CLUE WG
Internet-Draft
Intended status: Informational
Expires: January 16, 2014

M. Duckworth
Polycom

July 15, 2013

CLUE Switching Mixer Example
draft-duckworth-clue-switching-example-01

Abstract

This document presents an example multipoint use case scenario for CLUE. This example uses the media switching variety of the Topo-Mixer RTP topology. This example is intended to promote discussion about how to implement it using the CLUE Framework, and whether or not the framework as currently defined is sufficient to enable this use case.

This version is incomplete, and is intended to raise questions and prompt discussion.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

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

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 16, 2014.

Table of Contents

1. Introduction

This document presents an example multipoint use case scenario for CLUE. This example uses the media switching variety of the Topo-Mixer RTP topology. This example is intended to promote discussion about how to implement it using the CLUE Framework [I-D.ietf-clue-framework], and whether or not the framework as currently defined is sufficient to enable this use case.

From the requirements document [I-D.ietf-clue-telepresence-requirements]:

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

This example uses the switching approach.

[I-D.ietf-clue-rtp-mapping] says media-switching mixer is one of the RTP topologies relevant for CLUE. The media switching variety of Topo-Mixer is described in section 3.6.2 of [I-D.ietf-avtcore-rtp-topologies-update]. In this topology, the mixer provides one or more conceptual sources selecting one source at a time from the original sources. The mixer creates a conference-wide RTP session by sharing remote SSRC values as CSRCs to all conference participants.

The basic scenario for this example is a multipoint conference consisting of some traditional single-camera single-screen endpoints and some 3-camera multi-screen endpoints. Each endpoint receives multiple Capture Encodings that originated from several other endpoints. The multi-screen endpoints show the currently speaking endpoint's video using a large area of the display screens, and also show other recent speakers in smaller size using less screen space.

Since the middlebox (the mixer) is of the switching variety it is not doing any video composition. The endpoints are responsible for composing video streams to be rendered on the endpoint's display screens. The mixer sends several Capture Encodings to each endpoint, with those Capture Encodings originally coming from several other endpoints. So each endpoint receives many capture encodings, representing Media Captures that originate at other endpoints. The multi-camera endpoints send multiple Media Captures, while the single-camera endpoints send just one Media Capture. Each Media Capture could have multiple Capture Encodings, however.

The mixer selects which original sources it sends to the endpoints based on speech activity, using a policy defined by the mixer.

When completed, this example should be added to the examples in the Framework.

2. Scenario from user's point of view

From the human user's point of view, this example is a more specific case of the general multipoint scenario in the use cases document [I-D.ietf-clue-telepresence-use-cases]. Consider a conference with these endpoints:

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 cameras
Endpoint G - 1 screen, 1 camera

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

User A can also see video from other endpoints, in addition to the current talker, although much smaller in size. Endpoint A has 4 screens, so one of those screens shows up to 9 other Media Captures in a tiled fashion.

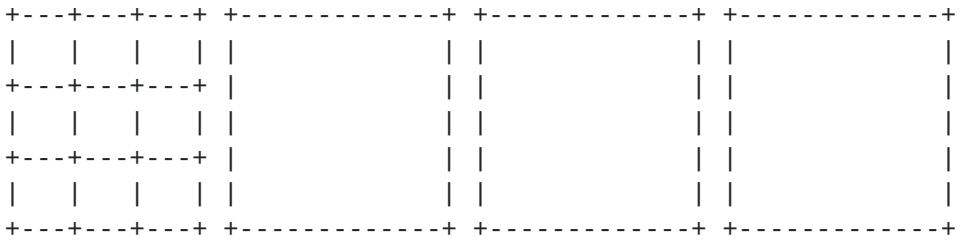

Figure 1: Endpoint A - 4 Screen Display

User B at Endpoint B sees a similar arrangement, except there are only 3 screens, so the 9 other Media Captures are spread out across the bottom of the 3 displays, in a picture-in-picture (PIP) format.

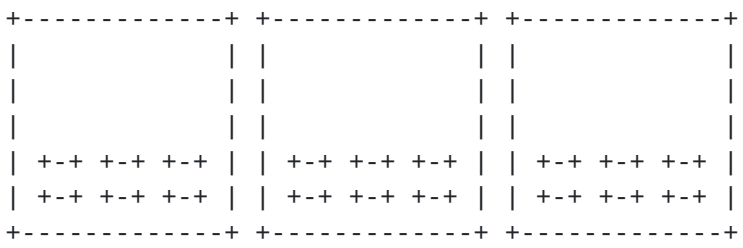

Figure 2: Endpoint B - 3 Screen Display with PiPs

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

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

3. Mixer Advertisement

The Media Provider in the mixer sends a CLUE Advertisement to each endpoint in the conference. There are different possibilities for how the mixer might construct advertisements. The mixer could advertise one Capture Scene with many captures, or many scenes with fewer captures. Each alternative has issues discussed below.

