Network Working Group
Request for Comments: 4579
BCP: 119
Category: Best Current Practice

A. Johnston
Avaya
O. Levin
Microsoft Corporation

August 2006

Session Initiation Protocol (SIP) Call Control - Conferencing for User Agents

Status of This Memo

This document specifies an Internet Best Current Practices for the Internet Community, and requests discussion and suggestions for improvements. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (2006).

Abstract

This specification defines conferencing call control features for the Session Initiation Protocol (SIP). This document builds on the Conferencing Requirements and Framework documents to define how a tightly coupled SIP conference works. The approach is explored from the perspective of different user agent (UA) types: conference-unaware, conference-aware, and focus UAs. The use of Uniform Resource Identifiers (URIs) in conferencing, OPTIONS for capabilities discovery, and call control using REFER are covered in detail with example call flow diagrams. The usage of the isfocus feature tag is defined.

Table of Contents

1. Introduction

This specification uses the concepts and definitions from the high level requirements [14] and the SIP conferencing framework [8] documents. This approach is applicable to tightly coupled SIP conferences. In this architecture, a user agent (UA), known as a participant, establishes a SIP dialog with another UA, known as a focus. The focus is the central point of control, authentication, and authorization. This specification defines the operation of a focus and participant UAs. Note that only the signalling (SIP) needs to be centralized in this model; the media can be centrally mixed, distributed, or even multicast. For a full discussion of this architecture, see the SIP conferencing framework document [8].

The approach described in this document implements key functions in the conferencing framework using SIP primitives only. This allows for conducting simple conferences with defined functionalities using SIP mechanisms and conventions. Many other advanced functions can be implemented using additional means, but they are not in the scope of this document.

This document presents the basic call control (dial-in and dial-out) conferencing building blocks from the UA perspective. Possible applications include ad-hoc conferences and scheduled conferences.

Note that a single conference can bridge participants that have different capabilities and who potentially have joined the conference by different means (i.e., dial-in, dial-out, scheduled, or ad-hoc).

The call control and dialog manipulation approach is based on the multiparty framework document [15]. That document defines the basic approach of service design adopted for SIP, which includes the following:

- Definition of primitives, not services
- Signaling model independent
- Invoker oriented
- Primitives make full use of URIs
- Include policies for authentication, authorization, logging, etc.
- Define graceful fallback to baseline SIP

The use of opaque URIs and the ability to communicate call control context information within a URI (as opposed to using service-related header fields), as discussed in RFC 3087 [11], is fundamental to this approach.

Capabilities discovery is an important feature of SIP systems, and conferencing systems can make use of such features. For a UA acting as a focus in a conference, this specification defines the usage of the 'isfocus' feature parameter.

2. Terminology

In this document, the key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" are to be interpreted as described in RFC 2119 and indicate requirement levels for compliant implementations [1].

3. SIP User Agent Conferencing Capability Types

From a conferencing perspective, the framework document outlines a number of possible different SIP components such as conference-unaware participant, conference-aware participant, and focus.

This document applies the concepts above to the SIP call control part of the conferencing components. It defines normative behavior of the SIP UAs in various conferencing situations (referred to later as "scenarios").

3.1. Focus UA

A focus, as defined in the framework, hosts a SIP conference and maintains a SIP signaling relationship with each participant in the conference. A focus contains a conference-aware user agent that supports the conferencing call control conventions as defined in this document.

A focus SHOULD support the conference package RFC 4575 [9], behave as a notifier for that package, and indicate its support in the Allow-Events header fields in requests and responses. A focus MAY include information about the conference in Session Description Protocol (SDP) bodies sent as part of normal SIP signaling by populating the Session Information, URI, Email Address, and Phone Number SDP fields.

In order to support advanced features, where a session established between two endpoints can migrate to a centralized conference, a focus SHOULD support the Replaces header field [6].

A user agent with focus capabilities could be implemented in end user equipment and would be used for the creation of ad-hoc conferences.

A dedicated conferencing server, whose primary task is to simultaneously host conferences of arbitrary type and size, may allocate and publish a conference factory URI (as defined in the next section) for creating an arbitrary number of ad-hoc conferences (and subsequently their focuses) using SIP call control means.

3.2. Conference Factory URI

According to the framework, there are many ways in which a conference can be created. A conferencing server implementation is free to choose from these methods, which include non-automated means (such as an Interactive Voice Response (IVR) system), SIP, or any conference control protocol.

In order to automatically create an arbitrary number of ad-hoc conferences (and subsequently their focuses) using SIP call control means, a globally routable Conference Factory URI can be allocated and published.

A successful attempt to establish a call to this URI would result in the automatic creation of a new conference and its focus. As a result, note that the Conference Factory URI and the newly created focus URI MAY resolve to different physical devices.

A scenario showing the use of the conference factory URI is shown in Section 5.4.

3.3. Conference-Unaware UA

The simplest user agent can participate in a conference ignoring all SIP conferencing-related information. The simplest user agent is able to dial in to a conference and to be invited to a conference. Any conferencing information is optionally conveyed to/from it using non-SIP means. Such a user agent would not usually host a conference (at least, not using SIP explicitly). A conference-unaware UA need only support RFC 3261 [2]. Call flows for conference-unaware UAs are not shown in general in this document as they would be identical to those in the SIP call flows document [13].

Note that the presence of an 'isfocus' feature tag in a Contact header field will not cause interoperability issues between a focus and a conference-unaware UA since it will be treated as an unknown header parameter and ignored, as per standard SIP behavior.

3.4. Conference-Aware UA

A conference-aware user agent supports SIP conferencing call control conventions defined in this document as a conference participant, in addition to support of RFC 3261 [2]. A conference-aware UA should be able to process SIP redirections such as described in Section 8.1.3.4 of RFC 3261.

A conference-aware UA MUST recognize the 'isfocus' feature parameter. A conference-aware UA SHOULD support REFER [4], SIP events [3], and the conferencing package [9].

A conference-aware UA SHOULD subscribe to the conference package if the 'isfocus' parameter is in the remote target URI of a dialog and if the conference package is listed by a focus in an Allow-Events header field. The SUBSCRIBE to the conference package SHOULD be sent outside any INVITE-initiated dialog. A termination of the INVITE dialog with a BYE does not necessarily terminate the SUBSCRIBE dialog.

