE

CLUE will permit a SIP call to include multiple 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.

Two main backwards compatibility issues exist: firstly, on an initial SIP offer we can not be sure that the far end will support CLUE, and therefore a CLUE endpoint must not offer a selection of RTP sessions which would confuse a CLUEless endpoint. Secondly, there exist many SIP devices in the network through which calls may be routed; even if we know that the far end supports CLUE, re-offering with a larger selection of RTP sessions may fall foul of one of these middle boxes.

We also desire to simplify NAT and firewall traversal by allowing endpoints to deal with only a single static address/port mapping per media type rather than multiple mappings which change dynamically over the duration of the call.

A SIP call in common usage today will typically offer one or two video RTP sessions (one for presentation, one for main video), and one audio session. Each of these RTP sessions will be used to send either zero or one media streams in either direction, with the presence of these streams negotiated in the SDP (offering a particular session as send only, receive only, or send and receive), and through BFCP (for presentation video).

In a CLUE environment this model -- sending zero or one source (in each direction) per RTP session -- doesn’t scale as discussed above, and mapping asymmetric numbers of sources to sessions is needlessly complex.

Therefore, telepresence systems SHOULD use a single RTP session per media type, as shown in Figure 1, except where there’s a need to give sessions different transport treatment. All sources of the same media type, although from distinct captures, are sent over this single RTP session.

```
   Camera 1   -.__                                        _,’Screen 1
                  ‘--._   , =-----------..........      ,’
                       ‘‘+.._‘\ _________________ _\,’
                        /    ’|        RTP          |
   Camera 2 ------------+----,’’’’’’’’’’’’’’’’’’’’’:-------- Screen 2
                        \  _ ----------------------.’.
                     _,.-’’-----------------------,/  ‘-._
                _,.-’                                     ‘.. Screen 3
   Camera 3  ,-’                                             ‘
```

Figure 1: Multiplexing multiple media streams into one RTP session

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 a single session are discussed in [I-D.westerlund-avtcore-multiplex-architecture] and [I-D.lennox-rtcweb-rtp-media-type-mux]. In the rest of this document we concentrate on examining the mapping of RTP streams to requested

CLUE captures in the specific context of telepresence systems.

The CLUE architecture requires more than simply source multiplexing, as defined by [RFC3550]. The key issue is how a receiver interprets the multiplexed 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 many 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 mapping streams for captures, showing their pros and cons. We then describe a third possible alternative.

4. RTCP requirements for CLUE

When sending media streams, we are also required to send corresponding RTCP information. However, while a unidirectional RTP stream (as identified by a single SSRC) will contain a single stream of media, the associated RTCP stream will include sender information about the stream, but will also include feedback for streams sent in the opposite direction. On a simple point-to-point case, it may be possible to naively forward on RTCP in a similar manner to RTP, but in more complicated use cases where multipoint devices are switching streams to multiple receivers, this simple approach is insufficient.

As an example, receiver report messages are sent with the source SSRC of a single media stream sent in the same direction as the RTCP, but contain within the message zero or more receiver report blocks for streams sent in the other direction. Forwarding on the receiver report packets to the same endpoints which are receiving the media stream tagged with that SSRC will provide no useful information to endpoints receiving the messages, and does not guarantee that the reports will ever reach the origin of the media streams on which they are reporting.

CLUE therefore requires devices to more intelligently deal with received RTCP messages, which will require full packet inspection, including SRTCP decryption. The low rate of RTCP transmission/reception makes this feasible to do.

RTCP also carries information to establish clock synchronization between multiple RTP streams. For CLUE, this information will be crucial, not only for traditional lip-sync between video and audio, but also for synchronized playout of multiple video streams from the same room. This information needs to be provided even in the case of switched captures, to provide clock synchronization for sources that are temporarily being shown for a switched capture.

5. Multiplexing multiple streams or multiple sessions?

It may not be immediately obvious whether this problem is best described as multiplexing multiple RTP sessions onto a single transport layer, or as multiplexing multiple media streams onto a single RTP session. Certainly, the different captures represent independent purposes for the media that is sent; however, as any stream may be switched into any of the multiplexed captures, we maintain the requirement that all media streams within a CLUE call must have a unique SSRC -- this is also a requirement for the above use of RTCP.

Because of this, CLUE’s use of RTP can best be described as multiplexing multiple streams onto one RTP session, but with additional data about the streams to identify their intended destinations. A solution to perform this multiplexing may also be sufficient to multiplex multiple RTP sessions onto one transport session, but this is not a requirement.

6. Use of multiple transport flows

Most existing videoconferencing systems use separate RTP sessions for main and presentation video sources, distinguished by the SDP content attribute [RFC4796]. The use of the CLUE telepresence framework [I-D.ietf-clue-framework] to describe multiplexed streams can remove the need to establish separate RTP sessions (and transport flows) for these sessions, as the relevant information can be provided by CLUE messaging instead.