3.1. Advertising one big scene

The Provider in the mixer can advertise one Capture Scene, with many Capture Scene Entries (CSE), each with a different number of Media Captures. Say the Provider wants to send up to 12 Media Captures, it could advertise one CSE with 12 switched captures, one with 11, one with 10, etc. These switched Media Captures are distinct from the Media Captures sent from the endpoints. But these switched media captures get their media from those endpoint Media Captures (really their encodings).

```
 Capture Scene 1:
 CSE1 (VC1, VC2, VC3, VC4, VC5, VC6, VC7, VC8, VC9, VC10, VC11, VC12)
 CSE2 (VC1, VC2, VC3, VC4, VC5, VC6, VC7, VC8, VC9, VC10, VC11)
 CSE3 (VC1, VC2, VC3, VC4, VC5, VC6, VC7, VC8, VC9, VC10)
 . . .
 CSE12 (VC1)
```

Figure 3: One Big Capture Scene

Each CSE is just a subset of the CSE above it, which is how the Provider is indicating that any of these subsets is considered a view of the entire scene. The swtiched attribute draft [I-D.pepperell-clue-switched-attribute] suggested this same information could be expressed more simply by adding a CSE attribute for "explicitly signalling that a subset of the constituent captures can be used to produce a valid representation of that scene." This seems like a useful attribute, if the group decides this approach makes sense in general. A Consumer could then pick the number of Media Captures the Consumer wants to receive.

Another possibility for handling subsets is for the Provider to use the media capture priority attribute, to indicate different priorities among the many captures in a single large CSE. The Consumer could pick the ones with the highest priority if it doesn't want to receive all of them.

With either way of simplifying subsets, the capture scene reduces to:

```
 Capture Scene 1:
 CSE1 (VC1, VC2, VC3, VC4, VC5, VC6, VC7, VC8, VC9, VC10, VC11, VC12)
```

Figure 4: One Large Capture Scene with Subset Mechanism

But how does the Consumer learn about the spatial information it needs to know in order to render these captures in the correct spatial relationship to each other? The consumer layout draft [I-D.hansen-clue-consumer-layout] discusses this exact issue, and some reasons why CLUE doesn't have a good solution for this type of use case. Here is a summary of issues:

1. The mixer could advertise all the other endpoint's captures, but as the size of the conference increases the number of captures that must be advertised will quickly become impractical.
2. If the renderer had all the spatial information about all possible original source captures it might receive, it would enable correct spatial rendering. But as the number of captures per endpoint and the number of endpoints in a conference rise caching all the data becomes impractical. Or would it be

practical by using something like an extension to SIP Event Package [RFC4575] and XCON Data Model [RFC6501]?

3. An alternative would be for A to request the originating capture information for streams it is receiving, or for the MCU to send it whenever it switches streams. However, because the RTP packets and the CLUE capture information will be sent in separate channels this will lead to cases where A is receiving RTP packets but has not yet received the corresponding capture data and the same problem occurs.
4. The mixer could advertise its own generated spatial information (with "no scale" coordinates) to express a relation among all the captures in a scene. But this is overconstrained, because all 12 captures in this example do not have a relation with each other. How would the Consumer know which spatial relationships are meaningful and which are not? What would this spatial information really mean?

The consumer layout draft [I-D.hansen-clue-consumer-layout] proposes a solution to overcome all these issues. The summary is to add a mechanism by which the Consumer can send "Area of Display" information to the Provider as part of the Configure message. The Provider can use that information to inform its choice when switching video, to ensure video captures with real spatial relationships maintain those relationships as best as possible at the rendering side. The swtiched attribute draft [I-D.pepperell-clue-switched-attribute] makes a very similar proposal.

3.2. Advertising multiple scenes

The Provider could advertise multiple scenes, each one representing a different level in the recent talker list, and also representing spatial information without the overconstrained problem of a single large capture scene. Each Scene could have several CSEs, with different numbers of captures.

Scene 1: current and most recent talkers
Scene 2: next most recent talkers
Scene 3: next most recent talkers
Scene 4: next most recent talkers

The grouping of Media Captures into the CSEs in each Scene indicates the mixer is responsible for maintaining a useful spatial relationship between the original source Media Captures it switches into these conceptual Media Captures. The mixer provides spatial information, using "no scale" coordinates. Captures have a spatial relationship only with other captures in the same scene.

The mixer should use the priority attribute to indicate the Media Captures in Scene 1 are highest priority, Scene 2 is next highest, and so on.

Each capture scene has entries with only a small number of captures in each entry. The number of captures in an entry needs to be only large enough to account for the maximum number of captures that would have real spatial relationships from their original source. In this example it is three, because no endpoint has more than 3 captures that are spatially related to each other.

```
Capture Scene 1:
  CSE1 (VC1, VC2, VC3)

Capture Scene 2:
  CSE1 (VC4, VC5, VC6)

Capture Scene 3:
  CSE1 (VC7, VC8, VC9)

Capture Scene 4:
  CSE1 (VC10, VC11, VC12)
```

Figure 5: Many Capture Scenes

As with the "one big scene" method, the Advertisement could use some mechanism to indicate a subset of captures from a CSE is okay to use, and still considered a representation of the whole scene.

