

CLUE protocol

draft-presta-clue-protocol-03

(to appear)

IETF 88 @ Vancouver

Outline

- Clue message types
- Extensions
 - Rationale
- CLUE session initiation
 - Version and extension negotiation
 - OPTIONS mechanism proposal
- Next steps

Message types

- Each CLUE message inherits the characteristics of the corresponding class
 1. Requests
 - Issued from a MC to a MP
 - Each request is answered by a response message
 - They are coupled by looking at the sequence number
 - Example: CONF, RE-ADV
 2. Responses
 - Answers to request messages, from the MP to the MC
 3. Notifications
 - Sent asynchronously from the MP to the MC to notify offers, as well as changes on the provider's side
 4. Acknowledgments
 - Sent from the MC to the MP to acknowledge notifications

Why notifications

- The advertisement is not semantically a request
- The configure is not semantically a response
 - There can be more than one configuration request referred to the same advertisement

Why acknowledgements (1/2)

- Needed to tell the MP the notification has been correctly received and understood by the MC...
- ...But:
 - DTLS/SCTP/ UDP channel is used by CLUE in reliable mode
 - ...hence there is no need to ack the delivery and the integrity of the message
 - A mechanism for negotiating the version and the extensions has been conceived
 - ...hence there can not be “Version incompatibility”, “Option incompatibility”, “Unsupported option” cases

Why acknowledgements (2/2)

- ACKs are useful to timely indicate:
 - XML Syntax errors in the notification
 - XML value errors in the notification
 - “Invalid value”: an invalid parameter value
 - “Conflicting parameters”: multiple values that can not be used together
 - ...

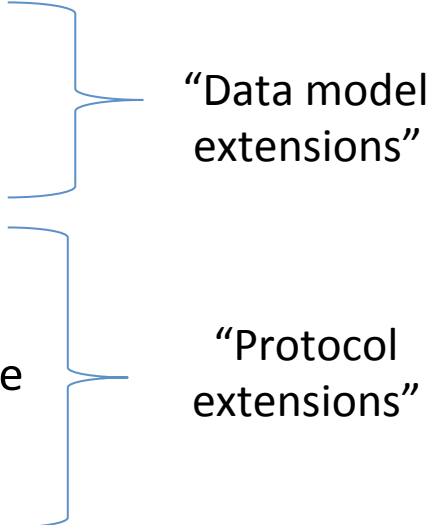
Demux rule

- A CLUE Participant can act as a MP and as a MC simultaneously on the same channel
- Messages can be demultiplexed on the basis of their type
 - The MC part receives only
 - Responses
 - Notifications
 - The MP part receives only
 - Requests
 - Acknowledgements

Extensions

- Something that is not envisioned in the current specification of the protocol...
- ...*and* something that is not envisioned in the current specification of the datamodel
 - since data model elements are included in CLUE messages
- Extensions are defined elsewhere
 - In other documents, in other XML schemas
 - An extension can be identified by the defining XML schema

Extension examples

- Extensions can be
 1. New data model elements
 - For example, a new audio capture attribute that can be used to provide an enhanced description
 2. New protocol message fields
 - For example, a new field in the request message identifying the sender of the message
 3. New protocol messages
 - For example, a new notification message
 - New information (1 and 2) can be passed in place of the “*any*” and “*any attribute*” fields of the existing schema
 - New messages (3) can be obtained by deriving the CLUE message types
- 

The OPTIONS proposal

- A mechanism for handling version and extensions negotiation ***as soon as the channel is ready***
- ...what happens as soon as the channel is instatiated between two *CLUE Participants*?
 - Reminder (from framework document):
 - A **CLUE Participant** is an entity able to use the CLUE protocol within a telepresence session
 - It can be an endpoint or an MCU able to use the CLUE protocol

