Abstract

This document justifies the need and lists the requirements for correlating SIP (Session Initiation Protocol) dialogs. The
correlated dialogs may or may not be related to the same multimedia session. Being able to logically correlate multiple SIP dialogs is useful for applications that, for different reasons, need to establish several SIP dialogs to provide a given service. The logical correlation of two SIP dialogs is also useful, for instance, to correlate an incoming with an outgoing dialog at a B2BUA.

Table of Contents

1. Introduction ....................................................... 3
2. Media Session correlation: Use Cases ............................. 4
   2.1. Using Two Separate UAs to Start a Conversation ....... 4
   2.2. Showing a Pre-recorded Video During a Conversation .... 4
   2.3. Sending a File from a PC During a Conversation ......... 5
   2.4. Including Live Video in a Conversation ................. 5
   2.5. Including Remote Live Video in a Conversation .......... 5
   2.6. Using Two Separate UAs in a Conversation .............. 6
3. Dialogs correlation: Use Cases .................................. 6
4. Requirements .......................................................... 7
   4.1. Media Session correlation .................................. 7
   4.2. Dialog correlation .......................................... 7
   4.3. Common requirements ....................................... 7
5. Existing Mechanisms ................................................ 8
6. Security Considerations ........................................... 9
7. IANA Considerations ............................................... 9
8. Informational References ......................................... 9
Authors' Addresses ................................................... 10
1. Introduction

This document shows the need to logically correlate multiple SIP (Session Initiation Protocol) dialogs. The correlated dialogs may or may not be related to the same multimedia session.

The SIP specification [RFC3261] defines a multimedia session as "an exchange of data between an association of participants". SDP (Session Description Protocol) is the default session description format in SIP. The SDP (Session Description Protocol) specification [RFC4566] defines a multimedia session as "a set of multimedia senders and receivers and the data streams flowing from senders to receivers".

Generally, a given participant uses a single SIP dialog to establish (or participate in) a given multimedia session. For example, two SIP user agents can use a SIP dialog between them to establish a multimedia session consisting of an audio stream and a video stream. Still, in some situations, several SIP dialogs are required to establish a single multimedia session. In order to treat the different media streams established by the different SIP dialogs as part of a single media session, there is a need to correlate those dialogs.

Also in many situations, the processing of a SIP call involves a number of different SIP dialogs that can or can not be related to the same multimedia session. A special case is the presence of the B2BUA's and other middle-boxes in the end-to-end path. A B2BUA is a logical entity that acts as both user agent server (UAS) and user agent client (UAC), and then breaks an end-to-end SIP dialog in two distinct SIP dialogs that are difficult to correlate outside the context of the B2BUA. In order to treat those two different SIP dialogs as part of a SIP call, there is a need to correlate those dialogs.

The requirements to correlate different SIP dialogs as part of a single media session and the requirements to correlate SIP dialogs that belong to the same SIP session but not necessary involve media seem to be semantically different. For this reason this document contains two distinct set of requirements one for correlating "media sessions" and different one to correlate "dialogs".

The remainder of this document is organized as follows. Section 2 contains use cases where multiple dialogs are required to establish a multimedia session. Section 3 contains use cases where multiple dialogs needs to be correlate using a unique identifier. Section 4 contains the two set of requirements for a mechanism to correlate SIP dialogs. Section 5 analyzes existing mechanisms against the
requirements we have identified and concludes that a new mechanism will need to be developed since there is no existing mechanism that meets all of them.

2. Media Session correlation: Use Cases

This section lists use cases where the UAC or the UAS is distributed. That is, the user initiating the session will use several UAs in parallel during the session or the user receiving the session invitation will use several UAs in parallel during the session.

2.1. Using Two Separate UAs to Start a Conversation

Laura is at her office. On her desk, she has a PC with a soft client and a desk phone. The PC has a low-quality built-in microphone and is connected to high-quality speakers.

Laura wants to establish a voice session with Bob using the desk phone as the microphone and the soft client as the speaker.

Two SIP dialogs will be established: one between the desk phone and Bob's UA and one between the soft client and Bob's UA. The former dialog will establish a send-only (from Laura's perspective) audio stream. The latter dialog will establish a receive-only (from Laura's perspective) audio stream.