A conference-aware UA MAY render to the user any information about the conference obtained from the SIP header fields and SDP fields from the focus.

A conference-aware UA SHOULD render to the user any information about the conference obtained from the SIP conference package.

4. Usage of the 'isfocus' Feature Parameter

4.1. General

The main design guidelines for the development of SIP extensions and conventions for conferencing are to define the minimum number of extensions and to have seamless backward compatibility with conference-unaware SIP UAs. The minimal requirement for SIP is being able to express that a dialog is a part of a certain conference referenced to by a URI. As a result of these extensions, it is possible to do the following using SIP:

- Create a conference
- Join a conference
- Invite a user to a conference
- Expel a user by third party
- Discover if a URI is a conference URI
- Delete a conference

The approach taken is to use the feature parameter 'isfocus' to express that a SIP dialog belongs to a conference. The use of feature parameters in Contact header fields to describe the characteristics and capabilities of a UA is described in the User Agent Capabilities document [5], which includes the definition of the 'isfocus' feature parameter.

4.2. Session Establishment

In session establishment, a focus MUST include the 'isfocus' feature parameter in the Contact header field unless the focus wishes to hide the fact that it is a focus. To a participant, the feature parameter will be associated with the remote target URI of the dialog. It is an indication to a conference-aware UA that the resulting dialog belongs to a conference, identified by the URI in the Contact header field, and that the call control conventions defined in this document can be applied.

By their nature, the conferences supported by this specification are centralized. Therefore, typically a conferencing system needs to allocate a SIP conference URI such that SIP requests to this URI are not forked and are routed to a dedicated conference focus. For example, a globally accessible SIP conference could be well constructed with a conference URI using a Globally Routable User Agent URI (GRUU) (defined in [16]), because of its ability to support the non-forking and global routability requirements.

4.3. Discovery

Using the mechanism described in this section, it is possible, given an opaque URI, to determine if it belongs to a certain conference (i.e., meaning that it is a conference URI) or not. This discovery function can be implemented in SIP using an OPTIONS request, and can be done either inside an active dialog or outside a dialog. A focus MUST include the 'isfocus' feature parameter in a 200 OK response to an OPTIONS unless the focus wishes to hide the fact that it is a focus.

5. SIP Conferencing Primitives

The SIP conferencing call control flows presented in this section are the call control building blocks for various SIP conferencing applications as described in the conferencing requirements [14] and framework [8] documents. The major design goal is that the same SIP conferencing primitives would be used by user agents having different conferencing capabilities and implementing different applications.

5.1. INVITE: Joining a Conference Using the Conference URI - Dial-In

In this section, a user knows the conference URI and "dials in" to join this conference. The focus will authenticate the participant and apply authorization policy before allowing the participant to join the conference.

If the UA is the first participant of the conference to dial-in, it is likely that this INVITE will activate the focus and hence the conference. However, the conference URI must have been reserved prior to its use.

If the conference is up and running already, the dialing-in participant is joined to the conference by its focus.

To join an existing specific conference, a UA will send an INVITE with the Request-URI set to the conference URI. The focus MUST include the 'isfocus' feature parameter in the Contact header field of the 200 OK response to the INVITE.

An example call flow for joining a conference is shown in Figure 1.

