Internet Engineering Task Force (IETF)
Request for Comments: 8845
Category: Standards Track
ISSN: 2070-1721

M. Duckworth, Ed.
A. Pepperell
Acano
S. Wenger
Tencent

January 2021

Framework for Telepresence Multi-Streams

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 Session Description Protocol (SDP) negotiation for setting up a telepresence session.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc8845.

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.

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, signaling that is ControLling mUltiple streams for tElepresence (CLUE) specific is required to exchange the information describing the multiple Media Streams. The motivation for this framework, an overview of the signaling, and the information required to be exchanged are described in subsequent sections of this document. Companion documents describe the signaling details [RFC8848], the data model [RFC8846], and the protocol [RFC8847].

2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Definitions

The terms defined below are used throughout this document and in companion documents. Capitalization is used in order to easily identify a defined term.

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 (AC): 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: 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-* refers to something that has been designed by the CLUE working group; for example, this document may be called the CLUE-framework document herein and elsewhere.

CLUE-capable device: A device that supports the CLUE data channel [RFC8850], the CLUE protocol [RFC8847] and the principles of CLUE negotiation; it also 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 that 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 "A Framework for Conferencing within the Session Initiation Protocol (SIP)" [RFC4353].

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 subdivided across potentially multiple Individual Encodings.

Endpoint: A CLUE-capable device that 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 that 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 CSVs 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 into one single multimedia Conference [RFC7667]. An MCU includes a Mixer like that described in [RFC4353], without the requirement of [RFC4353] 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 (MC): 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 (for example, 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 a Media Provider.

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

Scene: Same as a Capture Scene.

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

Single Media Capture: A Capture that contains Media from a single source Capture Device, e.g., an Audio Capture from a single microphone or a Video Capture from a single camera.

Spatial Relation: The arrangement of two objects in space, 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, bitrate, resolution, profile/level, etc.) as well as CLUE-specific attributes, such as the Capture ID or a spatial location.

Video Capture (VC): 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 Media Capture and Media Render 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 Media 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 then 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 to capture either the middle few seats only 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, which is 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 as 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 Binary Floor Control Protocol (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 they 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) what the Provider can send.

An Endpoint or MCU typically acts 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.)

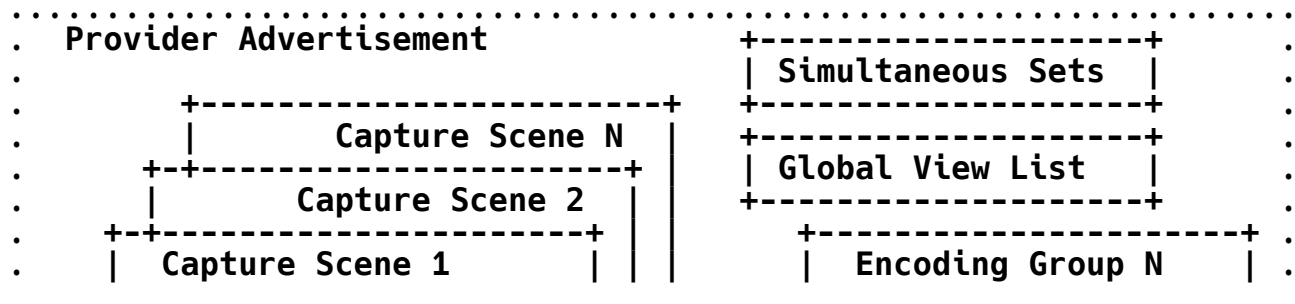
The data model [RFC8846] is based around two main concepts: a Capture and an Encoding. A Media Capture, such as of type audio or video, has attributes to describe the content a Provider can send. Media Captures are described in terms of CLUE-defined attributes, such as Spatial Relationships and purpose 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 that 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 device; 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.



