BLISS
Internet-Draft
Updates: 3261, 4235 (if approved)
Intended status: Standards Track
Expires: November 5, 2012

A. Johnston, Ed.
Avaya
M. Soroushnejad
V. Venkataramanan
Sylantro Systems Corp

May 4, 2012

Shared Appearances of a Session Initiation Protocol (SIP) Address of Record (AOR)
draft-ietf-bliss-shared-appearances-10

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 of an Address of Record (AOR) since SIP does not have the concept of lines. This feature is commonly offered in 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 feature allows several user agents (UAs) to share a common AOR, learn about calls placed and received by other UAs in the group, and pick up or join calls within the group. This document discusses use cases, lists requirements and defines extensions to implement this feature.

Status of this Memo

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

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

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

This Internet-Draft will expire on November 5, 2012.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the

document authors. All rights reserved.

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

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Table of Contents

1. Introduction 5
2. Conventions used in this document 6
3. Usage Scenarios 6
3.1. Executive/Assistant Arrangement 6
3.2. Call Group 7
3.3. Single Line Extension 7
3.4. Changing UAs 7
4. Requirements and Implementation 7
4.1. Requirements 7
4.2. Implementation 9
5. Normative Description 11
5.1. Elements 11
5.2. Shared Appearance Dialog Package Extensions 12
5.2.1. The <appearance> element 12
5.2.2. The <exclusive> element 12
5.2.3. The <joined-dialog> element 13
5.2.4. The <replaced-dialog> element 14
5.3. Shared Appearance User Agents 14
5.3.1. Appearance Numbers and Call Context 17
5.3.2. Appearance Numbers and Call Control 17
5.3.3. Appearance Numbers and Transfer 18
5.4. Appearance Agent 19
6. XML Schema Definition 21
7. Alert-Info Appearance Parameter Definition 23
8. User Interface Considerations 24
8.1. Appearance Number Rendering 24
8.1.1. Single Appearance UAs 24
8.1.2. Dual Appearance UAs 24
8.1.3. Shared Appearance UAs with Fixed Appearance Number . . 25
8.1.4. Shared Appearance UAs with Variable Appearance Number 25
8.1.5. Example User Interface Issues 25
8.2. Call State Rendering 26
9. Interoperability with non-Shared Appearance UAs 26
9.1. Appearance Assignment 26
9.2. Appearance Release 27
9.3. UAs Supporting Dialog Events but Not Shared Appearance . 27
10. Provisioning Considerations 27
11. Example Message Flows 28
11.1. Registration and Subscription 28
11.2. Appearance Selection for Incoming Call 32
11.3. Outgoing Call without Appearance Seizure 35
11.4. Outgoing Call with Appearance Seizure 38
11.5. Outgoing Call without using an Appearance Number 42
11.6. Appearance Release 44
11.7. Appearance Pickup 45

11.8. Calls between UAs within the Group 50
11.9. Consultation Hold with Appearances 52
11.10. Joining or Bridging an Appearance 55
11.11. Appearance Allocation - Loss of Appearance 58
11.12. Appearance Seizure Contention Race Condition 59
11.13. Appearance Agent Subscription to UAs 60
11.14. Appearance Pickup Race Condition Failure 62
11.15. Appearance Seizure Incoming/Outgoing Contention Race Condition 65
12. Security Considerations 66
13. IANA Considerations 66
13.1. SIP Event Package Parameter: shared 66
13.2. URN Sub-Namespace Registration: sa-dialog-info 67
13.3. XML Schema Registration 67
14. Acknowledgements 68
15. References 68
15.1. Normative References 68
15.2. Informative References 69
Authors' Addresses 70

1. Introduction

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], many of the services in a business environment require the support for SIP extensions such as REFER [RFC3515], SUBSCRIBE/NOTIFY primitives [I-D.ietf-sipcore-rfc3265bis] PUBLISH [RFC3903], the SIP Replaces [RFC3891], and Join [RFC3911] header fields, etc. Many of the popular business services have been documented in the SIP Service Examples [RFC5359].

This specification details a method for implementing a group telephony feature known variously 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] in conjunction with SIP events [I-D.ietf-sipcore-rfc3265bis] and publication [RFC3903] (carrying the SIP dialog state event package [RFC4235]) for exchanging status among user agents.

In traditional telephony, the line is physical. A common scenario in telephony is for a number of business telephones to share a single or a small number of lines. The sharing or appearance of these lines between a number of phones is what gives this feature its name. A common scenario in SIP is for a number of business telephones to share a single or a small number of Address of Record (AOR) URIs.

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 of an AOR has a visual display (lamp that can change color or blink or a screen icon) and a button (used to select the appearance) where each appearance number is associated with a different dialog to/from the AOR. The telephony concept of line appearance 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.

The purpose of the appearance number is to identify active calls to facilitate sharing between users (e.g. passing a call from one user to another). If a telephone has enough buttons/lamps, the appearance number could be the positional sequence number of the button. If not, it may still be desirable to present the call state, but the appearance number should be displayed so that users know which call, for example, is on hold on which key.

In this document, except for the usage scenarios in the next section, 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

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

3. Usage Scenarios

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 main differences relate to the user interface considerations of the device.

3.1. Executive/Assistant Arrangement

The appearances on the executive's UA 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. Call Group

Users with similar business needs or tasks can be assigned to specific groups and share an AOR. 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/joined/ bridged to an existing appearance resulting in a conference call.

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 seizes 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.

3.4. Changing UAs

A user is on a call on one UA and wishes to change devices and continue the call on another UA. They place the call on hold, note the appearance number of the call, then walk to another UA. They are able to identify the same appearance number on the other UA, pickup the call, and continue the conversation.

4. Requirements and Implementation

The next section details the requirements and discusses the implementation of the shared appearances of an AOR feature.

4.1. Requirements

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 A UA must be able to join (also called bridge or conference together) or pick up (take) an active call of 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 participate in the appearance group without manual configuration of group members.

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) should 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 status of all appearances of 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 utilize 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, seize 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 join or take the call. 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 shared appearance 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 seize 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 seize a particular appearance number and also send the request at the same time. This is needed when an automatic ringdown feature (a telephone configured to immediately dial a phone number when it goes off hook) is combined with shared appearances - in this case, seizing the line is the same thing as dialing.

4.2. Implementation

This section non-normatively discusses the implementation of the shared appearance feature. The normative description is in Section 5. 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 use of a State Agent for the Dialog Package meets REQ-4 and 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 that has taken on this functionality for a group. The Appearance Agent knows or is able to determine the dialog state of all members of the group.

While the appearance resource could be managed co-operatively by a group of UAs without any central control, this is outside the scope of this draft. 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.

To best meet REQ-9, the appearance number for an incoming INVITE needs to be contained in the INVITE, in addition to being delivered in the dialog package NOTIFY. Otherwise, if the NOTIFY is delayed or lost, a UA in the group might receive an incoming INVITE but might not know which appearance number to render during alerting.

This specification defines an extension parameter for the Alert-Info header field in RFC 3261 to carry the appearance number:

Alert-Info: <urn:alert:service:normal>;appearance=1

The next section discusses the operations used to implement parts of the shared appearance feature.

1. A UA is configured with the AOR of a shared appearance group. It registers against the AOR, then attempts a dialog state subscription to the AOR. If the subscription fails, loops back to itself, or returns an error, it knows there is no State Agent, and hence no Appearance Agent and this feature is not implemented.
2. If the subscription receives a 200 OK, the UA knows there is a State Agent and that the feature is implemented. The UA then follows the steps in this list.
3. Information learned about the dialog state of other UAs in the group is rendered to the user.
4. Incoming calls are forked to all UAs in the group, and any may answer. UAs receive the appearance number to use in rendering the incoming call in a NOTIFY from the Appearance Agent and in the INVITE itself. The UA will also receive a notification if the call is answered by another UA in the group so this information can be rendered to the user.
5. For outgoing calls, the operation depends on the implementation. If the user seizes a particular appearance number for the call, the UA publishes the trying state dialog information with the desired appearance number and waits for a 2xx response before sending the INVITE.
6. For outgoing calls, if the user does not seize a particular appearance or does not care, the INVITE can be sent immediately, and the appearance number learned as the call progresses from a notification from the Appearance Agent.
7. For outgoing calls, if the user does not want an appearance number assigned (such as during a consultation call or if a UA is fetching 'service media' such as music on hold [I-D.worley-service-example]), the UA also publishes prior to sending the INVITE but does not include an appearance number in the publication.

8. Established calls within the group may be joined (bridged) or taken (picked) by another UA. Information in the dialog package notifications can be used to construct Join or Replaces header fields. Since the same appearance number is used for these types of operations, this information is published prior to sending the INVITE Join or INVITE Replaces.
9. The Appearance Agent may not have direct access to the complete dialog state of some or all of the UAs in the group. If this is the case, the Appearance Agent will subscribe to the dialog state of individual UAs in the group to obtain this information. In any case, the Appearance Agent will send normal notifications (via the subscriptions established by the UAs in step 1) every time the aggregate dialog state of the AOR changes, including when calls are placed, answered, placed on and off hold, and hung up.

5. Normative Description

This section normatively describes the shared appearance feature extensions. The following definitions are used throughout this document:

Appearance number: An appearance number is a positive integer associated with one or more dialogs of an AOR. Appearance numbers are managed by an Appearance Agent and displayed and rendered to the user by UAs that support this specification. When an appearance number is assigned or requested, generally the assigned number is the smallest positive integer that is not currently assigned as an appearance number to a dialog for this AOR.

Seizing: An appearance can be reserved prior to a call being placed by seizing the appearance. An appearance can be seized by communicating an artificial state of "trying" prior to actually initiating a dialog (i.e. sending the INVITE), in order to appear as if it was already initiating a dialog.

Selecting(or Not-Seizing): An appearance is merely selected (i.e., not seized) if there is no such communication of artificial state of "trying" prior to initiating a dialog: i.e., the state is communicated when the dialog is actually initiated. The appearance number is learned after the INVITE is sent.

5.1. Elements

A complete system to implement this feature consists of:

1. User Agents that support publications, subscriptions, and notifications for the SIP dialog event package, and the shared appearance dialog package extensions and behavior.
2. An Appearance Agent consisting of a State Agent for the dialog event package that implements an Event State Compositor (ESC) and the shared appearance dialog package extensions and behavior. The Appearance Agent also has logic for assigning and releasing appearance numbers, and resolving appearance number contention.
3. A forking proxy server that can communicate with the State Agent
4. A registrar that supports the registration event package.

The behavior of these elements is described normatively in the following sections after the definitions of the dialog package extensions.

5.2. Shared Appearance Dialog Package Extensions

This specification defines four new elements as extensions to the SIP Dialog Event package [RFC4235]. The schema is defined in Section 6. The elements are <appearance>, <exclusive>, <joined-dialog>, and <replaced-dialog> which are sub-elements of the <dialog> element.

5.2.1. The <appearance> element

The <appearance> element, a child of the <dialog> element, is used to convey the appearance number of the dialog described by the parent <dialog> element. When sent by a UA in a PUBLISH with parent <dialog> with state attribute "trying" to the Appearance Agent, the UA is requesting assignment of the given appearance number to the current or future dialog with the given dialog identifiers. When an <appearance> element is sent by the Appearance Agent in a NOTIFY, it indicates that the appearance number has been assigned to the specified dialog.

Note that a <dialog-info> element describes the contained dialogs from the point of view of the UA (named by the "entity" attribute), regardless of whether the containing request is sent by the UA or the Appearance Agent. In particular, if the UA sent a request within the described dialog, the To header field URI would match the <remote> <identity> value and the to-tag parameter would match the remote-tag attribute. Similarly, the From header field URI would match the <local> <identity> value and the from-tag parameter would match the local-tag attribute.

5.2.2. The <exclusive> element

The <exclusive> element, a child of the <dialog> element, is a boolean, which when true, indicates that the UA is not willing to

accept an INVITE with a Join or Replaces header field targeted to the dialog described by the <dialog> element that is the parent of the <exclusive> element. For example, some shared appearance systems only allow call pickup when the call is on hold. In this case, the <exclusive> element should be set to "false" when the call is held and "true" when the call is not held, rather than having the "exclusive" value implied by the hold state.

It is important to note that this element is a hint. In order to prevent another UA from taking or joining a call, a UA can, in addition to setting the <exclusive> tag, not report full dialog information to the Appearance Agent. Not having the full dialog information (Call-ID, remote-tag, and local-tag) prevents another UA from constructing a Join or Replaces header field. Although a UA may set exclusive to true, the UA must still be ready to reject an INVITE Join relating to this dialog. 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 Replaces requests containing those dialog identifiers with a 403 Forbidden response.

> Note that exclusivity has nothing to do with appearance number selection or seizing - instead, it is about call control operations that can be performed on a dialog.

If the <exclusive> element is not present, it is assumed to be false.

5.2.3. The <joined-dialog> element

The <joined-dialog> element, a child of the <dialog> element, is used to convey dialog identifiers of any other dialogs which are joined (mixed or bridged) with the dialog. Only the UA which is the common endpoint of the mixed dialogs (and thus controlling the mixing operation) should include this element in publications 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. Joined dialogs will share the same appearance number.

If the <joined-dialog> element is not present, it is assumed that the dialog is not joined or to be joined to any other dialog.

5.2.4. The <replaced-dialog> element

The <replaced-dialog> element, a child of the <dialog> element, is used to convey dialog identifiers of any other dialogs which will be or have been replaced with this dialog. For example, a UA in the group picking up a call on another UA by sending an INVITE with Replaces would include this element for the replacing dialog. Replaced dialogs will share the same appearance number.

If the <replaced-dialog> element is not present, it is assumed that the dialog has not replaced or is not to replace to any other dialog.

5.3. Shared Appearance User Agents

User Agents that support the Shared Appearance feature MUST support the dialog state package [RFC4235] with the shared appearance extensions and the 'shared' dialog event package parameter defined in Section 13.

User Agents MUST support the dialog package extensions in Section 5.2 along with SUBSCRIBE and NOTIFY [I-D.ietf-sipcore-rfc3265bis] and PUBLISH [RFC3903]. SUBSCRIBE, NOTIFY, and PUBLISH requests for the dialog event package MUST include the 'shared' Event header field parameter.

> The presence of the 'shared' Event package parameter tells the Appearance Agent that this UA supports this specification.

Upon initialization, the UA MUST subscribe to the dialog event package of the AOR and refresh the subscription per the SIP Events Framework. If the SUBSCRIBE request fails, then no Appearance Agent may be present and this feature is not active for this AOR. The UA MAY periodically retry the subscription to see if conditions have changed at intervals no shorter than 4 hours.

> Four hours was chosen to limit the subscription test to 6 per day per UA. Increading this interval would reduce this failure traffic but take longer for a newly activated Appearance Agent to be discovered.

User Agents which implement the shared appearances feature and call pickup, joining and bridging MUST support sending an INVITE with Replaces [RFC3891] or Join [RFC3911]. The User Agent Client needs to include the to-tag and from-tag information in the Replaces or Join header so that the correct dialog will be matched by the User Agent Server per the rules in RFC 3891 and RFC 3911.

All User Agents which implement the shared appearances feature and

support INVITE MUST support receiving an INVITE with a Replaces [RFC3891] or a Join [RFC3911] header field.

When publishing or notifying 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.

> The accurate rendering of the idle/active/alerting/hold state of other UAs in the group is an important part of the shared appearance feature.

In the following cases, before sending the INVITE, A UA MUST send a dialog package PUBLISH request and wait for a 2xx response before proceeding:

1. When the user seizes a particular appearance number for an outgoing call (e.g. seizing the appearance and going "off-hook", if the UA's user interface uses this metaphor).
2. When the user has requested that an appearance number not be used for an outgoing call (i.e. during a consultation call, a 'service media' call such as for music on hold [I-D.worley-service-example] or for a call not considered part of the shared appearance group).
3. When the user has selected to join (or bridge) an existing call.
4. When the user has selected to replace (or take) an existing call.

An exception is an emergency call, when a UA MUST never wait for a confirmed seizure before sending an INVITE. Instead, the emergency call MUST proceed without waiting for the PUBLISH transaction.

Note that when a UA seizes an appearance prior to establishment of a dialog (#1 and #2 in above list), not all dialog information will be available. In particular, when a UA publishes an attempt to seize an appearance prior to knowing the destination URI, minimal or no dialog information may be available. For example, in some cases, only the local target URI for the call will be known and no dialog information. If the From tag and Call-ID were not present in the initial PUBLISH, a new PUBLISH MUST be sent as soon as this information is available.

> The first publication will cause the Appearance Agent to reserve the appearance number for this UA. If the publication does not have any dialog identifiers (e.g. Call-ID, or local tag) the

> Appearance Agent cannot assign the appearance number to a particular dialog of the UA until the second publication which will contain some dialog identifiers.

This publication state is refreshed as described in [RFC3903] during the early dialog state or the Appearance Agent may reassign the appearance number. Once the dialog has transitioned to the confirmed state, no publication refreshes are necessary.

> This specification assumes that the Appearance Agent has other means besides UA publication to learn about the state of UA dialogs. In this specification, PUBLISH is used to indicate desired and intended appearance number operations. Once a dialog transitions from early to confirmed, this role is over, and hence no publication refreshes are needed.