However, it can still be useful in many cases to establish multiple RTP sessions (and transport flows) for a single CLUE session. Two clear cases would be for disaggregated media (where media is being sent to devices with different transport addresses), or scenarios where different sources should get different quality-of-service treatment. To support such scenarios, the use of multiple RTP sessions, with SDP m lines with different transport addresses, would be necessary.

To support this case, CLUE messaging needs to be able to indicate the RTP session in which a requested capture is intended to be received.

7. Use Cases

There are three distinct use cases relevant for telepresence systems: static stream choice, dynamically changing streams chosen from a finite set, and dynamic changing streams chosen from an unbounded set.

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 2.

```
            ,'''''''''''|                           +-----------Y
            |           |                           |           |
            | +--------+|"""""""""""""""""""""""""""|+--------+ |
            | |EndPoint||---------------------------||EndPoint| |
            | +--------+|"""""""""""""""""""""""""""|+--------+ |
            |           |                           |           |
            "-----------'                           "------------
```

Figure 2: Point to Point Static Streams

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. (Note in particular, in this example, that the center camera stream could be sent as either the left or the right auto-switched capture.)

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

This is illustrated in Figure 3.

```
               ,’’’’’’’’’’’|                           +-----------Y
               |           |                           |+--------+ |
               | +--------+|"""""""""""""""""""""""""""||EndPoint| |
               | |EndPoint||                           |+--------+_|
               | +--------+’’’’’’’’’’                   ’’’’’’’’’’’
               |           |........
               "-----------’
```

Figure 3: Point to Point Finite Source Streams

Dynamic streams from an unbounded 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 4.

```
              +-+--+--+
              | |EP|  ‘-.
              | +--+  |‘.‘-.
              +-------‘. ‘. ‘.
                        ‘-.‘. ‘-.
                           ‘.‘-. ‘-.
                             ‘-.‘.  ‘-.-------+              +------+
              +--+--+---+       ‘.‘.|  +---+  ---------------| +--+ |
              |  |EP|   +----.....:=.  |MCU|  ...............| |EP| |
              |  +--+   |"""""""""--|  +---+  |______________| +--+ |
              +---------+"""""""""";’.’.’.’---+              +------+
                                 .’.’.’.’
                               .’.’.’.’
                              / /.’.’
                            .’.::-’
               +--+--+--+ .’.::’
               |  |EP|  .’.::’
               |  +--+  .::’
               +--------.’
```

Figure 4: Multipoint Unbounded 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 as simplifications of the third and most difficult case, that of dynamic streams from an unbounded set. Thus, this is the primary case we will consider.

8. Other implementation constraints

To cope with receivers with limited decoding resources, for example a hardware based telepresence endpoint with a fixed number of decoding modules, each capable of handling only a single stream, it is particularly important to ensure that the number of streams which the transmitter is expecting the receiver to decode never exceeds the maximum number the receiver has requested. In this case the receiver will be forced to drop some of the received streams, causing a poor user experience, and potentially higher bandwidth usage, should it be required to retransmit I-frames.

On a change of stream, such a receiver can be expected to have a one-out, one-in policy, so that the decoder of the stream currently being received on a given capture is stopped before starting the decoder for the stream replacing it. The sender MUST therefore indicate to the receiver which stream will be replaced upon a stream change.

9. Requirements of a solution

This section lists, more briefly, the requirements a media architecture for Clue telepresence needs to achieve, summarizing the discussion of previous sections. In this section, RFC 2119 language refers to requirements on a solution, not an implementation; thus, requirements keywords are not written in capital letters.

Media-1: It must not be necessary for a Clue session to use more than a single transport flow for transport of a given media type (video or audio).

Media-2: It must, however, be possible for a Clue session to use multiple transport flows for a given media type where it is considered valuable (for example, for distributed media, or differential quality-of-service).

Media-3: It must be possible for a Clue endpoint or MCU to simultaneously send sources corresponding to static, to composited, and to switched captures, in the same transport flow. (Any given device might not necessarily be able send all of these source types; but for those that can, it must be possible for them to be sent simultaneously.)

Media-4: It must be possible for an original source to move among switched captures (i.e. at one time be sent for one switched capture, and at a later time be sent for another one).

Media-5: It must be possible for a source to be placed into a switched capture even if the source is a "late joiner", i.e. was added to the conference after the receiver requested the switched source.

Media-6: Whenever a given source is assigned to a switched capture, it must be immediately possible for a receiver to determine the switched capture it corresponds to, and thus that any previous source is no longer being mapped to that switched capture.

Media-7: It must be possible for a receiver to identify the actual source that is currently being mapped to a switched capture, and correlate it with out-of-band (non-Clue) information such as rosters.

Media-8: It must be possible for a source to move among switched captures without requiring a refresh of decoder state (e.g., for video, a fresh I-frame), when this is unnecessary. However, it must also be possible for a receiver to indicate when a refresh of decoder state is in fact necessary.