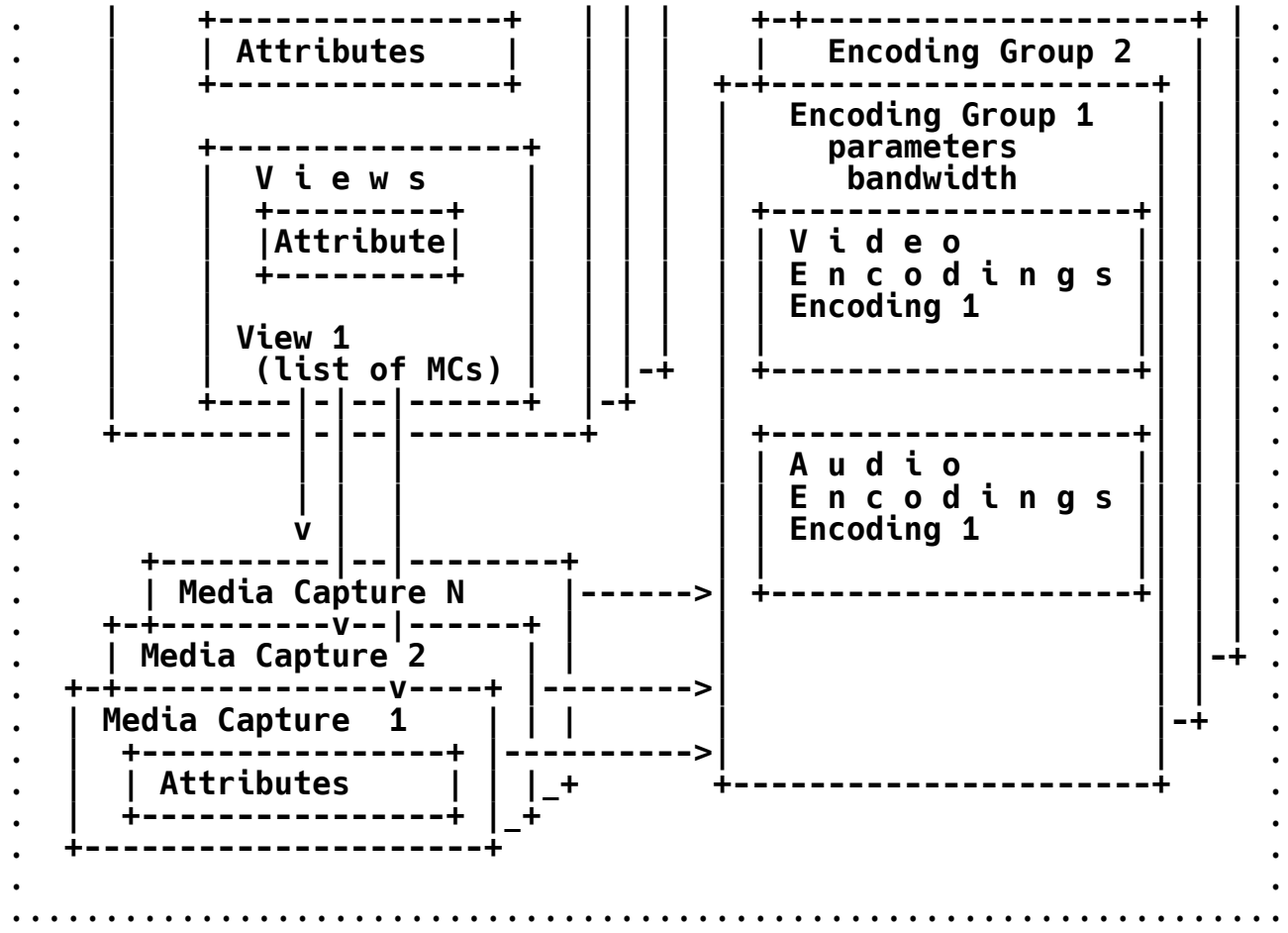
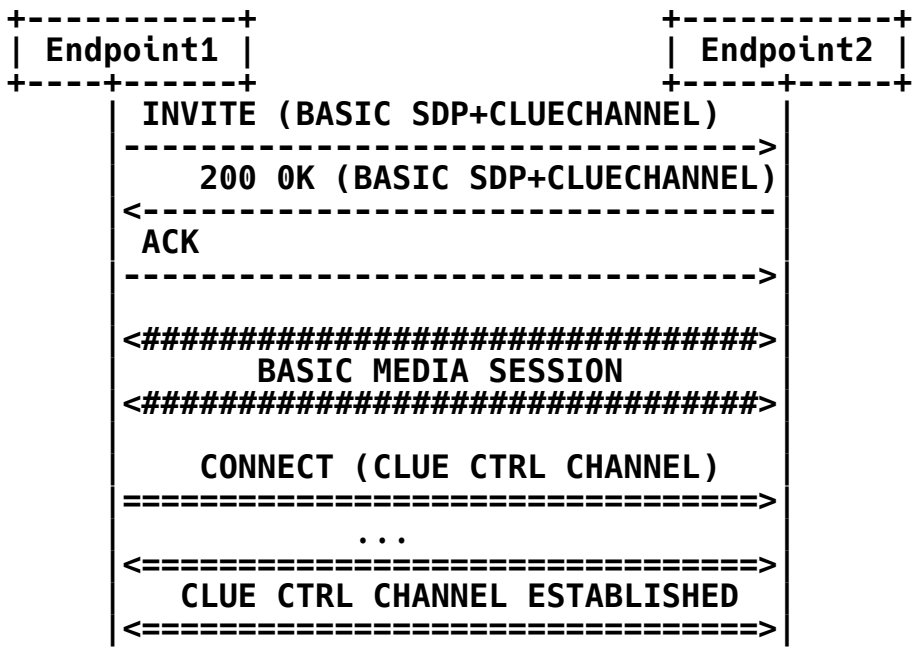


Figure 1: Advertisement Structure

Figure 2 illustrates the call flow used by a simple system (two Endpoints) in compliance with this document. A very brief outline of the call flow is described in the text that follows.