Appearance numbers are a shorthand label for active and pending dialogs related to an AOR. Many of the features and services built using this extension rely on the correct rendering of this information to the human user. In addition, the group nature of the feature means that the rendering must be similar between different vendors and different models. Failure to do so will greatly reduce the value and usefulness of these protocol extensions. The appearances number for each active and pending dialog SHOULD be explicitly (i.e. by appearance number) or implicitly (using a user interface metaphor that makes the numbering and ordering clear to the user) rendered to the user. The far end identity of each dialog (e.g. the remote party identity) MUST NOT be rendered in place of the appearance number. The state of each appearance SHOULD also be rendered (idle, active, busy, joined, etc.). 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. Appearance numbers of dialogs can be learned by dialog package notifications containing the <appearance> element from the Appearance Agent or from the 'appearance' Alert-Info parameter in an incoming INVITE. Should they conflict, the dialog package notification takes precedence.

A UA that does not need to seize a particular appearance number (or doesn't care) would just send an INVITE as normal to place an outbound call.

A user may select an appearance number but then abandon placing a call (go back on hook). In this case, the UA MUST free up the appearance number by removing the event state with a PUBLISH as described in [RFC3903].

A UA SHOULD register against the AOR only if it is likely the UA will

be answering incoming calls. If the UA is mainly going to be monitoring the status of the shared appearance group calls and picking or joining calls, the UA SHOULD only subscribe to the AOR and not register against the AOR.

All subscribed UAs will receive dialog package NOTIFYs of trying state for incoming INVITEs.

A UA MUST NOT insert an 'appearance' parameter into an Alert-Info header field in an INVITE or other request.

The Appearance Agent is solely responsible for doing this.

5.3.1. Appearance Numbers and Call Context

There are cases where two separate dialogs at a UA are not mixed but share the same 'context'. That is, they relate to each other and should not be treated the same as any other two dialogs within the group. One example of this is a 'consultation call' where a user puts an existing dialog on hold, then calls another user, before switching back to the original dialog. Another case, described below, occurs during transfer operations, where for a transient period, a UA is involved in dialogs with two other UAs, but the dialogs are related, and should not be treated as independent dialogs. These cases are best handled by not assigning an appearance number to a newly-created dialog when it shares a context with an existing dialog. But if the pre-existing dialog is terminated, its appearance number should be reassigned to the newly-created dialog.

A UA wanting to place a call but not have an appearance number assigned publishes before sending the INVITE without an 'appearance' element but with the 'shared' event package parameter present. If the Appearance Agent policy does not allow calls without an assigned appearance number, a 400 response is sent by the Appearance Agent and the UA will republish either selecting/seizing an appearance number or send the INVITE without publishing, in which case the Appearance Agent will assign one.

Note that if an Appearance Agent rejects calls without an appearance number, certain operations such as consultation calls, transfer, and music on hold may be negatively impacted.

5.3.2. Appearance Numbers and Call Control

When an INVITE is generated to attempt to bridge or take a call (i.e. contains Join or Replaces with a dialog identifier of another dialog in the shared appearance group), the UA MUST first send a PUBLISH to the Appearance Agent. This PUBLISH will contain:

1. The appearance number of the joined or replaced call in the <appearance> element
2. If the dialog is being joined, the <joined-dialog> element will contain the dialog information from the Join header field
3. If the dialog is being replaced, the <replaced-dialog> element will contain the dialog information from the Replaces header field

Note that this information is provided to the Appearance Agent so that it can provide proper appearance assignment behavior. If the INVITE Join or Replaces was sent without publishing first, the Appearance Agent might assign a new appearance number to this INVITE, which would be a mistake. With Join, the publication has the <joined-dialog> element to prevent the Appearance Agent from generating a 400 response due to the reuse of an appearance number. For Replaces, the purpose of the <replaced-dialog> is to prevent a race condition where the BYE could cause the appearance number to be released when it should stay with the replacing dialog.

5.3.3. Appearance Numbers and Transfer

During a transfer operation, it is important that the appearance number not change during the operation. Consider the example of Alice, a member of an appearance group, who is talking to Carol, who is outside the appearance group. Carol transfers Alice to David, who is also outside the appearance group. For example, if Alice is using appearance 3 for the session with Carol, the resulting session with David should also use appearance number 3. Otherwise, an appearance number change can cause a "jump" on the UI and confusion to the user. There are two possible scenarios using the terminology of RFC 5589: Alice is the transferee in any type of transfer (receives the REFER) or the transfer target in an attended transfer (receives the INVITE with Replaces).

If Alice is the transferee, the triggered INVITE from the REFER is treated as a consultation call. Alice SHOULD publish requesting that the Appearance Agent not assign an appearance number for this INVITE. When the transfer completes, Alice SHOULD publish again to move the appearance number from the dialog with Carol to the dialog with David. Note that this publication MUST be sent prior to sending the BYE to Carol to avoid a race condition where the Appearance Agent reassigns the appearance number after seeing the BYE.

If Alice is the target, the incoming INVITE will contain a Replaces header field. As a result, the Appearance Agent will have reused the appearance number of the dialog with Carol, and this appearance number will continue to be used after the dialog with Carol has been

terminated.

5.4. Appearance Agent

An Appearance Agent defined in this specification MUST implement a dialog package state agent for the UAs registered against the AOR. The Appearance Agent MUST support the appearance dialog package extensions defined in Section 5.2. The Appearance Agent MUST support publications and subscriptions for this event package.

The Appearance Agent MUST have a way of discovering the state of all dialogs associated with the AOR. If this information is not available from a call stateful proxy or B2BUA, the Appearance Agent MAY use the registration event package [RFC3680] to learn of UAs associated with the AOR and MAY subscribe to their dialog event state. Also, an Appearance Agent MAY subscribe to a UAs dialog event state in order to reconstruct state. As a result, the registrar MUST support the registration event package. Dialog package notifications are recommended by RFC 4235 to "only contain information on the dialogs whose state or participation information has changed." This specification extends RFC 4235 as follows. The Appearance Agent SHOULD send dialog event state notifications whenever the following events happen to UAs in the AOR group:

1. A call is received, placed, answered, or terminated.
2. A call is placed on or off hold.
3. A call is joined or replaced.
4. An appearance number is reserved or released.

The Appearance Agent MUST allocate an appearance number for all incoming calls and send immediate notifications to the UAs subscribed to the shared group AOR. A new appearance number is allocated except for an incoming INVITE with a Join or Replaces header field. For this case, the appearance number should match the appearance number of the dialog being joined or replaced. If the INVITE Replaces or Join comes from outside the appearance group, the Appearance Agent will include a <joined-dialog> or <replaced-dialog> element in the NOTIFY containing the dialog information from the Replaces or Joined header field.

The Appearance Agent MUST be able to communicate with the forking proxy to learn about incoming calls and also to pass the appearance number to the proxy to insert in the Alert-Info header field.

> Note that UAs need to be able to handle incoming INVITEs without an appearance number assigned. This could be caused by a failure of the Appearance Agent or other error condition. Although the proper rendering of the INVITE may not be possible, this is better

than ignoring or failing the INVITE.

An Appearance Agent SHOULD assign an appearance number to an outgoing dialog if a PUBLISH has not been received selecting/seizing a particular appearance number.

Note that if the appearance group has appearance-unaware UAs making calls, the Appearance Agent will still allocate appearance numbers for INVITEs sent by those UAs.

An Appearance Agent receiving a PUBLISH with an appearance number checks to make sure the publication is valid. An appearance number can be assigned to only one dialog unless there is a <joined-dialog> or <replaced-dialog> element indicating that the dialog will be/has been replaced or joined. A 400 response is returned if the chosen appearance number is invalid, and an immediate NOTIFY should be sent to the UA containing full dialog event state.

An Appearance Agent receiving a PUBLISH without an appearance number but with the 'shared' event package parameter present interprets this as a request by the UA to not assign an appearance number. If the Appearance Agent policy does not allow this, a 400 response is returned. If policy does allow this, a 200 OK response is returned and no appearance number is allocated. An Appearance Agent does not have to share this dialog information (i.e. send a NOTIFY) with other UAs in the group as the information will not be rendered by the other UAs.

The Appearance Agent allocates an apperance number to a dialog from the time the appearance is requested via a PUBLISH or from the receipt of an INVITE, to the time when the last dialog associated with the appearance is terminated, including all dialogs which are joined or replaced. During the early dialog state, the Appearance Agent controls the rate of dialog state publication using the Expires header field in 200 OK responses to PUBLISH requests. An interval of 3 minutes is RECOMMENDED. After the dialog associated with the publication has been confirmed, the expiration of the publication state has no effect on the appearance allocation. If the publication contains no dialog state information, the Appearance Agent MUST reserve the appearance number for the UA but can not assign the appearance to any particular dialog of the UA. When the publication state is updated with any dialog information, the appearance number can then be assigned to the particular dialog. A UA which has been allocated an appearance number using a PUBLISH MAY free up the appearance number by removing the event state with a PUBLISH as described in [RFC3903].

If an INVITE is sent by a member of the group to the shared AOR (i.e.

they call their own AOR), the Appearance Agent MUST assign two appearance numbers. The first appearance number will be the one selected or assigned to the outgoing INVITE. The second appearance number will be another one assigned by the Appearance Agent for the INVITE as it is forked back to the members of the group.

The is to preserve a common behavior in legacy systems.

If an INVITE is sent by a member of the group using the shared AOR or sent to the shared AOR and no appearance number is available, the proxy MAY reject the INVITE with a 403 Forbidden response code.

Appearance numbers are only used for dialogs in which one or more UAs associated with the group AOR is a participant. If an incoming INVITE to the group AOR is forwarded to another AOR, the appearance number is immediately freed up and can be assigned to another dialog.

6. XML Schema Definition

The 'appearance', 'joined-dialog', 'replaced-dialog', 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"
    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="replaced-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. Alert-Info Appearance Parameter Definition

This specification extends RFC 3261 [RFC3261] to add an 'appearance' parameter to the Alert-Info header field, and to also allow proxies to modify or delete the Alert-Info header field.

The changes to RFC 3261 ABNF are:

```
alert-param      =  LAQUOT absoluteURI RAQUOT *( SEMI
                      (generic-param / appearance-param) )
appearance-param = "appearance" EQUAL 1*DIGIT
```

A proxy inserting an 'appearance' Alert-Info parameter follows normal Alert-Info policies. To indicate the appearance number for this dialog, the proxy adds the Alert-Info header field with the 'appearance' parameter to the INVITE. If an Alert-Info is already present, the proxy adds the 'appearance' parameter to the Alert-Info header field. If an appearance number parameter is already present (associated with another AOR or by mistake), the value is rewritten adding the new appearance number. There MUST NOT be more than one appearance parameter in an Alert-Info header field.

If no special ringtone is desired, a normal ringtone should be indicated using the urn:alert:service:normal in the Alert-Info, as per [I-D.ietf-salud-alert-info-urns]. The appearance number present in an Alert-Info header field SHOULD be rendered by the UA to the user, following the guidelines in Section 5.3. If the INVITE is forwarded to another AOR, the appearance parameter in the Alert-Info SHOULD be removed before forwarding outside the group.

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 seize 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.

8. User Interface Considerations

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.

8.1. 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.

8.1.1. Single Appearance UAs

These devices are constrained by only having the capability of displaying status indications for a single appearance. The UA should still send messages annotated with appearance number "1". Any call indications for appearances other than for number "1" should be rejected with a 486 or 480 response.

8.1.2. Dual Appearance UAs

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. Only appearance numbers "1" and "2" will be used by these UAs.

8.1.3. Shared Appearance UAs with Fixed Appearance Number

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 rejected, and not mapped to a free button. Users of these devices often seize specific appearance numbers for outgoing calls, and the UA will need to seize the appearance number and wait for confirmation from the Appearance Agent before proceeding with calls.

8.1.4. 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. These devices can typically make calls without waiting for confirmation from the Appearance Agent on the appearance number.

8.1.5. Example User Interface Issues

The problems faced by each style of user interface are readily seen in this example:

1. A call arrives at the shared appearance group, and is assigned an appearance number of "1". All UAs should be able to render to the user the arrival of this call.
2. Another call arrives at the shared appearance group, and is assigned an appearance number of "2". 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 "1". 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 "1" is now free, but that there is still a call on appearance number "2".
4. A third call arrives, and is assigned the appearance number of "1". All UAs should be able to render the arrival of this new call to the user. Multiple appearance 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.

8.2. Call State Rendering

UAs that implement the shared appearance 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 shared appearance group is hold. This can be determined by the presence of the "+sip.rendering=no" feature tag [RFC3840] 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.

9. Interoperability with non-Shared Appearance UAs

It is desirable to allow a basic UA that does not directly support shared appearance to be part of a shared appearance group. To support this the Proxy must collaborate with the Appearance Agent. This is not required in the basic shared appearance architecture, consequently shared appearance interoperability with non-shared appearance UAs will not be available in all shared appearance deployments.

First, a UA which does not support dialog events or the shared appearance feature will be discussed. Then, a UA which does support dialog events but not the shared appearance feature will be discussed.

9.1. Appearance Assignment

A UA that has no knowledge of appearances must will only have appearance numbers for outgoing calls if assigned by the Appearance Agent. If the non-shared appearance UA does not support Join or Replaces, all dialogs could be marked "exclusive" to indicate that these options are not available.

9.2. Appearance Release

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-shared appearance aware UA, the Appearance Agent will know about the state of dialogs including hold, etc. This information could be determined from inspection of non-end-to-end-encrypted INVITE and re-INVITE messages and added to the dialog information conveyed to other UAs.

9.3. UAs Supporting Dialog Events but Not Shared Appearance

Interoperability with UAs which support dialog events but not the shared appearance feature is more straightforward. As before, all appearance number assignment must be done by the Appearance Agent. 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 need to reject such requests or the dialogs will get joined or taken.

10. Provisioning Considerations

UAs can automatically discover if this feature is active for an AOR by looking for the 'shared' Event header parameter in a response to a dialog package SUBSCRIBE to the AOR, so no provisioning for this is needed.

The registrar will need to be provisioned to accept either first or third party registrations for the shared AOR. First party registration means the To and From URIs in the REGISTER request are the shared AOR URI. Third party registration means the To URI is the shared AOR URI and the From URI is a different AOR, perhaps that of the individual user. Either the credentials of the shared AOR or the user MUST be accepted by the registrar and the Appearance Agent, depending on the authorization policy in place for the domain.

If the Appearance Agent needs to subscribe to the dialog state of the UAs, then the Appearance Agent and the UAs need to be provisioned with credentials so the UAs can authenticate the Appearance Agent.

In some cases, UAs in the shared appearance group might have a UI limitation on the number of appearances that can be rendered.

Typically this will be hard phones with buttons/lamps instead of more flexible UIs. In this case, it can be useful for the Appearance Agent to know this maximum number. This can allow the Appearance Agent to apply policy when this limit is reached, e.g. deny a call. However, this mechanism does not provide any way to discover this by protocol means.

11. Example Message Flows

The next section shows call flow and message examples. The flows and descriptions are non-normative. Note that in these examples, all INVITEs sent by a UA in the group will be From the shared AOR (sip:HelpDesk@example.com in this case), and all INVITES sent to the group will have a Request-URI of the shared AOR. Any other requests would not apply to this feature and would be handled using normal SIP mechanisms.

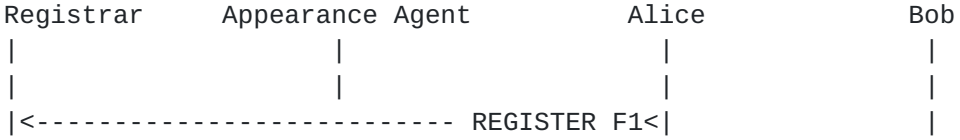
Note that the first twelve examples assume the Appearance Agent is aware of dialog state events. The example in Section 11.13 shows the case where this is not the case and as a result the Appearance Agent initiates a subscription to users of the shared AOR. Any of the other call flow examples could have shown this mode of operation as it is equally valid.

11.1. Registration and Subscription

Bob and Alice are in an appearance group identified by the shared appearance AOR sip:HelpDesk@example.com. Bob REGISTERs using contact sip:bob@ua2.example.com. Alice REGISTERs with contact sip:alice@ua1.example.com.

User Agents for Alice and Bob subscribe to the dialog package for the appearance AOR and publish dialog state to the Appearance Agent. Message exchanges between the Registrar, Appearance Agent, Alice, and Bob are shown below. The call flow examples below do not show the authentication of subscriptions, publications, and notifications. It should be noted that for security purposes, all publications and subscriptions must be authorized before the SUBSCRIBE is accepted.

Also note that registrations and subscriptions must all be refreshed by Alice at intervals determined by the expiration intervals returned by the Registrar or Appearance Agent.