Media-9: If a given source is being sent on the same transport flow for more than one reason (e.g. if it corresponds to more than one switched capture at once, or to a static capture), it should be possible for a sender to send only one copy of the source.

Media-10: On the network, media flows should, as much as possible, look and behave like currently-defined usages of existing protocols; established semantics of existing protocols must not be redefined.

Media-11: The solution should seek to minimize the processing burden for boxes that distribute media to decoding hardware.

Media-12: If multiple sources from a single synchronization context are being sent simultaneously, it must be possible for a receiver to associate and synchronize them properly, even for sources that are are mapped to switched captures.

10. Mapping streams to requested captures

The goal of any scheme is to allow the receiver to match the received streams to the requested captures. As discussed in Section 7, during the lifetime of the transmission of one capture, we may see one or multiple media streams which belong to this capture, and during the lifetime of one media stream, it may be assigned to one or more captures.

Topologically, the requirements in Section 9 are best addressed by implementing static and a switched captures with an RTP Media Translator, i.e. the topology that RTP Topologies [RFC5117] defines as Topo-Media-Translator. (A composited capture would be the topology described by Topo-Mixer; an MCU can easily produce either or both as appropriate, simultaneously.). The MCU selectively forwards certain sources, corresponding to those sources which it currently assigns to the requested switched captures.

Demultiplexing of streams is done by SSRC; each stream is known to have a unique SSRC. However, this SSRC contains no information about capture IDs. There are two obvious choices for providing the mapping from SSRC to captures: sending the mapping outside of the media stream, or tagging media packets with the capture ID. (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.)

(An alternative architecture would be to map all captures directly to SSRCs, and then to use a Topo-Mixer topology to represent switched captures as a "mixed" source with a single contributing CSRC. However, such an architecture would not be able to satisfy the requirements Media-8, Media-9, or Media-12 described in Section 9, without substantial changes to the semantics of RTP.)

10.1. Sending SSRC to capture ID mapping outside the media stream

Every RTP packet includes an SSRC, which can be used to demultiplex the streams. However, although the SSRC uniquely identifies a stream, it does not indicate which of the requested captures that stream is tied to. If more than one capture is requested, a mapping from SSRC to capture ID is therefore required so that the media receiver can treat each received stream correctly.

As described above, the receiver may need to know in advance of receiving the media stream how to allocate its decoding resources. Although implementations MAY cache incoming media received before knowing which multiplexed stream it applies to, this is optional, and other implementations may choose to discard media, potentially requiring an expensive state refresh, such as an 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 for e.g. MCUs processing large numbers of incoming streams.

Consider the choices for where to put the mapping from SSRC to capture ID. This mapping could be sent in the CLUE messaging. The use of a reliable transport means that it can be sure that the mapping will not be lost, but if this reliability is achieved through retransmission, the time taken for the mapping 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 mapping is in RTCP, for instance as a new SDES item. This is likely to follow the same path as media, and therefore if the mapping data 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 and, due to its timing rules, cannot always be sent immediately, the mapping 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 mapping.

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |  CaptureID=9  |   length=4    |          Capture ID           :
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    :                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 5: SDES item for encoding of the Capture ID

10.2. Sending capture IDs in the media stream

The second option is to tag each media packet with the capture 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 capture ID, it can be assumed that this media stream will replace the stream currently being received with that capture ID.

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

However, the disadvantage in using a capture ID in the stream that it introduces additional processing costs for every media packet, as capture IDs are scoped only within one hop (i.e., within a cascaded conference a capture 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 added or modified at every stage.

As capture IDs are chosen by the media sender, by offering a particular capture to multiple recipients with the same ID, this requires the sender to only produce one version of the stream (assuming outgoing payload type numbers match). This reduces the cost in the multicast case, although does not necessarily help in the switching case.

An additional issue with putting capture IDs in the RTP packets comes from cases where a non-CLUE aware endpoint is being switched by an MCU to a CLUE endpoint. In this case, we may require up to an additional 12 bytes in the RTP header, which may push a media packet over the MTU. However, as the MTU on either side of the switch may not match, it is possible that this could happen even without adding extra data into the RTP packet. The 12 additional bytes per packet could also be a significant bandwidth increase in the case of very low bandwidth audio codecs.

10.2.1. Multiplex ID shim

As in draft-westerlund-avtcore-transport-multiplexing

10.2.2. RTP header extension

The capture ID could be carried within the RTP header extension field, using [RFC5285]. This is negotiated within the SDP i.e.

```
a=extmap:1 urn:ietf:params:rtp-hdrex:clue-capture-id
```

Packets tagged by the sender with the capture ID will then contain a header extension as shown below

```
        0                   1                   2                   3
        0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |  ID=1 |  L=3  |                 capture id                    |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |  capture id   |
       +-+-+-+-+-+-+-+-+
```

Figure 6: RTP header extension for encoding of the capture ID

To add or modify the capture ID can be 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.