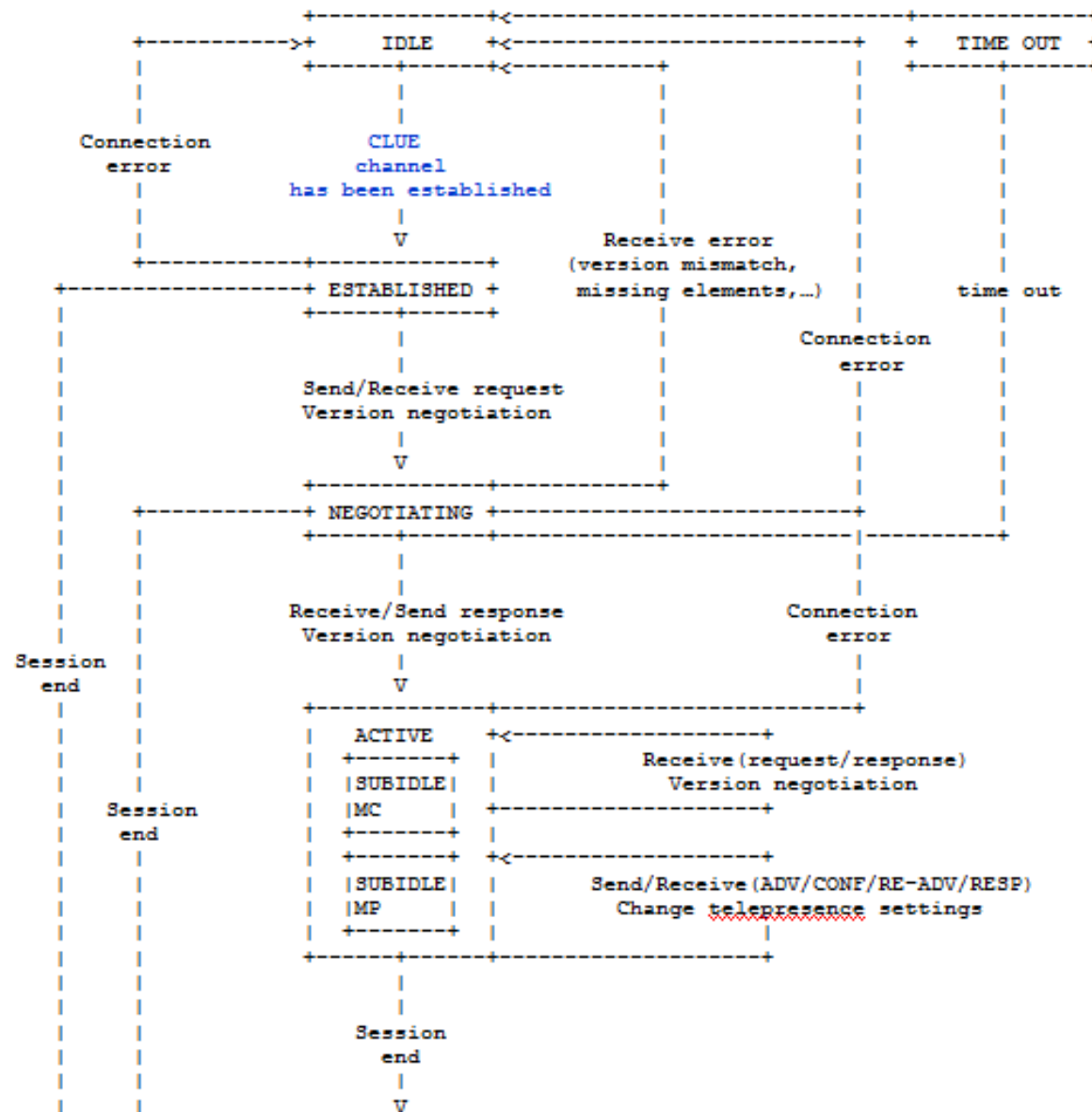
CLUE Session initiation

- Three main layers
 - Establishment of the CLUE channel
 - Considered in draft-clue-kyzivat-signaling
 - Negotiation of the CLUE protocol version and extensions
 - OPTIONS message
 - Media session description and negotiation
 - ADVERTISEMENT, CONFIGURE,...