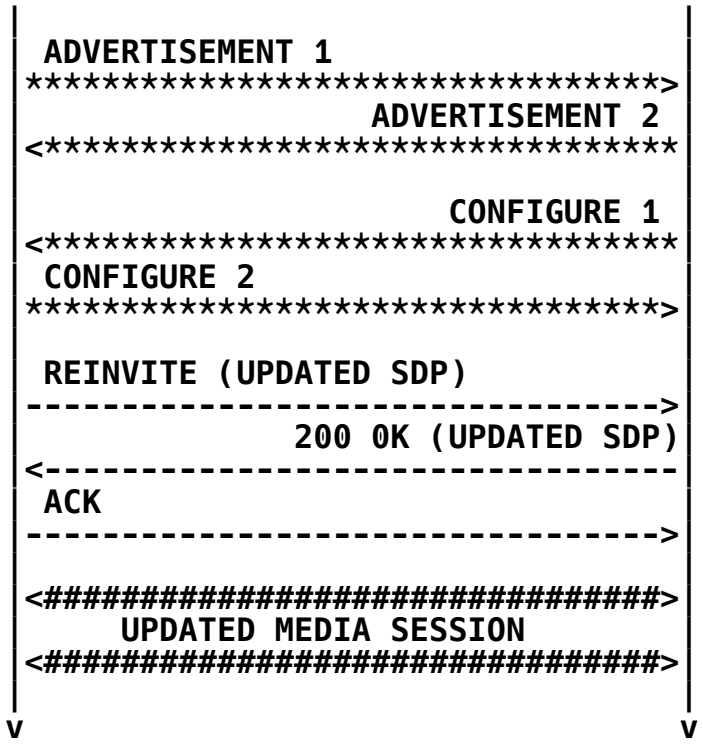


Figure 2: Basic Information Flow

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 also has the effect of

setting up the Media transmission itself, involving potentially security exchanges, Interactive Connectivity Establishment (ICE), and whatnot. This step is considered "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 that the Provider 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. In other cases, however, after the CLUE negotiation, an additional offer/answer exchange becomes necessary. For example, if both sides decide to upgrade the call from one 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 offer/answer 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 [RFC8848]. 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 that 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 either be in real, physical units (millimeters), have an unknown scale, or have no physical scale. Systems that 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 that 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 Scale" means the scale is not necessarily in 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 as follows: 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, 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, AC1, and 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 an MCC.

* A Media Capture may be associated with one or more CSVs.

* 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 subsections 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 that 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 midpoint 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 that 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 three-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 that 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, Point on Line 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 that 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 Point of Capture, Capture Axis, and Area of Capture change with time.

The Point of Capture of a static Capture MUST NOT move for the life of the CLUE session. The Point of Capture 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 Point of Capture that is constantly moving. If the Area of Capture, Point of Capture, and Point on 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 that is the source of the Capture. Possible values include patterns such as omni, shotgun, cardioid, and 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 [RFC8846] 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 as follows:

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 of [RFC6350]), Communication Information (Section 6.4 of [RFC6350]), and Organizational information (Section 6.6 of [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 of [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. On reception of a Capture with a Person Type attribute, a Consumer knows with some certainty 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 scheduled on the agenda to make a presentation in the meeting. Note: This is not related to any "active speaker" functionality.
Translator:	the person providing some form of translation or commentary in the meeting.
Timekeeper:	the person 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 that 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 that 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. An 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 an MCC. An 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, a Provider should 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 subset 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 subsections below describe the MCC-only attributes.

7.2.1.1. MaxCapture: Maximum Number of Captures within an MCC

The MaxCaptures 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 regard to space and time.

A Provider can indicate that the number of Captures in an 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 an MCC that purely represents a composition of sources and an MCC that represents switched sources or switched and composed sources.

MaxCaptures may be set to one so that only content related to one of the sources is 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 MaxCaptures versus the number of Captures in the MCC affects how sources appear in a Capture Encoding:

* A switched case occurs when MaxCaptures is set to <= 1 and the number of Captures in the MCC is greater than 1 (or not specified) in the MCC. Zero or one Captures may be switched into the Capture Encoding. Note: zero is allowed because of the "<=".

* A switched case occurs when MaxCaptures is set to = 1 and the number of Captures in the MCC is greater than 1 (or not specified) in the MCC. Only one Capture source is contained in a Capture Encoding at a time.

* A switched and composed case occurs when MaxCaptures is set to <= N (with N > 1) and the number of Captures in the MCC is greater than N (or not specified). The Capture Encoding may contain purely switched sources (i.e., <=2 allows for one source on its own), or it may contain composed and switched sources (i.e., a composition of two sources switched between the sources).

* A switched and composed case occurs when MaxCaptures is set to = N (with N > 1) and the number of Captures in the MCC is greater than N (or not specified). 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.

* A switched and composed case occurs when MaxCaptures is set <= to the number of Captures in the MCC. The Capture Encoding may contain Media switched between any number (up to the MaxCaptures) of composed sources.