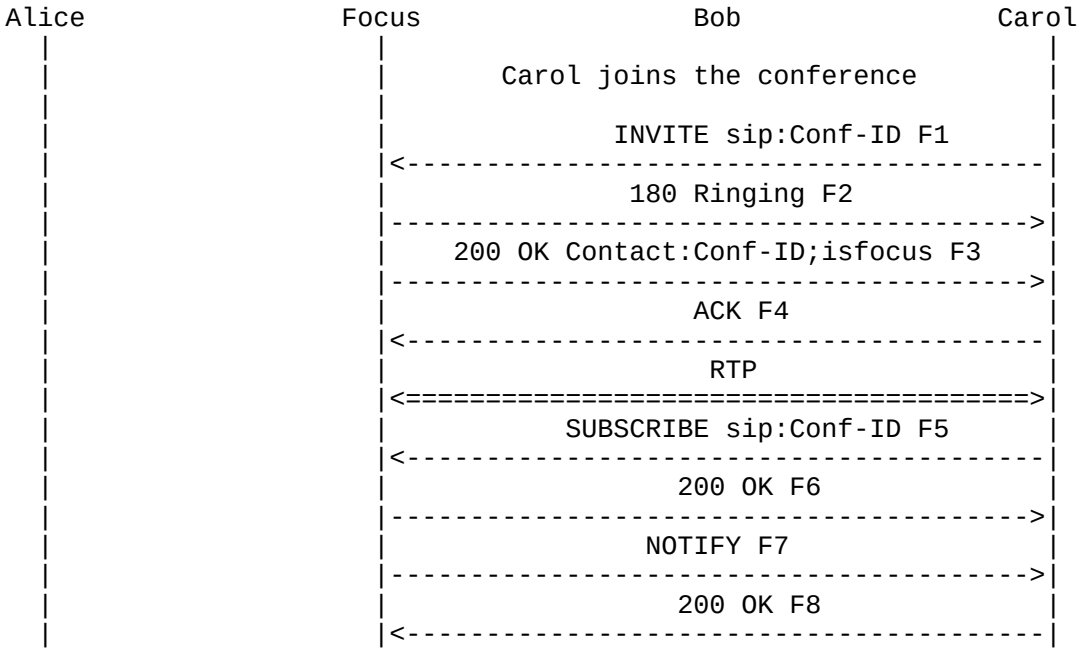


Figure 1. A Participant Joins a Conference Using the Conference URI.

```
F1    INVITE sip:3402934234@conf.example.com SIP/2.0
      Via: SIP/2.0/UDP client.chicago.example.com
       ;branch=z9hG4bKhjhs8ass83
      Max-Forwards: 70
      To: <sip:3402934234@conf.example.com>
      From: Carol <sip:carol@chicago.example.com>;tag=32331
      Call-ID: d432fa84b4c76e66710
      CSeq: 45 INVITE
      Contact: <sip:carol@client.chicago.example.com>
      Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER,
       SUBSCRIBE, NOTIFY
      Allow-Events: dialog
      Accept: application/sdp, message/sipfrag
      Supported: replaces
      Content-Type: application/sdp
      Content-Length: ...

      (SDP not shown)
```

```
   F3    SIP/2.0 200 OK
         Via: SIP/2.0/UDP client.chicago.example.com
          ;branch=z9hG4bKhjhs8ass83;received=192.0.2.4
         To: <sip:3402934234@conf.example.com>;tag=733413
         From: Carol <sip:carol@chicago.example.com>;tag=32331
         Call-ID: d432fa84b4c76e66710
         CSeq: 45 INVITE
         Contact: <sip:3402934234@conf.example.com>;isfocus
         Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER,
          SUBSCRIBE, NOTIFY
         Allow-Events: dialog, conference
         Accept: application/sdp, application/conference-info+xml,
          message/sipfrag
         Supported: replaces, join, gruu
         Content-Type: application/sdp
         Content-Length: ...

         v=0
         o=focus431 2890844526 2890842807 IN IP4 ms5.conf.example.com
         s=-
         i=Example Conference Hosted by Example.com
         u=http://conf.example.com/3402934234
         e=3402934234@conf-help.example.com
         p=+1-888-2934234
         c=IN IP4 ms5.conf.example.com
         t=0 0
         m=audio 49170 RTP/AVP 0
         m=video 51372 RTP/AVP 31


   F5    SUBSCRIBE sip:3402934234@conf.example.com SIP/2.0
         Via: SIP/2.0/UDP client.chicago.example.com
          ;branch=z9hG4bKdf334
         Max-Forwards: 70
         To: <sip:3402934234@conf.example.com>
         From: Carol <sip:carol@chicago.example.com>;tag=43524545
         Call-ID: k3l43id034kssereree
         CSeq: 22 SUBSCRIBE
         Contact: <sip:carol@client.chicago.example.com>
         Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER,
          SUBSCRIBE, NOTIFY
         Event: conference
         Accept: application/conference-info+xml
         Supported: replaces
         Content-Length: 0
```

```
F7    NOTIFY sip:carol@chicago.example.com SIP/2.0
      Via: SIP/2.0/UDP ms5.conf.example.com;branch=z9hG4bK3343d1
      Max-Forwards: 70
      To: Carol <sip:carol@chicago.example.com>;tag=43524545
      From: <sip:3402934234@conf.example.com>;tag=a3343df32
      Call-ID: k3l43id034kshereree
      CSeq: 34321 NOTIFY
      Contact: <sip:3402934234@conf.example.com>;isfocus
      Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER,
       SUBSCRIBE, NOTIFY
      Event: conference
      Accept: application/sdp, message/sipfrag
      Subscription-State: active;expires=3600
      Supported: replaces, join, gruu
      Content-Type: application/conference-info+xml

      Content-Length: ...

      <conference-info version="0" state="full"
       entity="sip:3402934234@conf.example.com">
       <conference-description>
        <conf-uris>
         <entry>
          <uri>tel:+18882934234</uri>
         </entry>
        </conf-uris>
       </conference-description>
       <users>
        <user entity="sip:carol@chicago.example.com" state="full">
         <display-text>Carol</display-text>
         <endpoint entity="sip:carol@client.chicago.example.com">
          <status>connected</status>
          <joining-method>dialed-in</joining-method>
          <media id="1">
           <display-text>Main Audio</display-text>
           <type>audio</type>
           <src-id>583398</src-id>
           <status>sendrecv</status>
          </media>
          <media id="2">
           <type>video</type>
           <src-id>345212</src-id>
           <status>sendrecv</status>
          </media>
         </endpoint>
        </user>
       </users>
      </conference-info>
```

5.2. INVITE: Adding a Participant by the Focus - Dial-Out

To directly add a participant to a conference, a focus SHOULD send an INVITE to the participant containing a Contact header field with the conference URI and the 'isfocus' feature parameter.

Note that a conference-unaware UA would simply ignore the conferencing information and treat the session (from a SIP perspective) as a point-to-point session. This is because standard RFC 3261 [2] behavior is to ignore unknown header parameters such as 'isfocus'.

An example call flow is shown in Figure 2. It is assumed that Alice is already a participant of the conference. The focus invites Carol to the conference by sending an INVITE. After the session is established, Carol subscribes to the conference URI. It is important to note that there is no dependency on Carol's SUBSCRIBE (F5) and the NOTIFY to Alice (F9) -- they occur asynchronously and independently.