For the multiple scene approach, the spatial relationships can be handled in a straightforward manner by the spatial attributes the mixer puts in its advertisement. The mixer can ensure that when it switches media captures from a multi-camera source into its outgoing captures, it puts them together in the correct order that it described in the advertisement. And when it switches captures from single-camera sources, it could also pick multiple single camera sources and assign them to a consistent conceptual spatial relation, even though they don't have a real physical relationship.

This approach has the advantage that the Provider can give spatial information that is not so overconstrained as in the one big scene approach. But it could still be somewhat overconstrained, for example when the provider switches in the captures from single camera endpoints E, F, and G into VC1, VC2, and VC3.

Author's Note: Are there issues with this approach that should be described here? At the interim meeting, people seemed to be leaning toward the "one big scene" approach, but I didn't come away with a

set of issues against this "multiple scenes" approach.

3.3. Other ways of advertising

What other ways should be considered?

4. Endpoint Selecting from Advertisement

This section describes how the Endpoint Consumer selects Media Captures from the advertisement.

4.1. One big scene

The multi-screen Consumer knows it wants to receive 12 captures, so it picks all 12 captures from the scene.

A single screen endpoint might choose to receive only 1 capture, or maybe 3 or 4, depending on how it wants to render video for showing to the user.

Issue: If the single screen endpoint wants to show 1 large image and three PiPs, then it must ask to receive 4 captures. But how could it ask for one capture that represents the whole scene, plus 3 others that are additional lower priority, and that the 3 others shouldn't have a spatial relationship with the one large one? The "Area of Display" idea would solve this issue. A 2 screen consumer would be similar to a single screen endpoint, regarding this issue.

A simple endpoint that does not want to do its own local compositing would simply request the number of captures it wants to receive, and place them on its displays according to the spatial information in the mixer advertisement.

4.2. Multiple scenes

The multi-screen Consumer knows it wants to receive 12 captures, and knows it is capable of putting up to 3 captures side by side for spatial relationship, so it picks the Media Captures with the highest priority first (one CSE of 3 captures), then the next highest (another CSE in another Scene with 3 captures), and so on until it has picked 12 captures. The spatial information in the mixer's advertisement is enough (and not too much) for the consumer to display the captures with correct spatial relationships.

The Consumer renderer assigns the scene with highest priority captures to the largest areas on its display screens, and it assigns each other scene to the smaller areas on its screens. These

assigments can remain static, they don't need to change when the mixer switches between sources for these media captures.

A simple endpoint that does not want to do its own local compositing would simply request the number of captures it wants to receive, and place them on its displays according to the spatial information in the mixer advertisement. It would choose between the multiple scenes based on the priority of captures, choosing the higher priority ones first.

5. Open issues

1. Add audio considerations - how to switch and render audio consistent with video. Add audio to the example
2. Consider how the scene-switch-policy attribute can be used with this scenario

6. Acknowledgements

Thanks to Stephan Wenger, Rob Hansen, and Andy Pepperell for contributing to the ideas in this example.

7. Informative References

[I-D.ietf-clue-telepresence-use-cases]
Romanow, A., Botzko, S., Duckworth, M., and R. Even, "Use Cases for Telepresence Multi-streams", draft-ietf-clue-telepresence-use-cases-05 (work in progress), April 2013.

[I-D.ietf-clue-telepresence-requirements]
Romanow, A. and S. Botzko, "Requirements for Telepresence Multi-Streams", draft-ietf-clue-telepresence-requirements-03 (work in progress), January 2013.

[I-D.ietf-clue-framework]
Duckworth, M., Pepperell, A., and S. Wenger, "Framework for Telepresence Multi-Streams", draft-ietf-clue-framework-11 (work in progress), July 2013.

[I-D.ietf-clue-rtp-mapping]
Even, R. and J. Lennox, "Mapping RTP streams to CLUE media captures", draft-ietf-clue-rtp-mapping-00 (work in

progress), February 2013.

[I-D.ietf-avtcore-rtp-topologies-update]
Westerlund, M. and S. Wenger, "RTP Topologies", draft-ietf-avtcore-rtp-topologies-update-00 (work in progress), April 2013.

[I-D.pepperell-clue-switched-attribute]
Pepperell, A., Romanow, A., Hansen, R., and B. Baldino, "Use of switched capture attribute & spatial co-ordinates in advanced cases", draft-pepperell-clue-switched-attribute-00 (work in progress), May 2012.

[I-D.hansen-clue-consumer-layout]
Hansen, R., Pepperell, A., Romanow, A., Baldino, B., and M. Duckworth, "The need for consumer spatial information in CLUE", draft-hansen-clue-consumer-layout-00 (work in progress), May 2012.

[RFC4575] Rosenberg, J., Schulzrinne, H., and O. Levin, "A Session Initiation Protocol (SIP) Event Package for Conference State", RFC 4575, August 2006.

[RFC6501] Novo, O., Camarillo, G., Morgan, D., and J. Urpalainen, "Conference Information Data Model for Centralized Conferencing (XCON)", RFC 6501, March 2012.

Author's Address

Mark Duckworth
Polycom

Email: mark.duckworth@polycom.com