

CLUE
Internet-Draft
Intended status: Standards Track
Expires: August 29, 2013

R. Hansen
Cisco Systems
February 25, 2013

SDP and CLUE message interactions
draft-hansen-clue-sdp-interaction-01

Abstract

This document attempts to help resolve some of the complexities of interaction between SDP and CLUE messages in call flows by providing some strategies and some suggested syntax.

Status of this Memo

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

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

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

This Internet-Draft will expire on August 29, 2013.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction	3
2. Terminology	3
3. Initial Assumptions	3
4. The CLUE framework: dividing the information between SDP and CLUE messaging	4
4.1. CLUE information principally in CLUE channel	4
4.2. Media encoding/decoding information in SDP, media content information in CLUE messaging	4
5. Interdependence of SDP and CLUE negotiation	6
6. Security Considerations	7
7. References	7
7.1. Normative References	7
7.2. Informative References	7
Appendix A. Changes From Draft -00	8
Author's Address	8

1. Introduction

One issue that has repeatedly come up in the development of CLUE is the interconnected nature of many of the issues - making decisions in any one area requires that decisions are made in other areas. One particularly problematic area has been that of producing call flows: many of the decisions that need to be made revolve around how offer/answer exchanges and CLUE messages will interact, but without a good understanding of what will be in SDP and what will be in CLUE these decisions have been difficult to make.

In the hope of resolving some of these issues and allowing us to make more progress on the subject of call flows and CLUE signalling generally this draft addresses two issues that are hopefully not dependent on decisions in other areas, both aspects of the relationship between CLUE signalling and SDP. Hopefully this draft will either provoke discussion, or document decisions that people feel are obvious but aren't currently reflected in writing.

2. Terminology

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

3. Initial Assumptions

This section enumerates a few assumptions based on previous discussion which are, at this stage, hopefully uncontroversial.

CLUE information such as capture descriptions are unsuitable for SDP, and as such there will be an alternate method for sending CLUE messages end to end. In a call scenario where both sides wish to send and receive this CLUE negotiation takes the form of two independent, uni-directional exchanges; on each exchange one device provides its send capabilities while the other side determines what it wishes to receive.

This CLUE negotiation will never enable or require a call to exceed boundaries negotiated in SDP. This most obviously applies to bandwidth, both for the total call and for negotiated sessions, but also means that codec-specific limitations such as the maximum number of H.264 macroblocks a second a receiver can process MUST be respected.

4. The CLUE framework: dividing the information between SDP and CLUE messaging

The CLUE Framework [I-D.ietf-clue-framework] defines the information that will be needed to successfully negotiate a CLUE call, but does not define the mechanism by which this information is conveyed. This section provides two options for dividing this information between SDP and CLUE signalling, without proposing explicit signalling for either channel (merely what information needs to be conveyed in each).

4.1. CLUE information principally in CLUE channel

One approach that has been a major part of CLUE discussions has been to make no significant additions to SDP, and continue to use it only for the negotiation of RTP sessions. The sessions are then potentially subdivided into multiple streams using CLUE signalling. In this model standard SDP signalling provides the envelope within which CLUE negotiates the number and content of multiple streams.

This method has a number of advantages - there is no need for additional SDP syntax, making interoperability with existing devices simple and concentrating new signalling in a single location (the CLUE negotiation). There is also clear separation of responsibilities between SDP and CLUE: as normal SDP negotiates the specifics of the RTP sessions: address and ports, supported codecs, receive maxima and so on, while CLUE messaging then specifies how many streams are to be multiplexed on a port, details for demultiplexing, content of those streams, encoding limits and so on. The only necessary addition to the SDP would be a label [RFC4574] attribute per media line to allow CLUE messaging to identify them.

Unfortunately, there are some downsides to this approach. The primary one is that all multiplexing of streams is entirely dependant on the CLUE channel - as such this is not a method applicable to other applications. Since other groups within the IETF have an interest in such multiplexing for reasons other than enabling telepresence scenarios they would have to invent other methods for negotiating similar multiplexing - both inefficient, and likely problematic when CLUE and some other solution involving multistreaming are both used in the call scenario.

4.2. Media encoding/decoding information in SDP, media content information in CLUE messaging

An alternative approach is to divide the information in the CLUE Framework [I-D.ietf-clue-framework] into the information specific to encoding and decoding RTP streams, and the content of those streams.

On the advertising side this split is fairly natural: most of the information in the framework relates to the number, content, physical dimensions and simultaneity of the media captures available, information related to the contents of the media streams rather than the streams themselves. In contrast, the encoder and encoder group information gives the limits on the media streams the sender can provide, with parameters such as bandwidth, max h.264 macroblocks per second and other parameters relevant to SDP. These are defined as sender limitations rather than receiver ones and so are not directly analogous to existing SDP parameters, but are better suited to SDP than CLUE.

When it comes to receiver selection the separation between parameters that logically should be in CLUE and should be in SDP is no longer so clean-cut, as the receiver must specify capture encodings, choosing both which captures they wish to receive and the media limitations on such streams. The latter limitations are obviously suited to SDP, but information about captures is more relevant to the CLUE channel. The CLUE-specific information, however, is limited to simply selecting a capture for the stream.