* A composed case occurs when MaxCaptures is set = to the number of Captures in the MCC. All the sources are composed into a single Capture Encoding.

If this attribute is not set, then as a 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 an MCC with three Video Captures, VC1, VC2, and VC3, would indicate that the Advertiser in the Capture Encoding would switch between VC1, VC2, and 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.

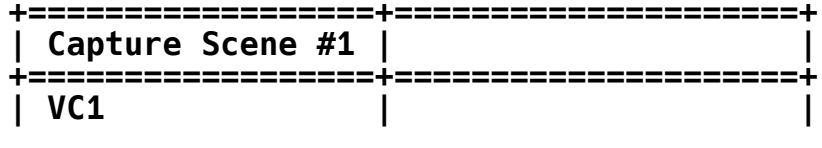
The tokens are as follows:

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.



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. SynchronizationID: Synchronization Identity

The SynchronizationID MCC attribute indicates how the individual Captures in multiple MCC Captures are synchronized. 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 SynchronizationID 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 SynchronizationID 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 SynchronizationID 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=Center
VC3	Description=Right
AC1	Description=Room
CSV(VC1,VC2,VC3)	
CSV(AC1)	
Capture Scene #2	
VC4	Description=Left
VC5	Description=Center

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)	SynchronizationID=1 MaxCaptures=1
MCC2(VC2,VC5,VC8)	SynchronizationID=1 MaxCaptures=1
MCC3(VC3,VC6)	MaxCaptures=1
MCC4(AC1,AC2,AC3,AC4)	SynchronizationID=1 MaxCaptures=1
CSV(MCC1,MCC2,MCC3)	
CSV(MCC4)	

Table 4: Example SynchronizationID 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 SynchronizationID, 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 synchronized with MCC1 or MCC2. As MCC4 also has the same SynchronizationID as MCC1 and MCC2, the content of the audio Encoding will be synchronized 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 (CSVs), 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 CSV. 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 CSVs. 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 CSVs for Video Captures and several CSVs for Audio Captures. A particular Capture MAY be included in more than one CSV.

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 in Section 6.

A Provider arranges Captures in a Capture Scene to help the Consumer choose which Captures it wants to Render. The CSVs in a Capture Scene are different alternatives the Provider is suggesting for representing the Capture Scene. Each CSV is given an advertisement-unique identity. The order of CSVs within a Capture Scene has no

significance. The Media Consumer can choose to receive all Media Captures from one CSV 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 CSVs. Different CSVs of the same Media type are not necessarily mutually exclusive alternatives. Also note that the presence of multiple CSVs (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 CSV MUST be of the same Media type -- it is not possible to mix audio and Video Captures in the same CSV, for instance. The Provider MUST be capable of encoding and sending all Captures (that have an Encoding Group) in a single CSV simultaneously. The order of Captures within a CSV has no significance. A Consumer can decide to receive all the Captures in a single CSV, but a Consumer could also decide to receive just a subset of those Captures. A Consumer can also decide to receive Captures from different CSVs, 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 CSV that most closely matched the Consumer's number of display devices or screen layout.

The following is an example of four potential CSVs 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 that 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 the following:

* Human-readable description of the Capture Scene, which could be in multiple languages;

* xCard Scene information

* Scale information ("Millimeters", "Unknown Scale", "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 Section 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 CSVs in addition to the Capture-Scene-wide attributes described above. CSV attributes apply to the CSV as a whole, i.e., to all Captures that are part of the CSV.

CSV attributes include the following:

* A human-readable description (which could be in multiple languages) of the CSV.

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:

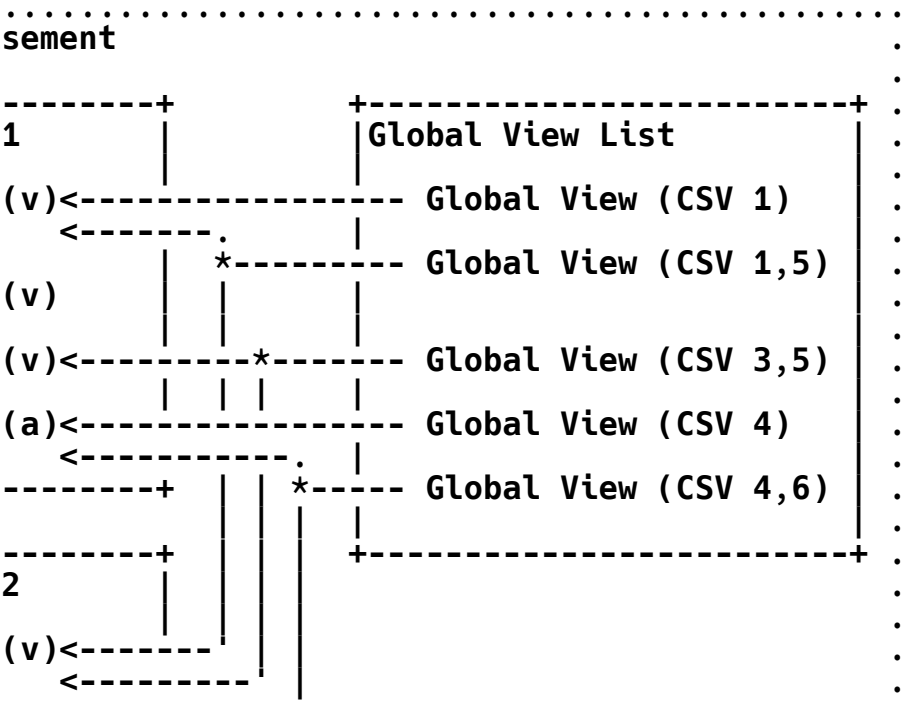
* The ordering of Global Views in the Global View list is insignificant.