10.2.3. Combined approach

The two major flaws of the above methods (high latency switching 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.

10.2.3.1. Behaviour of receivers

A receiver of a stream should demultiplex on SSRC if it knows the capture ID for the given SSRC, otherwise it should look within the packet for the presence of the stream ID. This has an issue where a stream switches from one capture to a second - for example, in the second use case described in Section 7, where the transmitter chooses to switch the center stream from the receiver’s right capture to the left capture, and so the receiver will already know an incorrect mapping from that stream’s SSRC to a capture ID.

In this case the receiver should, at the RTP level, detect the presence of the capture ID and update its SSRC to capture ID map. This could potentially have issues where the demultiplexer has now sent the packet to the wrong physical device - this could be solved by checking for the presence of a capture ID in every packet, but this will have speed implications. If a packet is received where the receiver does not already know the mapping between SSRC and capture ID, and the packet does not contain a capture ID, the receiver may discard it, and MUST request a transmission of the capture ID (see below).

10.2.3.2. Choosing when to send capture IDs

The updated capture ID needs to be known as soon as possible on a switch of SSRCs, as the receiver may be unable to allocate resources to decode the incoming stream, and may throw away the received packets. It can be assumed that the incoming stream is undecodable until the capture ID is received.

In common video codecs (e.g. H.264), decoder refresh frames (either IDR or GDR) also have this property, in that it is impossible to decode any video without first receiving the refresh point. It therefore seems natural to include the capture ID within every packet of an IDR or GDR.

For most audio codecs, where every packet can be decoded independently, there is not such an obvious place to put this information. Placing the capture ID within the first n packets of a stream on a switch is the most simple solution, where n needs to be sufficiently large that it can be expected that at least one packet will have reached the receiver. For example, n=50 on 20ms audio packets will give 1 second of capture IDs, which should give reasonable confidence of arrival.

In the case where a stream is switched between captures, for reasons of coding efficiency, it may be desirable to avoid sending a new IDR frame for this stream, if the receiver’s architecture allows the same decoding state to be used for its various captures. In this case, the capture ID could be sent for a small number of frames after the source switches capture, similarly to audio.

10.2.3.3. Requesting Capture ID retransmits

There will, unfortunately, always be cases where a receiver misses the beginning of a stream, and therefore does not have the mapping. One proposal could be to send the capture ID in SDES with every SDES packet; this should ensure that within ˜5 seconds of receiving a stream, the capture ID will be received. However, a faster method for requesting the transmission of a capture ID would be preferred.

Again, we look towards the present solution to this problem with video. RFC5104 provides an Full Intra Refresh feedback message, which requests that the encoder provide the stream such that receivers need only the stream after that point. A video receiver without the start of the stream will naturally need to make this request, so by always including the capture ID in refresh frames, we can be sure that the receiver will have all the information it needs to decode the stream (both a refresh point, and a capture ID).

For audio, we can reuse this message. If a receiver receives an audio stream for which it has no SSRC to capture mapping, it should send a FIR message for the received SSRC. Upon receiving this, an audio encoder must then tag outgoing media packets with the capture ID for a short period of time.

Alternately, a new RTCP feedback message could be defined which would explicitly request a refresh of the capture ID mapping.

10.3. Recommendations

We recommend that endpoints MUST support the RTP header extension method of sharing capture IDs, with the extension in every media packet. For low bandwidth situations, this may be considered excessive overhead; in which case endpoints MAY support the combined approach.

This will be advertised in the SDP (in a way yet to be determined); if a receiver advertises support for the combined approach, transmitters which support sending the combined approach SHOULD use it in preference.

11. Security Considerations

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

Capture IDs need to be integrity-protected in secure environments; however, they do not appear to need confidentiality.

12. IANA Considerations

Depending on the decisions, the new RTP header extension element, the new RTCP SDES item, and/or the new AVPF feedback message will need to be registered.

13. References

13.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.

13.2. Informative References

[I-D.ietf-clue-framework]
Romanow, A., Duckworth, M., Pepperell, A., and B. Baldino, "Framework for Telepresence Multi-Streams", draft-ietf-clue-framework-05 (work in progress), May 2012.

[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]
Romanow, A., Botzko, S., Duckworth, M., Even, R., and I. Communications, "Use Cases for Telepresence Multi-streams", draft-ietf-clue-telepresence-use-cases-02 (work in progress), January 2012.

[I-D.lennox-rtcweb-rtp-media-type-mux]
Lennox, J. and J. Rosenberg, "Multiplexing Multiple Media Types In a Single Real-Time Transport Protocol (RTP) Session", draft-lennox-rtcweb-rtp-media-type-mux-00 (work in progress), October 2011.