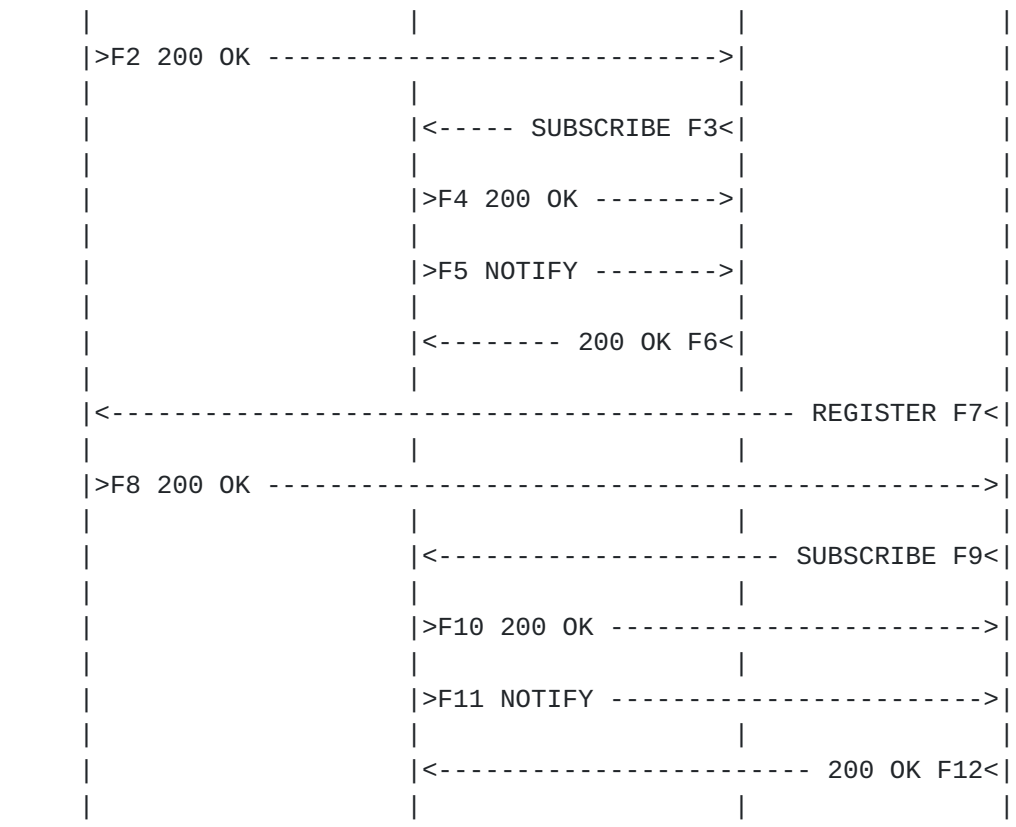


Figure 1.

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:HelpDesk@example.com>
CSeq: 2 REGISTER
Call-ID: d3281184-518783de-cc23d6bb
Contact: <sip:alice@ua1.example.com>
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=z9hG4bK527b54da8ACC7B09
CSeq: 2 REGISTER
```

```
Call-ID: d3281184-518783de-cc23d6bb
From: <sip:alice@example.com>;tag=CDF9A668-909E2BDD
To: <sip:HelpDesk@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 notifies Alice of the status.

F3 Alice ----> Appearance Agent

```
SUBSCRIBE sip:HelpDesk@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:HelpDesk@example.com>
CSeq: 91 SUBSCRIBE
Call-ID: ef4704d9-bb68aa0b-474c9d94
Contact: <sip:alice@ua1.example.com>
Event: dialog;shared
Accept: application/dialog-info+xml
Max-Forwards: 70
Expires: 3700
Content-Length: 0
```

F4 Appearance Agent ----> Alice

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP ua1.example.com;branch=z9hG4bKf10fac97E7A76D6A
CSeq: 91 SUBSCRIBE
Call-ID: ef4704d9-bb68aa0b-474c9d94
From: <sip:alice@example.com>;tag=925A3CAD-CEBB276E
To: <sip:HelpDesk@example.com>;tag=1636248422222257
Allow-Events: dialog
Expires: 3700
Contact: <sip:appearanceagent.example.com>
Content-Length: 0
```

F5 Appearance Agent ----> Alice

```
NOTIFY sip:alice@ua1.example.com SIP/2.0
From: <sip:HelpDesk@example.com>;tag=1636248422222257
To: <sip:alice@example.com>;tag=925A3CAD-CEBB276E
Call-ID: ef4704d9-bb68aa0b-474c9d94
CSeq: 232 NOTIFY
Via: SIP/2.0/UDP appearanceagent.example.com;branch=z9hG4bK1846
```

```
   Max-Forwards: 70
   Content-Type: application/dialog-info+xml
   Event: dialog;shared
   Subscription-State: active;expires=3000
   Contact: <appearanceagent.example.com>
   Content-Length: ...

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


   F6 Alice ----> Appearance Agent

   SIP/2.0 200 OK
   Via: SIP/2.0/UDP appearanceagent.example.com;branch=z9hG4bK1846
   From: <sip:HelpDesk@example.com>;tag=1636248422222257
   To: <sip:alice@example.com>;tag=925A3CAD-CEBB276E
   CSeq: 232 NOTIFY
   Call-ID: ef4704d9-bb68aa0b-474c9d94
   Contact: <sip:alice@ua1.example.com>
   Content-Length: 0


   F7 Bob ----> Registrar

   REGISTER sip:registrar.example.com SIP/2.0
   Via: SIP/2.0/UDP ua2.example.com;branch=z9hG4b53b54d87B
   From: <sip:bob@example.com>;tag=34831131
   To: <sip:HelpDesk@example.com>
   CSeq: 72 REGISTER
   Call-ID: 139490230230249348
   Contact: <sip:bob@ua2.example.com>
   Max-Forwards: 70
   Expires: 3600
   Content-Length: 0


   F8 Registrar ----> Bob

   SIP/2.0 200 OK
   Via: SIP/2.0/UDP ua2.example.com;branch=z9hG4b53b54d87B
   From: <sip:bob@example.com>;tag=34831131
   To: <sip:HelpDesk@example.com>;tag=fkwlwqi1
   CSeq: 72 REGISTER
```

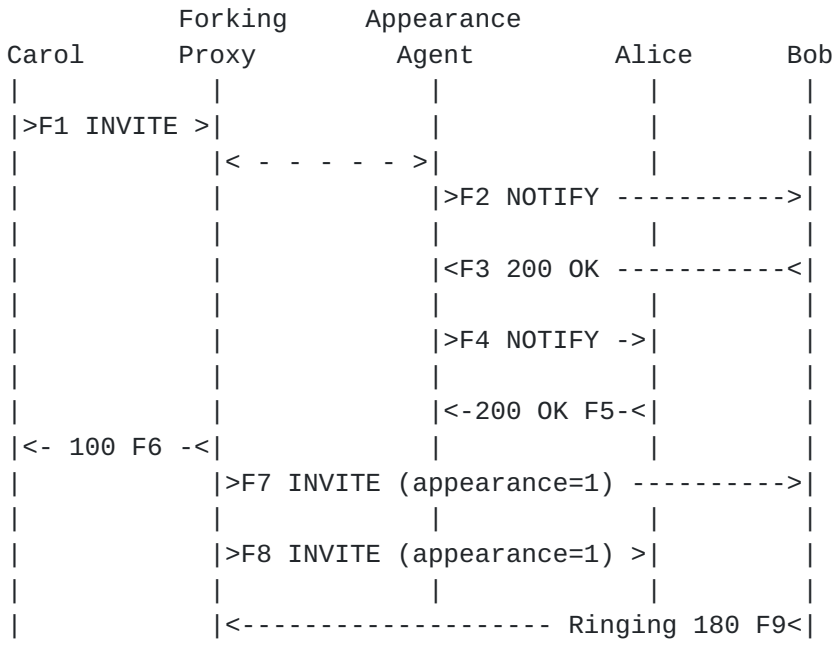
```
Call-ID: 139490230230249348
Contact: <sip:alice@ua1.example.com>;expires=3200
Contact: <sip:bob@ua2.example.com>;expires=3600
Content-Length: 0
```

11.2. Appearance Selection for Incoming Call

In the call flow below Bob and Alice are in an appearance group. Carol places a call to the appearance group AOR. The Appearance Agent sends NOTIFYs to Alice and Bob telling them what appearance the call is using. Both Alice and Bob's devices are alerted of the incoming call. Bob answers the call.

Note that it is possible that both Alice and Bob answer the call and send 200 OK responses to Carol. It is up to Carol to resolve this situation. Typically, Carol will send ACKs to both 200 OKs but send a BYE to terminate one of the dialogs. As a result, either Alice or Bob will receive the BYE and publish that their dialog is over. However, if Carol answers both Alice and Bob and keeps both dialogs active, then the Appearance Agent will need to resolve the situation by moving either Alice or Bob's dialog to a different appearance.

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. Note that the order of F2 - F5 and F7 - F8 could be reversed.



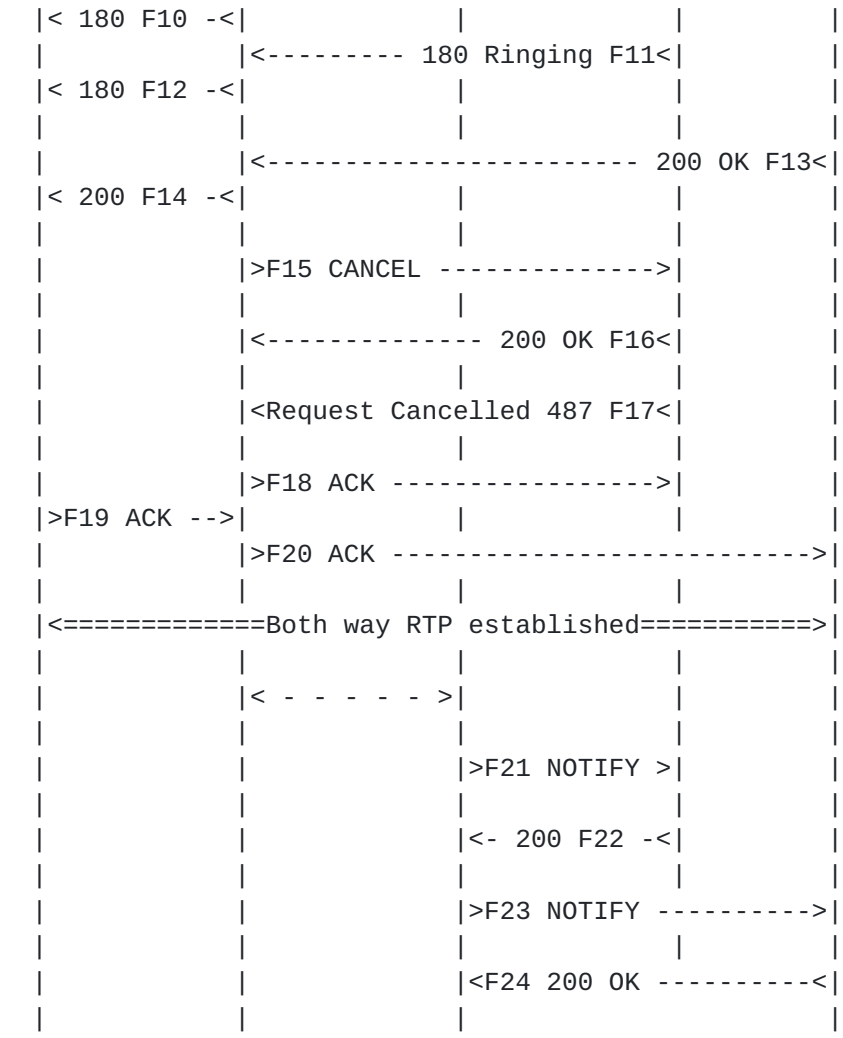


Figure 2.

F4 Appearance Agent ----> Alice

```
NOTIFY sip:alice@ua1.example.com SIP/2.0
From: <sip:HelpDesk@example.com>;tag=151702541050937
To: <sip:alice@example.com>;tag=18433323-C3D237CE
Call-ID: 1e361d2f-a9f51109-bafe31d4
CSeq: 12 NOTIFY
Via: SIP/2.0/UDP appearanceagent.example.com;branch=z9hG4bK1403
Max-Forwards: 70
Content-Type: application/dialog-info+xml
Event: dialog;shared
Subscription-State: active;expires=2800
Contact: <appearanceagent.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:HelpDesk@example.com">
   <dialog id="2a7294823093f5274e3fd2ec54a2d76c"
        call-id="14-1541707345"
        remote-tag="44BAD75D-E3128D42"
        direction="recipient">
      <sa:appearance>1</sa:appearance>
      <state>trying</state>
      <remote>
        <identity>sip:carol@ua.example.com</identity>
      </remote>
   </dialog>
</dialog-info>
```

F7 Proxy ----> Bob

```
INVITE sip:bob@ua2.example.com SIP/2.0
Via: SIP/2.0/UDP ua3.example.com;branch=z9hG4bK4324ea
Via: SIP/2.0/UDP proxy.example.com;branch=z9hG4bK38432ji
From: <sip:carol@example.com>;tag=44BAD75D-E3128D42
To: <sip:HelpDesk@example.com>
CSeq: 106 INVITE
Call-ID: 14-1541707345
Contact: <sip:carol@ua3.example.com>
Max-Forwards: 69
Alert-Info: <urn:alert:service:normal>;appearance=1
Content-Type: application/sdp
Content-Length: ...

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

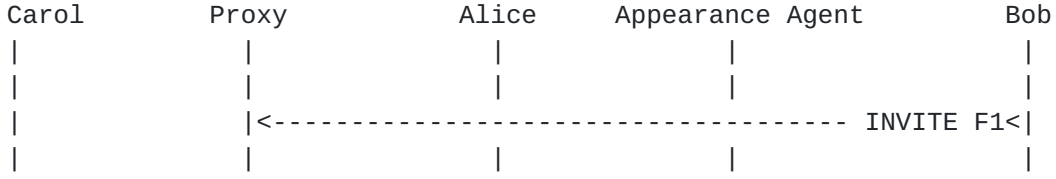
F21 Appearance Agent ----> Alice

```
NOTIFY sip:alice@ua1.example.com SIP/2.0
```

```
From: <sip:HelpDesk@example.com>;tag=151702541050937
To: <sip:alice@example.com>;tag=18433323-C3D237CE
Call-ID: 1e361d2f-a9f51109-bafe31d4
CSeq: 13 NOTIFY
Via: SIP/2.0/UDP appearanceagent.example.com;branch=z9hG4bK4164F03j
Max-Forwards: 70
Content-Type: application/dialog-info+xml
Event: dialog;shared
Subscription-State: active;expires=2500
Contact: <appearanceagent.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="17"
             state="partial"
             entity="sip:HelpDesk@example.com">
   <dialog id="2a7294823093f5274e3fd2ec54a2d76c"
        call-id="14-1541707345"
        remote-tag="44BAD75D-E3128D42"
        local-tag="7349dsfjkFD03s"
        direction="recipient">
      <sa:appearance>1</sa:appearance>
      <state>confirmed</state>
      <local>
        <target>sip:bob@ua2.example.com</target>
      </local>
      <remote>
        <identity>sip:carol@ua.example.com</identity>
      </remote>
   </dialog>
</dialog-info>
```

11.3. Outgoing Call without Appearance Seizure

In this scenario, Bob's UA places a call without first selecting/ seizing an appearance number. After Bob sends the INVITE, the appearance assigns an appearance number for it and notifies both Alice and Bob.



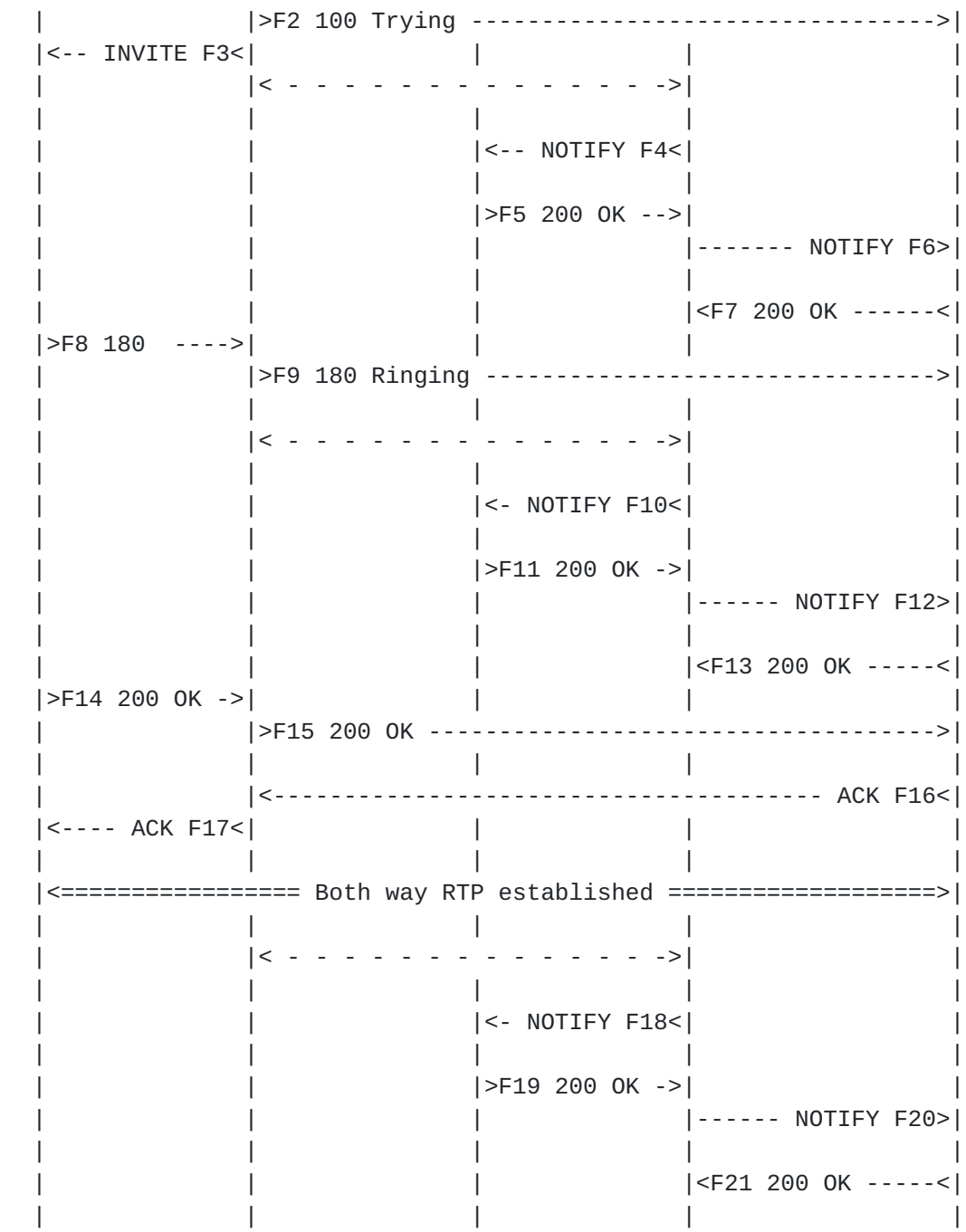


Figure 3.