* 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.



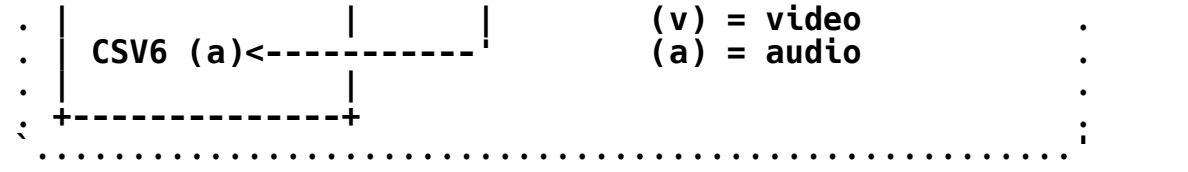


Figure 3: Global View List Structure

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, or 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 people each: VC0, VC1, and VC2. The middle camera can also zoom out (using an optical zoom lens) and show all six people, 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 CSV 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 CSVs and Capture Scenes. If a CSV is included in a Simultaneous Transmission Set, then all Media Captures in the CSV are included in the Simultaneous Transmission Set. If a Capture Scene is included in a Simultaneous Transmission Set, then all its CSVs (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 CSVs in a Capture Scene, then the Consumer MAY choose one CSV for each Media type, or it 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 follows:

* 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.

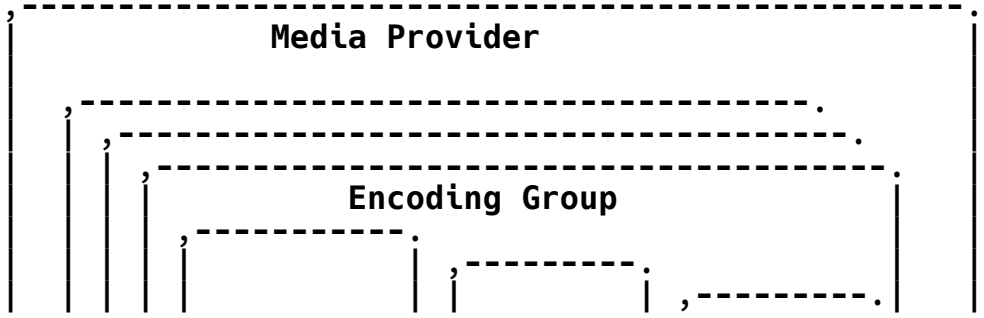
The Encoding Group data structure contains:

* 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.



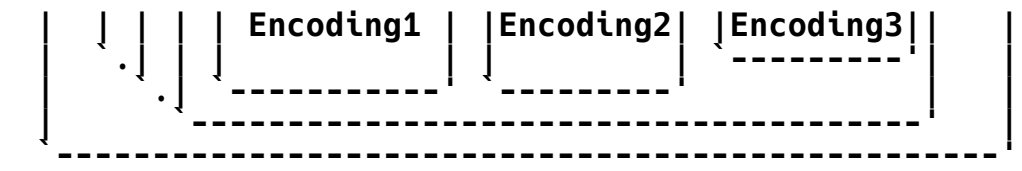


Figure 4: Encoding Group Structure

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 352x288p30, 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 CSV to be used simultaneously.

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

After receiving the Provider's Advertisement message (which 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/etc.) 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 is 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 utilize 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 subset 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 an 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, 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:

Three cameras, three displays, and a six-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 two 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 six 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 three cameras, three displays, and six seats. Each camera captures two people. The six seats are not all in a straight line.