[I-D.westerlund-avtcore-multiplex-architecture]
Westerlund, M., Burman, B., and C. Perkins, "RTP Multiplexing Architecture", draft-westerlund-avtcore-multiplex-architecture-01 (work in progress), March 2012.

[RFC4796] Hautakorpi, J. and G. Camarillo, "The Session Description Protocol (SDP) Content Attribute", RFC 4796, February 2007.

[RFC5104] Wenger, S., Chandra, U., Westerlund, M., and B. Burman, "Codec Control Messages in the RTP Audio-Visual Profile with Feedback (AVPF)", RFC 5104, February 2008.

[RFC5117] Westerlund, M. and S. Wenger, "RTP Topologies", RFC 5117, January 2008.

[RFC5285] Singer, D. and H. Desineni, "A General Mechanism for RTP Header Extensions", RFC 5285, July 2008.

Authors’ Addresses

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

Email: jonathan@vidyo.com

Paul Witty
England
UK

Email: paul.witty@balliol.oxon.org

Allyn Romanow
Cisco Systems
San Jose, CA 95134
USA

Email: allyn@cisco.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 13, 2012

S. Wenger
Vidyo
M. Eubanks
AmericaFree.TV
R. Even
Huawei
G. Camarillo
Ericsson

March 12, 2012

Transport Options for Clue
draft-wenger-clue-transport-02

Abstract

This memo describes the assumption and the proposed options for the coding and transport of CLUE messages as outlined in version 01 of the framework draft.

Requirements Language

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].

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 13, 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

The CLUE WG is chartered to design a protocol to enable communication about media streams for videoconferencing and telepresence working in conjunction with the IETFOs protocol suites of choice, namely SIP for basic call setup and control and RTP for media transport. (It should be noted that ITU-T Q.xx/16 has informally expressed a desire that parts or all of the work of the CLUE WG can be re-used in an H.323 environment. Therefore, occasionally, we comment on the re-use of CLUE work outside of SIP systems. This does not mean that we want to extent the charter; however, it seems sensible at least to us that if a cross protocol solution and a SIP-only solution to the CLUE problem could be devised, and both solutions are comparable in in their complexity etc etc, a solution with applicability beyond SIP may be the appropriate choice.)

This document describes options for the coding and transport of CLUE messages in a SIP / RTP environment. Specifically, three issues are addressed.

First, while the framework draft conceptually describes message flows, it does not specify how those messages are actually transferred "on the wire" and how they relate to the SIP offer/answer [RFC3264]. This document lists (hopefully all) the options that have been proposed in CLUE to date.

Second, the framework-01 draft describes three messages between the producer and the consumer in an abstract form, without specifying the details of the representation of those messages. This memo lists (some of) the options for the representation of the abstract messages of the framework draft.

Third, before any CLUE messages can be meaningfully exchanged, it is necessary to discover whether the involved systems are actually CLUE-capable. This memo discusses the proposed options for CLUE capability discovery.

In this memo we only present the options discussed to date in the working group. Deciding on the appropriate mechanism (or mechanisms, as it is not always appropriate to have a single solution for a given problem, though this is of course desirable from an interoperability viewpoint) is left for further discussion in the working group. That does not mean that the authors do not have preferences, and/or specific knowledge of certain mechanisms, and may as a result go in greater depth in describing one mechanism than another.

2. Assumptions

The Basic Clue data model is specified in the framework document. The framework defines three messages that carry the Clue data:

Consumer Capability Message

Provider Capabilities Announcement

Consumer Configure Request

(There is no clear consensus that the Consumer Capability Message is needed, but for the time being we attempt to document how it fits in the different options.)

CLUE messages may need to be sent at the initialization of a call, and possibly also at irregular intervals within a call, spaced in the order of seconds, minutes or even longer. There is also no hard real-time transmission requirement for CLUE messages; latencies in the seconds range are acceptable. More specifically, there appears not to be an issue with system reaction delay larger than the maximum round trip delay for reasonable operation of a telepresence system.

The Clue message handshake as required by the framework (independent from the issue related to the need of the Consumer Capability Message) is different from the offer/answer (o/A) exchange [RFC3264], primarily because the CLUE exchange is uni-directional, requiring a similar exchange for each side of the media flow, while one offer/answer exchange defines both sides of the media flow. (Note that asymmetry in SIP may require a second offer answer exchange, but this is not the typical case)

There is no hard requirement for synchronization of CLUE messages, though there may be a need for sequencing, (TBD).

CLUE messages may need to describe the characteristics of all endpoints in a conference (TBD), and that conference can potentially include dozens of endpoints.

It appears to be consensus within the CLUE WG that there will be an SDP offer/answer exchange as part of the solution. It further appears to be the consensus that the offer/answer will be used to establish the media channels and negotiate those SDP parameters negotiable with media types (i.e. as defined in RTP payload formats), as well as to allow interoperability with systems that do not support the CLUE protocol. It appears to be a sensible design goal that the CLUE data does not duplicate SDP attributes.