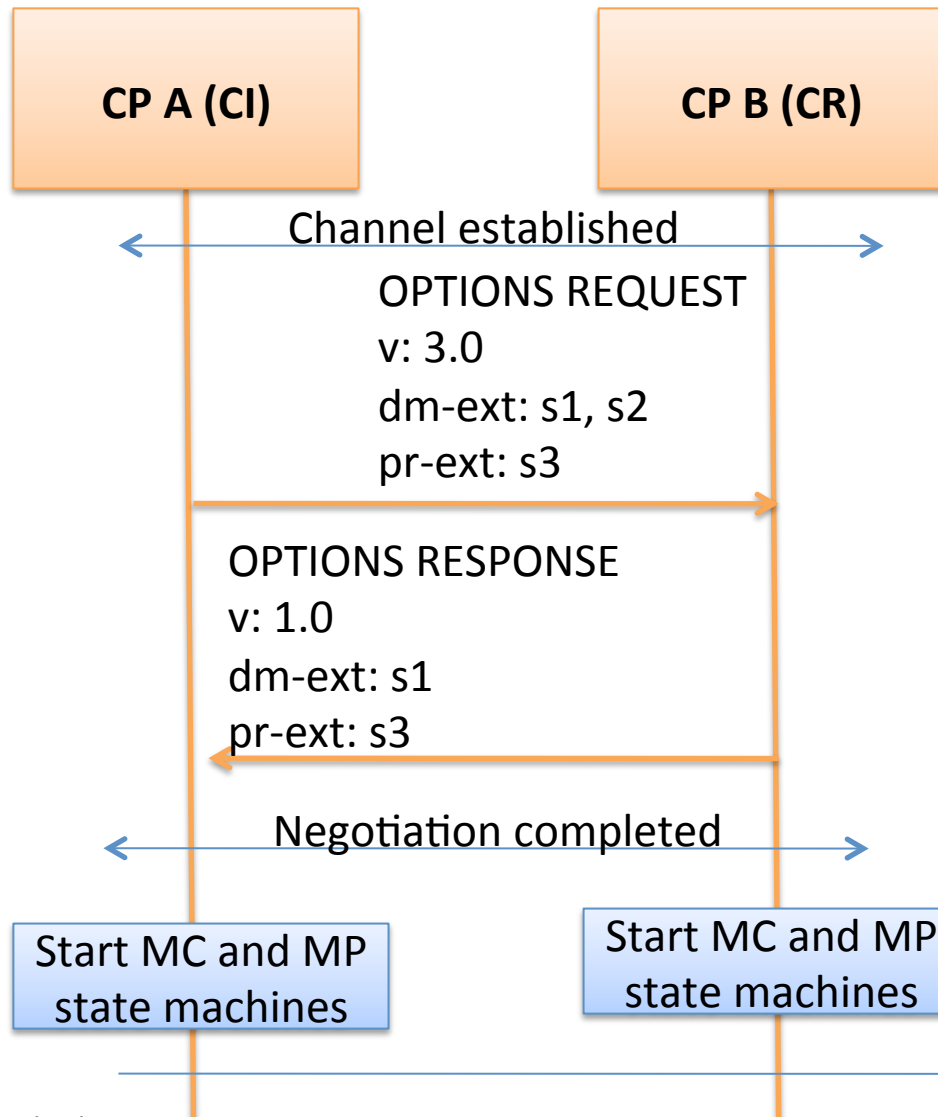
CLUE session initiation

- The CLUE Participant which is the ***Channel Initiator (CI)*** sends an OPTIONS message to the other party
 - OPTIONS request contains
 - the extensions supported by the CI
 - The version number of the CLUE protocol supported by the CI
- The CLUE Participant which is the ***Channel Receiver (CR)*** answers with an OPTIONS response
 - OPTIONS response contains
 - The extensions supported by the CR among those proposed by the CI
 - The version number of the CLUE protocol supported by the CR
 - Lower than or equal to the one proposed by the CI