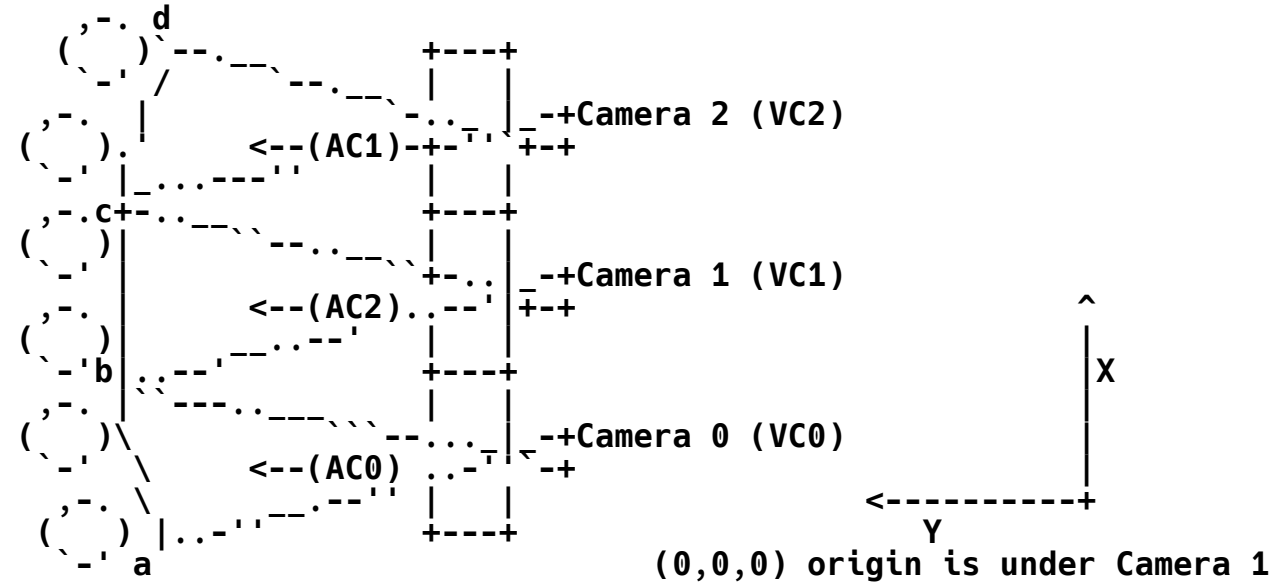


Figure 5: Room Layout Top View

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 it 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.

* AC0 (left), Encoding Group=EG3

* AC1 (right), Encoding Group=EG3

* AC2 (center), Encoding Group=EG3

* AC3 being a simple pre-mixed audio Stream from the room (mono), Encoding Group=EG3

* 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 that 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, and VC6 all use EG1 and EG1 has only three 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 three 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 three 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 CSVs 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 CSV.

Capture Scene #1
VC0, VC1, VC2
MCC3
MCC4
VC5
AC0, AC1, AC2
AC3
Capture Scene #2
VC6
AC4

Table 7: Example CSVs

Different Capture Scenes are distinct from each other and do not overlap. 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 to each other. Alternatively, another way of representing the Capture Scene is with the Capture MCC3, which automatically shows the person who is talking; this is the same 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 three Audio Captures (AC0, AC1, and 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 attribute's Area of Capture, Point of Capture, and Point on Line of Capture.

A Media Consumer would likely want to choose a CSV to receive, partially based 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, and 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 6 Mbit/s, no individual video Encoding can exceed 4 Mbit/s.

This Encoding Group also allows up to three 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 one-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, 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 CSV 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 on 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 "two-screen" system is really a "two-decoder" system, i.e., whether only one received Stream can be displayed per screen or whether more than two Streams can be received and spread across the available screen area. To enumerate three possible behaviors here for the two-screen system when it learns that the far end is "ideally" expressed via three Capture Streams:

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

2. Receive three Streams (VC0, VC1, and VC2) and display across two screens (either with each Capture being scaled to 2/3 of a screen and the center Capture being split across two 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 fourth "blank" panel. This fourth panel could potentially be used for any presentation that became active during the call.

3. Receive three Streams, decode all three, 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 three 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; 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. The Consumer would need to divide a single Encoding Group's capability by 3 either 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 coordination 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

Note: Endpoint B indicates that it sends two Streams.

Capture Scene #1	Description=USAConfRoom
VC1	Description=Audience EncodeGroupID=1
CSV(VC1)	

Table 12: Advertisement Received from Endpoint C

If the MCU wanted to provide a Multiple Content Captures containing a round-robin switched view of the audience from the three Endpoints and the speaker, it could construct the following Advertisement:

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

CSV(MCC1)	

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.

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 Endpoint 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=Center
	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=Center
	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 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 SynchronizationID=1 EncodingGroup=1
MCC2(VC2,VC5)	CaptureArea=Center MaxCaptures=1 SynchronizationID=1 EncodingGroup=1
MCC3(VC3,VC6)	CaptureArea=Right MaxCaptures=1 SynchronizationID=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 three-camera multi-screen Endpoints sees. Call this person User A, at Endpoint A. There are four 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 three-camera site, then the video from those three cameras fills three of the screens. If the person speaking 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 four loudest talkers.

User A can also see video from other Endpoints, in addition to the current person speaking, although much smaller in size. Endpoint A has four screens, so one of those screens shows up to nine other Media Captures in a tiled fashion. When video from a three-camera Endpoint appears in the tiled area, video from all three cameras appears together across the screen with correct Spatial Relationship among those three images.