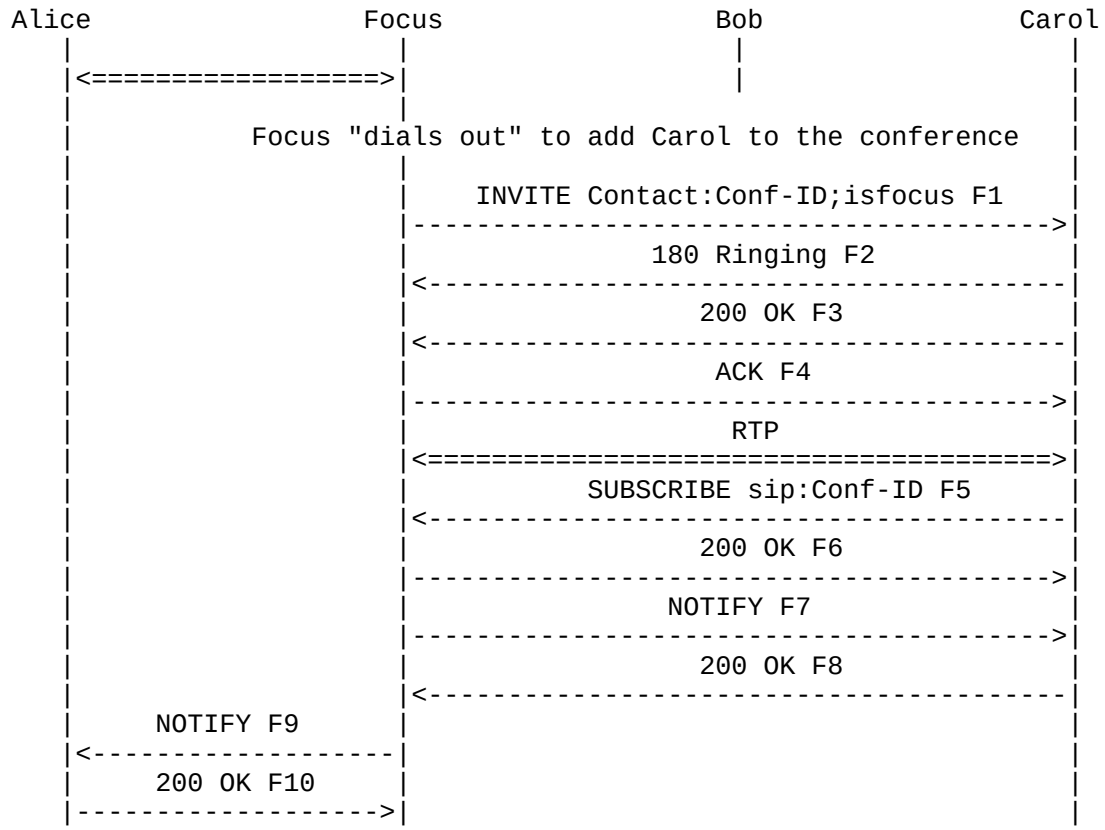


Figure 2. A Focus "Dials Out" to Add a Participant to the Conference.

```
F7    NOTIFY sip:carol@chicago.example.com SIP/2.0
      Via: SIP/2.0/UDP ms5.conf.example.com;branch=z9hG4bK3343d1
      Max-Forwards: 70
      To: Carol <sip:carol@chicago.example.com>;tag=43524545
      From: <sip:3402934234@conf.example.com>;tag=a3343df32
      Call-ID: k3l43id034kshereree
      CSeq: 34321 NOTIFY
      Contact: <sip:3402934234@conf.example.com>;isfocus
      Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER,
       SUBSCRIBE, NOTIFY
      Event: conference
      Accept: application/sdp, message/sipfrag
      Subscription-State: active;expires=3600
      Supported: replaces, gruu
      Content-Type: application/conference-info+xml
      Content-Length: ...
```

```
<conference-info version="0" state="full"
 entity="sip:3402934234@conf.example.com">
 <conference-description>
  <conf-uris>
   <entry>
    <uri>tel:+18882934234</uri>
   </entry>
  </conf-uris>
 </conference-description>
 <users>
  <user entity="sip:alice@atlanta.example.com" state="full">
   <display-text>Alice</display-text>
   <endpoint entity="sip:alice@client.atlanta.example.com">
    <status>connected</status>
    <joining-method>dialed-in</joining-method>
    <media id="3">
     <display-text>Main Audio</display-text>
     <type>audio</type>
     <src-id>647231</src-id>
     <status>sendrecv</status>
    </media>
    <media id="4">
     <type>video</type>
     <src-id>21345</src-id>
     <status>sendrecv</status>
    </media>
   </endpoint>
  </user>
  <user entity="sip:carol@chicago.example.com" state="full">
   <display-text>Carol</display-text>
   <endpoint entity="sip:carol@client.chicago.example.com">
    <status>connected</status>
    <joining-method>dialed-out</joining-method>
    <media id="1">
     <display-text>Main Audio</display-text>
     <type>audio</type>
     <src-id>583398</src-id>
     <status>sendrecv</status>
    </media>
    <media id="2">
     <type>video</type>
     <src-id>345212</src-id>
     <status>sendrecv</status>
    </media>
   </endpoint>
  </user>
 </users>
</conference-info>
```

```
   F9    NOTIFY sip:alice@atlanta.example.com SIP/2.0
         Via: SIP/2.0/UDP ms5.conf.example.com;branch=z9hG4bK3432
         Max-Forwards: 70
         To: Alice <sip:alice@atlanta.example.com>;tag=43524545
         From: <sip:3402934234@conf.example.com>;tag=a3343df32
         Call-ID: 8820450524545
         CSeq: 998 NOTIFY
         Contact: <sip:3402934234@conf.example.com>;isfocus
         Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER,
          SUBSCRIBE, NOTIFY
         Event: conference
         Accept: application/sdp, message/sipfrag
         Subscription-State: active;expires=2450
         Supported: replaces, gruu
         Content-Type: application/conference-info+xml
         Content-Length: ...

         <conference-info version="1" state="partial"
          entity="sip:3402934234@conf.example.com">
          <users>
           <user entity="sip:carol@chicago.example.com" state="full">
            <display-text>Carol</display-text>
            <endpoint entity="sip:carol@client.chicago.example.com">
             <status>connected</status>
             <joining-method>dialed-out</joining-method>
             <media id="1">
              <display-text>Main Audio</display-text>
              <type>audio</type>
              <src-id>583398</src-id>
              <status>sendrecv</status>
             </media>
             <media id="2">
              <type>video</type>
              <src-id>345212</src-id>
              <status>sendrecv</status>
             </media>
            </endpoint>
           </user>
          </users>
         </conference-info>
```

5.3. INVITE: Manually Creating a Conference by Dialing in to a Conferencing Application

In this section, a user sends an INVITE to a conference server application. The application (such as an IVR system or a web page) is implemented because the system requires additional input from the user before it is able to create a conference. After a normal dialog is established, additional information is received and the conference together with its focus are created. Since the UA is now in a dialog with a focus, the focus will re-INVITE the user with the conference URI in Contact with the 'isfocus' feature parameter.

Alternatively, the additional information can be provided by the user during an early dialog (see RFC 3261 [2] for a discussion of early dialogs in SIP). This could be accomplished by a 183 Session Progress response sent by the conferencing application. After the conference is created, the conference URI would then be returned in a Contact in the 200 OK.

Note that since this flow is all about human interaction with a conferencing application, any errors and failures will be returned to the human (recorded announcements, error tones, etc.).

As discussed in the conferencing framework, the conference URI must be unique across all distinct conferences within the same domain. In general, the user part of a conference URI will contain a pseudo random string.