In order to achieve interoperability with systems that do not support CLUE, the first offer answer exchange could be used to negotiate CLUE support.

An open issue is whether there needs to be a final offer answer exchange, after initial o/A exchange(s) as well as CLUE exchange(s), with an SDP reflecting the negotiated media flows, in order to address requirements imposed by intermediaries like Session Border Controllers (SBC). This topic was discussed in different contexts before, and there is some text about it in RFC5939 section 3.12 [RFC5939]

The size of a CLUE message is far from final yet but when selecting a solution the issue of message size and fragmentation (if applicable) needs to be addressed.

3. Transport for CLUE messages

CLUE messages need to be conveyed from one CLUE capable system to another, i.e., there needs to be "transport" of CLUE messages. It should be clear that the message transport can be based on a transport layer (layer 4 in ISO/OSI) protocol or other layers, such as the application layer.

In contrast to the "content representation", the transport of CLUE messages is somewhat more tightly bound to the environment. In some scenarios it may be possible to reuse most of the mechanisms defined in an option for transport between SIP and H.323, while in others this is not possible.

The selection of the transport may have some affect on the content representation, in that certain transports in the aforementioned sense are defined only to carry certain types of messages. For example, offer-answer is defined for the use in conjunction with SDP as content representation. In contrast, obviously, a CLUE-defined transport mechanism could carry any format specified by CLUE.

The CLUE protocol enables the CLUE systems to negotiate the semantic relationships of the media streams, mostly with respect to spatial relations. Another aspect that has recently risen to prominence is the negotiation of media codec settings, taking into account that in practical telepresence systems, certain combinations of codec settings may not be supported by the hardware ("codec alternatives" henceforth).

The apparently generally agreed need for interoperability with non CLUE systems requires defining an initial offer involving CLUE support, and guidance on how to progress the call setup based on the answer. The CLUE WG discussed a couple of options including two stage offer answer, using grouping similar to [I-D.ietf-mmusic-sdp-bundle-negotiation], and using the capability negotiation of [RFC5939].

We would like to consider the following options:

3.1. Option 1 : Piggy-pack on SIP

SIP includes a number of methods that can carry (directly or through content indirection) CLUE messages. Many of these messages can be exchanged during the lifetime of a session. Piggy-packing CLUE messages on SIP has the advantage that any built-in transport and reliability mechanisms of SIP can be re-used. (Whether this is an advantage in practice is somewhat questionable, considering that the vast majority of SIP systems use UDP for the transport of SIP messages, and that their SIP messages are typically small enough to fit into an MTU--something that like is not true for some CLUE messages.) It also has the feature (advantage?) that CLUE signaling is being conveyed in the signaling plane rather than in the media plane (making things such as decomposition potentially easier and certainly more intuitive).

There are three sub-options to consider

3.1.1. Option 1.1: Using SDP (in an offer-answer context) for CLUE information

In this option, the CLUE protocol is specified through the addition of CLUE-specific SDP codepoints in the (essentially unmodified) offer/answer process, for essentially all CLUE functionalities. The stream semantics associated with spatial relations of streams are represented as new SDP attributes . Codec alternatives may be negotiated based on draft-ietf-mmusic-sdp-media-capabilities.

The nature of spatial relations currently envisioned by some CLUE participants have some simultaneous restrictions due to the limitations of physical capture devices. For this reason, it may become necessary to separate the negotiation process into a session negotiation that defines RTP sessions, and a session negotiated that deals with the spatial relations.

It is noted that, at the time of writing, there is no proposal on the table that would suggest that offer-answer only is a sensible--or even possible--design choice.

3.1.2. Option 1.2: Using an SDP MIME body to carry the CLUE information in an INVITE or UPDATE exchange

In order to separate the RTP session negotiation from the CLUE media capture selection, a clean solution appears to be to carry the CLUE information in a body separate from the classic media negotiation information, with a parallel negotiation using INVITE and UPDATE for the CLUE information. A similar approach is proposed in [I-D.ietf-siprec-protocol].

There were concerns about using re-invite, claiming that it takes too long since that commonly implies codec boxes teardown of every existing media session during re-invites. [RFC3311]suggests that although UPDATE can be used on confirmed dialogs, it is RECOMMENDED that a re-INVITE be used instead. This is because an UPDATE needs to be answered immediately, ruling out the possibility of user approval. Such approval may be needed, and is possible only with a re-INVITE.

3.1.3. Option 1.3: Using a SIP INFO package

Another option may be to define a new SIP INFO package [RFC6086]. The SIP-INFO method is very flexible in that the package can define, at least to a large extent, the semantics of a SIP-INFO exchange. However, SIP-INFO is subject to SIPOs limitations, for example in terms of message size when SIP messages are transported over UDP (which, we understand, is the common operation point.