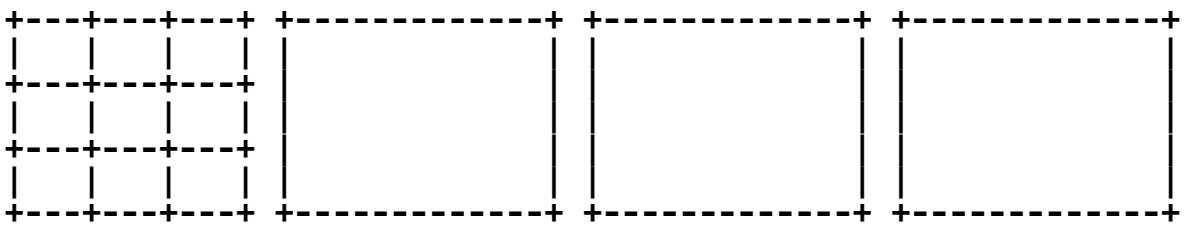


Figure 8: Endpoint A - Four-Screen Display

User B at Endpoint B sees a similar arrangement, except there are only three screens, so the nine other Media Captures are spread out across the bottom of the three displays, in a PiP format. When video from a three-camera Endpoint appears in the PiP area, video from all three cameras appears together across one screen with correct Spatial Relationship.

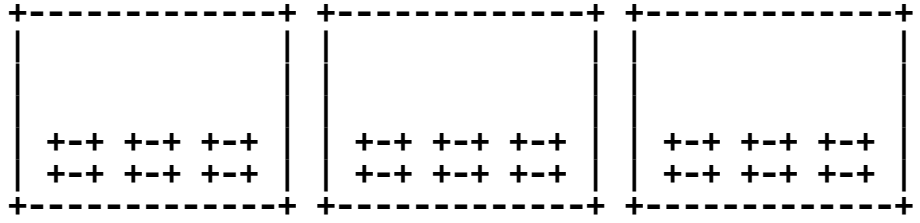


Figure 9: Endpoint B - Three-Screen Display with PiPs

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

One of the points of this example is that Endpoints A and B each want to receive three Capture Encodings for their large display areas, and nine 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, 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	CaptureArea=Left
VC11	CaptureArea=Center
VC12	CaptureArea=Right
AC3	
CSV(VC10,VC11,VC12)	
CSV(AC3)	
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 renumbered.

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 SynchronizationID=1 Policy=SoundLevel:0 EncodingGroup=1
MCC2(VC5,VC8,VC11, VC14)	CaptureArea=Center MaxCaptures=1 SynchronizationID=1 Policy=SoundLevel:0 EncodingGroup=1
MCC3(VC6,VC9,VC12,	CaptureArea=Right

VC15)	MaxCaptures=1 SynchronizationID=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 Sent to Endpoint A - Switching Parts

The above part describes the three main switched Streams that relate to site switching. MaxCaptures=1 indicates that only one Capture from the MCC is sent at a particular time. SynchronizationID=1 indicates that the source sending is synchronized. The Provider can choose to group together VC13, VC14, and VC15 for the purpose of switching according to the SynchronizationID. 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 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 the 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 Parts

The above part indicates that there are nine 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 nine 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 Parts

As MaxCaptures is 9, it indicates that the Capture Encoding contains information from nine sources at a time.

The Advertisement to Endpoint B is identical to the above, other than the fact that 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 source description (SDES) item and associated RTP header extension, as described in [RFC8849], 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 Endpoints A and B to the MCU, and from the MCU to Endpoint 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 that 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 SynchronizationID=1 Policy=SoundLevel:0 EncodingGroup=1
MCC2()	CaptureArea=Center MaxCaptures=1 SynchronizationID=1 Policy=SoundLevel:0 EncodingGroup=1
MCC3()	CaptureArea=Right MaxCaptures=1 SynchronizationID=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 Endpoint C

The MCU advertises one Scene, with four video MCCs. Three of them in CSV1 give a left, center, and 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 of the MCU is not giving all the information about all the other Endpoints' Scenes and which of those Captures are included in the MCCs. The MCU could include all that 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 Endpoints 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 Endpoints 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 Endpoint C.

13. IANA Considerations

This document has no IANA actions.

14. Security Considerations

There are several potential attacks related to telepresence, specifically the protocols used by CLUE. This is the case due to conferencing sessions, 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 a conferencing system such as the one enabled by the Conference Information Data Model for Centralized Conferencing (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 in Section 11 of [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. Datagram Transport Layer Security (DTLS) / Secure Real-time Transport Protocol (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 [RFC8848] and [RFC8849]. 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 [RFC8846] defines, through an XML schema, the syntax to be used. One type of information that 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 CSV (Section 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 CSV attributes is that while the Endpoints involved in the session have been authenticated, there are 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 that is secure against both message recovery and message modification. Further details on this are provided in the CLUE data channel solution document [RFC8850].