Laura wants Bob to treat the send-only audio stream from her deskphone and the receive-only audio stream from her softclient as part of the same communication space. That is, Laura wants Bob to treat both streams as if both had been established using a single SIP dialog.

2.2. Showing a Pre-recorded Video During a Conversation

Bob has a voice-only phone and an IP-TV device. Laura has an integrated advanced multimedia phone with camera.

Bob is talking on his voice-only phone with Laura, who is on her multimedia phone.

Bob wants to show Laura part of the TV show he recorded last night. A SIP dialog between Bob's IP-TV device and Laura's UA needs to be established. This dialog will establish a video stream.

Bob talks about the show with Laura on his voice-only phone while Laura watches the show.
Bob wants Laura to treat the video stream from his IP-TV device and the voice stream from his voice-only phone as part of the same communication space. That is, Bob wants Laura to treat both streams as if both had been established using a single SIP dialog.

### 2.3. Sending a File from a PC During a Conversation

Bob has a voice-only phone and a PC with a soft client. Laura has an integrated advanced multimedia phone with support for file transfers.

Bob wants to send a file to Laura from his PC during his conversation with Laura on his voice-only phone.

A SIP dialog between Bob's PC and Laura's multimedia phone needs to be established. This dialog will establish a file transfer session.

Bob wants Laura to treat the file transfer from his PC and the voice stream from his voice-only phone as part of the same communication space. That is, Bob wants Laura to treat both streams as if both had been established using a single SIP dialog.

### 2.4. Including Live Video in a Conversation

Bob has a voice-only phone and a PC which has a soft client, a Webcam, and a low-quality built-in microphone. Laura has an integrated advanced multimedia phone with camera.

Bob wants to send a live video to Laura from his PC during his conversation with Laura.

A SIP dialog between Bob's PC and Laura's multimedia phone needs to be established. This dialog will establish a video stream.

Bob wants Laura to treat the live video stream from his PC and the voice stream from his voice-only phone as part of the same communication space. That is, Bob wants Laura to treat both streams as if both had been established using a single SIP dialog.

### 2.5. Including Remote Live Video in a Conversation

Bob, who is at his office, has a multimedia phone. At his summer cottage, Bob has a webcam-phone (e.g. a video-surveillance system). Laura has an integrated advanced multimedia phone with a camera.

During his conversation with Laura, Bob wants to show her the new summer cottage he just bought. A SIP dialog between Bob's webcam phone at this summer cottage and Laura's multimedia phone needs to be established. This dialog will establish a video stream.
Bob wants Laura to treat the live video stream from his webcam-phone and the voice stream from his voice-only phone as part of the same communication space. That is, Bob wants Laura to treat both streams as if both had been established using a single SIP dialog.

2.6. Using Two Separate UAs in a Conversation

Laura has a PC with a softclient and a desk phone. Bob has an integrated advanced multimedia phone with camera.

Laura receives on her desk phone an incoming voice-video call from Bob.

Laura decides to answer the Bob's session invitation by establishing a voice session with Bob using the desk phone and the video session using her multimedia phone. Two SIP dialogs need to be established: one between Bob's UA and Laura's desk phone and one between Bob's UA and Laura's multimedia phone.

Laura wants Bob to treat the audio stream from her deskphone and the video stream from her softclient as part of the same communication space. That is, Laura wants Bob to treat both streams as if both had been established using a single SIP dialog.

3. Dialogs correlation: Use Cases

This section lists use cases where multiple dialogs need to be correlate.

1. Certain SIP applications need to reference dialogs in out-of-dialog requests at a layer above the SIP message dialog matching rules, and wish it to work across B2BUA domains. For example, the SIP Media Control Channel Framework [I-D.ietf-mediactrl-sip-control-framework] needs to reference a dialog identifier used between a UAC and UAS by a third party. The mechanism originally used the Call-ID and remote/local-tags for such matching, but changed due to concerns over B2BUA's changing them, and now uses a new "cfw-id" SDP attribute instead which does not rely on the Call-ID value.