3.1.4. Option 1.4: SIP signaling options

There may be other options using SIP signaling, such as subscribe/ notify or Message method, see [RFC6086] section 8.4.1. Note that, in those cases, a subscribe creates a separate dialog usage and is normally sent outside of existing dialog. Within this document, we are not discussing the implications of such a possible implementation path.

3.2. Option 2: CLUE control channel on the media plane over UDP

During the initial SIP handshake, a secure(?) CLUE channel is established (if both systems are CLUE capable). This channel may be UDP or TCP based. Using UDP may require an additional reliability mechanism, perhaps using a mechanism similar to BFCP over UDP, and addressing fragmentation is likely to be necessary due to message size. These complications are not required for a TCP based solution. On the other hand, using ICE to address firewall and NAT traversal as well as working with intermediaries like SBCs works better with UDP. Note that even under this option, we assume that the actual protocol exchange to negotiate and open media channels is being conducted using an SDP content representation, quite possibly through a "fincal" offer-answer exchange that nails down the actual media flows to be used, for the benefit of SBCs and similar middleboxes.

3.3. Option 3: Other Work

At least three other individual submissions address similar topics as this section, and the the readerOs attention is drawn to those. Specifically:

[I-D.hansen-clue-protocol-choices-evaluation] goes into some detail in analyzing the pros and cons of a previous version of our document. The authors arrive at a conclusion that can be summarized as that there is a need for a transport mechanism that is not based on SIP, but using a UDP session negotiated using SIP and Offer/Answer for the transport of CLUE messages. CLUE messages in this case probably ought to be interpreted narrowly in that they relate to spatial relationships and related issues, in contrast to codec parameter negotiation.

[I-D.romanow-clue-sdp-usage] arrives at a similar conclusion. The draft lists those codepoints that could be conveyed using SDP in an offer-answer setting: video properties (bandwidth and resolution), and bandwidth-related group settings. Everything else, including spatial relationship of captures, is suggested to be conveyed over a CLUE-specific protocol, conveyed over a UDP(?) session negotiated in SIP during the early (first) offer/answer exchange.

[I-D.cazeaux-clue-sip-signaling] signaling appears to advocate a solution in which SDP based O/A is used to negotiate media. The negotiated media appear to be a superset of the media later being used. CLUE specific information, such as spatial relationships, but also the details of the media sessions (including restrictions of provider content selection based on consumer capabilities), appear to be relegated to signaling conveyed over a SIP/OA negotiated CLUE channel.

All three aforementioned drafts appear to acknowledge the need for a CLUE signaling channel, possibly conveyed directly over UDP (in contrast to a being conveyed over SIP-info or something similar), although these drafts vary in the degree to which they use the CLUE signaling channel.

4. Content Representation

The data model in the framework-03 draft does not include a specification of the representation of the data. Many different representation languages, for example XML, possibly SDP, ASN.1, and others can be used, and we need to decide on one, possibly for each data structure defined in a CLUE solution (that is, for example, it’s possible that some data points of CLUE can be conveyed in SDP, whereas others use XML). Depending on the transport decision, we may be restricted to certain representations for certain data structures, or we may have freedom of choice. Referring to the options suggested above, it is clear that option 3.1.1 mandates SDP for representing CLUE. However, all other options appear not to require any pre-defined choice, at least for some (though not necessarily for all) of the CLUE-defined codepoints.

One observation that has to be made at this point (described in greater detail above) is that the framework-01 draft’s message exchange system requires more than one end to end exchange due to the asymmetry. Another observation is that the advertisement describes the sending options, which makes the CLUE exchange different from the offer/answer mechanism SIP videoconferencing endpoints use today. For this reason we do not think that the option in 3.1.1 is a good direction. Therefore, there appears not to be a hard requirement to use SDP exclusively for the representation of CLUE messages. For some messages, SDP may be an appropriate choice, but for others, there is no precedence: We have a freedom of choice here, which is why this section exists.

It is very well possible that even moderately complex CLUE messages may exceed MTU sizes commonly found in todayOs Internet. There has been discussion in CLUE of sessions with thousands of participantsNa very real requirement for at least one of the authors of this draft, who routinely participates in multipoint videoconferences with 200+ participants. Even if a CLUE message can be compressed into a few bytes for each endpoint, such sessions will violate the commonly found Ethernet 1492-byte MTU. Accordingly, message transport protocols will have to be prepared to split CLUE messages into fragments, which has implications on the design complexity of those protocols. This problem is especially an issue for verbose representations, such as XML.

4.1. Option 1 : SDP

SDP and its various extensions are used in SIP based systems for the offer/answer exchange, and, therefore, those systems include SDP parsers that could probably be extended to support CLUE messages. SDP is also a fairly compact, but still (though barely) human readable . Even though it does not appear to us to make overly much sense to use SDP for CLUE, since it will require a separate blob for describing the CLUE relations between the media captures, it still viable to use text based representation for CLUE if using any of the options which is not 3.1.1. [I-D.romanow-clue-sdp-usage] suggests that an SDP-only representation of CLUE based parameters is an (impossible/suboptimal) bad choice. We concur. As mentioned before, though, those parameters that can reasonably be negotiated using SDP o/A (with however many round trip it takes) should in our opinion be represented in SDP. We shouldn’t be in SDP-ng’s business.

