BLISS	A. Johnston, Ed.
Internet-Draft	Avaya
Expires: March 29, 2009	M. Soroushnejad
	V. Venkataramanan
	Sylantro Systems Corp
	P. Pepper
	Citel Technologies
	A. Kumar
	Yahoo Inc.
	September 25, 2008

Shared Appearances of a Session Initiation Protocol (SIP) Address of Record (AOR)
draft-ietf-bliss-shared-appearances-00

Status of this Memo

Abstract

This document describes the requirements and implementation of a group telephony feature commonly known as Bridged Line Appearance (BLA) or Multiple Line Appearance (MLA), or Shared Call/Line Appearance (SCA). When implemented using the Session Initiation Protocol (SIP), it is referred to as Shared Appearances (SA) of an Address of Record (AOR) since SIP does not have the concept of lines. This feature is commonly offered in the IP Centrex services and IP-PBX offerings and is likely to be implemented on SIP IP telephones and SIP feature servers used in a business environment. This document lists requirements and compares implementation options for this feature. Extensions to the SIP dialog event package are proposed.

Table of Contents

---

1. Introduction

TOC

The feature and functionality requirements for SIP user agents (UAs) supporting business telephony applications differ greatly from basic SIP user agents, both in terms of services and end user experience. In addition to basic SIP support [RFC3261] (Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, “SIP: Session Initiation Protocol,” June 2002.), many of the services in a business environment require the support for SIP extensions such as REFER [RFC3515] (Sparks, R., “The Session Initiation Protocol (SIP) Refer Method,” April 2003.), SUBSCRIBE/NOTIFY primitives [RFC3265] (Roach, A., “Session Initiation Protocol (SIP)-Specific Event Notification,” June 2002.), PUBLISH [RFC3903] (Niemi, A., “Session Initiation Protocol (SIP) Extension for Event State Publication,” October 2004.), the SIP Replaces [RFC3891] (Mahy, R., Biggs, B., and R. Dean, “The Session Initiation Protocol (SIP) "Replaces" Header,” September 2004.), and Join [RFC3911] (Mahy, R. and D. Petrie, “The Session Initiation Protocol (SIP) "Join" Header,” October 2004.), header fields, etc. Many of the popular business services have been documented in the SIP Service Examples [I-D.ietf-sipping-service-examples] (Johnston, A., Sparks, R., Cunningham, C., Donovan, S., and K. Summers, “Session Initiation Protocol Service Examples,” July 2008.).
This specification details a method for implementing a group telephony feature known in telephony as Bridged Line Appearance (BLA) or Multiple Line Appearances (MLA), one of the more popular advanced features expected of SIP IP telephony devices in a business environment. Other names for this feature include Shared Call/Line Appearance (SCA), Shared Call Status and Multiple Call Appearance (MCA). A variant of this feature is known as Single Line Extension.

This document looks at how this feature can be implemented using standard SIP [RFC3261] (Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, “SIP: Session Initiation Protocol,” June 2002.) in conjunction with [RFC3265] (Roach, A., “Session Initiation Protocol (SIP)-Specific Event Notification,” June 2002.) and [RFC3903] (Niemi, A., “Session Initiation Protocol (SIP) Extension for Event State Publication,” October 2004.) for exchanging status among user agents, and the SIP dialog state event package [RFC4235] (Rosenberg, J., Schulzrinne, H., and R. Mahy, “An INVITE-Initiated Dialog Event Package for the Session Initiation Protocol (SIP),” November 2005.) to exchange dialog state information to achieve the same. Different approaches will be discussed including the use of URI parameters, feature tags, and dialog package extensions along with the strengths and weaknesses of the various approaches.

A call flow for Single Line Extension was formerly included in the SIP Service Examples [I-D.ietf-sipping-service-examples] (Johnston, A., Sparks, R., Cunningham, C., Donovan, S., and K. Summers, “Session Initiation Protocol Service Examples,” July 2008.). However, the attempt to implement using standard SIP primitives ultimately failed, leading to its removal from that document. This document defines SIP extensions to implement this service.

In traditional telephony, the line is physical. A common scenario is for a number of business telephones to share a single or a small number of Address of Record (AOR) URIs. The sharing of this AOR between multiple UAs is what gives this feature its name. In addition, an AOR can have multiple appearances on a single UA in terms of the user interface. The appearance number relates to the user interface for the telephone - typically each appearance or an AOR has a visual display (lamp that can change color or blink) and a button (used to select the appearance). The telephony concept of line aappearance is still relevant to SIP due to the user interface considerations. It is important to keep the appearance number construct because:

1. Human users are used to the concept and will expect it in replacement systems (e.g. an overhead page announcement says "Joe pickup line 3").

2. It is a useful structure for user interface representation.

In this document, we will use the term "appearance" rather than "line appearance" since SIP does not have the concept of lines. Note that this does not mean that a conventional telephony user interface (lamps and buttons) must be used - implementations may use another metaphor as long as the appearance number is readily apparent to the user. Each AOR has a separate appearance numbering space. As a result, a given UA user interface may have multiple occurrences of the same appearance number, but they will be for different AORs.

2. Conventions used in this document

TOC

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] (Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels,” March 1997.) and indicate requirement levels for compliant mechanisms.

3. Usage Scenarios

TOC

The following examples are common applications of the Shared Appearances feature and are mentioned here as informative use cases. All these example usages can be supported by the Shared Appearances feature described in this document. The differences relate to the user interface considerations of the device.

3.1. Executive/Assistant Arrangement

TOC

The appearances on the executive's UA may also appear on the assistant's UA. The assistant may answer incoming calls to the executive and then place the call on hold for the executive to pick up. The assistant can always see the state of all calls on the executive's UA.

3.2. BLA Call Group

TOC

Users with similar business needs or tasks can be assigned to specific groups and share the line appearances of each other on each others SIP telephony devices. For example, an IT department staff of five might answer a help line which has three appearances on each phone in the IT work area. A call answered on one phone can be put on hold and picked up on another phone. A shout or an IM to another staff member can result in them taking over a call on a particular appearance. Another phone can request to be added to an appearance resulting in a conference call.

TOC

3.3. Single Line Extension

In this scenario, incoming calls are offered to a group of UAs. When one answers, the other UAs are informed. If another UA in the group selects the line (i.e. goes off hook), it is immediately bridged or joined in with the call. This mimics the way residential telephone extensions usually operate.

---

4. Requirements

TOC

The basic requirements of the shared appearance feature can be summarized as follows:
REQ-1 Incoming calls to the AOR must be offered to a group of UAs and can be answered by any of them.
REQ-2 Each UA in the group must be able to learn the call status of the others in the group for the purpose of rendering this information to the user.
REQ-3 Calls can be joined (also called bridged or conferenced together) or can be picked up (taken) by another UA in the group in a secure way.
REQ-4 The mechanism should require the minimal amount of configuration. UAs registering against the group AOR should be able to learn about each other and join the appearance group.
REQ-5 The mechanism must scale for large numbers of appearances, n, and large numbers of UAs, N, without introducing excessive messaging traffic.
REQ-6 Each call or session (incoming or outgoing) must be assigned a common "appearance" number from a managed pool administered for the AOR group. Once the session has terminated, the appearance number is released back into the pool and can be reused by another incoming or outgoing session.
REQ-7 Each UA in the group must be able to learn the appearance status of the the group.
REQ-8 There must be mechanisms to resolve appearance contention among the UAs in the group.
REQ-9 The mechanism must allow all UAs receiving an incoming session request to select the same appearance number at the time of alerting.
REQ-10 The mechanism must have a way of reconstructing appearance state after an outage that does not result in excessive traffic and processing.
REQ-11 The mechanism must have backwards compatibility such that a UA which is unaware of the feature can still register against the group AOR and make and receive calls.
REQ-12 The mechanism must not allow UAs outside the group to select or manipulate appearance numbers.

REQ-13 For privacy reasons, there must be a mechanism so that appearance information is not leaked outside the group of UAs. (e.g. "So who do you have on line 1?")
REQ-14 The mechanism must support a way for UAs to request exclusivity on a line appearance. Exclusivity means that the UA requesting it desires to have a private conversation with the external party and other UAs must not be allowed to barge-in. Exclusivity may be requested at the start of an incoming or outgoing session or during the session. An exclusivity request may be accepted or rejected by the entity providing the SA service. Therefore, the mechanism must provide a way of communicating the result back to the requester UA.
REQ-15 The mechanism should support a way for a UA to select a particular appearance number for outgoing requests prior to sending the actual request. This is often called seizure.
REQ-16 The mechanism should support a way for a UA to select a particular appearance number and also send the request at the same time. This is needed when a ringdown feature is combined with shared appearances - in this case, seizing the line is the same thing as dialing.

---

5. Normative Description

TOC

This section normatively describes the SA feature extensions.

5.1. Implementation

TOC

Many of the requirements for this service can be met using standard SIP mechanisms such as:
- A SIP Forking Proxy and Registrar/Location Service meets REQ-1.
- The SIP Dialog Package meets REQ-2.
- The SIP Replaces and Join header fields meets REQ-3.
- The SIP Registration Package meets REQ-4.
- The use of a State Agent for the Dialog Package meets REQ-5.

REQ-6 suggests the need for an entity which manages the appearance resource. Just as conferencing systems commonly have a single point of control, known as a focus, a Shared Appearance group has a single point of control of the appearance shared resource. This is defined as an Appearance Agent for a group. While an Appearance Agent can be part of a centralized server, it could also be co-resident in a member User Agent who has taken on this functionality for a group. The Appearance Agent learns the group state either by subscribing to the dialog state of each member UA individually or by dialog state publications from members.

While the appearance resource could be managed co-operatively by a group of UAs without any central control, this is not discussed in this draft, but instead is left as a research project for future standardization. It is also possible that the Appearance Agent logic could be distributed in all UAs in the group. For example, rules that govern assigning appearance numbers for incoming requests (e.g. lowest available appearance number) and rules for contention handling (e.g. when two UAs request the use of the same appearance number, hash dialog identifiers and compare with the lowest hash winning) would need to be defined and implemented.

REQs 6-13 can be implemented using a number of approaches, as discussed in the following sections.

Figure 1 illustrates the SIP components involved in supporting these common requirements of the Shared Appearance using standard SIP messages including REGISTER, INVITE, SUBSCRIBE, NOTIFY, and PUBLISH.

```
     +----------------------------+                            +----+
     |                            |                            |    |
     |     Appearance Agent       |                            | UA |
     |                            |                            |    |
     +----------------------------+                            +----+
         ^ ^ |1)SUBSCRIBE  ^  ^    4)NOTIFY            INVITE     |
         | | |(Event:reg)  |  | registration sip:alice@example.com|
         | | V             |  |     events                        V
         | | +--------------------+        +----------+7)Query+--------+
         | | |    (example.com)   |        |          |<===== |        |
         | | |                    |3) Store| Location |       | Proxy  |
         | | |     Registrar      |=======>|  Service |       |        |
         | | |                    |        |          |=====> |        |
         | | +--------------------+        +----------+8)Resp +--------+
         | |     ^       ^                                       |  |
         | |     |       |  2) REGISTER (alice)                  |  |
         | |     |       |                                       |  |
         | |   +----+ +----+                                     |  |
         | |   |    | |    |                                     |  |
         | |   |UA1 | |UA2 |                                     |  |
         | |   |    | |    |                                     |  |
         | |   +----+ +----+                                     |  |
         | |    ^  ^    ^ ^                                      |  |
         | |    |  |    | |                                      |  |
         | +----+  |    | |                                      |  |
         |         |    | +--------------------------------------+  |
         |         +----+-------------------------------------------+
         |              |             8) INVITE
         +--------------+            sip:alice@example.com
         5-7) SUBSCRIBE and/or PUBLISH
              (Event:dialog)
```

Figure 1.
The next section discusses normal SIP operations used to implement parts of the shared appearance feature.

1. The Appearance Agent SUBSCRIBES to the registration event package as outlined in [RFC3680] (Rosenberg, J., "A Session Initiation Protocol (SIP) Event Package for Registrations," March 2004.) for contacts registered to the group AOR. Thus, it has knowledge of all User Agents registered against the AOR at any point of time.

2. UAs (UA1 and UA2 in Figure 1) belong to the appearance group and, after authentication, register against the same AOR (e.g., sip:alice@example.com).