An example call flow is shown in Figure 3. In this example, Alice uses a conference application that is triggered when Alice sends an INVITE to the conference application. In this example, Conf-App is used to represent the conference application URI. Alice's conference-aware UA learns of the existence of the conference from the 'isfocus' feature parameter and subscribes to the conference package to receive notifications of the conference state.

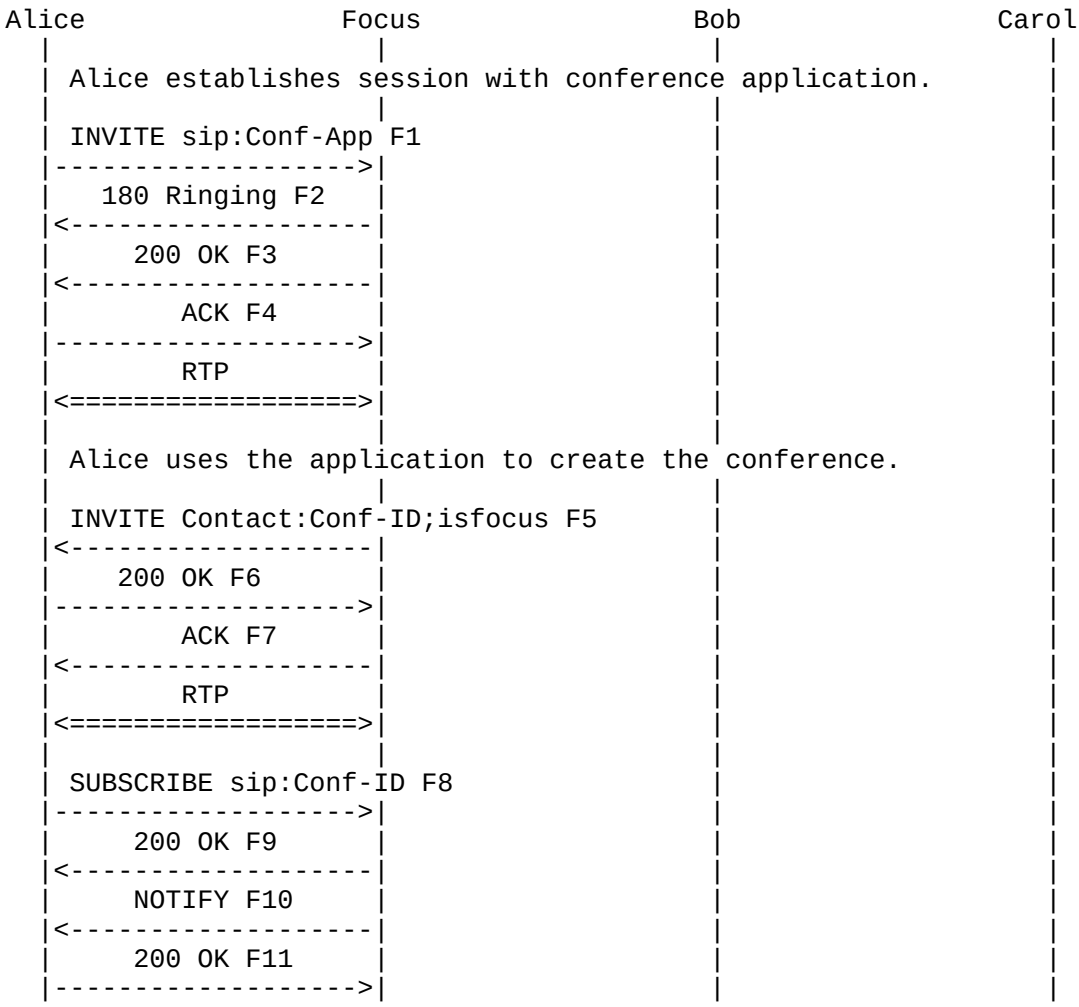


Figure 3. A Participant Creates a Conference Using an Application.

5.4. INVITE: Creating a Conference Using Ad-Hoc SIP Methods

This section addresses creating a conference by using ad-hoc SIP means. The conference factory URI (as defined in Section 3.2) is used to automatically create the conference in this example. This is different from the previous scenario in that no human intervention is required -- an automaton can create the conference and add participants. Since the conference does not need to be scheduled or reserved, but is created "on the fly", it is an "ad-hoc" conference creation.

The benefit of this approach is that the conference URI need not be known to the user; instead it is created by a focus and used by the participants' UAs. The main difference between this scenario and Section 5.3 is that no user intervention (IVR, web page form, etc.) is required to create the conference.

The SIP URI of the conference factory can be provisioned in the UA (as in a "create new conference" button on a SIP phone) or can be discovered using other means.

A SIP entity (such as conferencing server) can distinguish this INVITE request as a request to create a new ad-hoc conference from a request to join an existing conference by the Request-URI. That is, although both requests may route to the same application, the differing services requested can be identified by the differing URIs in the request itself.

Assuming that all security and policy requirements have been met, a new conference will be created with the Contact URI returned in the 200 OK being the conference URI. The Contact header field MUST contain the 'isfocus' feature parameter to indicate that this URI is for a conference.

An example call flow is shown in Figure 4. Note that Conf-Factory is shorthand for the conference factory URI and Conf-ID Is short for the conference URI. In this flow, Alice has a conference-aware UA and creates a conference by sending an INVITE to the conference factory URI. The conference factory application creates the conference and redirects Alice to the focus using a 302 Moved Temporarily response. Note that with proxy recursion as part of normal RFC 3261 [2] behavior, Alice may never see the redirect but may just receive the responses from the focus starting with message F5. Once the media session is established, Alice subscribes to the conference URI obtained through the Contact in the 200 OK response from the focus.