F1 Bob ----> Proxy

```
INVITE sip:carol@example.com SIP/2.0
Via: SIP/2.0/UDP ua2.example.com;branch=z9hG4bK98c87c52123A08BF
From: <sip:HelpDesk@example.com>;tag=15A3DE7C-9283203B
To: <sip:carol@example.com>
```

```
   CSeq: 1 INVITE
   Call-ID: f3b3cbd0-a2c5775e-5df9f8d5
   Contact: <sip:bob@ua2.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


   F4 Appearance Agent ----> Alice

   NOTIFY sip:alice@ua1.example.com SIP/2.0
   Via: SIP/2.0/UDP appearanceagent.example.com;branch=z9hG4bK81d84f62
   From: <sip:HelpDesk@example.com>;tag=1636248422222257
   To: <sip:alice@example.com>;tag=925A3CAD-CEBB276E
   Call-ID: ef4704d9-bb68aa0b-474c9d94
   CSeq: 233 NOTIFY
   Max-Forwards: 70
   Content-Type: application/dialog-info+xml
   Event: dialog;shared
   Subscription-State: active;expires=2200
   Contact: <appearanceagent.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:HelpDesk@example.com">
       <dialog id="fa02538339df3ce597f9e3e3699e28fc"
            call-id="f3b3cbd0-a2c5775e-5df9f8d5"
            local-tag="15A3DE7C-9283203B"    direction="initiator">
               <sa:appearance>1</sa:appearance>
               <sa:exclusive>false</sa:exclusive>
                  <state>trying</state>
                   <local>
                       <target uri="sip:bob@ua2.example.com">
```

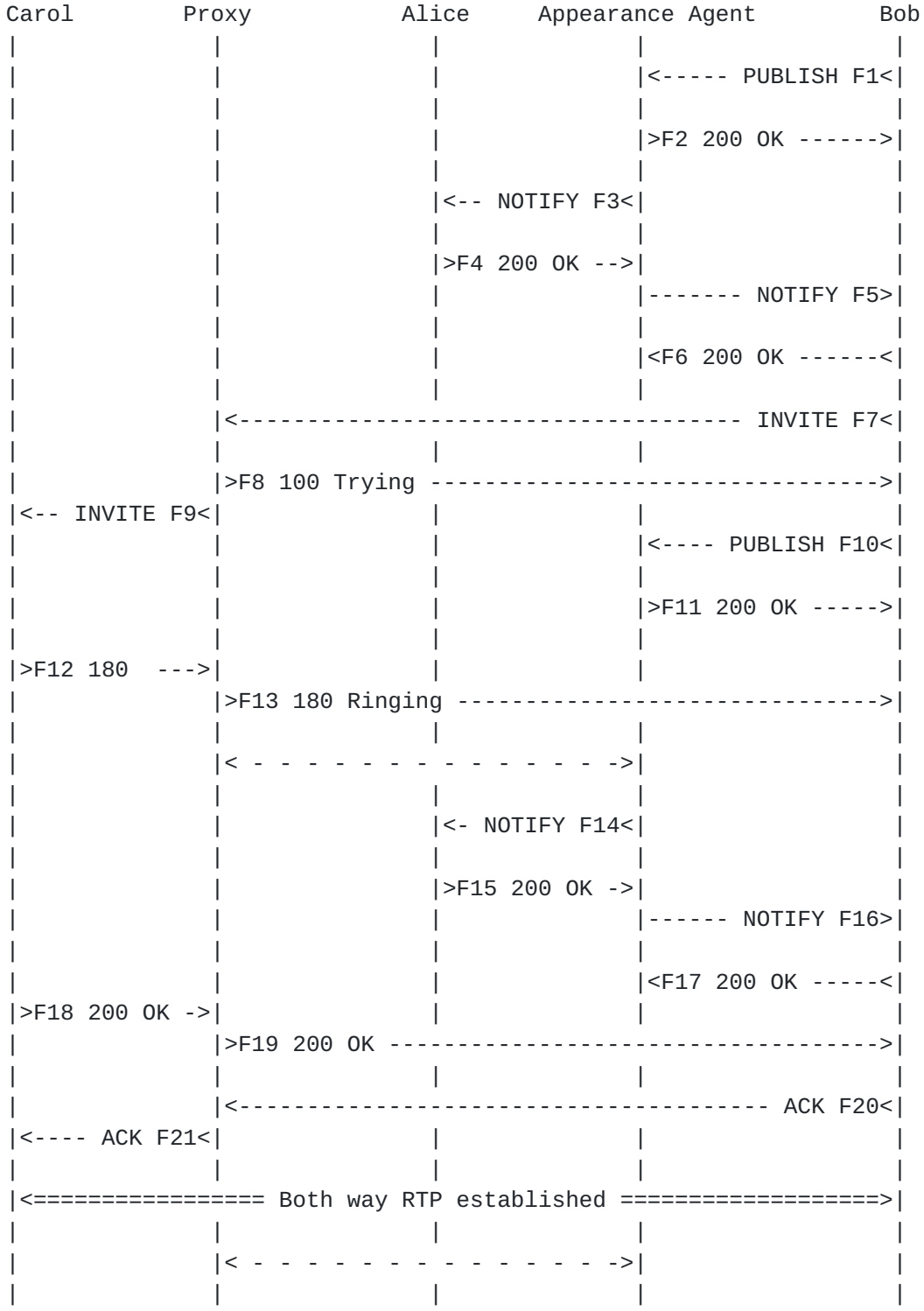
```
                    </target>
                   </local>
          </dialog>
   </dialog-info>
```

```
   F6 Appearance Agent ----> Bob

   NOTIFY sip:bob@ua1.example.com SIP/2.0
   From: <sip:HelpDesk@example.com>;tag=497585728578386
   To: <sip:bob@example.com>;tag=633618CF-B9C2EDA4
   Call-ID: a7d559db-d6d7dcad-311c9e3a
   CSeq: 7 NOTIFY
   Via: SIP/2.0/UDP appearanceagent.example.com
    ;branch=z9hG4bK1711759878512309
   Max-Forwards: 70
   Content-Type: application/dialog-info+xml
   Event: dialog;shared
   Subscription-State: active;expires=2000
   Contact: <sip:appearanceagent.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="78"
                state="partial"
                entity="sip:HelpDesk@example.com">
       <dialog id="02538339hfgdf3ce597f9e3egkl3699e28fc"
            call-id="f3b3cbd0-a2c5775e-5df9f8d5"
            local-tag="15A3DE7C-9283203B"    direction="initiator">
               <sa:appearance>1</sa:appearance>
               <sa:exclusive>false</sa:exclusive>
                  <state>trying</state>
                   <local>
                       <target uri="sip:bob@ua2.example.com">
                       </target>
                      </local>
           </dialog>
   </dialog-info>
```

11.4. Outgoing Call with Appearance Seizure

In this scenario, Bob's UA sends out a dialog event PUBLISH with state (trying) selecting/seizing an appearance number before sending the INVITE. After receiving the 200 OK 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. Bob's UA puts as much of the dialog information from F7 as can be determined in advance. In this case, the minimum of the Contact URI is included which allows the Appearance Agent to correlate the INVITE with the PUBLISH.



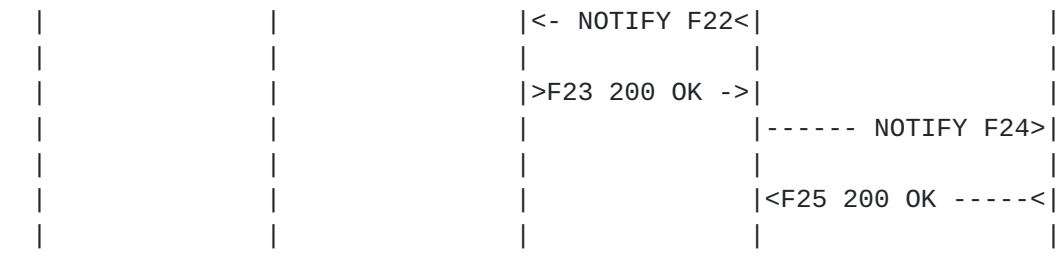


Figure 4.

F1 to F4: Bob uses the shared appearance of the Help Desk on his UA to place an outgoing call (e.g., he goes off-hook). Before sending the outgoing INVITE request, Bob publishes to the Appearance Agent reserving appearance number 1. The Appearance Agent notifies Alice (and all other UAs, including Bob) of the event by sending NOTIFYs.

F1 Bob ----> Appearance Agent

```
PUBLISH sip:HelpDesk@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:HelpDesk@example.com>
CSeq: 7 PUBLISH
Call-ID: 44fwF144-F12893K38424
Contact: <sip:bob@ua2.example.com>
Event: dialog;shared
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="full"
             entity="sip:HelpDesk@example.com">
    <dialog id="id3d4f9c83" direction="initiator">
        <sa:appearance>1</sa:appearance>
        <sa:exclusive>false</sa:exclusive>
        <state>trying</state>
        <local>
            <target uri="sip:bob@ua2.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
From: <sip:bob@example.com>;tag=44150CC6-A7B7919D
To: <sip:HelpDesk@example.com>
CSeq: 7 PUBLISH
Call-ID: 44fwF144-F12893K38424
Contact: <sip:bob@ua2.example.com>
Event: dialog;shared
SIP-Etag: 482943245
Allow-Events: dialog
Expires: 60
Content-Length: 0
```

F7 Bob ---> Proxy

```
INVITE sip:carol@example.com SIP/2.0
Via: SIP/2.0/UDP ua2.example.com;branch=z9hG4bK342122
Max-Forwards: 70
From: <sip:HelpDesk@example.com>;tag=15A3DE7C-9283203B
To: <sip:carol@example.com>
Call-ID: f3b3cbd0-a2c5775e-5df9f8d5
CSeq: 31 INVITE
Contact: <sip:bob@ua2.example.com>
Content-Type: application/sdp
Content-Length: ...

(SDP Not Shown)
```

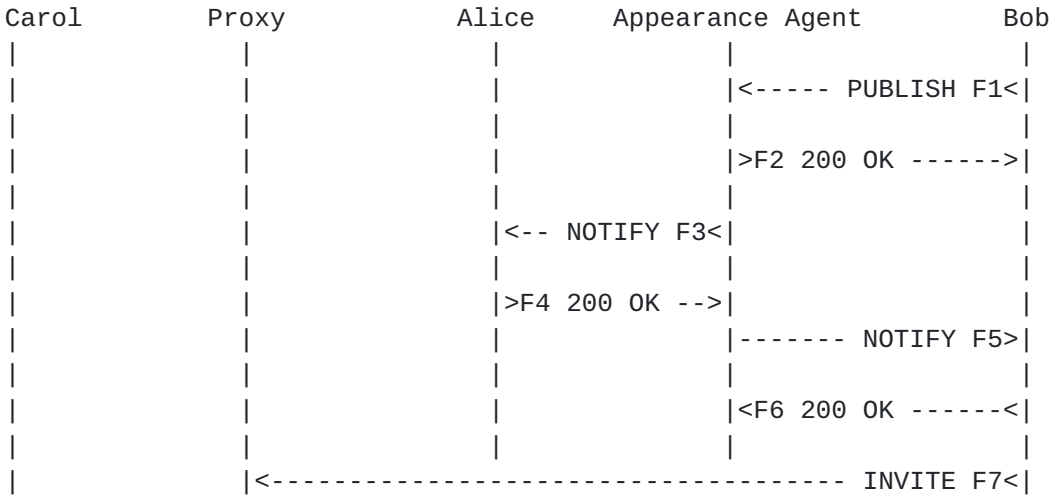
F10 Bob ----> Appearance Agent

```
PUBLISH sip:HelpDesk@example.com SIP/2.0
Via: SIP/2.0/UDP ua2.example.com;branch=z9hG4bK6d644638E7
From: <sip:bob@example.com>;tag=0CCf6-A7FdsB79D
To: <sip:HelpDesk@example.com>
CSeq: 437 PUBLISH
Call-ID: fwF14d4-F1FFF2F2893K38424
Contact: <sip:bob@ua2.example.com>
Event: dialog;shared
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="full"
                entity="sip:HelpDesk@example.com">
        <dialog id="id3d4f9c83"
             call-id="f3b3cbd0-a2c5775e-5df9f8d5"
             local-tag="15A3DE7C-9283203B"
                                        direction="initiator">
            <sa:appearance>1</sa:appearance>
            <sa:exclusive>false</sa:exclusive>
            <state>trying</state>
            <local>
                <target uri="sip:bob@ua2.example.com">
                </target>
            </local>
            <remote>
                <identity uri="sip:carol@example.com">
                </identity>
            </remote>
        </dialog>
    </dialog-info>
```

11.5. Outgoing Call without using an Appearance Number

In this scenario, Bob's UA sends out a dialog event PUBLISH with state (trying) indicating that he does not want to utilize an appearance number for this dialog. The PUBLISH does not have an appearance element but does have the 'shared' dialog event parameter. As a result, the Appearance Agent knows the UA does not wish to use an appearance number for this call. If the Appearance Agent does not wish to allow this, it would reject the PUBLISH with a 400 response and the UA would know to re-PUBLISH selecting/seizing an appearance number.



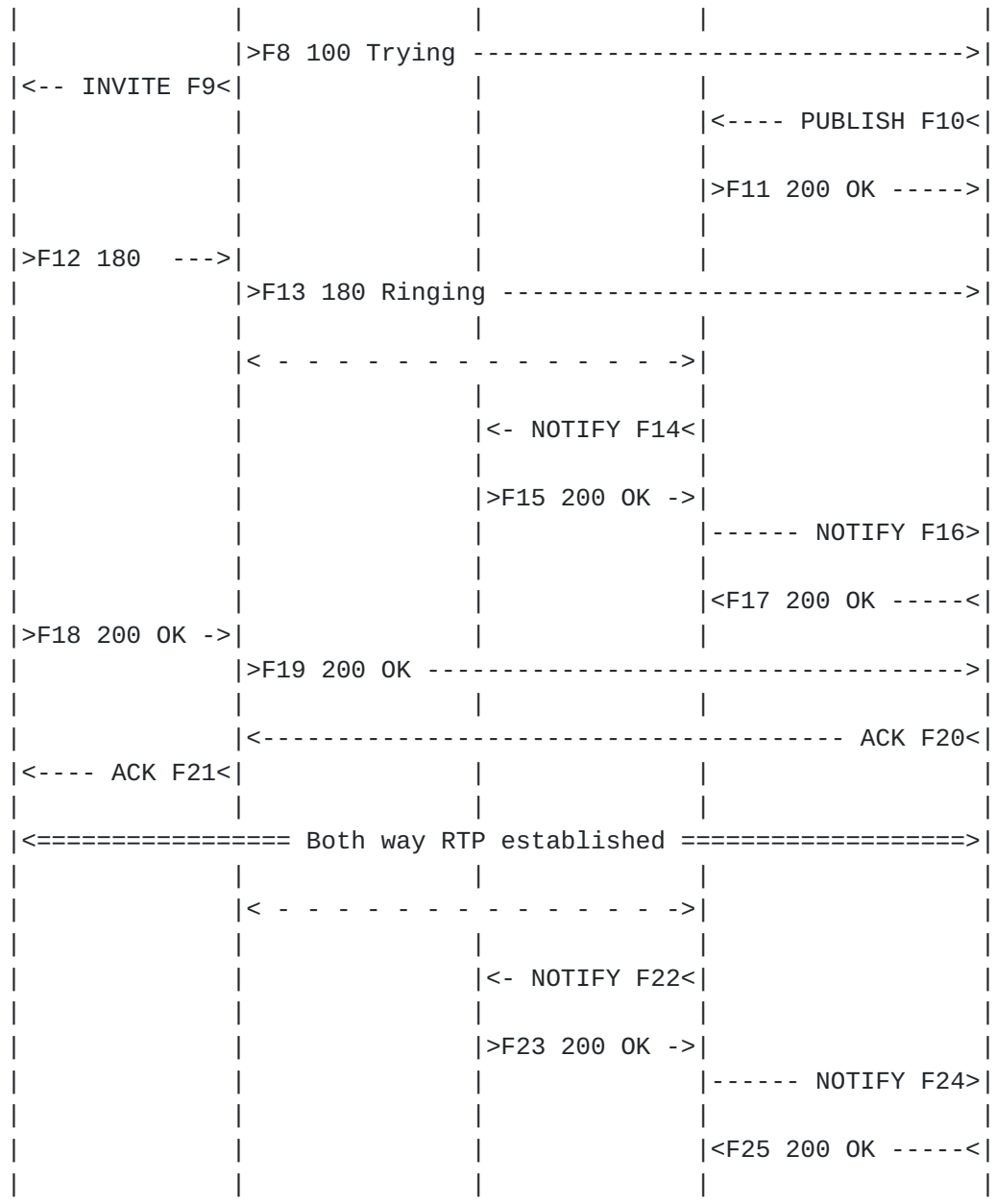


Figure 5.