2. Multiple RFC 3265 Event packages use the Call-ID value in their package bodies to reference established sessions, even though they don't actually need to match a Call-ID per se - and should work across B2BUA domains.
3. The call admission control (CAC) only allows SIP INVITE requests if the network has sufficient bandwidth for the given SDP. However if a call request is forked by B2BUA's, or crosses them, the CAC model treats each fork as a separate call because there is no mechanism to tie them together. This leads to rejected forks due to CAC, when they should have been allowed to proceed. Currently proprietary SIP headers are used for this purpose.

5. Troubleshooting SIP sessions becomes complicated when multiple legs of the session are on different sides of B2BUA's, because it is impossible to correlate the legs together. Currently proprietary mechanisms are used to achieve this.

6. When SIP requests cross B2BUA's, the only form of loop detection that will stop a loop is the Max-Forwards hop-count limit being reached (value reaching zero). Via header values are removed by B2BUA's, so both spirals and simple loops cannot be detected by Via branch-parameter matching. A mechanism to correlate legs of the sessions on both sides of B2BUA's could be used to limit or avoid the loop.

4. Requirements

4.1. Media Session correlation

REQ1 It must be possible to correlate an already existing dialog or dialogs with a new dialog or dialogs as relating to the same media session.

4.2. Dialog correlation

REQ1 It must be possible to identify a set of dialogs which have a direct correlation with each other such that they represent the same SIP session, with as high a probability as possible.

4.3. Common requirements

REQ1 It must be possible, when establishing a dialog, to specify that it be correlated with one or more already existing dialogs, which dialogs, at the time they were created, did not need to specify that they might be correlated with in the future.
REQ2. The state information associated to the correlation among a set of SIP dialogs must expire (i.e., can be discarded) when the last of the SIP dialogs is terminated.

REQ3. UAs that do not implement the correlation mechanism and, thus, do not understand the correlation information they received should be able to handle the individual SIP dialogs that were supposed to be correlated as well as possible. That is, the correlation mechanism should not keep them from trying to handle the SIP dialogs.

REQ4. It must be possible to correlate outside the context of the B2BUA the dialog entering with the one leaving a B2BUA. This requirement drives the following requirements:

REQ4.1. It must be possible for correlated dialogs that pass through B2BUA’2 to continue to be correlate.

REQ4.2. The correlation mechanism must not reveal any information related to any SIP device or domain identity, including IP Address, port, hostname, domain name, username, Address-of-Record, MAC address, IP address family, transport type, etc.

REQ4.3. The correlation mechanism not reveal to the receiver of it that the Call-ID, tags, or any other SIP header or body portion have been changed by middle-boxes, with as high a probability as possible.

5. Existing Mechanisms

This section analyses existing mechanisms against the requirements we have identified. Currently, there is no mechanism that meets those requirements. Thus, a new mechanism will need to be develop in order to meet those requirements.

SIP Service Control [RFC3087] proposes to communicate context information using a distinctive Request URI. However, dialogs to be correlated do not necessary share the same Request URI.

The Join header field [RFC3911] allows inserting a new participant into a given conversation space. However, the Join operation is normally used to create or join a conference. It adds a dialog to the conversation space associated with the matched dialog and performs a media mixing or media combining. Therefore, while the mechanics of the Join mechanism are suitable to correlate dialogs,
the semantics for the correlations created by Join are different than the ones described in the previous requirements.

The Third Part Call Control mechanism [RFC3725] allows the caller to make transparent for the callee the fact that it is using separate devices for different media, and then making everything in one dialog for the callee. However there are use cases where it is preferable not to use a centralized and heavy mechanism as 3pcc but instead using a distributed and lighter mechanism. Moreover in 3pcc, the Controller is a central point for signaling, it has complete control over the call and then it needs to be involved for all the session time. With a correlation mechanism the UA that establishes the first dialog does not necessarily need to be involved all the session time. It could leave the session before the whole session ends.

6. Security Considerations

To be done.

7. IANA Considerations

This document does not require any actions by the IANA.

8. Informational References


Authors' Addresses

Gonzalo Camarillo
Ericsson
Hirsalantie 11
Jorvas  02420
Finland

Email: gonzalo.camarillo@ericsson.com

Salvatore Loreto
Ericsson
Hirsalantie 11
Jorvas  02420
Finland

Email: salvatore.loreto@ericsson.com