3. Each registration is stored in the Location Service.

4. The registrar notifies the Appearance Agent of successful registration at each UA.

5. UAs PUBLISH their dialog state to the State Agent in the Appearance Agent.

6. The UAs SUBSCRIBE to the Appearance Agent for the state of all dialogs as defined in [RFC3265] (Roach, A., "Session Initiation Protocol (SIP)-Specific Event Notification," June 2002.) . The Request-URI of the SUBSCRIBE could be either the AOR of the group, the Contact URI information it received in the incoming subscription from the Appearance Agent, or a provisioned URI.

7. The UAs PUBLISH their dialog information to the Appearance Agent every time their dialog state changes (i.e. receive an INVITE, enter alerting state, answer a call, terminate a call, generate an INVITE, etc.)

8. Forking Proxy forks an incoming INVITE for the AOR address to the registered user agents.

The User Agents in the group could SUBSCRIBE to each other and NOTIFY dialog state events, but in a large group the User Agents have to manage a larger number of SUBSCRIPTIONS and NOTIFICATIONS. The State Agent in the Appearance Agent helps in managing large groups better. Further, the State Agent can filter dialog state events and NOTIFY User Agents of the dialog state events which are required for the application or feature. The State Agent can also SUBSCRIBE to dialog state events with filters to reduce the number of NOTIFY messages exchanged between the State Agent and the user agents in the group. This allows a group of N UAs to each only establish a pair of dialog state subscriptions (one in each direction) to learn the dialog state of all other group members. This results in 2N total subscriptions for

the entire group. A full mesh of subscriptions without a state agent would result in N(N-1) total subscriptions.
The Appearance Agent can select the appearance number for an incoming call
OPEN ISSUE: Do we want to define another mode of operation in which UAs only PUBLISH to seize a line appearance? This assumes the Appearance Agent already knows about all dialogs related to the AOR and could publish that information to the UAs in the SA group. This approach would simply UA operation and cleanly resolve some race conditions. Should we define this mode in a separate draft?

5.2. SA Dialog Package Extensions

TOC

This specification defines three new elements as extensions to the SIP Dialog Event package [RFC3265] (Roach, A., “Session Initiation Protocol (SIP)-Specific Event Notification,” June 2002.) . The schema is defined in Section 7. The elements are <appearance>, <exclusive>;, and <joined-dialog>. All three elements are sub-elements of the <dialog> element.

5.2.1. The <appearance> element

TOC

The <appearance> element is used convey the appearance number. The appearance number is a non-negative integer. When sent in a notification in state Trying to the Appearance Agent, it is used to request an appearance number. When sent by the Appearance Agent, it indicates that the appearance number is associated with a dialog.

5.2.2. The <exclusive> element

TOC

The <exclusive> element is a boolean used indicate whether the UA will accept Join or Replaces INVITEs for this dialog. For example, some SA systems only allow call pickup when the call is on hold. In this case, the <exclusive> element should be used to explicitly indicate this, rather than implicitly by the hold state.
It is important to note that this element is a hint. Although a UA may set exclusive to true, the UA must still be ready to reject an INVITE Join relating to this dialog. Also, an INVITE Replaces might be sent to the non-SA UA to take the call. For this reason, a UA MAY also not report full dialog identifier information to the Appearance Agent for calls set to exclusive. If these dialog identifiers have already been shared with the Appearance Agent, the UA could send an INVITE Replaces

to change them and then not report the new ones to the Appearance Agent.
If the proxy knows which dialogs are marked exclusive, the proxy MAY enforce this exclusivity by rejecting INVITE Join and INVITE Replace requests containing those dialog identifiers with a 403 Forbidden response.

5.2.3. The <joined-dialog> element

TOC

The <joined-dialog> element is used convey dialog identifiers of any other dialogs which are joined (mixed or bridged) with the dialog. Only the UA which is performing the actual media mixing should include this element in notifications to the Appearance Agent. Note that this element should still be used even when the Join header field was not used to join the dialogs. For example, two separate dialogs on a UA could be joined without any SIP call control operations. The UA would report this using this header field.

5.3. Shared Appearance User Agents

TOC

User Agents that support the Shared Appearance feature MUST support the dialog state package [RFC4235] (Rosenberg, J., Schulzrinne, H., and R. Mahy, “An INVITE-Initiated Dialog Event Package for the Session Initiation Protocol (SIP),” November 2005.) with the SA extensions and the "sa" dialog event package parameter defined in this draft.
User Agents MUST use the extended package in conjunction with PUBLISH [RFC3903] (Niemi, A., “Session Initiation Protocol (SIP) Extension for Event State Publication,” October 2004.) to send local status to the Appearance Agent, and in conjunction with SUBSCRIBE and NOTIFY [RFC3265] (Roach, A., “Session Initiation Protocol (SIP)-Specific Event Notification,” June 2002.) to learn the status or other User Agents.
The publication URI is either a provisioned value or the username 'appearance-agent'. For example, a UA in the domain of example.com would publish dialog state to sip:appearance-agent@example.com
OPEN ISSUE: Do we need to specify a publish URI or should it be only by provisioning? Should a UA be able to publish to its own URI for this?
User Agents SHOULD support sending and receiving an INVITE with a Replaces [RFC3891] (Mahy, R., Biggs, B., and R. Dean, “The Session Initiation Protocol (SIP) "Replaces" Header,” September 2004.) header to allow the Call Pickup operation. User Agents MUST support sending an INVITE a Join [RFC3911] (Mahy, R. and D. Petrie, “The Session Initiation Protocol (SIP) "Join" Header,” October 2004.) header to initiate bridging. User Agents MUST support receiving an INVITE with a Join header, though they are not obligated to support bridging of

calls, either through policy, preference, or implementation limitations such as bandwidth or hardware constraints.

When publishing dialog package information, a UA MUST include all dialog identification available at the time of publication, with the exception that a UA may omit information if it wishes to prevent other UAs from joining or picking up a call. Dialog identification includes local and remote target URIs, call-id, to-tag, and from-tag. When calls are placed on hold, the "+sip.rendering=no" feature tag MUST be included in dialog package notifications.

There are two exceptions to the requirement for seizing an appearance before sending an INVITE. One is an emergency call, which should proceed regardless of the status of PUBLISH transaction. This requirement applies to both clients and servers implementing this feature. The other is when the INVITE is an attempt to bridge or take a call (i.e. contains Join or Replaces with a dialog identifier of another member of the SA group). In this case, the INVITE Join or Replaces should be sent without selecting an appearance. After the INVITE succeeds, the PUBLISH MUST be sent, reusing the appearance number of the dialog that has been bridged or taken.

UAs can tell that a set of dialogs are joined (bridged or mixed) together by the presence of one or more <joined-dialog> elements containing other SIP dialog identifiers.

Prior to placing an outbound call, UAs may select or "seize" an appearance number by sending a PUBLISH to the Appearance Agent with an <appearance> identifying the appearance number selected. In traditional telephony terms, this corresponds to "going off hook" with an analog telephone. Note that when a UA selects an appearance prior to establishment of a dialog, not all dialog information will be available. In particular, when a UA publishes an attempt to select an appearance prior to knowing the destination very minimal dialog information may be available.

A UA that does not need to select a particular appearance number (or doesn't care) would just send an INVITE as normal to place an outbound call. The Appearance Agent would select an available appearance and notify all subscribed UAs of the selection. The UA placing the call then publishes the use of this appearance number.

If an INVITE is sent and no appearance number is available, the proxy would reject the INVITE with a suitable response code and perhaps a header field indication.

For inbound calls which contain a reference to an appearance number in the INVITE, a UA MUST PUBLISH the use of an appearance number after it responds with a 2xx to establish a dialog. A UA SHOULD NOT PUBLISH the use of an appearance number for incoming requests which are in the early state or which are rejected with 4xx, 5xx or 6xx responses, as this would create excessive traffic when a large number of UAs are sharing an appearance.

The dialog state package defined in [RFC4235] (Rosenberg, J., Schulzrinne, H., and R. Mahy, “An INVITE-Initiated Dialog Event Package for the Session Initiation Protocol (SIP),” November 2005.) defines the set of messages that MAY be provided by a UA to publish state information of dialogs.

A UA should only register against the AOR if it is likely the UA will be answering incoming calls. If the UA is mainly going to be monitoring the status of the SA group calls and picking or joining calls, the UA SHOULD only subscribe to the Appearance Agent and not register against the AOR.

5.4. Appearance Agent

TOC

An Appearance Agent defined in this specification MUST implement a dialog package state agent for the UAs registered against the AOR.

The dialog package XML element dialog id (NOT the SIP dialog identifier consisting of the Call-ID, To tag, and From tag) is used for partial update processing. RFC 4235 only requires the dialog id be unique for the UA, not unique across all UAs associated with the same AOR. As a result, is possible that two UAs in a SA group might choose the same XML element dialog id for different dialogs. If this is the case, the Appearance Agent, acting as the dialog package event state compositor may need to produce different notification documents for the UAs that have conflicting dialog package XML element dialog ids.

The Appearance Agent MUST support the appearance dialog package extensions defined in this specification.

Even in trivial deployments of two shared appearance enabled user agents, race conditions can exist when both user agents attempt to utilize an appearance simultaneously (i.e. when the NOTIFY messages, that each user agent sends to the other, are active within the network during an overlapping time period). The Appearance Agent can use any policy deemed necessary to resolve races and ensure no two user agents are not allocated the same appearance number at the same time.

Appearance Agents are responsible for resolving conflicts in the appearance resource. If a UA requests the use of an appearance number that is in use by another UA, the Appearance Agent will respond based on the presence of the selection attribute in the <appearance> element as described in the previous section.

In the case where a UA is bridging two calls, the Appearance Agent may receive dialog package PUBLISHes that contain multiple dialogs with the same appearance number. This is valid and does not represent appearance number contention.

A critical aspect for reliable operation of this feature is the ability for all user agents in the BLA group to recover from network failures caused at a single UA. For example, one of the user agents in the BLA group may have answered an incoming call, notified the dialog state to other members and then experienced a network outage. The calling UA could have detected the same (using RTP or some other means) and could have hung up. However, none of the other user agents in the BLA group would get notification of this change in dialog state and their BLA appearances could stay out of sync for a long time; depending on when the network is restored, or when the Appearance Agent attempts to refresh its dialog-state subscription with the failed UA. To recover from such a failure, UAs MUST PUBLISH with a shorter expiration (expiration interval not smaller than 300 seconds is RECOMMENDED) following the notification of a "confirmed" dialog or when a Appearance Agent honors a "trying" for call origination, with the user agents that notified it of this information.

---

6. XML Schema Definition

TOC

The 'appearance' and 'exclusive' elements are defined within a new XML namespace URI. This namespace is "urn:ietf:params:xml:ns:sa-dialog-info". The schema for these elements is:

```
<?xml version="1.0" encoding="UTF-8"?>
  <xs:schema
    targetNamespace="urn:ietf:params:xml:ns:sa-dialog-info-info"
    xmlns="urn:ietf:params:xml:ns:sa-dialog-info"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="qualified">


   <xs:element name="joined-dialog" minOccurs="0" maxOccurs="unbounded">
          <xs:complexType>
            <xs:attribute name="call-id" type="xs:string"
              use="mandatory"/>
            <xs:attribute name="local-tag" type="xs:string"
              use="mandatory"/>
            <xs:attribute name="remote-tag" type="xs:string"
              use="mandatory"/>
           </xs:complexType>
        </xs:element>

        <xs:element name="appearance" minOccurs="0" maxOccurs="1">
          <xs:simpleType type="xs:integer">
          </xs:simpleType>
        </xs:element>

        <xs:element name="exclusive" minOccurs="0" maxOccurs="1">
         <xs:simpleType type="xs:boolean">
         </xs:simpleType>
        </xs:element>
  </xs:schema>
```

---

7. User Interface Considerations

TOC

The "appearance number" allocated to a call is an important concept that enables calls to be handled by multiple devices with heterogeneous user interfaces in a manner that still allows users to see a consistent model. Careful treatment of the appearance number is essential to meet the expectations of the users. Also, rendering the correct call/appearance state to users is also important.