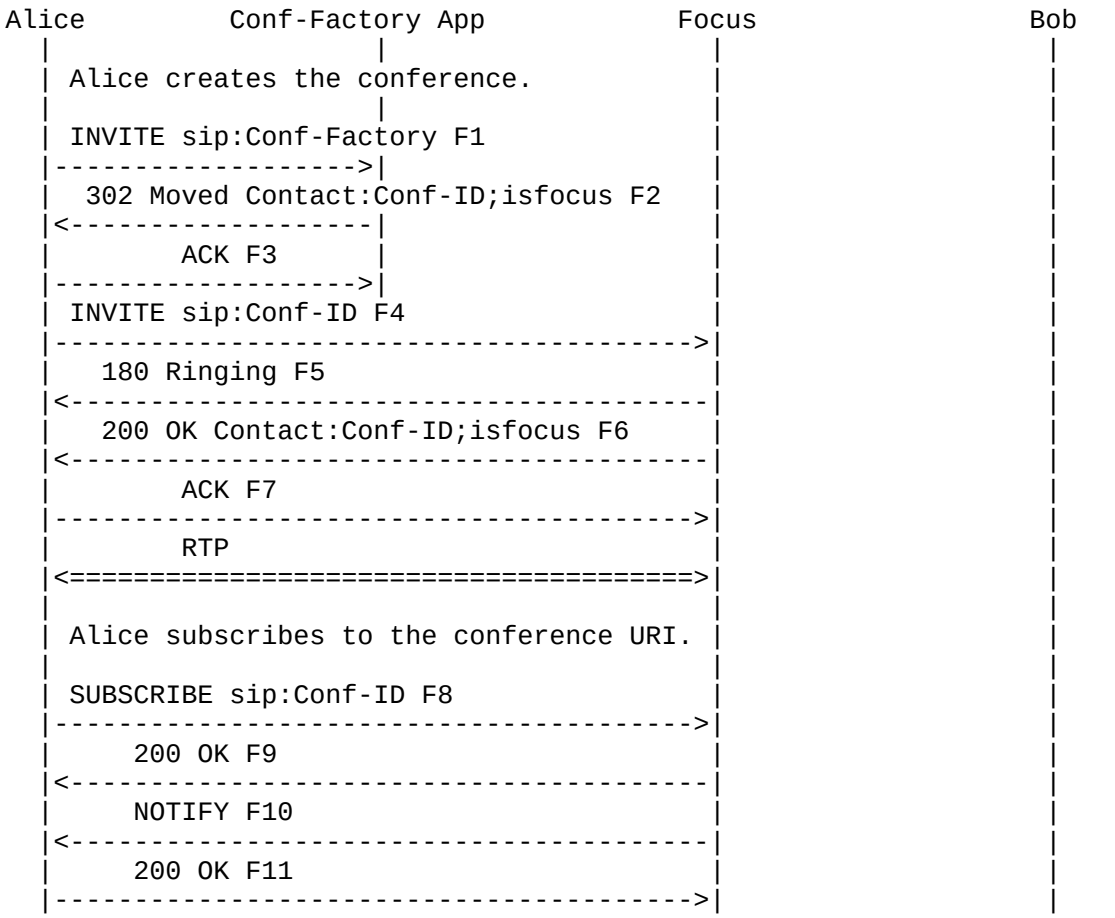


Figure 4. Creation of a Conference Using SIP Ad-Hoc Methods.

5.5. REFER: Requesting a Focus to Add a New Resource to a Conference (Dial Out to a New Participant)

A SIP conference URI can be used to inject different kinds of information into the conference. Examples include new participants, new real-time media sources, new IM messages, and pointers to passive information references (such as HTTP URIs).

To request that the focus add a new information resource to the specified conference, any SIP UA can send a REFER to the conference URI with a Refer-To containing the URI of the new resource. Since this REFER is sent to the conference URI and not the conference factory URI, the semantics to the focus are to bring the resource into the conference and make it visible to the conference

participants. The resultant focus procedures are dependent both on the nature of the new resource (as expressed by its URI) and the policy of the focus regarding IM, central vs. distributed real-time media processing, and so on.

The scenario for adding a new UA participant is important to support because it works even if the new participant does not support REFER and transfer call control -- only the requesting participant and the focus need to support the REFER and transfer call control.

Upon receipt of the REFER containing a Refer-To header with a SIP URI, the focus SHOULD send an INVITE to the new participant identified by the Refer-To SIP URI containing a Contact header field with the conference URI and the 'isfocus' feature parameter.

A conference-unaware UA would simply ignore the conferencing information and treat the session (from a SIP perspective) as a point-to-point session.

An example call flow is shown in Figure 5. While this flow shows the use of REFER to add a new participant to the conference, the mechanism can generally add a resource as identified by a URI to the conference. It is assumed that Alice is already a participant of the conference. Alice sends a REFER to the conference URI. The focus invites Carol to the conference by sending an INVITE. After the session is established, Carol subscribes to the conference URI. It is important to note that there is no dependency on Carol's SUBSCRIBE (F11) and the NOTIFY to Alice (F15) -- they occur asynchronously and independently.