F1 Bob ----> Appearance Agent

```
PUBLISH sip:appearanceagent.example.com SIP/2.0
Via: SIP/2.0/UDP ua2.example.com;branch=z9hG4bK61314d6446383E79
From: <sip:bob@example.com>;tag=4415df82k39sf
To: <sip:HelpDesk@example.com>
CSeq: 7 PUBLISH
Call-ID: 44fwF144-F12893K38424
Contact: <sip:bob@ua2.example.com>
```

```
Event: dialog;shared
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="full"
             entity="sip:HelpDesk@example.com">
    <dialog id="id3d4f9c83" direction="initiator">
        <sa:exclusive>false</sa:exclusive>
        <state>trying</state>
        <local>
            <target uri="sip:bob@ua2.example.com">
            </target>
        </local>
    </dialog>
</dialog-info>
```

Note that F7 would be the same as the previous example.

11.6. Appearance Release

Bob and Carol are in a dialog, created, for example as in Section 11.3. Carol sends a BYE to Bob to terminate the dialog and the Appearance Agent de-allocates the appearance number used, sending notifications out to the UAs in the shared group.

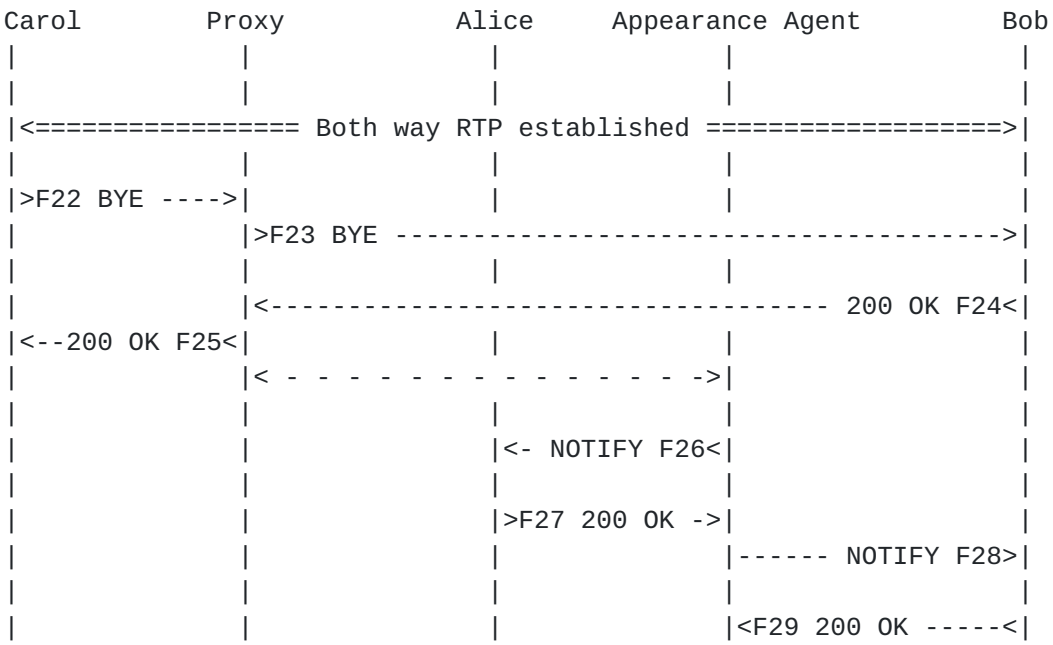


Figure 6.

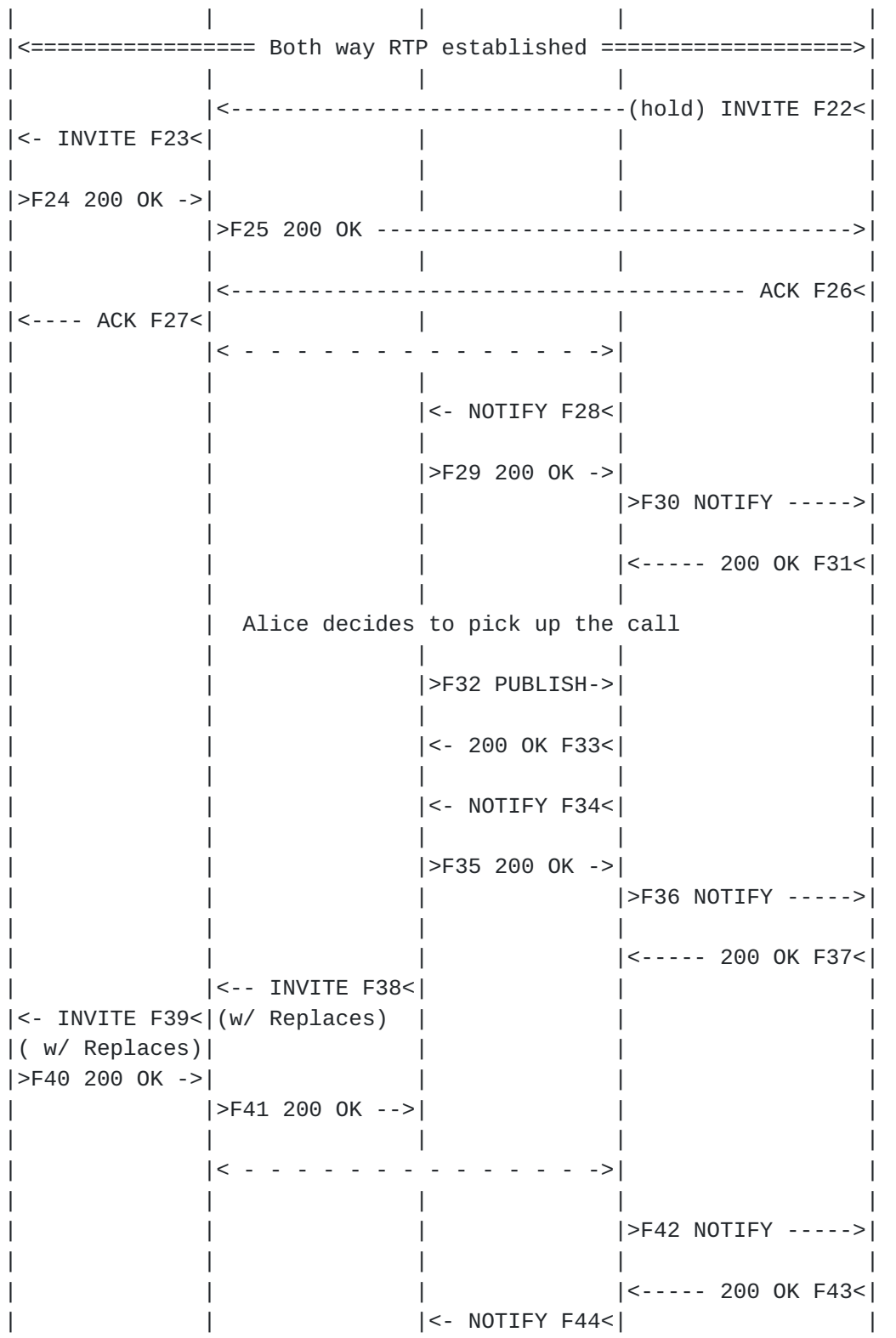
F28 Appearance Agent ----> Bob

```
NOTIFY sip:bob@ua1.example.com SIP/2.0
From: <sip:HelpDesk@example.com>;tag=497585728578386
To: <sip:bob@example.com>
Call-ID: a7d559db-d6d7dcad-311c9e3a
CSeq: 7 NOTIFY
Via: SIP/2.0/UDP appearanceagent.example.com
 ;branch=z9hG4bK759878512309
Max-Forwards: 70
Content-Type: application/dialog-info+xml
Event: dialog;shared
Subscription-State: active;expires=1800
Contact: <sip:appearanceagent.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:HelpDesk@example.com">
    <dialog id="fa02538339df3ce597f9e3e3699e28fc"
         call-id="f3b3cbd0-a2c5775e-5df9f8d5"
         local-tag="15A3DE7C-9283203B"
         remote-tag="65a98f7c-1dd2-11b2-88c6-b0316298f7c"
                                      direction="initiator">
            <sa:appearance>1</sa:appearance>
            <sa:exclusive>false</sa:exclusive>
               <state>terminated</state>
                <local>
                    <target uri="sip:bob@ua2.example.com">
                    </target>
                  </local>
        </dialog>
</dialog-info>
```

11.7. Appearance Pickup

In this scenario, Bob has an established dialog with Carol created using the call flows of Figure 1 or Figure 2. Bob then places Carol on hold. Alice receives a notification of this and renders this on Alice's UI. Alice subsequently picks up the held call and has a established session with Carol. Finally, Carol hangs up. Alice must PUBLISH F32 to indicate that the INVITE F38 will be an attempt to

pickup the dialog between Carol and Bob, and hence may use the same appearance number.



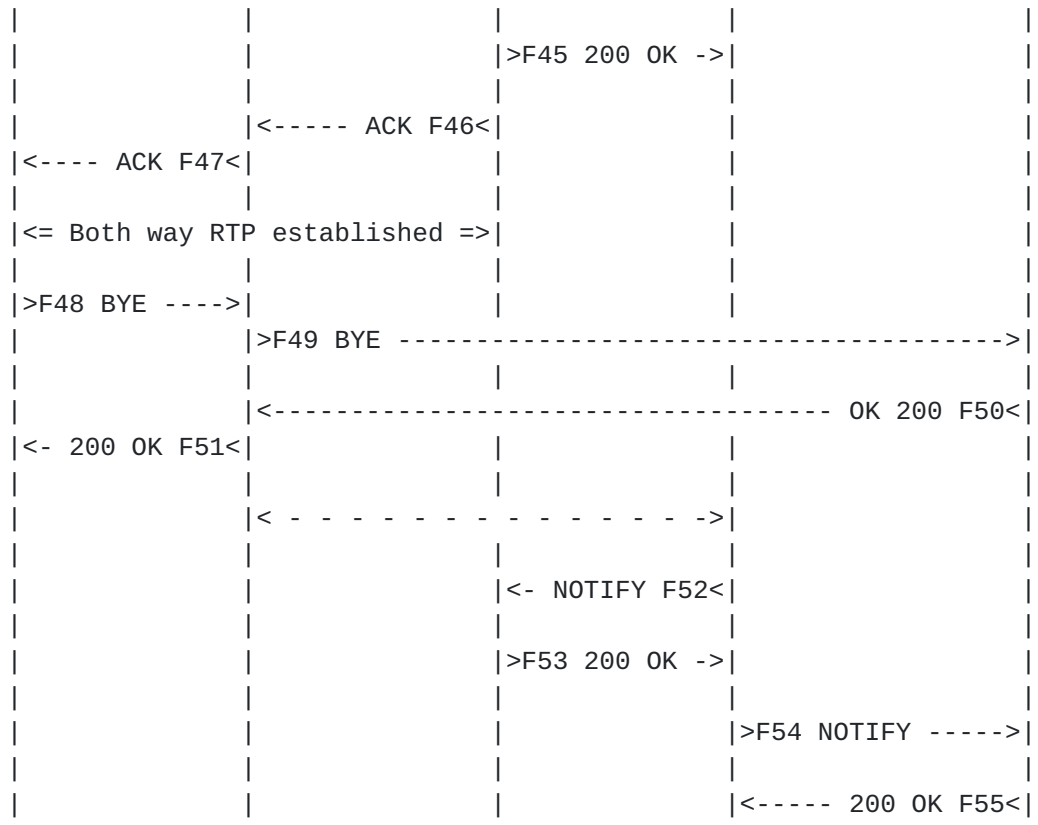


Figure 7.

F28 Appearance ----> Alice

```
NOTIFY sip:alice@ua1.example.com SIP/2.0
From: <sip:HelpDesk@example.com>;tag=151702541050937
To: <sip:alice@example.com>;tag=18433323-C3D237CE
Call-ID: 1e361d2f-a9f51109-bafe31d4
CSeq: 12 NOTIFY
Via: SIP/2.0/UDP appearanceagent.example.com
 ;branch=z9hG4bK1403
Max-Forwards: 70
Content-Type: application/dialog-info+xml
Event: dialog;shared
Subscription-State: active;expires=1800
Contact: <sip:appearanceagent.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="10"
             state="partial"
```

```
                entity="sip:HelpDesk@example.com">
      <dialog id="id3d4f9c83"
            call-id="f3b3cbd0-a2c5775e-5df9f8d5"
            local-tag="15A3DE7C-9283203B"
            remote-tag="65a98f7c-1dd2-11b2-88c6-b0316298f7c"
            direction="initiator">
            <sa:appearance>1</sa:appearance>
            <sa:exclusive>false</sa:exclusive>
            <state>active</state>
            <local>
              <target uri="sip:bob@ua2.example.com">
                <param pname="+sip.rendering" pval="no"/>
              </target>
            </local>
            <remote>
              <identity>sip:carol@example.com</identity>
              <target uri="sip:carol@ua3.example.com" />
            </remote>
       </dialog>
   </dialog-info>


   F32 Alice ----> Appearance Agent

   PUBLISH sip:HelpDesk@example.com SIP/2.0
   Via: SIP/2.0/UDP ua2.example.com;branch=z9hG4bKa5d6cf61F5FBC05A
   From: <sip:HelpDesk@example.com>;tag=44150CC6-A7B7919D
   To: <sip:alice@example.com>;tag=428765950880801
   CSeq: 11 PUBLISH
   Call-ID: 87837Fkw87asfds
   Contact: <sip:alice@ua2.example.com>
   Event: dialog;shared
   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="full"
                entity="sip:HelpDesk@example.com">
      <dialog id="id3d4f9c83"
          call-id="3d57cd17-47deb849-dca8b6c6"
                        local-tag="8C4183CB-BCEAB710" >
            <sa:appearance>1</sa:appearance>
            <sa:exclusive>false</sa:exclusive>
            <sa:replaced-dialog
```

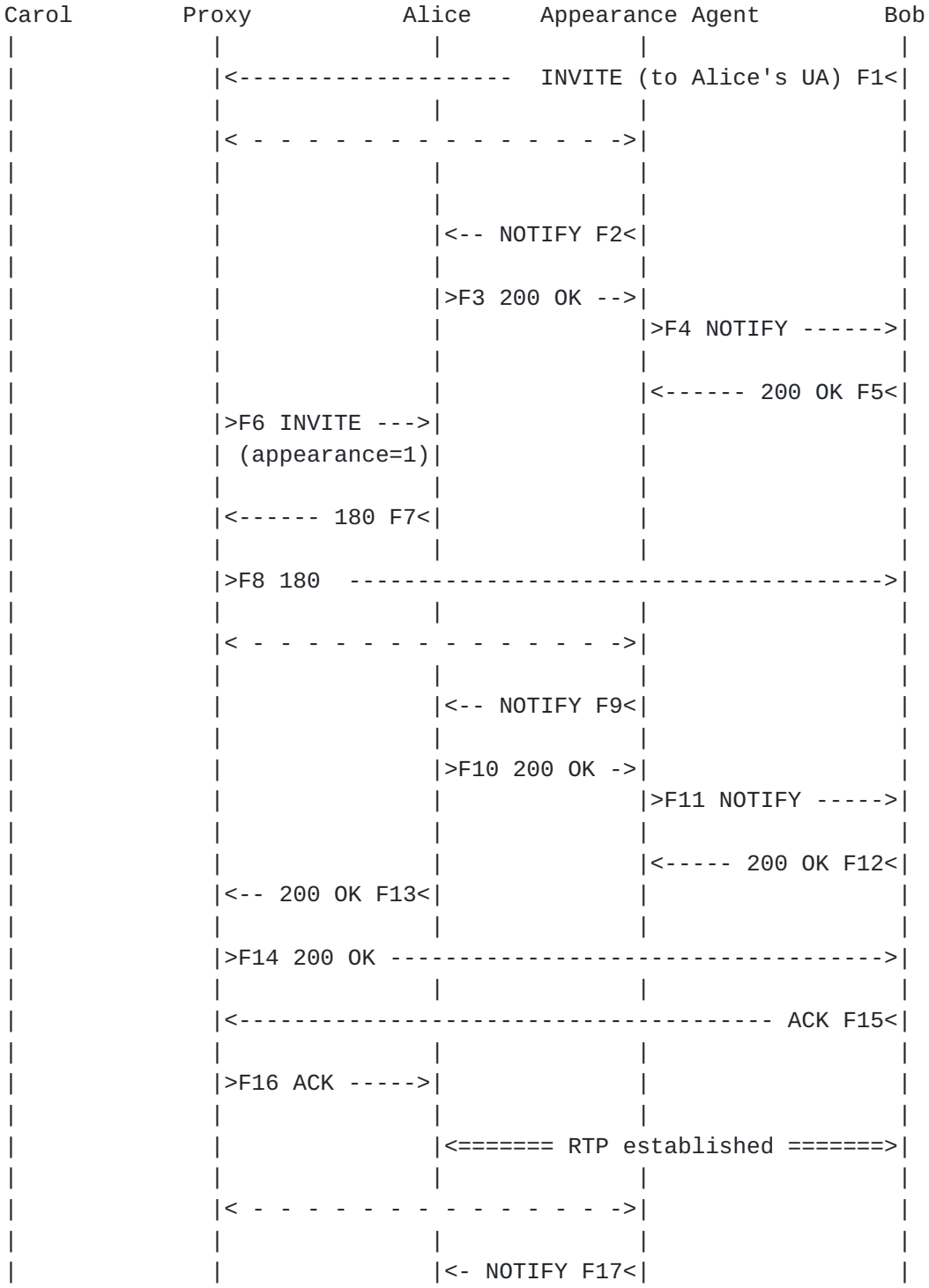
```
              call-id="f3b3cbd0-a2c5775e-5df9f8d5"
              from-tag="15A3DE7C-9283203B"
              to-tag="65a98f7c-1dd2-11b2-88c6-b03162323164+65a98f7c" />
            <state>trying</state>
            <local>
              <target uri="sip:alice@ua1.example.com">
                <param pname="+sip.rendering" pval="yes"/>
              </target>
            </local>
            <remote>
                <target uri="sip:carol@ua3.example.com" />
            </remote>
        </dialog>
    </dialog-info>
```

F38 Alice ----> Proxy

```
INVITE sip:carol@example.com SIP/2.0
Via: SIP/2.0/UDP ua1.example.com;branch=z9hG4bK4ea695b5B376A60C
From: <sip:HelpDesk@example.com>;tag=8C4183CB-BCEAB710
To: <sip:carol@example.com:5075>
CSeq: 1 INVITE
Call-ID: 3d57cd17-47deb849-dca8b6c6
Contact: <sip:alice@ua1.example.com>
<all-one-line>
Replaces: f3b3cbd0-a2c5775e-5df9f8d5;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
```

11.8. Calls between UAs within the Group

In this scenario, Bob calls Alice who is also in the Appearance group. Only one appearance number is used for this dialog. This example also shows the use of the 'exclusive' tag to indicate that other UAs in the group can not join or take this dialog.