TOC

7.1. Appearance Number Rendering

Since different UAs have different user interface capabilities, it is usual to find that some UAs have restrictions that others do not. Perfect interoperability across all UAs is clearly not possible, but by careful design, interoperability up to the limits of each UA can be achieved.
The following guidelines suggest how the appearance number should be handled in three typical user interface implementations.

7.1.1. Single Appearance UAs

TOC

These devices are constrained by only having the capability of displaying status indications for a single appearance. Despite this, it is important that devices of this type do not ignore the appearance number. The UA should still send messages annotated with an appropriate appearance number (i.e. "0"). Any call indications for appearances other than for number "0" should be rejected with a 486 or 480 response.

7.1.2. Dual Appearance UAs

TOC

These devices are essentially single appearance phones that implement call waiting. They have a very simple user interface that allows them to switch between two appearances (toggle or flash hook) and perhaps audible tones to indicate the status of the other appearance.

7.1.3. Shared Appearance UAs with Fixed Appearance Number

TOC

This UA is the typical 'business-class' hard-phone. A number of appearances are typically configured statically and labeled on buttons, and calls may be managed using these configured appearances. Any calls outside this range should be ignored, and not mapped to a free button. Users of these devices often select specific appearance numbers for outgoing calls, and the UA will need to select the appearance number and wait for confirmation from the Appearance Agent before proceeding with calls.

TOC

7.1.4. Shared Appearance UAs with Variable Appearance Number

This UA is typically a soft-phone or graphically rich user interface hard-phone. In these cases, even the idea of an appearance index may seem unnecessary. However, for these phones to be able to interwork successfully with other phone types, it is important that they still use the appearance index to govern the order of appearance of calls in progress. No specific guidance on presentation is given except that the order should be consistent. Thought should also be given to how an appearance number that has no call associated with it should be rendered to the user. These devices can typically make calls without waiting for confirmation from the Appearance Agent on the appearance number.
The problems faced by each style of user interface are readily seen in this example:

1. A call arrives at the SA group, and is assigned an appearance number of 0. All UAs should be able to render to the user the arrival of this call.

2. Another call arrives at the SA group, and is assigned an appearance number of 1. The single appearance UA should not present this call to the user. Other user agents should have no problems presenting this call distinctly from the first call.

3. The first call clears, releasing appearance number "0". The single appearance UA should now be indicating no calls since it is unable to manage calls other than on the first appearance. Both shared appearance UAs should clearly show that appearance number 0 is now free, but that there is still a call on appearance number 1.

4. A third call arrives, and is assigned the appearance number of 0. All UAs should be able to render the arrival of this new call to the user. Multiple appearnce UAs should continue to indicate the presence of the second call, and should also ensure that the presentation order is related to the appearance number and not the order of call arrival.

7.2. Call State Rendering

TOC

UAs that implement the SA feature typically have a user interface that provides the state of other appearances in the group. As dialog state NOTIFYs from the Appearance Agent are processed, this information can be rendered. Even the simplest user interface typically has three states: idle, active, and hold. The idle state, usually indicated by lamp off, is indicated for an appearance when the appearance number is not associated with any dialogs, as reported by the Appearance Agent. The active state, usually indicated by a lamp on, is indicated by an appearance number being associated with at least one dialog, as reported by the Appearance Agent. The hold state, often indicated by a blinking lamp, means the call state from the perspective of the UA in the SA group is hold. This can be determined by the presence of the "sip+rendering=no" feature tag [RFC3840] (Rosenberg, J., Schulzrinne, H., and P. Kyzivat, “Indicating User Agent Capabilities in the Session Initiation Protocol (SIP),” August 2004.) with the local target URI. Note that the hold state of the remote target URI is not relevant to this display. For joined dialogs, the state is rendered as hold only if all local target URIs are indicated with the "sip+rendering=no" feature tag.

---

8. Interop with non-SA UAs

TOC

It is desirable to allow a basic UA that does not directly support SA to be part of a SA group. To support this the Proxy must collaborate with the Appearance Agent. This is not required in the basic SA architecture, consequently SA interop with non-SA UAs will not be available in all SA deployments.
First, a UA which does not support dialog events or the SA feature will be discussed. Then, a UA which does support dialog events but not the SA feature will be discussed.

8.1. Appearance Assignment

TOC

A UA that has no knowledge of appearances must have appearance numbers assigned by the Appearance Agent for both incoming and outgoing calls. If the non-SA UA does not support Join or Replaces, all dialogs could be marked "exclusive" to indicate that these options are not available. OPEN ISSUE: How can an Appearance Agent know that a given UA does not support the feature? Do we need a SIP option tag for this purpose? Do we need to be able to correlate a registration with a subscription?

8.2. Appearance Release

TOC

In all cases the Appearance Agent must be aware of dialog lifetime to release appearances back into the group.

It is also desirable that any dialog state changes (such as hold, etc) be made available to other UAs in the group through the Dialog Event Package. If the Appearance Agent includes a proxy which Record-Routes for dialogs from the non-SA aware UA, the Appearance Agent will know about the state of dialogs including hold, etc. This information could be determined from inspection of INVITE and re-INVITE messages and added to the dialog information conveyed to other UAs.

8.3. UAs Supporting Dialog Events but Not SA

TOC

Interoperability with UAs which support dialog events but not the SA feature is more straightforward. As before, all appearance number assignment must be done by the Appearance Agent. This type of UA will be detected by the Appearance Agent by the absence of the ma event parameter in SUBSCRIBE or PUBLISH messages. The Appearance Agent can include appearance information in NOTIFYs - this UA will simply ignore this extra information. This type of UA will ignore appearance number limitations and may attempt to Join or Replace dialogs marked exclusive. As a result, the Proxy or UAs may need to reject such requests.

The need for close cooperation between the Proxy and the Appearance Agent is not needed as the Appearance Agent will learn about all dialogs from the UA itself.

---

9. Provisioning Considerations

TOC

TBD.

---

10. Example Message Flows

TOC

The next section shows call flow and message examples. The flows and descriptions are non-normative.

EDITOR'S NOTE: These flows need to be gone over closely to make sure they reflect the latest protocol design.

10.1. Registration and Subscription

TOC

Bob and Alice are in an appearance group identified by Alice's AOR. Bob REGISTERs using contact sip:bob@ua2.example.com. Furthermore, Bob

REGISTERs his primary address with contact sip: bob@ua2.example.com.
Alice REGISTERs with contact sip:alice@ua1.example.com.
The Appearance Agent subscribes to dialog states of the SA AOR (i.e., sip:alice@example.com) at the User Agents for Alice and Bob. Message exchange between the Registrar, Appearance Agent, Alice, and Bob are shown below. The call flow examples below do not show challenging of subscriptions or notifications. It should be noted that for security purposes, all subscriptions must be authorized before the same is accepted.

```
Registrar     Appearance Agent          Alice
|                    |                    |
|                    |                    |
|<--------------------------- REGISTER F1<|
|                    |                    |
|>F2 200 OK ----------------------------->|
|                    |                    |
|                    |<----- SUBSCRIBE F3<|
|                    |                    |
|                    |>F4 202 Accepted -->|
|                    |                    |
|                    |>F5 NOTIFY -------->|
|                    |                    |
|                    |<-------- OK 200 F6<|
|                    |                    |
```

Figure 2.

F1-F2: Alice registers AOR with contact: <sip:alice@ua1.example.com>

F1 Alice ----> Registrar

```
REGISTER sip:registrar.example.com SIP/2.0
Via: SIP/2.0/UDP ua1.example.com;branch=z9hG4bK527b54da8ACC7B09
From: <sip:alice@example.com>;tag=CDF9A668-909E2BDD
To: <sip:alice@example.com>
CSeq: 2 REGISTER
Call-ID: d3281184-518783de-cc23d6bb@ua1.example.com
Contact: <sip:alice@ua1.example.com>
User-Agent: ABC-UA/1.2.3
Max-Forwards: 70
Expires: 3600
Content-Length: 0
```

F2 Registrar ----> Alice

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP ua1.example.com;branch=z9hG4bKfbf176ef7F1D5FA2
CSeq: 2 REGISTER
Call-ID: d3281184-518783de-cc23d6bb@ua1.example.com
From: <sip:alice@example.com>;tag=CDF9A668-909E2BDD
To: <sip:alice@example.com>;tag=1664573879820199
Contact:  <sip:alice@ua1.example.com>
Expires:  3600
Content-Length: 0
```

F3 to F6: Alice also subscribes to the events associated with the Appearance AOR. Appearance Agent also notifies Alice of the status.

F3 Alice ----> Appearance Agent

```
SUBSCRIBE sip:sa@stateagent.example.com SIP/2.0
Via: SIP/2.0/UDP ua1.example.com;branch=z9hG4bKf10fac97E7A76D6A
From: <sip:alice@example.com>;tag=925A3CAD-CEBB276E
To: <sip:sa@stateagent.example.com>
CSeq: 1 SUBSCRIBE
Call-ID: ef4704d9-bb68aa0b-474c9d94@ua1.example.com
Contact: <sip:alice@ua1.example.com>
Event: dialog
User-Agent: ABC-UA/1.2.3
Accept: application/dialog-info+xml
Max-Forwards: 70
Expires: 3700
Content-Length: 0
```

F4 Appearance Agent ----> Alice

```
SIP/2.0 202 Accepted
Via: SIP/2.0/UDP ua1.example.com;branch=z9hG4bKf10fac97E7A76D6A
CSeq: 1 SUBSCRIBE
Call-ID: ef4704d9-bb68aa0b-474c9d94@ua1.example.com
From: <sip:alice@example.com>;tag=925A3CAD-CEBB276E
To: <sip:sa@stateagent.example.com>;tag=1636248422222257
Allow-Events: dialog
Expires: 3700
Contact: <sip:sa@stateagent.example.com>
Content-Length: 0
```

F5 Appearance Agent ----> Alice

```
NOTIFY sip:alice@ua1.example.com SIP/2.0
From: <sip:sa@stateagent.example.com>;tag=1636248422222257
To: <sip:alice@example.com>;tag=925A3CAD-CEBB276E
Call-ID: ef4704d9-bb68aa0b-474c9d94@ua1.example.com
CSeq: 2 NOTIFY
Via: SIP/2.0/UDP stateagent.example.com;branch=z9hG4bK1846984327225734
Max-Forwards: 70
Content-Type: application/dialog-info+xml
Event: dialog
Subscription-State: active
Contact: <sip:sa@stateagent.example.com>
Content-Length: ...
```

```
<?xml version="1.0"?>
<dialog-info xmlns="urn:ietf:params:xml:ns:dialog-info"
             version="40"
             state="full"
             entity="sip:alice@example.com">
</dialog-info>


F6 Alice ----> Appearance Agent

SIP/2.0 200 OK
Via: SIP/2.0/UDP stateagent.example.com;branch=z9hG4bK1846984327225734
From: <sip:sa@stateagent.example.com>;tag=1636248422222257
To: <sip:alice@example.com>;tag=925A3CAD-CEBB276E
CSeq: 2 NOTIFY
Call-ID: ef4704d9-bb68aa0b-474c9d94@ua1.example.com
Contact: <sip:alice@ua1.example.com>
Event: dialog
User-Agent: ABC-UA/1.2.3
Content-Length: 0
```

10.2. Appearance Selection for Outgoing Call

TOC

In this scenario, Bob's UA sends out a dialog event NOTIFY with state (trying) selecting an appearance number. After receiving the NOTIFY from the Appearance Agent confirming the appearance number, Bob's UA sends the INVITE to Carol and establishes a session. For brevity, details of some of the messages are not included in the message flows.