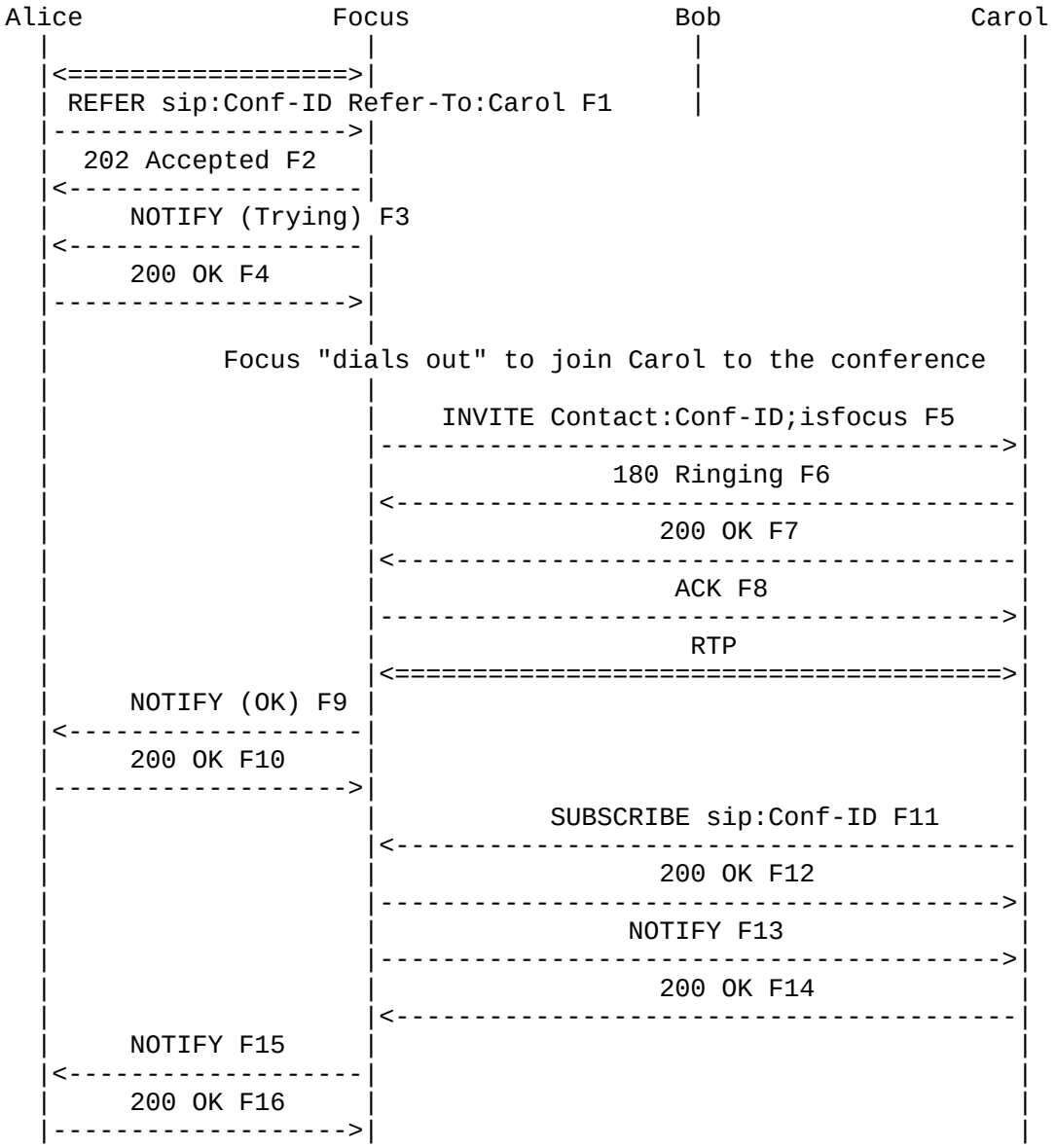


Figure 5. Participant Requests That the Focus Add a Participant to the Conference.

```
F1   REFER sip:3402934234@conf.example.com SIP/2.0
     Via: SIP/2.0/UDP client.atlanta.example.com;branch=z9hG4bKg4534
     Max-Forwards: 70
     To: <sip:3402934234@conf.example.com>
     From: Alice <sip:alice@atlanta.example.com>;tag=5534562
     Call-ID: 849392fklgl43
     CSeq: 476 REFER
     Contact: <sip:alice@alice.example.com>
     Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER,
      SUBSCRIBE, NOTIFY
     Accept: application/sdp, message/sipfrag
     Refer-To: <sip:carol@chicago.example.com>
     Supported: replaces
     Content-Length: 0
```

5.6. REFER: Requesting a User to Dial in to a Conference Using a Conference URI

A participant wishing to add a new participant will request this participant to send an INVITE to the conference URI. This can be done using a non-SIP means (such as passing or publishing the conference URI in an email, IM, or web page). If a non-SIP means is used, then the flow and requirements are identical to Section 5.1.

The SIP mechanism to do this utilizes the REFER method.

A UA wishing to add a new participant SHOULD send a REFER request to the participant with a Refer-To header containing the conference URI.

The requirements are then identical to the dial-in case of Section 5.1. The inviting participant MAY receive notification through the REFER action that the new participant has been added in addition to the notification received through the conference package.

An example is shown in Figure 6. In this call flow, it is assumed that Alice is already a participant of the conference. Alice sends Bob an "out of band" REFER - that is, a REFER outside of an established dialog. Should Bob reject the REFER, Alice might try sending an INVITE to Bob to establish a session first, then send a REFER within the dialog, effectively transferring Bob into the conference [17].