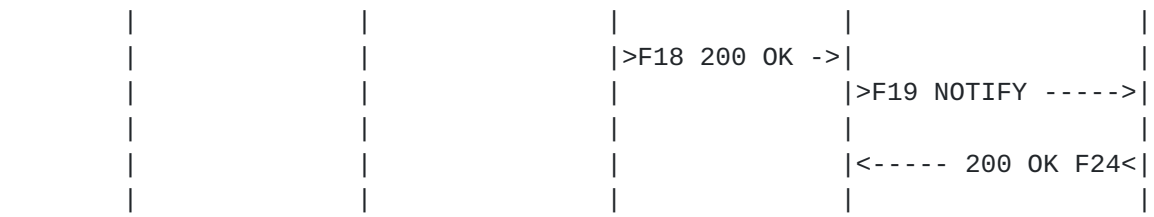


Figure 8.

F19 Appearance Agent ----> Bob

```
NOTIFY sip:bob@ua1.example.com SIP/2.0
From: <sip:HelpDesk@example.com>;tag=497585728578386
To: <sip:bob@example.com>;tag=633618CF-B9C2EDA4
Call-ID: a7d559db-d6d7dcad-311c9e3a
CSeq: 7 NOTIFY
Via: SIP/2.0/UDP appearanceagent.example.com
 ;branch=z9hG4bK1711759878512309
Max-Forwards: 70
Content-Type: application/dialog-info+xml
Event: dialog;shared
Subscription-State: active;expires=1500
Contact: <sip:appearanceagent.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="10"
             state="partial"
             entity="sip:HelpDesk@example.com">
   <dialog id="3xdsd4f9c83"
           call-id="b3cbd0-ad2c5775e-5df9f8d5"
              local-tag="34322kdfr234f"
           remote-tag="3153DE7C-928203B"
           direction="initiator">
         <sa:exclusive>true</sa:exclusive>
         <sa:appearance>1</sa:appearance>
         <state>confirmed</state>
         <local>
           <target uri="sip:bob@ua2.example.com">
           </target>
         </local>
         <remote>
           <identity>sip:HelpDesk@example.com</identity>
           <target uri="sip:alice@ua1.example.com" />
         </remote>
```

```
      </dialog>

      <dialog id="4839589"
             call-id="b3cbd0-ad2c5775e-5df9f8d5"
             local-tag="3153DE7C-928203B"
                remote-tag="34322kdfr234f"
             direction="responder">
           <sa:exclusive>true</sa:exclusive>
           <sa:appearance>1</sa:appearance>
           <state>confirmed</state>
           <local>
              <target uri="sip:alice@ua1.example.com" />
           </local>
           <remote>
             <identity>sip:HelpDesk@example.com</identity>
             <target uri="sip:bob@ua2.example.com" />
           </remote>
      </dialog>

   </dialog-info>
```

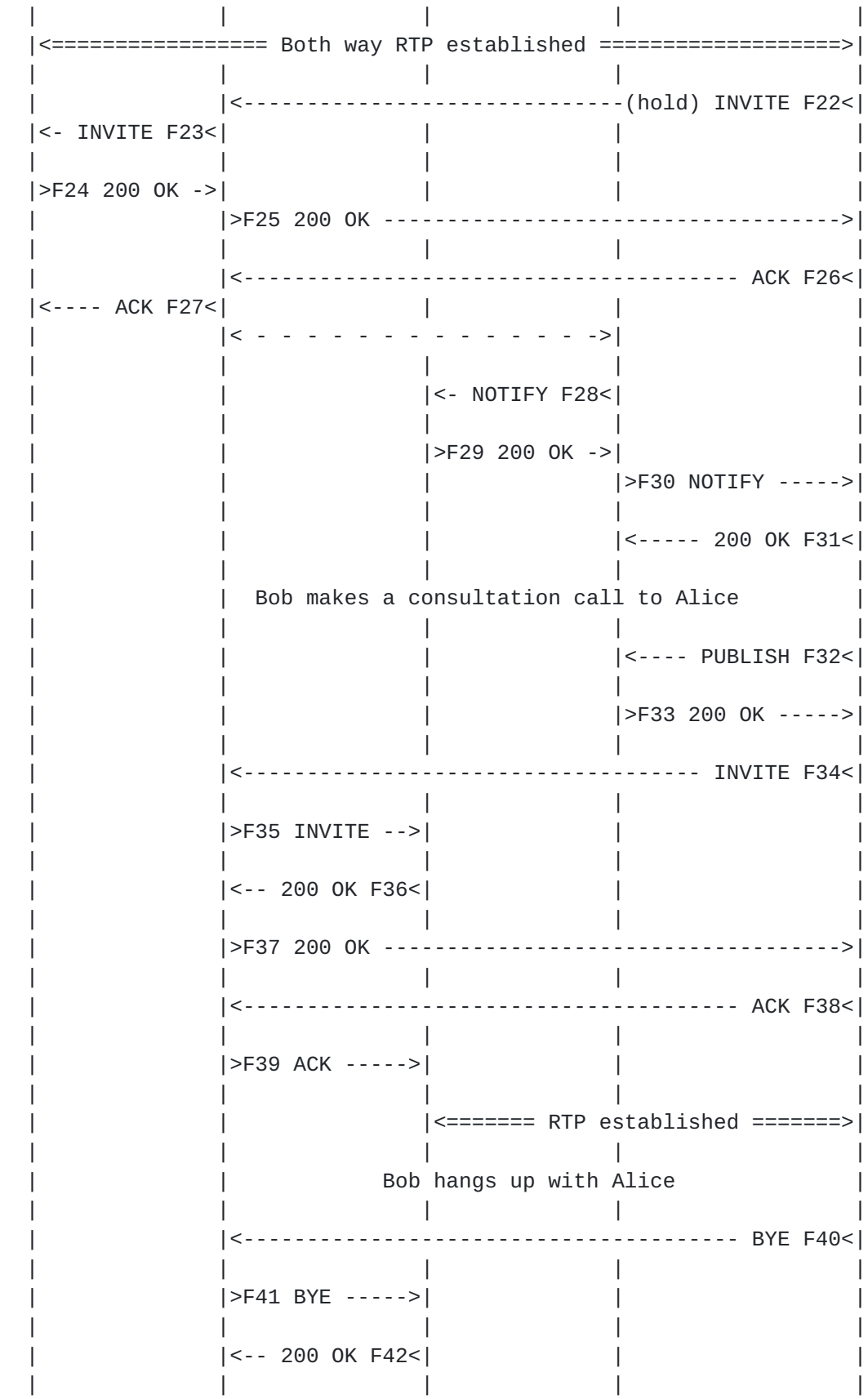
11.9. Consultation Hold with Appearances

In this scenario, Bob has a call with Carol. Bob makes a consultation call to Alice by putting Carol on hold and calling Alice. Bob's UA chooses not to have an appearance number for the call to Alice since it is treating it as part of the call to Carol. He indicates this in the PUBLISH F32 which contains the 'shared' Event parameter but no <appearance> element. The PUBLISH is sent before the INVITE to Alice to ensure no appearance number is assigned by the Appearance Agent. Finally, Bob hangs up with Alice and resumes the call with Carol. Dialog notifications of the consultation call are not shown, as they are not used.

Note that if Carol hangs up while Bob is consulting with Alice, Bob can decide if he wants to reuse the appearance number used with Carol for the call with Alice. If not, Bob publishes the termination of the dialog with Carol and the Appearance Agent will re-allocate the appearance. If he wants to keep the appearance, Bob will publish the termination of the dialog with Carol and also publish the appearance with the dialog with Alice. This will result in Bob keeping the appearance number until he reports the dialog with Alice terminated.

Note that the call flow would be similar if Bob called a music on hold server instead of Alice to implement a music on hold service as described in [I-D.worley-service-example].

Carol Proxy Alice Appearance Agent Bob


Both way RTP established
(hold) INVITE F22
<- INVITE F23
>F24 200 OK ->
>F25 200 OK
ACK F26
<---- ACK F27
< - - - - - - - - - - - - - ->
<- NOTIFY F28
>F29 200 OK ->
>F30 NOTIFY ----->
<----- 200 OK F31
Bob makes a consultation call to Alice
<---- PUBLISH F32
>F33 200 OK ----->
INVITE F34
>F35 INVITE -->
<-- 200 OK F36
>F37 200 OK
ACK F38
>F39 ACK ----->
<======= RTP established =======
Bob hangs up with Alice
BYE F40
>F41 BYE ----->
<-- 200 OK F42

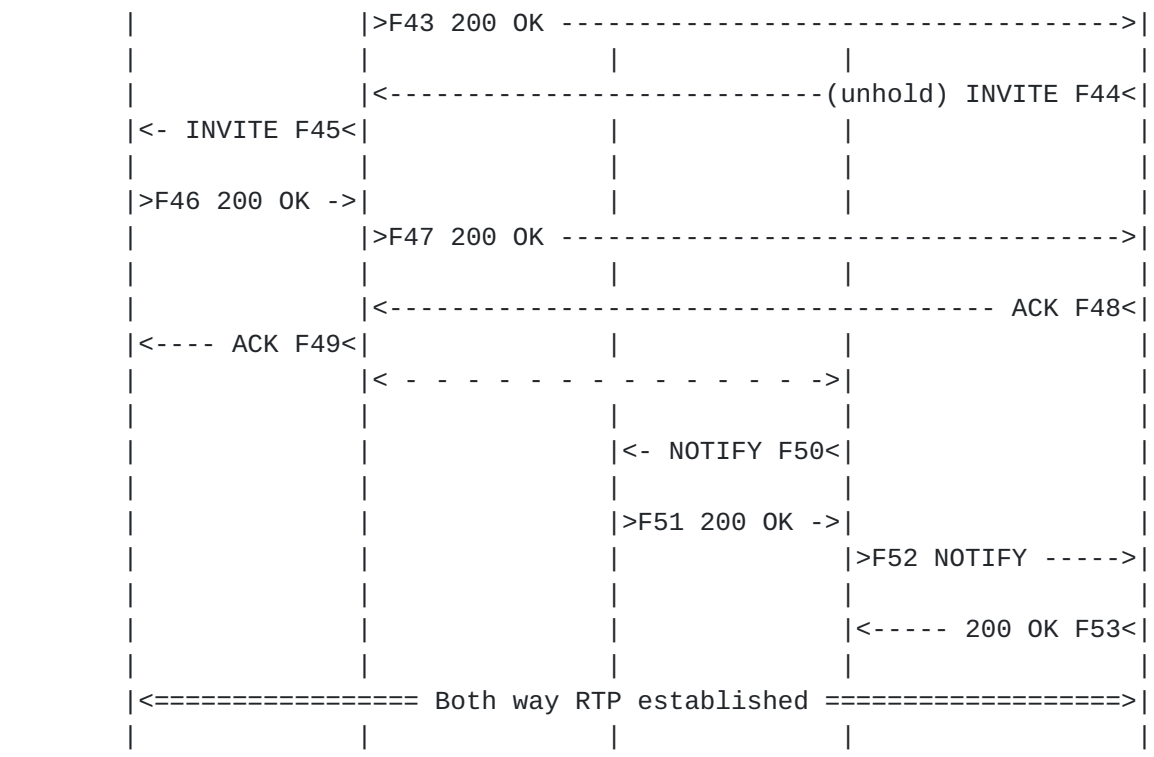


Figure 9.

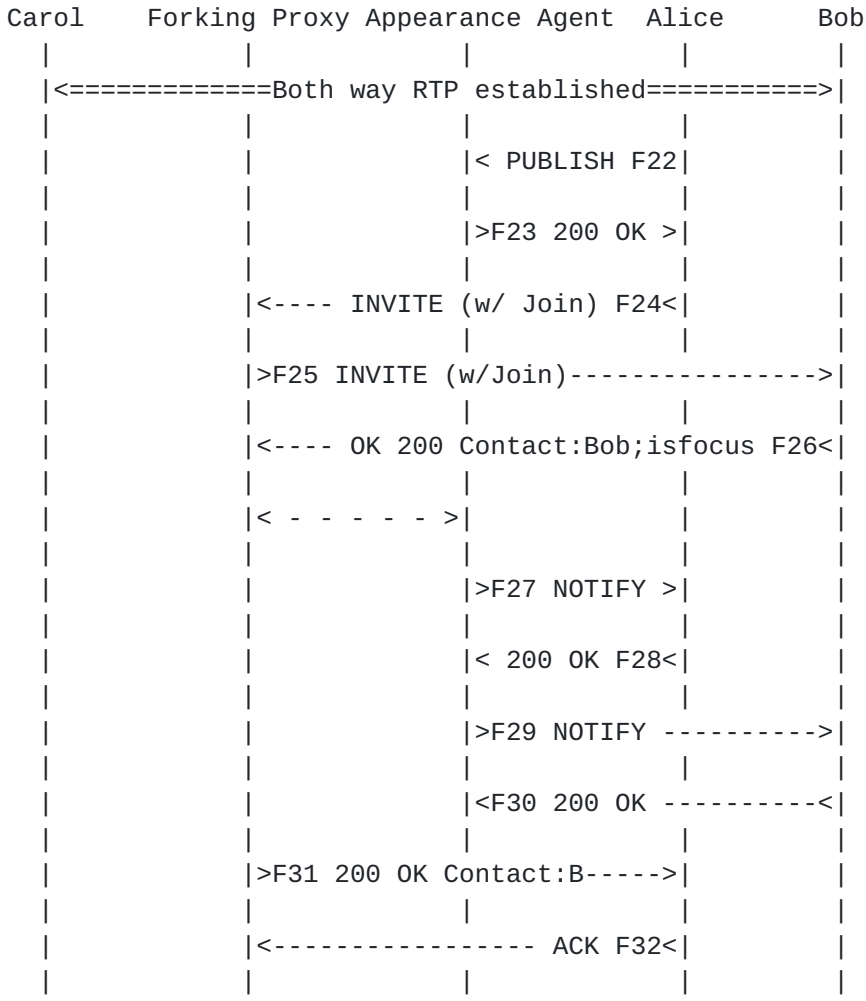
F32 Bob ----> Appearance Agent

```
PUBLISH sip:HelpDesk@example.com SIP/2.0
Via: SIP/2.0/UDP ua2.example.com;branch=z9hG4bKa5d6cf61F5FBC05A
From: <sip:bob@example.com>;tag=44150CC6-A7B7919D
To: <sip:HelpDesk@example.com>;tag=428765950880801
CSeq: 11 PUBLISH
Call-ID: 44fwF144-F12893K38424
Contact: <sip:bob@ua2.example.com>
Event: dialog;shared
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="full"
             entity="sip:HelpDesk@example.com">
   <dialog id="id3d4f9c83"
           call-id="b3cbd0-ad2c5775e-5df9f8d5"
           local-tag="3153DE7C-928203B"
           direction="initiator">
```

```
            <sa:exclusive>true</sa:exclusive>
            <state>trying</state>
            <local>
              <target uri="sip:bob@ua2.example.com">
              </target>
            </local>
            <remote>
              <identity>sip:HelpDesk@example.com</identity>
              <target uri="sip:alice@ua1.example.com" />
            </remote>
        </dialog>
    </dialog-info>
```

11.10. Joining or Bridging an Appearance

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].