```
Carol        Proxy           Alice     Appearance Agent         Bob
|              |               |              |                  |
|              |               |              |<----- PUBLISH F1<|
|              |               |              |                  |
|              |               |              |>F2 200 OK ------>|
|              |               |              |                  |
|              |               |<-- NOTIFY F3<|                  |
|              |               |              |                  |
|              |               |>F4 200 OK -->|                  |
|              |               |              |------- NOTIFY F5>|
|              |               |              |                  |
|              |               |              |<F6 200 OK ------<|
|              |               |              |                  |
|              |               |              |                  |
|              |<------------------------------------- INVITE F11<|
|              |               |              |                  |
|<- INVITE F12<|               |              |                  |
|              |               |              |                  |
|>F13 180  --->|               |              |                  |
|              |>F14 180 Ringing ------------------------------->|
|>F15 200 OK ->|               |              |                  |
|              |>F16 200 OK ------------------------------------>|
|              |               |              |                  |
|<------------------------------------------------------ ACK F17<|
|              |               |              |                  |
|<================= Both way RTP established ===================>|
|              |               |              |                  |
|              |               |              |<---- PUBLISH F18<|
|              |               |              |                  |
|              |               |              |>F19 200 OK ----->|
|              |               |<- NOTIFY F20<|                  |
|              |               |              |                  |
|              |               |>F21 200 OK ->|                  |
|              |               |              |------ NOTIFY F22>|
|              |               |              |                  |
|              |               |              |<F23 200 OK -----<|
|              |               |              |                  |
```

Figure 3.

F1 to F4: Bob uses the BLA appearance of Alice on his UA to place an outgoing call (e.g

</t>

<figure><artwork><![CDATA[

F1 Bob ----> Appearance Agent

```
PUBLISH sip:appearance-agent@example.com SIP/2.0
```

```
Via: SIP/2.0/UDP ua2.example.com;branch=z9hG4bK61314d6446383E79
From: <sip:bob@example.com>;tag=44150CC6-A7B7919D
To: <sip:appearance-agent@example.com>;tag=428765950880801
CSeq: 7 PUBLISH
Call-ID: 144-1289338424@example.com
Contact: <sip:bob@ua2.example.com>
Event: dialog
User-Agent: XYZ-UA/4.5.6
Max-Forwards: 70
Content-Type: application/dialog-info+xml
Content-Length: ...

<?xml version="1.0"?>
<dialog-info xmlns="urn:ietf:params:xml:ns:dialog-info"
             xmlns:sa="urn:ietf:params:xml:ns:sa-dialog-info"
             version="6"
             state="partial"
             entity="sip:alice@example.com">
    <dialog id="id3d4f9c83" direction="initiator">
        <sa:appearance>0</appearance>
        <sa:exclusive>false</exclusive>
        <state>trying</state>
        <local>
            <target uri="sip:bob@example.com">
            </target>
        </local>
    </dialog>
</dialog-info>
```

F2 Appearance Agent ----> Bob

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP ua2.example.com;branch=z9hG4bK61314d6446383E79
CSeq: 7 PUBLISH
Call-ID: 144-1289338424@example.com
From: <sip:bob@example.com>;tag=44150CC6-A7B7919D
To: <sip:appearance-agent@example.com>;tag=428765950880801
Allow-Events: dialog
Contact: <sip:appearance-agent@stateagent.example.com>
Content-Length: 0
```

F3 Appearance Agent ----> Alice

```
NOTIFY sip:alice@ua1.example.com SIP/2.0
From: <sip:alice@example.com>;tag=497585728578386
To: <sip:alice@example.com>;tag=633618CF-B9C2EDA4
Call-ID: a7d559db-d6d7dcad-311c9e3a@ua1.example.com
```

```
CSeq: 7 NOTIFY
Via: SIP/2.0/UDP stateagent.example.com;branch=z9hG4bK1711759878512309
Max-Forwards: 70
Content-Type: application/dialog-info+xml
Event: dialog
Subscription-State: active
Contact: <sip:sa@stateagent.example.com>
Content-Length: ...

<?xml version="1.0"?>
<dialog-info xmlns="urn:ietf:params:xml:ns:dialog-info"
             xmlns:sa="urn:ietf:params:xml:ns:sa-dialog-info"
             version="27"
             state="partial"
             entity="sip:alice@example.com">
    <dialog id="fa02538339df3ce597f9e3e3699e28fc" direction="initiator">
            <sa:appearance>0</appearance>
            <sa:exclusive>false</exclusive>
               <state>trying</state>
                <local>
                    <target uri="sip:bob@example.com">
                    </target>
                   </local>
        </dialog>
</dialog-info>
```

F4 Alice ----> Appearance Agent

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP stateagent.example.com;branch=z9hG4bK1711759878512309
From: <sip:alice@example.com>;tag=497585728578386
To: <sip:alice@example.com>;tag=633618CF-B9C2EDA4
CSeq: 7 NOTIFY
Call-ID: a7d559db-d6d7dcad-311c9e3a@ua1.example.com
Contact: <sip:alice@ua1.example.com>
Event: dialog
User-Agent: ABC-UA/1.2.3
Content-Length: 0
```

F5 and F6: The Appearance Agent sends a NOTIFY to Bob confirming appearance number.

F11 to F17: Bob places a call to Carol by sending the INVITE request towards the Proxy. The INVITE (see F5 message below) includes a P-Preferred-Identity header <xref target="RFC3325" /> to designate the identity to be used as the calling party for this call (i.e., Alice instead of Bob).

F11 Bob ----> Proxy

```
INVITE sip:carol@example.com SIP/2.0
```

```
Via: SIP/2.0/UDP ua2.example.com;branch=z9hG4bK98c87c52123A08BF
From: <sip:bob@example.com>;tag=15A3DE7C-9283203B
To: <sip:carol@example.com>
CSeq: 1 INVITE
Call-ID: f3b3cbd0-a2c5775e-5df9f8d5@ua2.example.com
Contact: <sip:bob@ua2.example.com>
User-Agent: XYZ-UA/4.5.6
P-Preferred-Identity: <sip:alice@example.com>
Max-Forwards: 70
Content-Type: application/sdp
Content-Length: 223

v=0
o=- 1102980499 1102980499 IN IP4 ua2.example.com
s=IP SIP UA
c=IN IP4 ua2.example.com
t=0 0
a=sendrecv
m=audio 2236 RTP/AVP 0 8 101
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:101 telephone-event/8000
```

F18 to F21: Bob notifies the Appearance Agent of the status of the dialog (i.e., confirmed). Appearance Agent notifies Alice of the same.

F18 Bob ----> Appearance Agent

```
NOTIFY sip:sa@stateagent.example.com SIP/2.0
Via: SIP/2.0/UDP ua2.example.com;branch=z9hG4bKa39d3f69D4E20602
From: <sip:bob@example.com>;tag=44150CC6-A7B7919D
To: <sip:alice@example.com>;tag=428765950880801
CSeq: 9 NOTIFY
Call-ID: 144-1289338424@example.com
Contact: <sip:bob@ua2.example.com>
Event: dialog
User-Agent: XYZ-UA/4.5.6
Subscription-State: active;expires=3342
Max-Forwards: 70
Content-Type: application/dialog-info+xml
Content-Length: ...

<?xml version="1.0"?>
<dialog-info xmlns="urn:ietf:params:xml:ns:dialog-info"
             xmlns:sa="urn:ietf:params:xml:ns:sa-dialog-info"
             version="8"
             state="partial"
             entity="sip:alice@example.com">
    <dialog id="id3d4f9c83"
            call-id="f3b3cbd0-a2c5775e-5df9f8d5@ua2.example.com"
            local-tag="15A3DE7C-9283203B"
            remote-tag="65a98f7c-1dd2-11b2-88c6-b03162323164+65a98f7c"
            direction="initiator">
          <state>confirmed</state>
          <sa:appearance>0</appearance>
          <sa:exclusive>false</exclusive>
          <local>
            <target uri="sip:bob@example.com">
               <param pname="+sip.rendering" pval="yes"/>
            </target>
          </local>
          <remote>
            <identity>sip:carol@example.com</identity>
              <target uri="sip:carol@example.com;user=phone" />
          </remote>
    </dialog>
</dialog-info>
```

10.3. Taking an Appearance

TOC

In this scenario, Bob has an established dialog with Carol. Bob then places Carol on hold. Alice subsequently picks up the held call and has a established session with Carol. Finally, Carol hangs up. The details of the notifications amongst the user agents and the Appearance Agent in updating the status of the BLA group members are shown below. For brevity, details of some of the messages are not included in the message flows.

```
Carol        Proxy           Alice     Appearance Agent         Bob
|              |               |              |                  |
|              |               |              |                  |
|<================= Both way RTP established ===================>|
|              |               |              |                  |
|              |               |              |                  |
|              |<------------------------------(hold) INVITE F16<|
|<- INVITE F17<|               |              |                  |
|              |               |              |                  |
|>F18 200 OK ->|               |              |                  |
|              |>F19 200 OK ------------------------------------>|
|              |               |              |                  |
|<------------------------------------------------------ ACK F20<|
|              |               |              |                  |
|              |               |              |<----- NOTIFY F21<|
|              |               |              |                  |
|              |               |              |>F22 200 OK ----->|
|              |               |<- NOTIFY F23<|                  |
|              |               |              |                  |
|              |               |>F24 200 OK ->|                  |
|              |<-- INVITE F25<|              |                  |
|<- INVITE F26<|(w/ Replaces)  |              |                  |
|( w/ Replaces)|               |              |                  |
|>F27 200 OK ->|               |              |                  |
|              |>F28 200 OK -->|              |                  |
|              |               |              |                  |
|<-------------------- ACK F29<|              |                  |
|              |               |              |                  |
|<= Both way RTP established =>|              |                  |
|              |               |              |                  |
|>F30 BYE ---->|               |              |                  |
|              |>F31 BYE --------------------------------------->|
|              |               |              |                  |
|              |<------------------------------------ OK 200 F32<|
|<- 200 OK F33<|               |              |                  |
|              |               |              |                  |
|              |               |              |<----- NOTIFY F34<|
|              |               |              |                  |
|              |               |              |>F35 200 OK ----->|
|              |               |<- NOTIFY F36<|                  |
|              |               |              |                  |
|              |               |>F37 200 OK ->|                  |
|              |               |              |                  |
|              |               |>F38 NOTIFY ->|                  |
|              |               |              |                  |
|              |               |<- 200 OK F39<|                  |
|              |               |              |>F40 NOTIFY ----->|
|              |               |              |                  |
|              |               |              |<----- 200 OK F41<|
|>F42 BYE ---->|               |              |                  |
|              |>F43 BYE ----->|              |                  |
|              |               |              |                  |
|              |<-- 200 OK F44<|              |                  |
|<--200 OK F45<|               |              |                  |
|              |               |>F46 NOTIFY ->|                  |
|              |               |              |                  |
|              |               |<- 200 OK F47<|                  |
|              |               |              |>F48 NOTIFY ----->|
|              |               |              |                  |
|              |               |              |<----- OK 200 F49<|
```

Figure 4.

F16 to F20: Bob places Carol on hold.

F22 to F24: Bob notifies Appearance Agent of the status of the dialog to indicate the held state. It indicates this by setting the sip.rendering parameter in the NOTIFY payload to (no). Appearance Agent notifies Alice of the same.

F22 Bob ----> Appearance Agent

```
NOTIFY sip:sa@stateagent.example.com SIP/2.0
Via: SIP/2.0/UDP ua2.example.com;branch=z9hG4bK6c78a6c5CA00520E
From: <sip:bob@example.com>;tag=44150CC6-A7B7919D
To: <sip:alice@example.com>;tag=428765950880801
CSeq: 10 NOTIFY
Call-ID: 144-1289338424@example.com
Contact: <sip:bob@ua2.example.com>
Event: dialog
User-Agent: XYZ-UA/4.5.6
Subscription-State: active;expires=3338
Max-Forwards: 70
Content-Type: application/dialog-info+xml
Content-Length: ...