The ability to describe the sender's encoder limitations for multiplexed streams along with the receiver's selection of those streams and the media limitations, SSRCS and other demultiplexing information are all requirements that are not specific to CLUE; having them in SDP means that a consistent mechanism can be used by CLUE as well as by other call scenarios wishing to support additional media streams in this fashion. Capture information, in contrast, is CLUE-specific and as such is sensible to keep in the CLUE channel. The CLUE channel will also reference the SDP, linking captures to encoding capabilities and identifying which capture is desired for each stream. This split of information means that any change in capture information on the part of a sender does not necessitate an offer/answer exchange of SDP if there is no corresponding change to the encoding capabilities of that sender - only a new CLUE advertisement is required.

This approach leads to a number of dependencies between the SDP and CLUE messages - the sender must define which captures and capture scenes are usable with which streams/encodings, while the receiver must define what capture they wish to receive with a particular encoding. These could take the form of references in SDP to CLUE, references in CLUE to SDP or references in each to the other. However, this draft proposes that all such references MUST be from CLUE messages to SDP, not the other way around. By ensuring all dependencies are unidirectional it reduces the complexity of integrating the two signalling methods. There are multiple reasons for having references be CLUE->SDP and not the other way around: one

is that logically CLUE is providing metadata about the contents of streams that are negotiated in SDP, so it makes sense for CLUE to be dependent on SDP and not visa-versa. Another is that middle boxes wishing to monitor or alter SDP can then do so without necessarily needing to involve themselves in the CLUE channel as SDP remains self-contained.

5. Interdependence of SDP and CLUE negotiation

With separate negotiation of SDP and CLUE there is the question of how to deal with dependencies between these two channels. The number of dependencies depends on how the information defined in the CLUE Framework [I-D.ietf-clue-framework] is split between SDP and CLUE, as discussed in the previous section, but even in the case where all new information is in CLUE there will still be some dependencies as it will be necessary to determine which m-lines the CLUE signalling is referring to. However, because we have two signalling methods changes that require alterations in both CLUE and SDP are no longer atomic: one message will be processed before the other. There has been debate within the working group about how this will be dealt with, as such a decision has significant effect upon call flows.

This draft proposes that CLUE messages and SDP messages should be independent: parameters in CLUE messages MAY exceed values negotiated in SDP, or may make reference to SDP contents not present in the most recent offer/answer exchange. Without this provision, SDP and CLUE messages become part of a single negotiation, and a change on either by either side may necessitate an exchange of the other message type. For instance, removing stream information from SDP might first necessitate sending a new CLUE message removing the references to this stream. The state machine required to ensure validity of negotiation will be complicated, and there will be a number of invalid states which must be avoided. This is further complicated by the fact that, even if both ends of a call obey the constraints to ensure validity, a middle box may choose to rewrite an SDP such that an invalid state is reached.

Making the two message types independent significantly reduces the complexity of the state machines required. And with the message flows independent there is no way for an invalid state to occur when the two negotiations contain contradictory information. A cost of this is that endpoints will now need to deal with the fact that CLUE messages may contain parameters exceeding those negotiated in SDP, or referencing SDP content that does not exist. However, this is analogous to an issue endpoints already deal with in SDP. For instance, the sum of bandwidth parameters for various m-lines can exceed the overall session bandwidth. Not only is this not invalid,

but it can be desirable, as it allows the sender to prioritise streams. What can be sent for any device is simply the intersection of what is permitted by the most recent SDP offer/answer, and the outcome of the CLUE negotiation; implementations should ignore references to entities in the other negotiation that do not exist.

This does not mean that there will be no interaction between SDP and CLUE messaging - a device wishing to add a new stream may well need to update both their SDP and their CLUE negotiations. However, there is no fixed order in which this must be done and no requirement for them to be updated in a particular order or fashion; it is left to the implementation to renegotiate the channels as it sees fit. If updates to both negotiations are required for a new stream to be added, then the new stream will not be available until both renegotiations are complete - the completion of the first renegotiation will have no effect.

6. Security Considerations

This draft only addresses how best to split information between SDP and CLUE signalling and the interdependencies between these two methods of signalling, it does not define the signalling or information itself. As such this draft should require no additional security considerations.

7. References

7.1. Normative References

- [I-D.ietf-clue-framework]
Duckworth, M., Pepperell, A., and S. Wenger, "Framework for Telepresence Multi-Streams", draft-ietf-clue-framework-09 (work in progress), February 2013.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

7.2. Informative References

- [RFC4574] Levin, O. and G. Camarillo, "The Session Description Protocol (SDP) Label Attribute", RFC 4574, August 2006.

Appendix A. Changes From Draft -00

- o Reordered main sections, as in the discussion about interdependence of SDP and CLUE is useful to reference the split between CLUE and SDP.
- o Added more detail to the argument of why dependencies should be CLUE->SDP and not the other way around or in both directions.
- o Fixed spelling issues and did some minor rewording.

Author's Address

Robert Hansen
Cisco Systems
San Jose, CA 95134
USA

Email: rohanse2@cisco.com