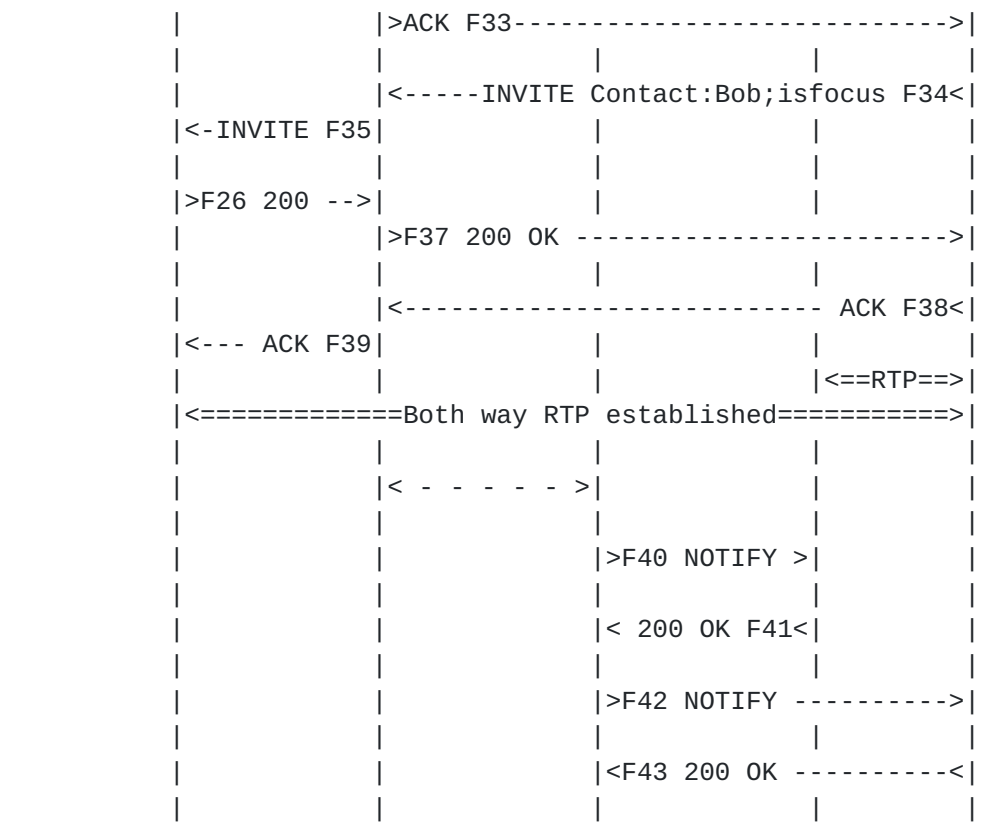


Figure 10.

F22 Alice ----> Appearance Agent

```
PUBLISH sip:HelpDesk@example.com SIP/2.0
Via: SIP/2.0/UDP ua2.example.com;branch=z9hG4bKa5d6cf61F5FBC05A
From: <sip:alice@example.com>;tag=44150CC6-A7B7919D
To: <sip:HelpDesk@example.com>;tag=428765950880801
CSeq: 11 PUBLISH
Call-ID: 87837Fkw87asfds
Contact: <sip:alice@ua2.example.com>
Event: dialog;shared
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="full"
             entity="sip:HelpDesk@example.com:5060">
   <dialog id="id3d4f9c83"
       call-id="dc95da63-60db1abd-d5a74b48"
```

```
          local-tag="605AD957-1F6305C2" >
            <sa:appearance>1</sa:appearance>
            <sa:exclusive>false</sa:exclusive>
            <sa:joined-dialog
              call-id="14-1541707345"
              from-tag="44BAD75D-E3128D42"
              to-tag="d3b06488-1dd1-11b2-88c5-b03162323164+d3e48f4c" />
            <state>trying</state>
            <local>
              <target uri="sip:alice@ua1.example.com">
              </target>
            </local>
            <remote>
                <target uri="sip:bob@example.com" />
            </remote>
        </dialog>
    </dialog-info>
```

F24 Alice ----> Proxy

```
INVITE sip:bob@ua.example.com SIP/2.0
Via: SIP/2.0/UDP ua1.example.com;branch=z9hG4bKcc9d727c2C29BE31
From: <sip:HelpDesk@example.com>;tag=605AD957-1F6305C2
To: <sip:bob@ua.example.com>
CSeq: 2 INVITE
Call-ID: dc95da63-60db1abd-d5a74b48
Contact: <sip:alice@ua1.example.com>
<all-one-line>
Join: 14-1541707345;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
```

11.11. Appearance Allocation - Loss of Appearance

Bob reserves an appearance with a PUBLISH, sends an INVITE to Carol, then becomes unreachable. When he fails to refresh his publication to the appearance agent, the Appearance Agent declares the dialog terminated and frees up the appearance using NOTIFYs F14 and F16. After retransmitting the NOTIFY to Bob (in not shown messages F17, F18, etc.), the subscription is terminated.

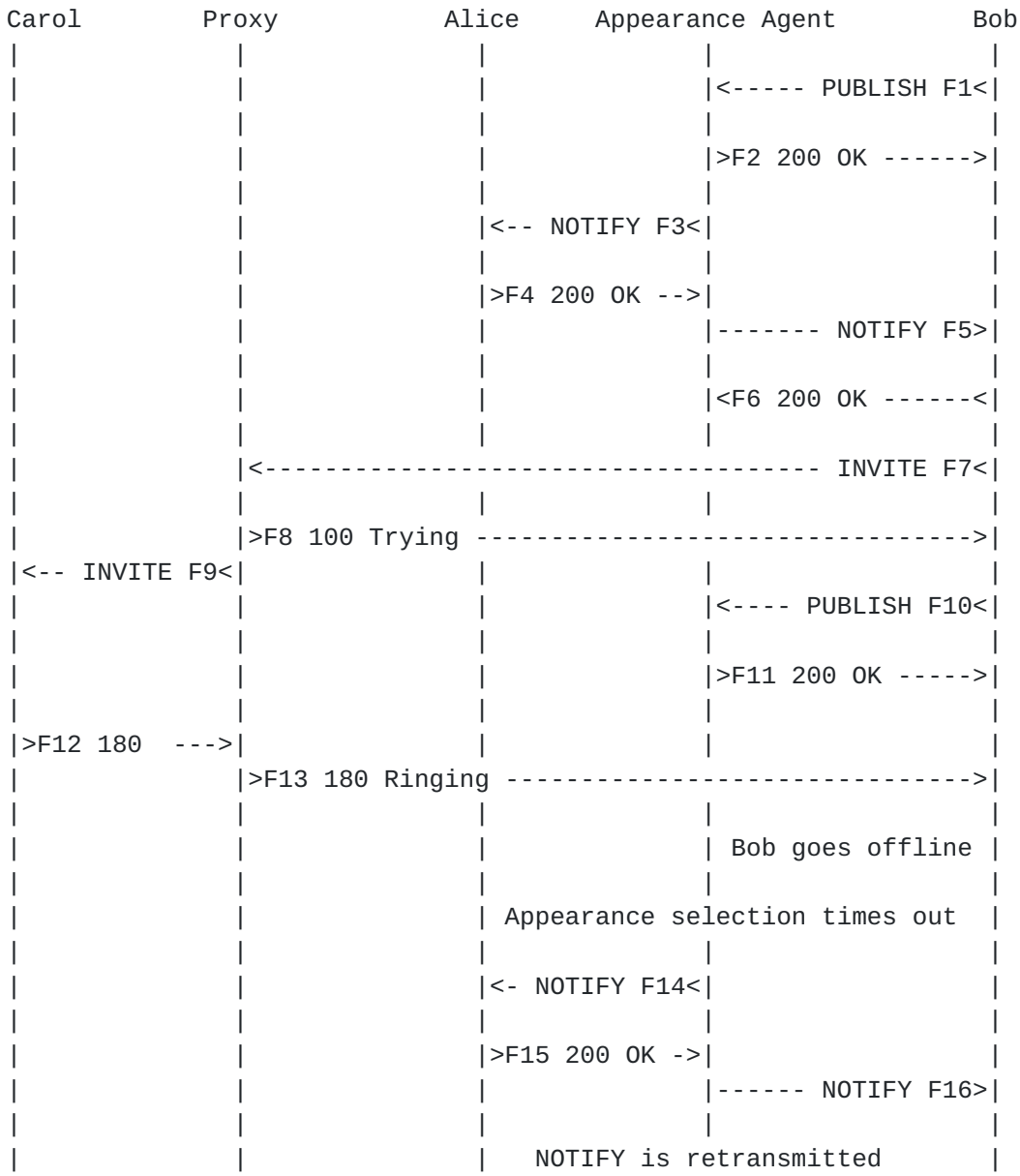


Figure 11.

11.12. Appearance Seizure Contention Race Condition

Bob and Alice both try to reserve appearance 2 by publishing at the same time. The Appearance Agent allocates the appearance to Bob by sending a 200 OK and denies it to Alice by sending a 400 response. After the NOTIFY F5, Alice learns that Bob is using appearance 2. Alice then attempts to reserve appearance 3 by publishing, which is then accepted.

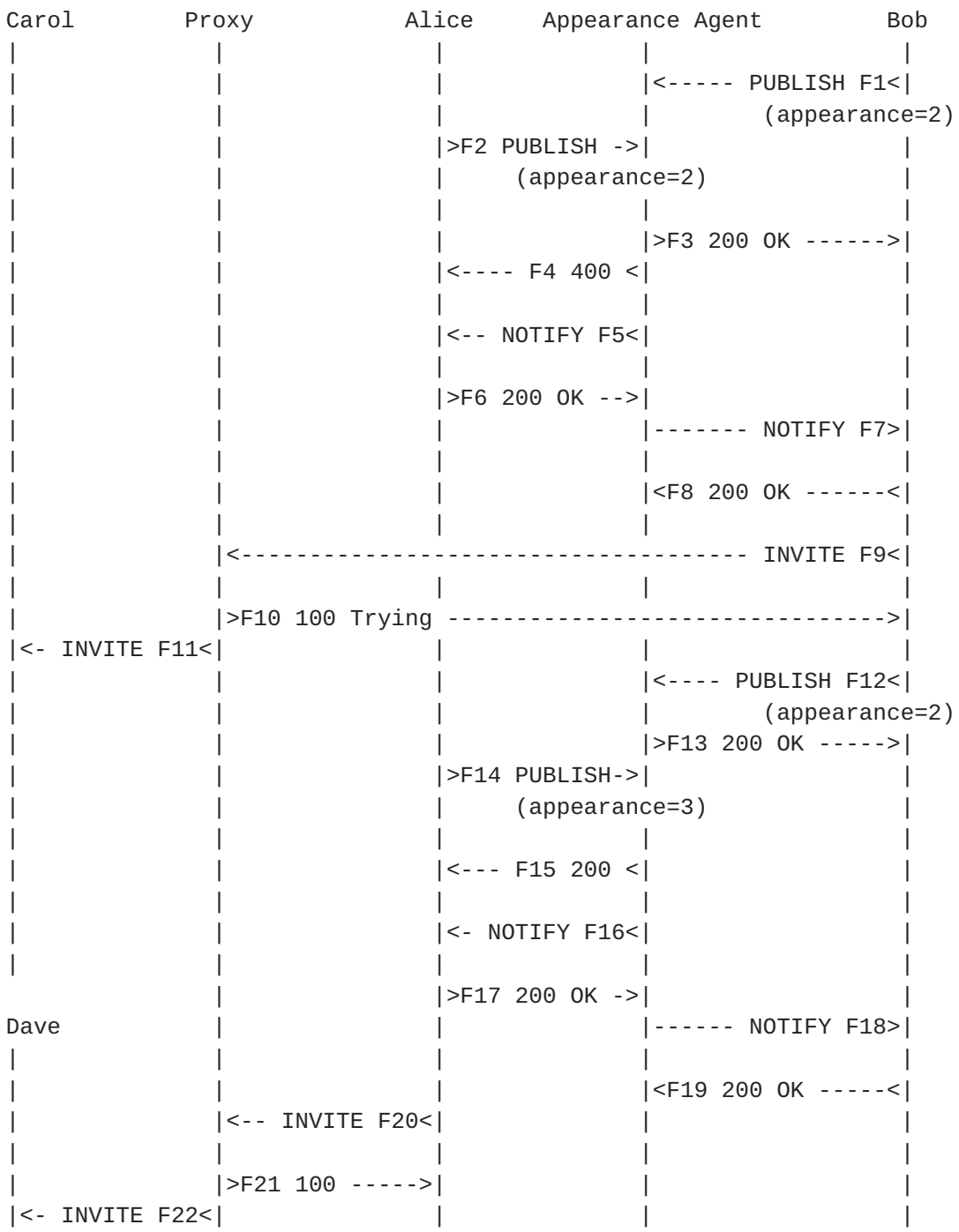
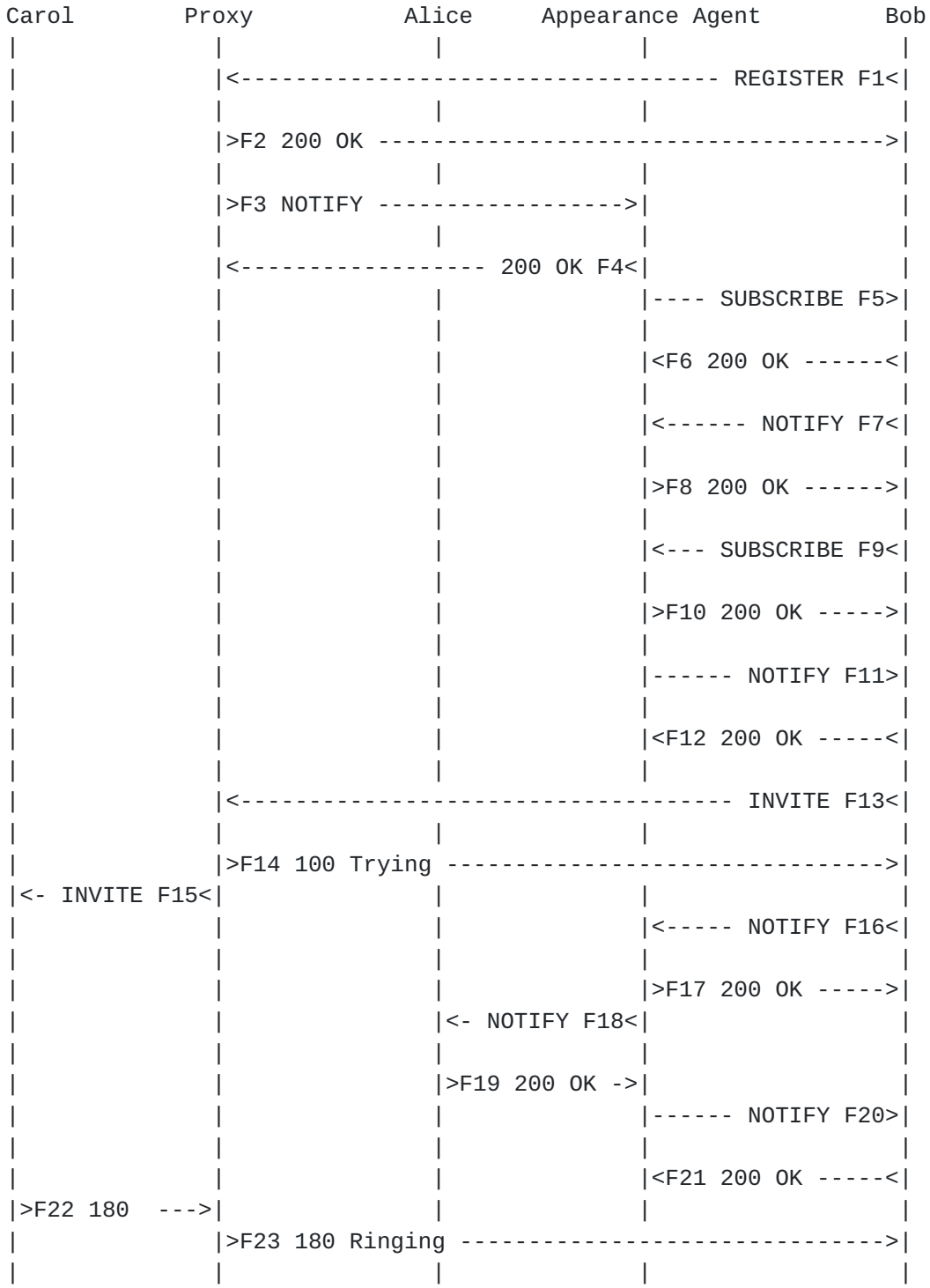


Figure 12.

11.13. Appearance Agent Subscription to UAs

In this scenario, the Appearance Agent does not have any way of knowing Bob's dialog state information, except through Bob. This could be because the Appearance Agent is not part of a B2BUA, or

perhaps Bob is remotely registering. When Bob registers, the Appearance Agent receives a registration event package notification from the registrar. The Appearance Agent then SUBSCRIBEs to Bob's dialog event state using Event:dialog in the SUBSCRIBE. Whenever Bob's dialog state changes, Bob's UA sends a NOTIFY to the Appearance Agent which then notifies the other other UAs in the group.