<?xml version="1.0"?>
<dialog-info xmlns="urn:ietf:params:xml:ns:dialog-info"
             xmlns:sa="urn:ietf:params:xml:ns:sa-dialog-info"
             version="9"
             state="partial"
             entity="sip:alice@example.com">
    <dialog id="id3d4f9c83"
       call-id="f3b3cbd0-a2c5775e-5df9f8d5@ua2.example.com"
       local-tag="15A3DE7C-9283203B"
       remote-tag="65a98f7c-1dd2-11b2-88c6-b03162323164+65a98f7c"
       direction="initiator">
       <state>confirmed</state>
       <sa:appearance>0</appearance>
       <sa:exclusive>false</exclusive>
       <local>
         <target uri="sip:bob@example.com">
             <param pname="+sip.rendering" pval="no"/>
         </target>
       </local>
       <remote>
         <identity>sip:carol@example.com</identity>
           <target uri="sip:carol@example.com" />
        </remote>
    </dialog>
</dialog-info>
```

F26 to F34 : Alice picks up the held call by sending an INVITE with Replaces: header (F26). Session is established between Alice and Carol. The dialog between Carol and Bob is terminated.

F26 Alice ----> Proxy

```
INVITE sip:carol@example.com SIP/2.0
Via: SIP/2.0/UDP ua1.example.com;branch=z9hG4bK4ea695b5B376A60C
From: <sip:alice@example.com>;tag=8C4183CB-BCEAB710
To: <sip:carol@example.com:5075>
CSeq: 1 INVITE
Call-ID: 3d57cd17-47deb849-dca8b6c6@ua1.example.com
Contact: <sip:alice@ua1.example.com>
User-Agent: ABC-UA/1.2.3
P-Preferred-Identity: <sip:alice@example.com>
<all-one-line>
Replaces: f3b3cbd0-a2c5775e-5df9f8d5@ua2.example.com;to-tag=65a98f7c
-1dd2-11b2-88c6-b03162323164+65a98f7c;from-tag=15A3DE7C-9283203B
</all-one-line>
Max-Forwards: 70
Content-Type: application/sdp
Content-Length: 223

v=0
o=- 1102980497 1102980497 IN IP4 ua1.example.com
s=IP SIP UA
c=IN IP4 ua1.example.com
t=0 0
a=sendrecv
m=audio 2238 RTP/AVP 0 8 101
```

```
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:101 telephone-event/8000
```

F34 to F41: Bob notifies the Appearance Agent of the termination of dialog at his UA. Alice notifies the Appearance Agent of the confirmed state of the dialog at her UA.

F34 Bob ----> Appearance Agent

```
NOTIFY sip:sa@stateagent.example.com SIP/2.0
Via: SIP/2.0/UDP ua2.example.com;branch=z9hG4bKa5d6cf61F5FBC05A
From: <sip:bob@example.com>;tag=44150CC6-A7B7919D
To: "State_Agent" <sip:alice@example.com>;tag=428765950880801
CSeq: 11 NOTIFY
Call-ID: 144-1289338424@example.com
Contact: <sip:bob@ua2.example.com>
Event: dialog
User-Agent: XYZ-UA/4.5.6
Subscription-State: active;expires=3334
Max-Forwards: 70
Content-Type: application/dialog-info+xml
Content-Length: ...

<?xml version="1.0"?>
<dialog-info xmlns="urn:ietf:params:xml:ns:dialog-info"
             xmlns:sa="urn:ietf:params:xml:ns:sa-dialog-info"
             version="10"
             state="partial"
             entity="sip:alice@example.com:5060">
   <dialog id="id3d4f9c83"
         call-id="f3b3cbd0-a2c5775e-5df9f8d5@ua2.example.com"
         local-tag="15A3DE7C-9283203B"
         remote-tag="65a98f7c-1dd2-11b2-88c6-b03162323164+65a98f7c"
         direction="initiator">
         <sa:appearance>0</appearance>
         <sa:exclusive>false</exclusive>
         <state>terminated</state>
         <local>
           <target uri="sip:bob@example.comsip:bob@example.com">
           </target>
         </local>
         <remote>
           <identity>sip:carol@example.com</identity>
           <target uri="sip:carol@example.com" />
         </remote>
    </dialog>
</dialog-info>
```

F38 Alice ----> Appearance Agent

```
NOTIFY sip:sa@stateagent.example.com SIP/2.0
Via: SIP/2.0/UDP ua1.example.com;branch=z9hG4bK93f44af3518A1572
From: <sip:alice@example.com>;tag=5861255C-14C04045
To: "State_Agent" <sip:alice@example.com>;tag=920163082722420
CSeq: 10 NOTIFY
Call-ID: 143-1840952798@example.com
Contact: <sip:alice@ua1.example.com>
Event: dialog
User-Agent: ABC-UA/1.2.3
Subscription-State: active;expires=3315
Max-Forwards: 70
Content-Type: application/dialog-info+xml
Content-Length: ...

<?xml version="1.0"?>
<dialog-info xmlns="urn:ietf:params:xml:ns:dialog-info"
            xmlns:sa="urn:ietf:params:xml:ns:sa-dialog-info"
            version="9"
             state="partial"
             entity="sip:alice@example.com:5060">
    <dialog id="id402f024e"
        call-id="3d57cd17-47deb849-dca8b6c6@ua1.example.com"
        local-tag="8C4183CB-BCEAB710"
        remote-tag="65a98f7c-1dd2-11b2-88c6-b03162323164+65a98f7c"
        direction="initiator">
        <state>confirmed</state>
        <sa:appearance>0</appearance>
        <sa:exclusive>false</exclusive>
        <local>
          <target uri="sip:alice@example.com">
             <param pname="+sip.rendering" pval="yes"/>
          </target>
        </local>
        <remote>
           <identity>sip:carol@example.com</identity>
           <target uri="sip:carol@example.com" />
        </remote>
    </dialog>
</dialog-info>
```

F42 to F59: Carol terminates the dialog with Alice. Alice notifies the Appearance Agent of the dialog state (terminated). The Appearance Agent notifies Bob of the same.

```
F46 Alice ----> Appearance Agent

NOTIFY sip:sa@stateagent.example.com SIP/2.0
Via: SIP/2.0/UDP ua1.example.com;branch=z9hG4bKa46c2f85F29F839C
From: <sip:alice@example.com>;tag=5861255C-14C04045
To: "State_Agent" <sip:alice@example.com>;tag=920163082722420
CSeq: 11 NOTIFY
Call-ID: 143-1840952798@example.com
Contact: <sip:alice@ua1.example.com>
Event: dialog
User-Agent: ABC-UA/1.2.3
Subscription-State: active;expires=3311
Max-Forwards: 70
Content-Type: application/dialog-info+xml
Content-Length: ...

<?xml version="1.0"?>
<dialog-info xmlns="urn:ietf:params:xml:ns:dialog-info"
             xmlns:sa="urn:ietf:params:xml:ns:sa-dialog-info"
             version="10"
             state="partial"
             entity="sip:alice@example.com">
   <dialog id="id402f024e"
      call-id="3d57cd17-47deb849-dca8b6c6@ua1.example.com"
      local-tag="8C4183CB-BCEAB710"
      remote-tag="65a98f7c-1dd2-11b2-88c6-b03162323164+65a98f7c"
      direction="initiator">
      <state>terminated</state>
      <sa:appearance>0</appearance>
      <sa:exclusive>false</exclusive>
      <local>
        <target uri="sip:alice@example.com">
        </target>
      </local>
      <remote>
        <identity>sip:carol@example.com</identity>
        <target uri="sip:carol@example.com" />
      </remote>
   </dialog>
</dialog-info>
```

10.4. Appearance Selection for Incoming Call

TOC

In the call flow below Bob and Alice are in an appearance group identified by Alice's AOR. Carol places a call to Alice. Both Alice and

Bob's devices are alerted of the incoming call. Bob answers the call. He then places Carol on hold. Alice picks up the held call and has a established session with Carol. Finally, Carol terminates the session. All NOTIFY messages in the call flow below carry dialog events and only dialog states are mentioned for simplicity. For brevity, the details of some messages are not shown below.

```
           Forking     Appearance
Carol      Proxy         Agent         Alice      Bob
|            |             |             |         |
|>F1 INVITE >|             |             |         |
|            |< - - - - - >|             |         |
|            |             |>F2 NOTIFY ----------->|
|            |             |             |         |
|            |             |<F3 200 OK -----------<|
|            |             |             |         |
|            |             |>F4 NOTIFY ->|         |
|            |             |             |         |
|            |             |<-200 OK F5-<|         |
|            |             |             |         |
|            |>F6 INVITE ------------------------->|
|            |             |             |         |
|            |>F7 INVITE --------------->|         |
|            |             |             |         |
|<- 100 F8 -<|             |             |         |
|            |             |             |         |
|            |<-------------------- Ringing 180 F9<|
|< 180 F10 -<|             |             |         |
|            |<--------- 180 Ringing F11<|         |
|< 180 F12 -<|             |             |         |
|            |<------------------------ 200 OK F13<|
|< 200 F14 -<|             |             |         |
|            |             |             |         |
|            |>F14 CANCEL -------------->|         |
|            |             |             |         |
|            |<-------------- 200 OK F15<|         |
|            |             |             |         |
|            |<Request Cancelled 487 F16<|         |
|            |             |             |         |
|            |>F17 ACK ----------------->|         |
|>F18 ACK -->|             |             |         |
|            |>F19 ACK --------------------------->|
|            |             |             |         |
|<=============Both way RTP established===========>|
|            |             |             |         |
|            |             |<---------- NOTIFY F20<|
|            |             |             |         |
|            |             |>F21 200 OK ---------->|
|            |             |             |         |
|            |             |>F22 NOTIFY >|         |
|            |             |             |         |
|            |             |<- 200 F22 -<|         |
|            |             |             |         |
|            |             |-------- SUBSCRIBE F23>|
|            |             |                       |
|            |             |<F24 200 OK ----------<|
|            |             |                       |
|            |             |<---------- NOTIFY F25<|
|            |             |                       |
|            |             |>F26 200 OK ---------->|
|            |             |                       |
```

Figure 5.

F1 to F16: An incoming call from Carol to Alice is forked to Bob and Alice. Both Alice and Bob indicate an incoming call (e.g., ringing) from Carol. Bob answers the call and two-way media is established between Carol and Bob.

F2 Proxy ----> Bob

```
INVITE sip:alice@ua3.example.com SIP/2.0
Via: SIP/2.0/UDP ua3.example.com;branch=z9hG4bK4324ea695b5B376A
Via: SIP/2.0/UDP proxy.example.com;branch=z9hG4bK38432ji
From: <sip:carol@example.com>;tag=94183CB-BCEAB7
To: <sip:alice@example.com>
CSeq: 106 INVITE
Call-ID: 47deb849-dca8b6c6-3d342
Contact: <sip:carol@ua3.example.com>
Max-Forwards: 69
Alert-Info: <file://ring.pcm>;alert=normal;appearance=0
Content-Type: application/sdp
Content-Length: 223

v=0
o=- 1102980499 1102980499 IN IP4 ua3.example.com
s=
c=IN IP4 ua3.example.com
t=0 0
a=sendrecv
m=audio 2238 RTP/AVP 0 8 101
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:101 telephone-event/8000
```

F3 Proxy ----> Alice

```
INVITE sip:alice@ua1.example.com SIP/2.0
Via: SIP/2.0/UDP ua3.example.com;branch=z9hG4bK4324ea695b5B376A
Via: SIP/2.0/UDP proxy.example.com;branch=z9hG4bK348281
From: <sip:carol@example.com>;tag=94183CB-BCEAB7
To: <sip:alice@example.com>
CSeq: 106 INVITE
Call-ID: 47deb849-dca8b6c6-3d342
```

```
Contact: <sip:carol@ua3.example.com>
Max-Forwards: 69
Alert-Info: <file://ring.pcm>;alert=normal;appearance=0
Content-Type: application/sdp
Content-Length: 223

v=0
o=- 1102980499 1102980499 IN IP4 ua3.example.com
s=
c=IN IP4 ua3.example.com
t=0 0
a=sendrecv
m=audio 2238 RTP/AVP 0 8 101
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:101 telephone-event/8000
```

F17 - F20: Bob notifies the Appearance Agent with dialog state payload indicating the dialog in confirmed state. Appearance Agent notifies Alice of the status of the dialog at Bob.

F17 Bob ----> Appearance Agent

```
NOTIFY sip:sa@stateagent.example.com SIP/2.0
Via: SIP/2.0/UDP ua2.example.com;branch=z9hG4bK58a0dd68C2D63263
From: <sip:bob@example.com>;tag=558C18F7-DB9DF7BC
To: <sip:alice@example.com>;tag=1894685100249086
CSeq: 14 NOTIFY
Call-ID: 77-505889516@example.com
Contact: <sip:bob@ua2.example.com>
Event: dialog
User-Agent: XYZ-UA/4.5.6
Subscription-State: active;expires=3427
Max-Forwards: 70
Content-Type: application/dialog-info+xml
Content-Length: ...