CLUE Participant's state machine



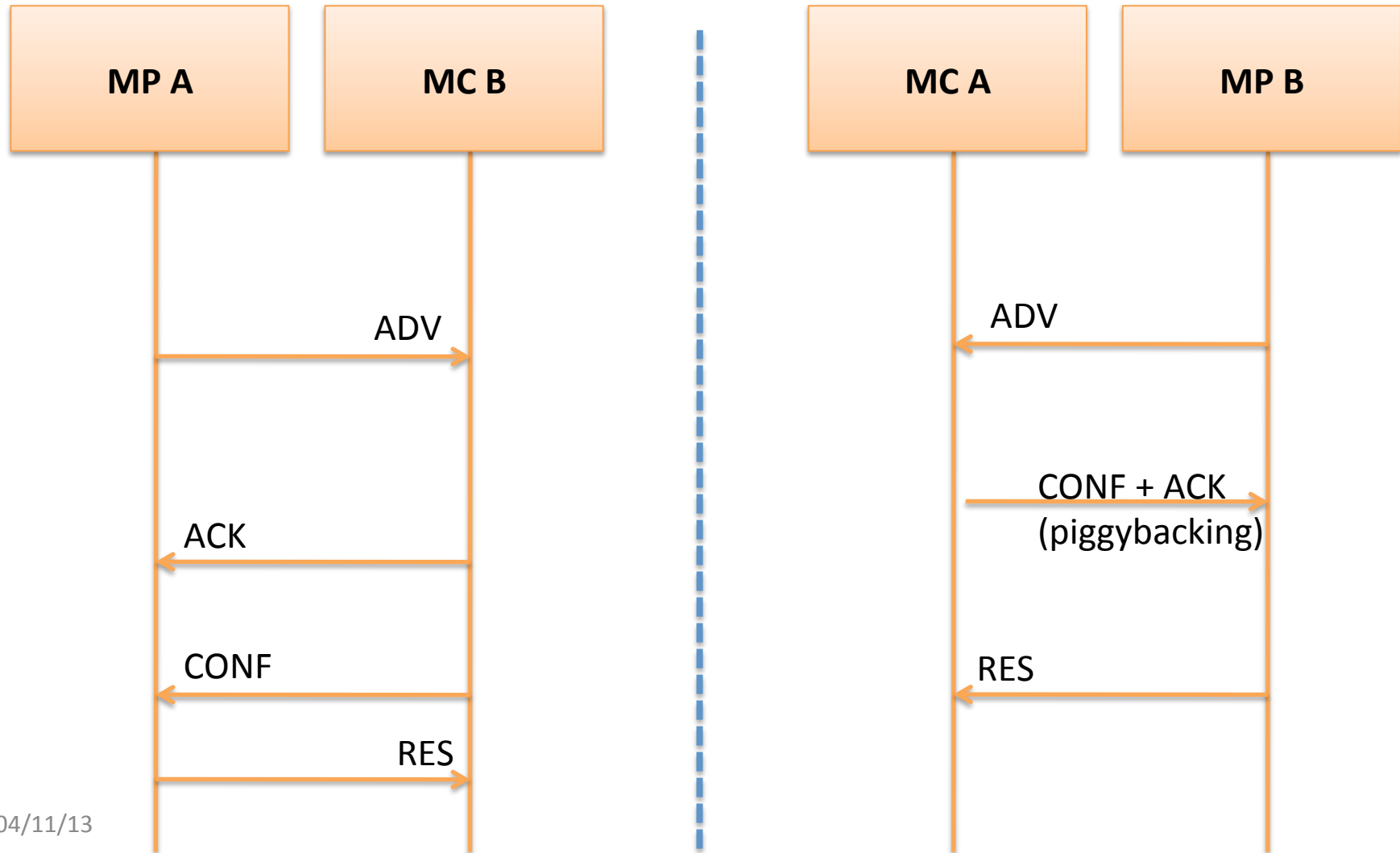
OPTIONS



From this moment on, we consider separately the two following dialogues:

1. the one btw A's MC and B's MP
2. the one btw B's MC and A's MP

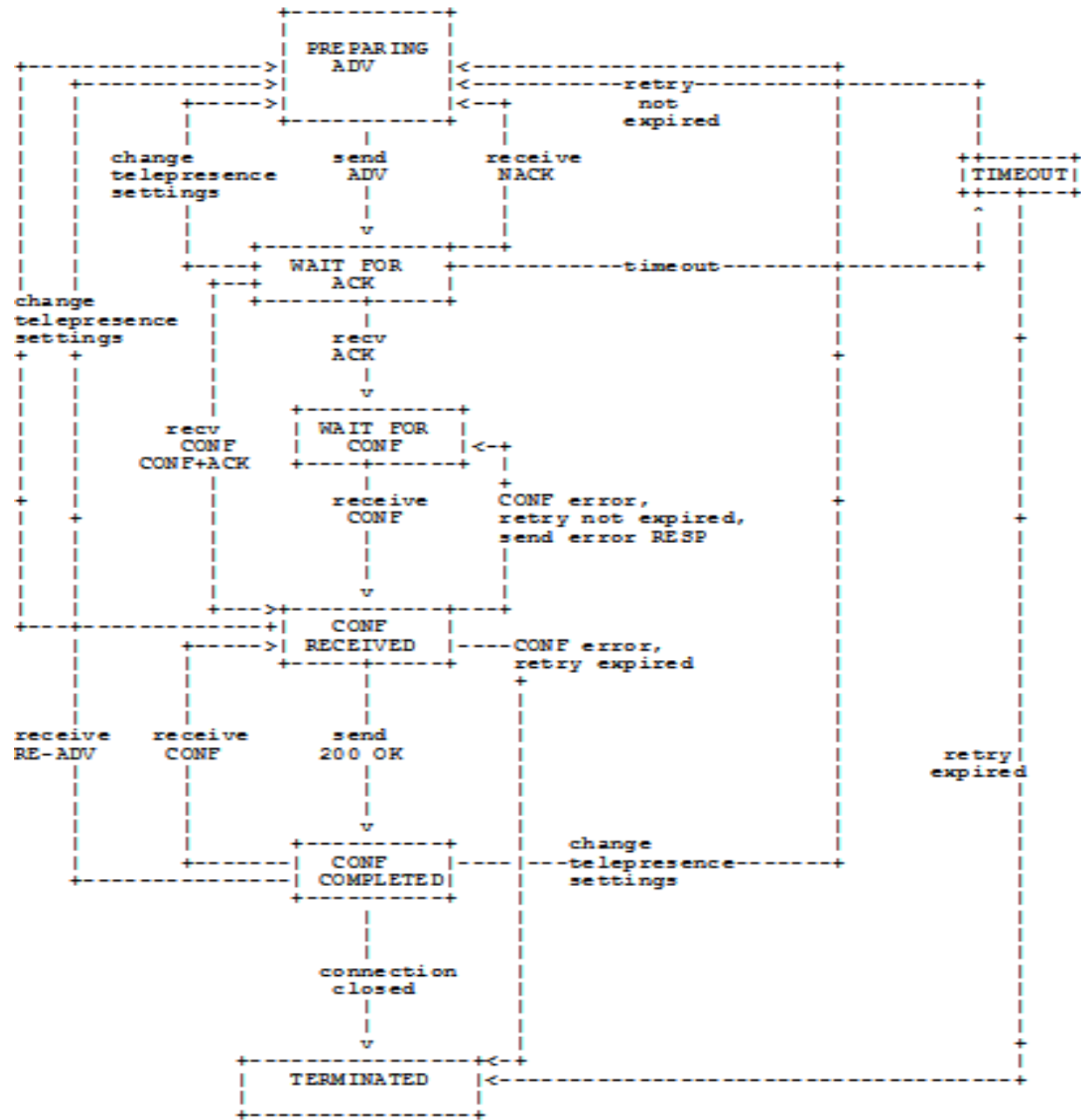
Successful establishment of a bidirectional session



Limiting cases

- Both A and B don't want to send anything
 - There will be no ADVs on the channel on both directions
 - ...until something changes
- Both A and B don't want to receive any stream
 - They use a void CONF after the ADVs
 - “I don't need anything, thanks”
 - When things change (“I want to consume media streams”), a READV request can be issued

MP's State Machine



Next steps

- Gathering feedbacks
- Update the protocol document accordingly
- Update and validate the XML Schema definitions