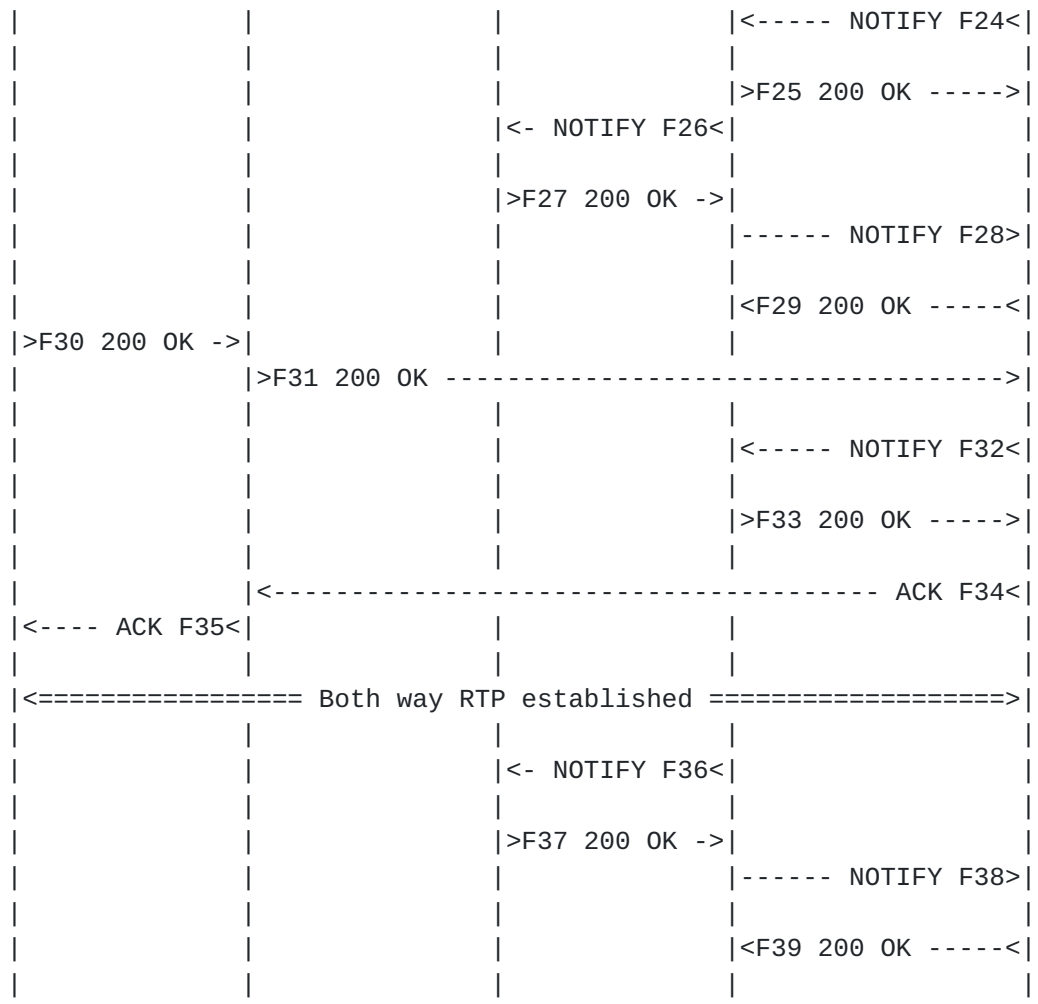
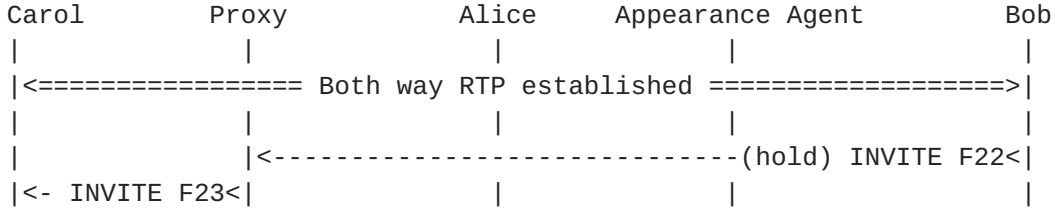
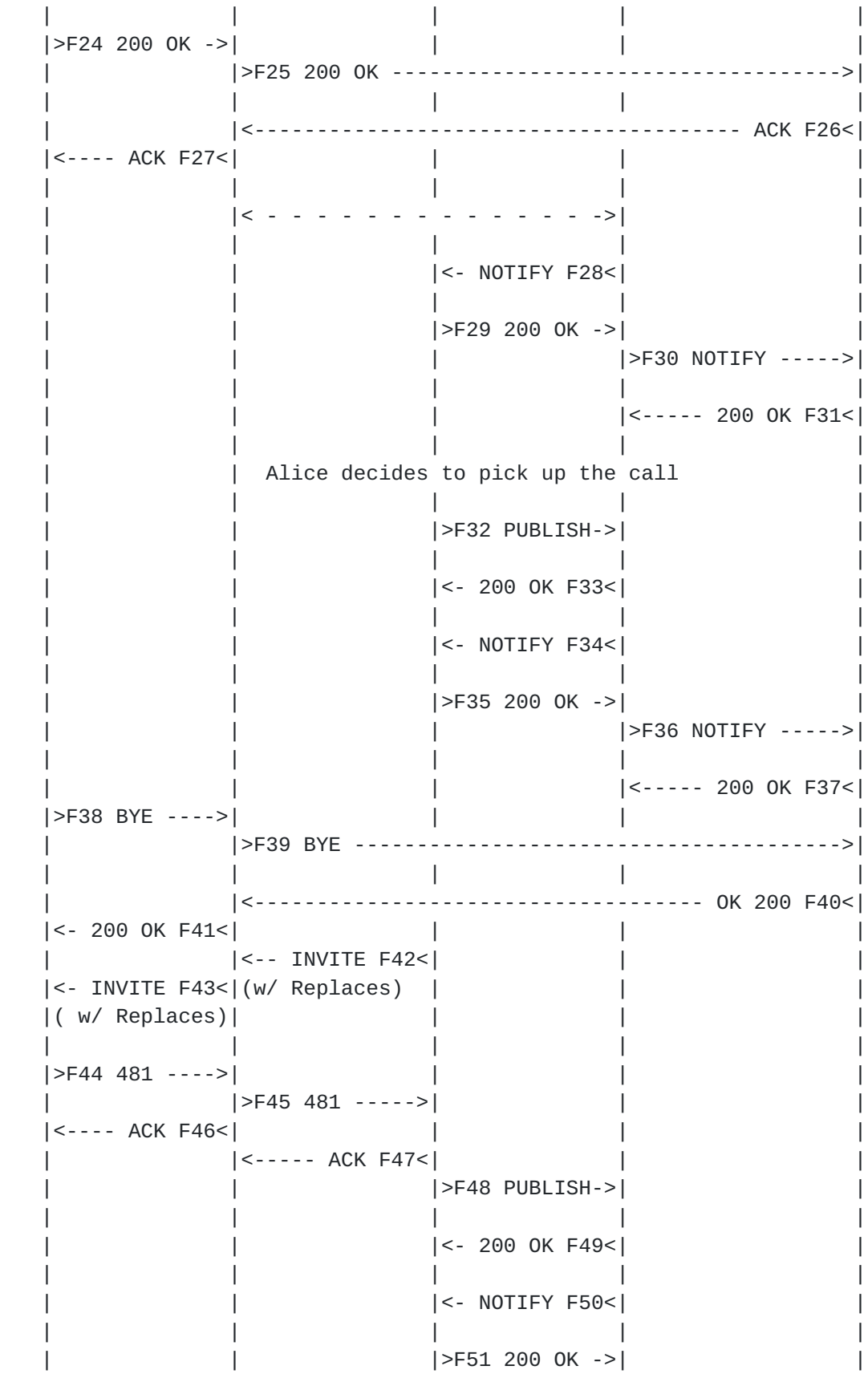


Figure 13.

11.14. Appearance Pickup Race Condition Failure

In this scenario, Bob has an established dialog with Carol created using the call flows of Figure 1 or Figure 2. Bob then places Carol on hold. Alice receives a notification of this and renders this on Alice's UI. Alice attempts to pick up the call but Carol hangs up before the pickup can complete. Alice cancels the pickup attempt with the PUBLISH F48. Note that the call flow for a failed Join would be almost identical.




|>F24 200 OK ->|
|>F25 200 OK ----------->|
|<----------- ACK F26<|
|<---- ACK F27<|
|< - - - - - - - - - - ->|
|<- NOTIFY F28<|
|>F29 200 OK ->|
|>F30 NOTIFY ----->|
|<----- 200 OK F31<|
Alice decides to pick up the call
|>F32 PUBLISH->|
|<- 200 OK F33<|
|<- NOTIFY F34<|
|>F35 200 OK ->|
|>F36 NOTIFY ----->|
|<----- 200 OK F37<|
|>F38 BYE ---->|
|>F39 BYE ----------->|
|<----------- OK 200 F40<|
|<- 200 OK F41<|
|<-- INVITE F42<|
|<- INVITE F43<|(w/ Replaces)
|(w/ Replaces)|
|>F44 481 ---->|
|>F45 481 ----->|
|<---- ACK F46<|
|<----- ACK F47<|
|>F48 PUBLISH->|
|<- 200 OK F49<|
|<- NOTIFY F50<|
|>F51 200 OK ->|

```
   |              |               |              |>F52 NOTIFY ----->|
   |              |               |              |                  |
   |              |               |              |<----- 200 OK F53<|
```

Figure 14.

F48 Alice ----> Appearance Agent

```
PUBLISH sip:HelpDesk@example.com SIP/2.0
Via: SIP/2.0/UDP ua2.example.com;branch=z9hG4bKa5d6cf61F5FBC05A
From: <sip:alice@example.com>;tag=44150CC6-A7B7919D
To: <sip:HelpDesk@example.com>;tag=428765950880801
CSeq: 11 PUBLISH
Call-ID: 87837Fkw87asfds
Contact: <sip:alice@ua2.example.com>
Event: dialog;shared
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="full"
             entity="sip:HelpDesk@example.com">
   <dialog id="id3d4f9c83"
       call-id="dc95da63-60db1abd-d5a74b48"
       local-tag="605AD957-1F6305C2" >
         <sa:appearance>1</sa:appearance>
         <sa:exclusive>false</sa:exclusive>
         <sa:replaced-dialog
           call-id="14-1541707345"
           from-tag="44BAD75D-E3128D42"
           to-tag="d3b06488-1dd1-11b2-88c5-b03162323164+d3e48f4c" />
         <state>terminated</state>
         <local>
           <target uri="sip:alice@ua1.example.com">
           </target>
         </local>
         <remote>
             <target uri="sip:carol@ua3.example.com" />
         </remote>
    </dialog>
</dialog-info>
```

11.15. Appearance Seizure Incoming/Outgoing Contention Race Condition

Alice tries to seize appearance 2 at the same time appearance 2 is allocated to an incoming call. The Appearance Agent resolves the conflict by sending a 400 to Alice. After the NOTIFY F6, Alice learns that the incoming call is using appearance 2. Alice republishes for appearance 3, which is accepted. Note that this example shows the INVITE being received before the NOTIFY from the Appearance Agent.

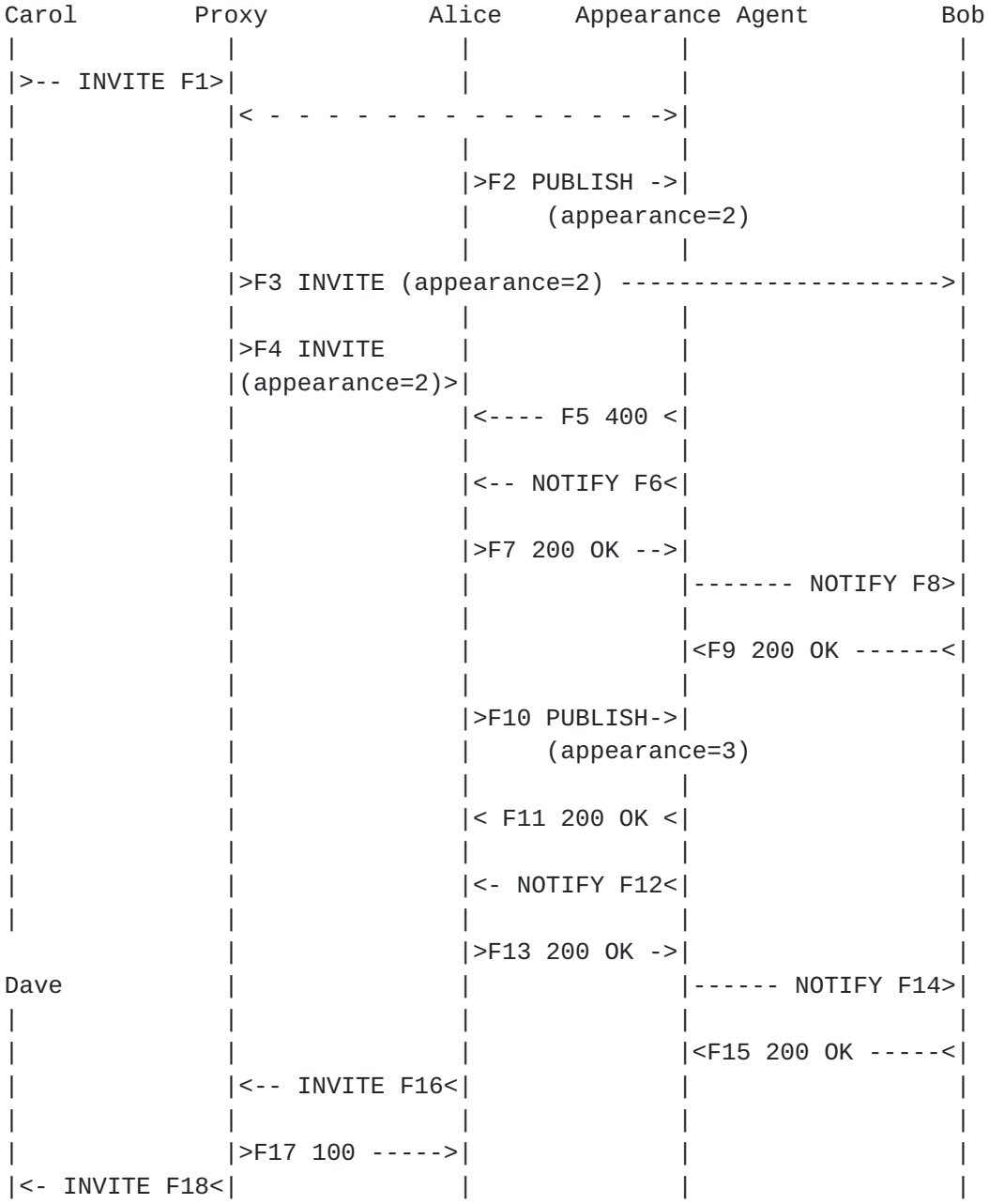


Figure 15.

12. Security Considerations

Since multiple line appearance features are implemented using semantics provided by [RFC3261], Event Package for Dialog State as define in , and Event Notification [I-D.ietf-sipcore-rfc3265bis], [RFC3903], security considerations in these documents apply to this draft as well.

NOTIFY or PUBLISH message bodies that provide the dialog state information and the dialog identifiers MAY be encrypted end-to-end using the standard mechanisms. All SUBSCRIBES and PUBLISHES between the UAs and the Appearance Agent MUST be authenticated.

All INVITEs with Replaces or Join header fields MUST only be accepted if the peer requesting dialog replacement or joining has been properly authenticated using a standard SIP mechanism (such as Digest or S/MIME), and authorized to request a replacement.

For an emergency call, a UA MUST never wait for a confirmed seizure of an appearance before sending an INVITE. Instead, the emergency call MUST proceed regardless of the status of the PUBLISH transaction.

13. IANA Considerations

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

13.1. SIP Event Package Parameter: shared

This specification defines a new event parameter 'shared' for the Dialog Package. When used in a NOTIFY, it indicates that the notifier supports the shared appearance feature. When used in a PUBLISH, it indicates that the publisher has explicit appearance information contained in the message body. If not present in a PUBLISH, the Appearance Agent MAY assign an appearance number to any new dialogs in the message body.

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

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.b.johnston@gmail.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

13.3. XML Schema Registration

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.b.johnston@gmail.com>

The XML for this schema can be found in Section 6.

14. Acknowledgements

The following individuals were part of the shared appearance 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, Martin Dolly, and Brett Tate for their comments.

Thanks to Carolyn Beeton, Francois Audet, Andy Hutton, Tim Ross, Raji Chinnappa, and Harsh Mendiratta for their detailed review of the document.

15. References

15.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[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.

[RFC3515] Sparks, R., "The Session Initiation Protocol (SIP) Refer Method", RFC 3515, April 2003.

[I-D.ietf-sipcore-rfc3265bis] Roach, A., "SIP-Specific Event Notification", draft-ietf-sipcore-rfc3265bis-09 (work in progress),

April 2012.

[RFC3903] Niemi, A., "Session Initiation Protocol (SIP) Extension for Event State Publication", RFC 3903, October 2004.

[RFC3891] Mahy, R., Biggs, B., and R. Dean, "The Session Initiation Protocol (SIP) "Replaces" Header", RFC 3891, September 2004.

[RFC4235] Rosenberg, J., Schulzrinne, H., and R. Mahy, "An INVITE-Initiated Dialog Event Package for the Session Initiation Protocol (SIP)", RFC 4235, November 2005.

[RFC3911] Mahy, R. and D. Petrie, "The Session Initiation Protocol (SIP) "Join" Header", RFC 3911, October 2004.

[RFC3840] Rosenberg, J., Schulzrinne, H., and P. Kyzivat, "Indicating User Agent Capabilities in the Session Initiation Protocol (SIP)", RFC 3840, August 2004.

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, January 2004.

[I-D.ietf-salud-alert-info-urns] Liess, L., Jesske, R., Johnston, A., Worley, D., and P. Kyzivat, "Alert-Info URNs for the Session Initiation Protocol (SIP)", draft-ietf-salud-alert-info-urns-06 (work in progress), April 2012.

15.2. Informative References

[RFC5359] Johnston, A., Sparks, R., Cunningham, C., Donovan, S., and K. Summers, "Session Initiation Protocol Service Examples", BCP 144, RFC 5359, October 2008.

[RFC4579] Johnston, A. and O. Levin, "Session Initiation Protocol (SIP) Call Control - Conferencing for User Agents", BCP 119, RFC 4579, August 2006.

[RFC3680] Rosenberg, J., "A Session Initiation Protocol (SIP) Event Package for Registrations", RFC 3680, March 2004.

[I-D.worley-service-example] Worley, D., "Session Initiation Protocol Service Example -- Music on Hold", draft-worley-service-example-09 (work in progress), February 2012.

Authors' Addresses

Alan Johnston (editor)
Avaya
St. Louis, MO 63124

Email: alan.b.johnston@gmail.com

Mohsen Soroushnejad
Sylantro Systems Corp

Email: mohsen.soroush@sylantro.com

Venkatesh Venkataramanan
Sylantro Systems Corp

Email: vvenkatar@gmail.com