<?xml version="1.0"?>
<dialog-info xmlns="urn:ietf:params:xml:ns:dialog-info"
       xmlns:sa="urn:ietf:params:xml:ns:sa-dialog-info"
       version="13"
       state="partial"
       entity="sip:alice@example.com">
   <dialog id="ida0f8dc17"
        call-id="14-1541707345@example.com"
        local-tag="44BAD75D-E3128D42"
        remote-tag="d3b06488-1dd1-11b2-88c5-b03162323164+d3e48f4c"
        direction="recipient">

      <sa:appearance>0</appearance>
      <sa:exclusive>false</exclusive>
      <state>confirmed</state>
      <local>
        <target uri="sip:alice@example.com">
            <param pname="+sip.rendering" pval="yes"/>
        </target>
      </local>
      <remote>
        <identity>sip:carol@ua.example.com</identity>
      </remote>
   </dialog>
</dialog-info>
```

```
F19 Appearance Agent ----> Alice

NOTIFY sip:alice@ua1.example.com SIP/2.0
From: <sip:alice@example.com>;tag=151702541050937
To: <sip:alice@example.com>;tag=18433323-C3D237CE
Call-ID: 1e361d2f-a9f51109-bafe31d4@ua1.example.com
CSeq: 12 NOTIFY
Via: SIP/2.0/UDP stateagent.example.com;branch=z9hG4bK14031499568413
Max-Forwards: 70
Content-Type: application/dialog-info+xml
Event: dialog
Subscription-State: active
Contact: <sip:sa@stateagent.example.com>
Content-Length: ...

<?xml version="1.0"?>
<dialog-info xmlns="urn:ietf:params:xml:ns:dialog-info"
             xmlns:sa="urn:ietf:params:xml:ns:sa-dialog-info"
             version="13"
             state="partial"
             entity="sip:alice@example.com">
   <dialog id="2a7294823093f5274e3fd2ec54a2d76c"
        call-id="14-1541707345@example.com"
        local-tag="44BAD75D-E3128D42"
        remote-tag="d3b06488-1dd1-11b2-88c5-b03162323164+d3e48f4c"
        direction="recipient">
      <sa:appearance>0</appearance>
      <sa:exclusive>false</exclusive>
      <state>confirmed</state>
         <local>
        <target uri="sip:alice@example.com">
           <param pname="+sip.rendering" pval="yes"/>
```

```
            </target>
          </local>
          <remote>
            <identity>sip:carol@ua.example.com</identity>
          </remote>
       </dialog>
    </dialog-info>
```

10.5. Appearance Publication

TOC

This call flow shows the use of PUBLISH between the members of the appearance group and the Appearance Agent.

```
Carol        Proxy           Alice      Appearance Agent         Bob
  |              |               |              |                  |
  |              |               |              |<----- PUBLISH F1<|
  |              |               |              |                  |
  |              |               |              |>F2 200 OK ------>|
  |              |               |              |                  |
  |              |               |<-- NOTIFY F3<|                  |
  |              |               |              |                  |
  |              |               |>F4 200 OK -->|                  |
  |              |               |              |                  |
  |              |<------------------------------------- INVITE F5<|
  |              |               |              |                  |
  |<-- INVITE F6<|               |              |                  |
  |              |               |              |                  |
  |>F7 180 Ring >|               |              |                  |
  |              |>F8 180 Ringing -------------------------------->|
  |>F9 200 OK -->|               |              |                  |
  |              |>F10 200 OK ------------------------------------>|
  |              |               |              |                  |
  |<------------------------------------------------------ ACK F11<|
  |              |               |              |                  |
  |<================= Both way RTP established ===================>|
  |              |               |              |                  |
  |              |               |              |<---- PUBLISH F12<|
  |              |               |              |                  |
  |              |               |              |>F13 200 OK ----->|
  |              |               |<- NOTIFY F14<|                  |
  |              |               |              |                  |
  |              |               |>F15 200 OK ->|                  |
  |              |               |              |                  |
  |              |<------------------------------(hold) INVITE F16<|
  |<- INVITE F17<|               |              |                  |
  |              |               |              |                  |
  |>F18 200 OK ->|               |              |                  |
  |              |>F19 200 OK ------------------------------------>|
  |              |               |              |                  |
  |<------------------------------------------------------ ACK F20<|
  |              |               |              |                  |
  |              |               |              |<---- PUBLISH F21<|
  |              |               |              |                  |
  |              |               |              |>F22 200 OK ----->|
  |              |               |<- NOTIFY F23<|                  |
  |              |               |              |                  |
  |              |               |>F24 200 OK ->|                  |
  |              |<-- INVITE F25<|              |                  |
  |<- INVITE F26<|(w/ Replaces)  |              |                  |
  |( w/ Replaces)|               |              |                  |
  |>F27 200 OK ->|               |              |                  |
  |              |>F28 200 OK -->|              |                  |
  |              |               |              |                  |
  |<-------------------- ACK F29<|              |                  |
  |              |               |              |                  |
  |<= Both way RTP established =>|              |                  |
  |              |               |              |                  |
  |>F30 BYE ---->|               |              |                  |
  |              |>F31 BYE --------------------------------------->|
  |              |               |              |                  |
  |              |<------------------------------------ OK 200 F32<|
  |<- 200 OK F33<|               |              |                  |
  |              |               |              |                  |
  |              |               |              |<---- PUBLISH F34<|
  |              |               |              |                  |
  |              |               |              |>F35 200 OK ----->|
  |              |               |<- NOTIFY F36<|                  |
  |              |               |              |                  |
  |              |               |>F37 200 OK ->|                  |
  |              |               |              |                  |
  |              |               |>F38 PUBLISH >|                  |
  |              |               |              |                  |
  |              |               |<- 200 OK F39<|                  |
  |              |               |              |>F40 NOTIFY ----->|
  |              |               |              |                  |
  |              |               |              |<----- 200 OK F41<|
  |>F42 BYE ---->|               |              |                  |
  |              |>F43 BYE ----->|              |                  |
  |              |               |              |                  |
  |              |<-- 200 OK F44<|              |                  |
  |<--200 OK F45<|               |              |                  |
  |              |               |>F46 PUBLISH >|                  |
  |              |               |              |                  |
  |              |               |<- 200 OK F47<|                  |
  |              |               |              |>F48 NOTIFY ----->|
  |              |               |              |                  |
  |              |               |              |<----- OK 200 F49<|
```

Figure 6.

10.6. Joining an Appearance

TOC

In this call flow, a call answered by Bob is joined by Alice or "bridged". The Join header field is used by Alice to request this bridging. If Bob did not support media mixing, Bob could obtain conferencing resources as described in [RFC4579] (Johnston, A. and O. Levin, “Session Initiation Protocol (SIP) Call Control - Conferencing for User Agents,” August 2006.).

```
Carol    Forking Proxy Appearance Agent  Alice      Bob
  |            |             |             |         |
  |>F1 INVITE >|             |             |         |
  |            |>F2 INVITE ------------------------->|
  |            |             |             |         |
  |            |>F3 INVITE --------------->|         |
  |            |             |             |         |
  |<-100Try F4<|             |             |         |
  |            |             |             |         |
  |            |<-------------------- Ringing 180 F5<|
  |<180Ring F6<|             |             |         |
  |            |<---------- Ringing 180 F7<|         |
  |<180Ring F8<|             |             |         |
  |            |<------------------------- 200 OK F9<|
  |<-200OK F10<|             |             |         |
  |            |             |             |         |
  |            |>F11 CANCEL -------------->|         |
  |            |             |             |         |
  |            |<-------------- 200 OK F12<|         |
  |            |             |             |         |
  |            |<Request Cancelled 487 F13<|         |
  |            |             |             |         |
  |            |>F14 ACK ----------------->|         |
  |>F15 ACK -->|             |             |         |
  |            |>F16 ACK ---------------------------->|
  |            |             |             |         |
  |<=============Both way RTP established===========>|
  |            |             |             |         |
  |            |             |<---------- NOTIFY F17<|
  |            |             |             |         |
  |            |             |>F18 200 OK ---------->|
  |            |             |             |         |
  |            |             |>F19 NOTIFY >|         |
  |            |             |             |         |
  |            |             |<- 200OK F20<|         |
  |            |             |             |         |
  |            |<---- INVITE (w/ Join) F21<|         |
  |            |             |             |         |
  |            |>F22 INVITE (w/Join)---------------->|
  |            |             |             |         |
  |            |<---- OK 200 Contact:Bob;isfocus F23<|
  |            |             |             |         |
  |            |>F24 200 OK Contact:Bob;isfocus----->|
  |            |             |             |         |
  |            |<----------------- ACK F25<|         |
  |            |             |             |         |
  |            |>ACK F26---------------------------->|
  |            |             |             |         |
  |            |<-----INVITE Contact:Bob;isfocus F27<|
  |<-INVITE F28|             |             |         |
  |>F29 200 -->|             |             |         |
  |            |>F30 200 OK ------------------------>|
  |            |             |             |         |
  |            |<--------------------------- ACK F31<|
  |<--- ACK F32|             |             |         |
  |            |             |             |<==RTP==>|
  |<=============Both way RTP established===========>|
```

Figure 7.

F21 Alice ----> Proxy

```
INVITE sip:bob@ua.example.com SIP/2.0
Via: SIP/2.0/UDP ua1.example.com;branch=z9hG4bKcc9d727c2C29BE31
From: <sip:alice@example.com>;tag=605AD957-1F6305C2
To: <sip:bob@ua.example.com>
CSeq: 2 INVITE
Call-ID: dc95da63-60db1abd-d5a74b48@ua1.example.com
Contact: <sip:alice@ua1.example.com>
User-Agent: ABC-UA/1.2.3
P-Preferred-Identity: <sip:alice@example.com>
<all-one-line>
Join: 14-1541707345@example.com;to-tag=d3b06488-1dd1-11b2-88c5
-b03162323164+d3e48f4c;from-tag=44BAD75D-E3128D42
</all-one-line>
Max-Forwards: 70
Content-Type: application/sdp
Content-Length: 223

v=0
o=- 1103061265 1103061265 IN IP4 ua1.example.com
s=IP SIP UA
c=IN IP4 ua1.example.com
t=0 0
a=sendrecv
m=audio 2236 RTP/AVP 0 8 101
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:101 telephone-event/8000
```

10.7. Appearance Allocation - Loss of Subscription with UA

```
        UA           Appearance Agent        UA1              UA2
        |                |                |                |
        |                | F1: NOTIFY (trying)             |
        |                |<---------------|                |
        |                | F2: 200 OK     |                |
        |                |--------------->|                |
        |                | F3: NOTIFY (trying)             |
        |                |----------------+--------------->|
        |                | F4: 200 OK     |                |
        |                |<---------------+----------------|
        | F5: INVITE     |                |                |
        |<--------------------------------|                |
        | F6: 180 Ringing|                |                |
        |-------------------------------->|                |
        |                |                |                |
        |                |               End               |
        |                |                                 |
        |                |                                 |
        |                | F7: SUBSCRIBE x 6 retries       |
        |                |--------------->                 |
        |                |                                 |
        |                | F8: NOTIFY (terminated)         |
        |                |-------------------------------->|
        |                | F9: 200 OK                      |
        |                |<--------------------------------|
        |                |                                 |
```

Figure 8.

The flow shown in this figure illustrates the failure of a user agent after it has obtained an appearance number (F1-F2). Messages used to refresh the subscription from Appearance Agent to UA1 are shown at F7. When the Appearance Agent attempts to refresh its subscription but receives no response. In this case, the Appearance Agent may apply policy and free up the appearance number as it wishes. In this case, after a delay, the Appearance Agent frees up the appearance number and sends NOTIFY messages (F8) indicating the termination of the dialog associated with the shared line.

TOC

10.8. Appearance Selection Contention Race Condition