There are also security issues associated with the authorization to perform actions at the CLUE Endpoints to invoke specific capabilities (e.g., rearranging 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.

15. References

15.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, DOI 10.17487/RFC3261, June 2002, <https://www.rfc-editor.org/info/rfc3261>.

[RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, DOI 10.17487/RFC3264, June 2002, <https://www.rfc-editor.org/info/rfc3264>.

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, DOI 10.17487/RFC3550, July 2003, <https://www.rfc-editor.org/info/rfc3550>.

[RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, DOI 10.17487/RFC4566, July 2006, <https://www.rfc-editor.org/info/rfc4566>.

[RFC4579] Johnston, A. and O. Levin, "Session Initiation Protocol (SIP) Call Control - Conferencing for User Agents", BCP 119, RFC 4579, DOI 10.17487/RFC4579, August 2006, <https://www.rfc-editor.org/info/rfc4579>.

[RFC5239] Barnes, M., Boulton, C., and O. Levin, "A Framework for Centralized Conferencing", RFC 5239, DOI 10.17487/RFC5239, June 2008, <https://www.rfc-editor.org/info/rfc5239>.

[RFC5646] Phillips, A., Ed. and M. Davis, Ed., "Tags for Identifying Languages", BCP 47, RFC 5646, DOI 10.17487/RFC5646, September 2009, <https://www.rfc-editor.org/info/rfc5646>.

[RFC6350] Perreault, S., "vCard Format Specification", RFC 6350, DOI 10.17487/RFC6350, August 2011, <https://www.rfc-editor.org/info/rfc6350>.

[RFC6351] Perreault, S., "xCard: vCard XML Representation", RFC 6351, DOI 10.17487/RFC6351, August 2011, <https://www.rfc-editor.org/info/rfc6351>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8846] Presta, R. and S P. Romano, "An XML Schema for the Controlling Multiple Streams for Telepresence (CLUE) Data Model", RFC 8846, DOI 10.17487/RFC8846, January 2021, <http://www.rfc-editor.org/info/rfc8846>.

[RFC8847] Presta, R. and S P. Romano, "Protocol for Controlling Multiple Streams for Telepresence (CLUE)", RFC 8847, DOI 10.17487/RFC8847, January 2021, <https://www.rfc-editor.org/info/rfc8847>.

[RFC8848] Hanton, R., Kyzivat, P., Xiao, L., and C. Groves, "Session Signaling for Controlling Multiple Streams for Telepresence (CLUE)", RFC 8848, DOI 10.17487/RFC8848, January 2021, <https://www.rfc-editor.org/info/rfc8848>.

[RFC8850] Holmberg, C., "Controlling Multiple Streams for Telepresence (CLUE) Protocol Data Channel", RFC 8850, DOI 10.17487/RFC8850, January 2021, <https://www.rfc-editor.org/info/rfc8850>.

15.2. Informative References

[RFC4353] Rosenberg, J., "A Framework for Conferencing with the Session Initiation Protocol (SIP)", RFC 4353, DOI 10.17487/RFC4353, February 2006, <https://www.rfc-editor.org/info/rfc4353>.

[RFC7201] Westerlund, M. and C. Perkins, "Options for Securing RTP Sessions", RFC 7201, DOI 10.17487/RFC7201, April 2014, <https://www.rfc-editor.org/info/rfc7201>.

[RFC7202] Perkins, C. and M. Westerlund, "Securing the RTP Framework: Why RTP Does Not Mandate a Single Media Security Solution", RFC 7202, DOI 10.17487/RFC7202, April 2014, <https://www.rfc-editor.org/info/rfc7202>.

[RFC7205] Romanow, A., Botzko, S., Duckworth, M., and R. Even, Ed., "Use Cases for Telepresence Multistreams", RFC 7205, DOI 10.17487/RFC7205, April 2014, <https://www.rfc-editor.org/info/rfc7205>.

[RFC7262] Romanow, A., Botzko, S., and M. Barnes, "Requirements for Telepresence Multistreams", RFC 7262, DOI 10.17487/RFC7262, June 2014, <https://www.rfc-editor.org/info/rfc7262>.

[RFC7667] Westerlund, M. and S. Wenger, "RTP Topologies", RFC 7667, DOI 10.17487/RFC7667, November 2015, <https://www.rfc-editor.org/info/rfc7667>.

[RFC8849] Even, R. and J. Lennox, "Mapping RTP Streams to Controlling Multiple Streams for Telepresence (CLUE) Media Captures", RFC 8849, DOI 10.17487/RFC8849, January 2021, <https://www.rfc-editor.org/info/rfc8849>.

Acknowledgements

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

Authors' Addresses

Mark Duckworth (editor)

Email: mrducky73@outlook.com

Andrew Pepperell
Acano
Uxbridge
United Kingdom

Email: apeppere@gmail.com

Stephan Wenger
Tencent
2747 Park Blvd.
Palo Alto, CA 94306
United States of America

Email: stewe@stewe.org