4.2. Option 2 : XML

XML is very flexible, and the representation of choice for many IETF technologies not bound to a certain legacy. It certainly allows for all flexibility needed to represent all CLUE messages currently considered. It also is naturally extensible in a way SDP is not. On the downside, XML is fairly verbose, which has implications on the transport. Even considering this verboseness, we believe that XML may be an appropriate representation for CLUE messages that cannot be represented in SDP.

4.3. Option 3 : ASN.1

ASN.1 is similarly flexible and extensible as XML, and (in its binary representation) fairly compact. While it is commonly used in H.323, and while the video conferencing industry certainly has access to the tools necessary to deploy ASN.1 (a major obstacle in other industries), it is not widely used by SIP implementations.

4.4. Option 4 : Clue Defined Format

It is, of course, possible that the CLUE WG defines its own format, possibly compact, possibly binary and possibly extensible representation language or format for CLUE messages.

4.5. Examples

An example or examples should be added here when possible

4.6. Proposal

The preferred solution can be XML-based for codepoints not easily (currently?) representable in SDP, and SDP based for everything else.] With respect to XMLOs verboseness, fragmentation support in the transport protocol may be needed and the transport probably should include a fragmentation and reassembly support beyond IP fragmentation/re-assembly. Such support may require an encapsulation of the message with headers that will allow fragmentation and reassembly support.

5. Clue Discovery

This section summarizes ways to discover whether systems involved are CLUE-capable. For simplicity, point-to-point scenarios are assumed. Multipoint scenarios are similar since we are considering centralized conference models only.

Discovery appears to be necessarily bound to the capability exchange of the involved systems.

5.1. Option 1 : CLUE discovery as a side effect of opening a CLUE control channel

If, for the transport of CLUE messages (or at least a subset thereof), a media plane control channel were used (section 3.2), then the discovery of CLUE capability would be a side effect of the opening of this control channel during the initial offer/answer exchange. At this point in time, there is no proposal on the table that suggest that we can avoid a CLUE control channel.

5.2. Option 2 : SIP Message Transport

Very roughly speaking, if we use the INFO message for the transport of all CLUE messages, then by using the Recv-Info header field the support for the CLUE package can be signaled. If using a second MIME body the support of the MIME body in the offer answer can be used.

6. IANA Considerations

This document makes no request of IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

7. Security Considerations

Any method for bypassing NAT/Firewall protections of course brings security issues, which need to be dealt with.

8. Acknowledgements

The list of authors needs to grow.

9. Informative References

[I-D.cazeaux-clue-sip-signaling]
Cazeaux, S. and E. Bertin, "Requirements for ControLling mUltiple streams for tElepresence (CLUE) signaling.", draft-cazeaux-clue-sip-signaling-00 (work in progress), March 2012.

[I-D.hansen-clue-protocol-choices-evaluation]
Hansen, R. and A. Romanow, "Evaluation of using SIP or an independent protocol for CLUE messaging", draft-hansen-clue-protocol-choices-evaluation-00 (work in progress), November 2011.

[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.ietf-siprec-protocol]
Portman, L., Lum, H., Johnston, A., and A. Hutton, "Session Recording Protocol", draft-ietf-siprec-protocol-03 (work in progress), March 2012.

[I-D.romanow-clue-sdp-usage]
Romanow, A., Andreasen, F., and A. Krishna, "Investigation of Session Description Protocol (SDP) Usage for ControLling mUltiple streams for tElepresence (CLUE)", draft-romanow-clue-sdp-usage-00 (work in progress), March 2012.

[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.

[RFC3311] Rosenberg, J., "The Session Initiation Protocol (SIP) UPDATE Method", RFC 3311, October 2002.

[RFC5939] Andreasen, F., "Session Description Protocol (SDP) Capability Negotiation", RFC 5939, September 2010.

[RFC6086] Holmberg, C., Burger, E., and H. Kaplan, "Session Initiation Protocol (SIP) INFO Method and Package

Framework", RFC 6086, January 2011.

Authors' Addresses

Dr. Stephan Wenger
Vidyo
433 Hackensack Ave
Hackensack, NJ 07601
USA

Email: stewe@stewe.org

Marshall Eubanks
AmericaFree.TV
P.O. Box 141
Clifton, Virginia 20124
USA

Phone: +1-703-501-4376
Email: marshall.eubanks@gmail.com

Roni Even
Huawei

Email: ron.even.tlv@gmail.com

Gonzalo Camarillo
Ericsson

Email: Gonzalo.Camarillo@ericsson.com