```
        UA       Appearance Agent        UA1              UA2
        |                |                |                |
        |                | F1 NOTIFY (trying appearance=1) |
        |                |<---------------|                |
        |                | F2 NOTIFY (trying appearance=1) |
        |                |<---------------+----------------|
        |                | F3 200 OK      |                |
        |                |--------------->|                |
        |                | F4 200 OK      |                |
        |                |----------------+--------------->|
        |                | F5 NOTIFY  (trying appearance=1)|
        |                |--------------->|                |
        |                | F6 200 OK      |                |
        |                |<---------------|                |
        |                | F7 NOTIFY (trying)              |
        |                |----------------+--------------->|
        |                | F8 200 OK      |                |
        |                |<---------------+----------------|
        | F9 INVITE      |                |                |
        |<--------------------------------|                |
        |                | F10 NOTIFY (trying appearance=2)|
        |                |<---------------+----------------|
        |                | F11 200 OK     |                |
        |                |----------------+--------------->|
        |                | F12 NOTIFY (trying appearance=2)|
        |                |----------------+--------------->|
        |                | F13 200 OK      |               |
        |                |<---------------+----------------|
        | F14 INVITE     |                |                |
        |<-------------------------------------------------|
        |                |                |                |
```

Figure 9.

This figure illustrates two user agents, UA1 and UA2, attempting to select the same appearance number (i.e. seize the same line number) simultaneously. This type of race condition is often referred to in telephony as a glare condition. Appearance Agent may use any desired policy to decide which UA receives the appearance and which does not. In this case UA1 obtains the appearance number, as indicated by the NOTIFY from the Appearance Agent with the appearance number. UA2 learns that it did not obtain the appearance number since its NOTIFY does not contain the appearance number from its NOTIFY.

11. IANA Considerations

TOC

This section registers the SIP Alert-Info header field parameter "appearance" and the XML namespace extensions to the SIP Dialog Package.

11.1. SIP Event Package Parameter: sa

TOC

This specification also defines a new event parameter "sa" for the Dialog Package.

11.2. URN Sub-Namespace Registration: sa-dialog-info

TOC

This section registers a new XML namespace per the procedures in [RFC3688].

URI: urn:ietf:params:xml:ns:sa-dialog-info.

Registrant Contact: IETF BLISS working group, <bliss@ietf.org>, Alan Johnston <alan@sipstation.com>

XML:

```
BEGIN
   <?xml version="1.0"?>
   <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML Basic 1.0//EN"
             "http://www.w3.org/TR/xhtml-basic/xhtml-basic10.dtd">
   <html xmlns="http://www.w3.org/1999/xhtml">
   <head>
     <meta http-equiv="content-type"
        content="text/html;charset=iso-8859-1"/>
     <title>Shared Appearance Dialog Information Namespace</title>
   </head>
   <body>
     <h1>Namespace for Shared Appearance Dialog Information</h1>
     <h2>urn:ietf:params:xml:ns:dialog-info</h2>
     <p>See <a href="ftp://ftp.rfc-editor.org/in-notes/rfcXXXX.txt">
          RFCXXXX</a>.</p>
   </body>
   </html>
END
```

11.3. XML Schema Registration

TOC

This section registers an XML schema per the procedures in [RFC3688].

URI: urn:ietf:params:xml:schesa:sa-dialog-info.

Registrant Contact: IETF BLISS working group, <bliss@ietf.org>,
Alan Johnston <alan@sipstation.com>

The XML for this schema can be found in Section 7.

---

12. Appendix A - Incoming Appearance Assignment

TOC

To best meet REQ-9, the appearance number for an incoming INVITE should be contained in the INVITE itself.
For the dialog package parameter approach, REQ-9 could be met in two ways. When an incoming request is received, the Appearance Agent could send out a NOTIFY with state trying and include the appearance number to be used for this request. Upon receipt of this NOTIFY, the UAs could begin alerting using the appearance number selected. This approach is sub-optimal since the UAs could receive the INVITE but be unable to begin alerting if the NOTIFY from the Appearance Agent is delayed or lost
An alternative approach is to define an extension parameter for the Alert-Info header field in RFC 3261 such as:
Alert-Info: <file://ring.pcm>;alert=normal;appearance=0
This Alert-Info header would indicate to place the call on the first line appearance instance.
The determination as to what value to use in the appearance parameter can be done at the proxy that forks the incoming request to all the registered UAs. There are a variety of ways the proxy can use to determine what value it should use to populate this parameter. For example, the proxy could fetch this information by initiating a SUBSCRIBE request with Expires: 0 to the Appearance Agent for the AOR to fetch the list of lines that are in use. Alternatively, it could act like a UA that is a part of the appearance group and SUBSCRIBE to the State-Agent like any other UA. This would ensure that the active dialog information is available without having to poll on a need basis. It could keep track of the list of active calls for the appearance AOR based on how many unique INVITE requests it has forked to or received

from the appearance AOR. Another approach would be for the Proxy to first send the incoming INVITE to the Appearance Agent which would redirect to the appearance group URI and escape the proper Alert-Info header field for the Proxy to recurse and distribute to the other UAs in the group.

The Appearance Agent needs to know about all incoming requests to the AOR in order to select the appearance number. One way in which this could be done is for the Appearance Agent to register against the AOR with a higher q value. This will result in the INVITE being sent to the Appearance Agent first, then being offered to the UAs in the group.

The changes to RFC 3261 ABNF would be:

```
alert-param = LAQUOT absoluteURI RAQUOT *( SEMI (generic-param /
appearance-param) )
appearance-param = "appearance" EQUAL *DIGIT
```

---

13. Appendix B - Implementation Options Discussion

TOC

This section discusses some options on how to implement the Shared Appearances feature in SIP. This section is non-normative.

13.1. Appearance Implementation Options

TOC

This section discusses and compares two methods of implementing, conveying, and selecting appearances in SIP while meeting the requirements of Section 4. One approach involves a URI parameter and is discussed in section 5.1.1. The other approach uses a SIP dialog package extension parameter and is discussed in section 5.1.2. Both approaches assume the common elements and operations of Figure 1. In addition, this section discusses approaches for incoming appearance indication, REQ-9, and appearance contention, REQ-8. These approaches will be discussed for an example appearance group of N phones each with n line appearances. The usage of the word phone does not imply that this feature is limited to telephony devices.

13.1.1. URI parameter Approach

TOC

Some implementations of this feature utilize a URI parameter such as "line=3" on the Contact URI. Each appearance is effectively a logical UA, so each line appearance requires a separate registration. The number of line appearances needs to be provisioned on each phone. Each appearance also requires a separate dialog package subscription. Even using a State Agent for the dialog package, each phone must maintain n subscriptions to the dialog package.

This results in 2nN total subscriptions and nN registrations for this implementation.

Since Contact URI parameters will be conveyed by the dialog package, REQ-7 is met.

REQ-10 can be met by having the Appearance Agent send a SUBSCRIBE to each UA and line number to obtain the current dialog state - this will result in nN SUBSCRIBEs and NOTIFYs.

It is not obvious how to meet REQ-11 with this approach. A UA registering against the AOR but does not implement the appearance URI parameter will not include a line appearance number in Contact URIs and dialog package NOTIFYs. The Appearance Agent will have no way of indicating to the other UAs the appearance number being used by this UA, as adding a parameter to the Contact URI would cause call control operations such as Replaces and Join to fail.

REQs 12 and 13 are difficult to meet with this approach as the line appearance number will be present in the Request-URI of incoming requests and the Contact URI in INVITE and 200 OK messages. This approach will require integrity protection of all dialog creating requests and responses, and privacy mechanisms to hide the Contact URI from other UAs.

Also, this approach will require mechanisms to protect against another UA sending an INVITE directly to a group member with the line appearance number already set.

13.1.2. Dialog Package Parameter

TOC

Instead of the URI parameter approach, consider an extension parameter "appearance" to the SIP dialog package. The e.g.:

```
<?xml version="1.0"?>
<dialog-info xmlns="urn:ietf:params:xml:ns:dialog-info"
             xmlns:sa="urn:ietf:params:xml:ns:sa-dialog-info"
             version="6"
             state="partial"
             entity="sip:alice@example.com">
   <dialog id="id3d4f9c83" from-tag="3423" to-tag="a3f423j88uju1"
                                         direction="initiator">
       <sa:appearance>2</appearance>
       <sa:exclusive>false</exclusive>
       <sa:joined-dialog call-id="sdfg" from-tag="832d1" to-tag="4542454" />
       <sa:joined-dialog call-id="873287876" from-tag="433" to-tag="jwjwuf5" />
       <state>connected</state>
       <local>
           <target uri="sip:bob@pc.example.com" />
       </local>
   </dialog>
</dialog-info>
...
```

In this approach, the appearance number is never carried in a Request-URI or Contact URI. Instead, it is only present in dialog package NOTIFY and PUBLISH messages. As a result, only a single registration per AOR is required. Also, only a single dialog package subscription in each direction per AOR.

This results in 2N total subscriptions and N registrations for this approach.

If the dialog package is extended to carry the appearance number, then REQ-7 is met.

REQ-10 can be met by having the Appearance Agent send a SUBSCRIBE to each UA and line number to obtain the current dialog state - this will result in N SUBSCRIBEs and NOTIFYs.

REQ-11 can be met by this approach. Even though a UA does not provide an appearance number in dialog package NOTIFYs, the Appearance Agent can assign one and include it in NOTIFYs to the other UAs. This parameter would simply be ignored by the UAs that did not understand the parameter, and have no impact on call control operations.

REQs 12 and 13 are met because the appearance number is only conveyed in dialog package NOTIFYs. Integrity and privacy of NOTIFY bodies can be achieved using normal SIP mechanisms independent of the security mechanisms used for other requests.

The dialog-package [RFC3265] (Roach, A., “Session Initiation Protocol (SIP)-Specific Event Notification,” June 2002.) describes a mechanism whereby shared-line privacy REQ-14 can be accomplished by suppressing certain dialog information from being presented to the UAs. The reasoning behind that is if the UAs were unaware of a dialog's call-id, local-tag and remote-tag then they will be unable to create requests such as INVITE with Replaces [RFC3891] (Mahy, R., Biggs, B., and R. Dean, “The Session Initiation Protocol (SIP) "Replaces" Header,” September 2004.) and Join [RFC3911] (Mahy, R. and D. Petrie, “The Session Initiation Protocol (SIP) "Join" Header,” October 2004.) header fields to barge-in or pickup the line appearance. Below is a quote from section 3.6 of dialog-package[RFC3265] (Roach, A., “Session Initiation Protocol (SIP)-Specific Event Notification,” June 2002.) that describes this approach:

Note that many implementations of "shared-lines" have a feature that allows details of calls on a shared address-of-record to be made private. This is a completely reasonable authorization policy that could result in notifications that contain only the id attribute of the dialog element and the state element when shared-line privacy is requested, and notifications with more complete information when shared-line privacy is not requested.

There are certain fundamental drawbacks in the privacy-by-obscurity approach described in [RFC3265] (Roach, A., “Session Initiation Protocol (SIP)-Specific Event Notification,” June 2002.) . It models exclusivity as a static property of the appearance AOR. There are situations where exclusivity needs to be a dynamic property (e.g. boss does not want secretary to listen-in on a particular part of the conversation). In addition, [RFC3265] (Roach, A., “Session Initiation Protocol (SIP)-Specific Event Notification,” June 2002.) does not address how a UA can request exclusivity at the start of a session or mid-session and how that request will be granted or rejected.

Exclusivity being a dynamic property means that a UA can request it to be turned on or off in the middle of a session. When exclusivity is turned off all the UAs that share the line AOR will need to see the complete dialog information. Once they have that information it can not be taken back from them. This will not allow exclusivity to be turned on later on in the dialog lifetime. Therefore, there needs to be a centralized entity that will actually enforce exclusivity.

The approach proposed for meeting REQ-14 is to include an exclusivity parameter to the dialog package. This allows a UA to request exclusivity, by setting the exclusive parameter in notifications. This could be done prior to a call being made or answered, or during a call at any time. A UA can remove exclusivity by sending a notification at any time during a call and setting "exclusive=no". It also allows a UA to learn that a particular dialog is exclusive by the presence of this parameter in a NOTIFY. In addition, a UA can still apply policy to any INVITE Join or Replaces requests it receives, as per normal SIP call control mechanisms.

With this approach, the number of appearances is centrally managed and controlled by the Appearance Agent. For UAs with soft keys or buttons, this gives a great deal of flexibility in system management.

TOC

13.1.3. Appearance Selections Mechanisms

Regardless of how the appearance number is conveyed by UAs, there is still the issue of how appearance numbers are selected. For example, some UAs might have actual buttons and lamps, and pressing a particular button requires the UA to reserve a particular appearance number. For devices with this type of user interface, the selection must be done before the user continues with the call and dials digits or a URI. Other UAs with different user interfaces can be flexible at the time of dialing, updating the display with the appearance number at a later date. For devices which require advance appearance selection, there are three options discussed in the following sections for meeting REQ-15.

13.1.3.1. Floor Control Appearance Selection Mechanism

TOC

This approach models each appearance number as a floor (shared resource) and uses a floor control server to arbitrate exclusive access (seizure of a particular appearance number). This approach uses a standard SIP Event State Compositor (ESC), a standard Floor Control Server that uses the Appearance Agent as Moderator. The Binary Floor Control Protocol (BFCP) is used between the UAs and the Floor Control Server. A Registrar/Forking Proxy Server talks to Appearance Agent about incoming calls. The Appearance Agent acts as a Moderator for the floor control server and tells forking proxy to insert the appearance number in incoming and outgoing requests.
Appearance numbers are allocated/selected/reserved in two ways:
For incoming calls, the Forking Proxy interacts with the Appearance Agent. The Appearance Agent selects an appearance by taking a particular floor and marking it "moderator controlled". This appearance number is then included by the Forking Proxy in INVITEs using the Alert-Info parameter. When a UA answers the call, it takes the appearance number from the Alert-Info and includes it in the dialog state publication. It then requests the floor associated with the appearance number from the floor control server, which forwards the request to the Appearance Agent (moderator). The Appearance Agent correlates the floor control request with the dialog state notification with the dialog ID from the INVITE with the Alert-Info. If they match, the floor is granted. If they do not match, it means the floor request is not an answer of the call but is a random appearance selection by the UA and will be rejected.
For outgoing calls, the UA sends an INVITE and requests a particular floor from the floor control server. Depending on the User Interface requirements, the floor request can be done before or after sending the INVITE. The floor grant policy for most appearances is set to "first come first serve". Once the floor has been granted and the call

answered, the dialog state publication by the UA will include the appearance number.
When a call has ended, the UA releases the floor to the floor control server and this appearance is now available for incoming and outgoing calls.
When a UA in the group which does not support BFCP is in a call, the Appearance Agent will grant the floor associated with that appearance to that UA. When that call is over, the Appearance Agent will release the floor. Since the UA will not publish the appearance number to the ESC, the Appearance Agent will need to do that on their behalf. If the UA does publish dialog state but without the appearance number, the Appearance Agent will still need to re-publish the dialog state including the appearance number. UAs in the group will be able to recognize these two dialogs as one since they will have the same SIP dialog ID.

13.1.3.2. INVITE Appearance Selection Mechanism

TOC

This is an alternative approach that utilizes sending an INVITE to select/reserve/seize an appearance number.
A UA that does not need to select a particular appearance number (or doesn't care) would just send an INVITE as normal. The Appearance Agent would tell the proxy which appearance number was being used by inserting this information in a header field in the first non-100 provisional response sent back to the calling UA. The UA would then PUBLISH this appearance number to the Dialog Event State Compositor for the AOR which would distribute details of the dialog and the appearance number to the other UAs in the group.
If an INVITE is sent and no appearance number is available, the proxy would reject the INVITE with a suitable response code and perhaps a header field indication.
A UA that does need to select a particular appearance number would use an approach similar to overlap dialing (multi-stage dialing). An INVITE would be sent when the appearance number is requested (i.e. when the button is pressed, before dialing begins). The appearance number selected would be carried in the INVITE, in a header field or in the Request-URI, for example. The proxy would reject the INVITE with a 484 Address Incomplete response (see RFC 3578) if the appearance number is Available and start a timer. The UA could then resend the INVITE after the URI has been dialed and then PUBLISH this appearance number to the ESC. If the appearance number is not available, another response code such as 403 would be sent. The user could then select a different appearance number and resend the INVITE. If no INVITE with a matching Call-ID is received before the timer expires, the appearance seizure is cancelled and is made available for other calls.

Note that this approach does not actually require a B2BUA, but it does require a proxy that can act as a UAS and communicate with an Appearance Agent which keeps track of appearance number allocations.

13.1.3.3. PUBLISH Appearance Selection Mechanism

TOC

The approach used in previous versions of this draft is to use the PUBLISH to the event state compositor to select an appearance number. This approach requires a special event state compositor and special behavior on the part of the UA.
In the selection of an appearance for requests initiated by UAs in the group, there is the possibility of contention where more than one UA select the same appearance number.
One way to solve this and meet REQ-8 is to require UAs to send a notification (trying) to the Appearance Agent indicating the appearance number to be used for the session. The Appearance Agent would confirm the allocation of the appearance number in a NOTIFY sent to the group UAs. Should the appearance number be unavailable or otherwise not allowed, there are two options:
- The notification could be rejected with a 500 response and a Retry-After header field. The Appearance Agent would send an immediate NOTIFY indicating that the appearance is unavailable. If the NOTIFY is received before the expiration of the Retry-After time, the notification state information would become out of date and would be discarded without resending. The UA would select another appearance number and send another notification.
- The notification could be accepted but an immediate NOTIFY generated by the Appearance Agent indicating that the appearance is unavailable. The UA would then select another appearance number and PUBLISH again.
UAs would wait for a notification from the Appearance Agent before sending the INVITE.

13.2. Comparison

TOC

In comparing the URI parameter and the dialog package parameter, there are clear differences in the number of registrations and subscriptions, with the dialog package approach requiring n times fewer in both cases. The security model for the dialog package parameter approach is much cleaner, since only NOTIFY and PUBLISH requests need integrity and privacy. The security model for the URI parameter approach would likely require a B2BUA which introduces many undesirable properties.
The dialog package parameter approach has better backwards compatibility than the URI parameter approach.

In summary, the dialog package parameter approach better meets REQs 5, 10, 11, 12, and 13 while the URI parameter approach better meets REQ-9. However, the combined dialog package parameter approach and the Alert-Info parameter approach meets REQ-9.

13.2.1. Comparison of Appearance Selection Methods

TOC

All three approaches meet REQ-15 and REQ-16.

Previous versions of this draft proposed the publish/notify method of appearance selection. The advantage of this approach is that the appearance number is only carried in one place (dialog package XML documents) and the same protocol/mechanism is used to select and learn appearance numbers. The disadvantage of this approach is that a specialized event state compositor must be used, since it is aware of appearance numbers. Also, concerns have been raised about whether this approach defines new semantics for publish/notify beyond that in RFC 3265.

The floor control approach makes good reuse of existing protocols such as Binary Floor Control Protocol (BFCP) and cleanly models the state. However, while BFCP can be used in conferencing applications, it is unlikely most UAs implementing shared appearances would utilize the protocol. Also, having appearance state in two places (dialog package XML documents and floor control messages) complicates the application. Also, BFCP only runs over TCP and requires a separate offer/answer exchange to establish the connection, making operation through NATs and firewalls more difficult. The BFCP approach is also radically different from all current implementations of this feature. As a result, standardizing this approach would likely result in an increase in feature interoperability rather than a decrease.

The INVITE selection mechanism is based on overlap dialing. Overlap dialing is supported in very few SIP UAs and is regarded as a somewhat archaic leftover from the PSTN. As such, it is not regarded as a good starting point for a common feature such as shared appearances.

The PUBLISH selection mechanism reuses the SIP events extensions which already must be implemented by UAs supporting this feature. In fact, it results in no additional messages or round trips. It is also very similar to many current feature implementations today. Standardizing this approach is likely to increase overall interoperability of this feature.

The rest of this document will only discuss the PUBLISH appearance selection mechanism.

TOC

14. Acknowledgements

The following individuals were part of the SA Design team and have provided input and text to the document (in alphabetical order):
Martin Dolly, Andrew Hutton, Raj Jain, Fernando Lombardo, Derek MacDonald, Bill Mitchell, Michael Procter, Theo Zowzouvillys.
Thanks to Chris Boulton for helping with the XML schema.
Much of the material has been drawn from previous work by Mohsen Soroushnejad, Venkatesh Venkataramanan, Paul Pepper and Anil Kumar, who in turn received assistance from:
Kent Fritz, John Weald, and Sunil Veluvali of Sylantro Systems, Steve Towlson, and Michael Procter of Citel Technologies, Rob Harder and Hong Chen of Polycom Inc, John Elwell, J D Smith of Siemens Communications, Dale R. Worley of Pingtel, Graeme Dollar of Yahoo Inc.
Also thanks to Geoff Devine, Paul Kyzivat, Jerry Yin, John Elwell, Dan York, Spenser Dawkins, and Martin Dolly for their comments.

---

15. Security Considerations

TOC

Since multiple line appearance features are implemented using semantics provided by [RFC3261] (Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, “SIP: Session Initiation Protocol,” June 2002.), Event Package for Dialog State as define in , and Event Notification [RFC3265] (Roach, A., “Session Initiation Protocol (SIP)-Specific Event Notification,” June 2002.), [RFC3903] (Niemi, A., “Session Initiation Protocol (SIP) Extension for Event State Publication,” October 2004.), security considerations in these documents apply to this draft as well.

Specifically, since dialog state information and the dialog identifiers are supplied by UA's in an appearance group to other members, the same is prone to "call hijacks". For example, a rogue UA could snoop for these identifiers and send an INVITE with Replaces header containing these call details to take over the call. As such INVITES with Replaces header MUST be authenticated using the standard mechanism (like Digest or S/MIME) described in [RFC3261] (Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, “SIP: Session Initiation Protocol,” June 2002.). before it is accepted. NOTIFY message bodies that provide the dialog state information and the dialog identifiers MAY be encrypted end-to-end using the standard mechanics. All SUBSCRIBES between the UA's and the Appearance Agent MUST be authenticated.

---

16. Informative References

TOC

[RFC2119]

	Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels,” BCP 14, RFC 2119, March 1997 (TXT, HTML, XML).
[RFC3261]	Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, “SIP: Session Initiation Protocol,” RFC 3261, June 2002 (TXT).
[RFC3515]	Sparks, R., “The Session Initiation Protocol (SIP) Refer Method,” RFC 3515, April 2003 (TXT).
[RFC3265]	Roach, A., “Session Initiation Protocol (SIP)-Specific Event Notification,” RFC 3265, June 2002 (TXT).
[RFC3903]	Niemi, A., “Session Initiation Protocol (SIP) Extension for Event State Publication,” RFC 3903, October 2004 (TXT).
[RFC3891]	Mahy, R., Biggs, B., and R. Dean, “The Session Initiation Protocol (SIP) "Replaces" Header,” RFC 3891, September 2004 (TXT).
[I-D.ietf-sipping-service-examples]	Johnston, A., Sparks, R., Cunningham, C., Donovan, S., and K. Summers, “Session Initiation Protocol Service Examples,” draft-ietf-sipping-service-examples-15 (work in progress), July 2008 (TXT).
[RFC4235]	Rosenberg, J., Schulzrinne, H., and R. Mahy, “An INVITE-Initiated Dialog Event Package for the Session Initiation Protocol (SIP),” RFC 4235, November 2005 (TXT).
[RFC3680]	Rosenberg, J., “A Session Initiation Protocol (SIP) Event Package for Registrations,” RFC 3680, March 2004 (TXT).
[RFC3911]	Mahy, R. and D. Petrie, “The Session Initiation Protocol (SIP) "Join" Header,” RFC 3911, October 2004 (TXT).
[RFC3325]	Jennings, C., Peterson, J., and M. Watson, “Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks,” RFC 3325, November 2002 (TXT).
[RFC4579]	Johnston, A. and O. Levin, “Session Initiation Protocol (SIP) Call Control - Conferencing for User Agents,” BCP 119, RFC 4579, August 2006 (TXT).
[RFC3840]	Rosenberg, J., Schulzrinne, H., and P. Kyzivat, “Indicating User Agent Capabilities in the Session Initiation Protocol (SIP),” RFC 3840, August 2004 (TXT).

Authors' Addresses

[TOC]

	Alan Johnston (editor)
	Avaya
	St. Louis, MO 63124
Email:	alan@sipstation.com
	Mohsen Soroushnejad
	Sylantro Systems Corp
Email:	mohsen.soroush@sylantro.com
	Venkatesh Venkataramanan
	Sylantro Systems Corp
Email:	vvenkatar@gmail.com
	Paul Pepper
	Citel Technologies
Email:	paul.pepper@citel.com
	Anil Kumar
	Yahoo Inc.
Email:	anil@yahoo-inc.com

Full Copyright Statement

[TOC]

Intellectual Property