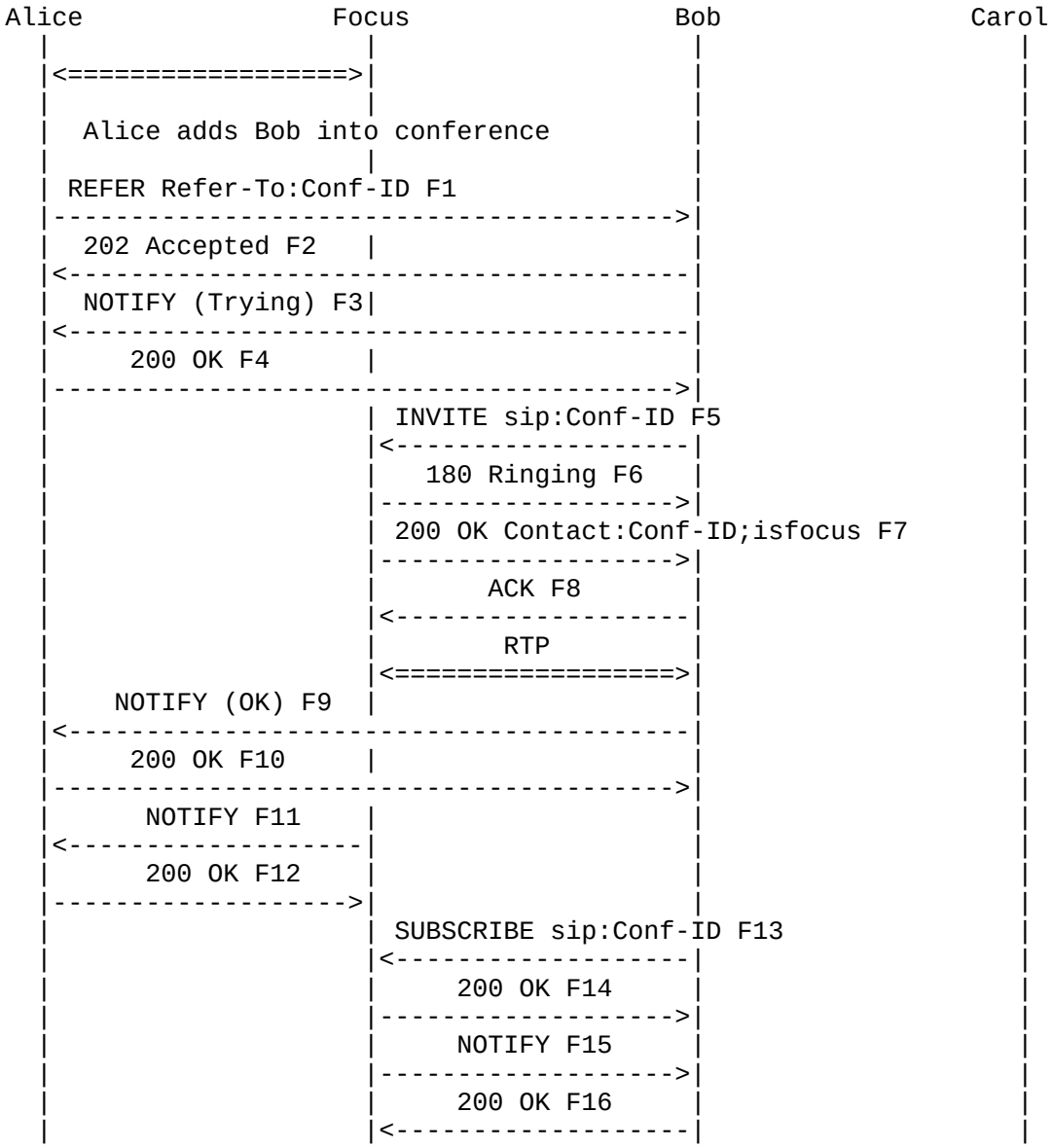


Figure 6. Adding a Participant to an Existing Conference.

5.7. REFER with REFER: Requesting a Focus to Refer a Participant to Dial in to the Conference

A participant may request that the focus refer a participant into the conference by sending a REFER method. The Refer-To header field will have the method set to REFER and an escaped Refer-To header field containing the conference URI.

Note that in Message F1 below, the Refer-To header field is shown as continuing across two lines -- this would not be the case in an actual message; the URI would have continued beyond the formatting limitations of this document.

This scenario is shown in Figure 7.

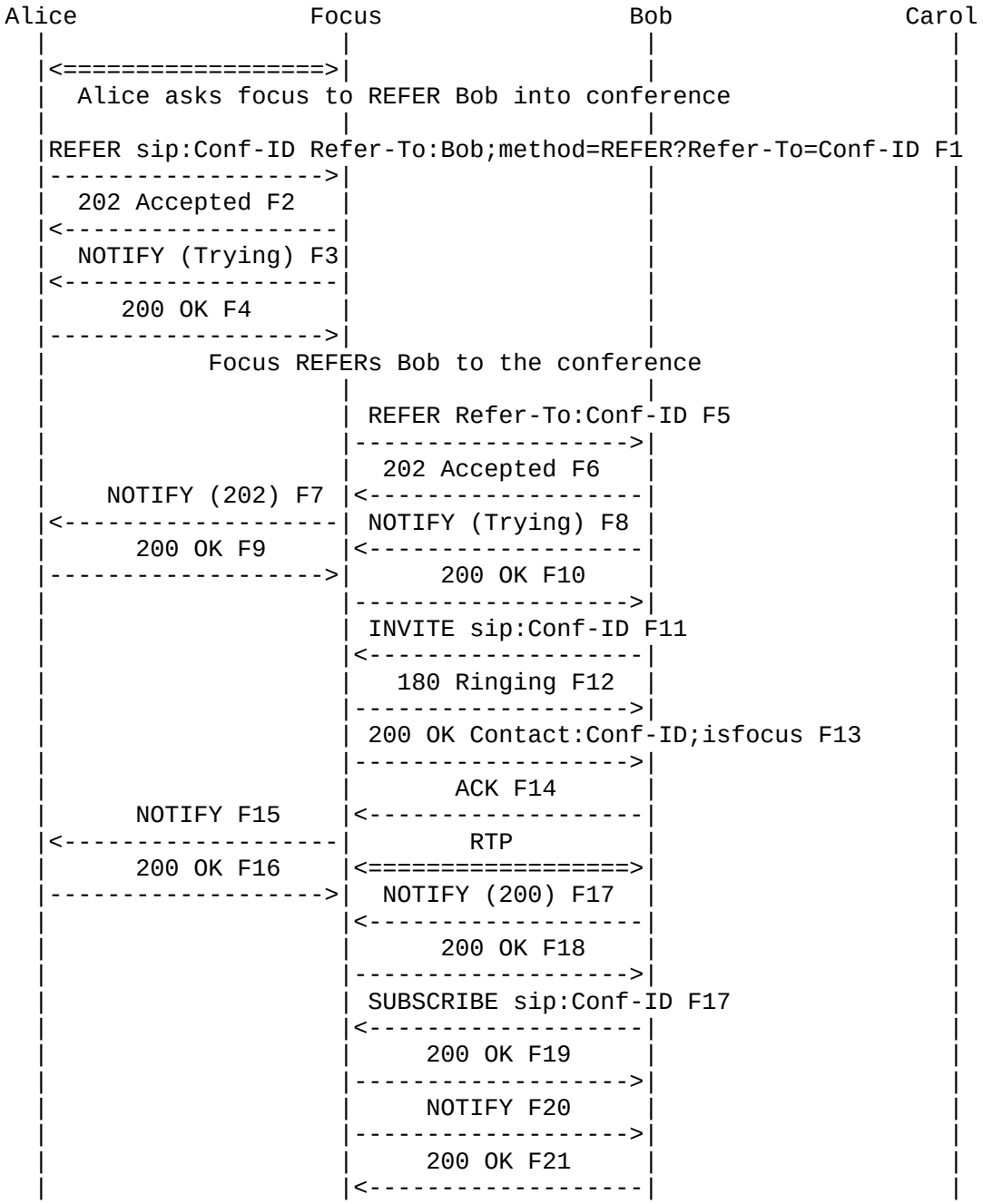


Figure 7. Requesting That the Focus Refer a Participant to a Conference.

```
F1    REFER sip:3402934234@conf.example.com SIP/2.0
      Via: SIP/2.0/UDP client.atlanta.example.com;branch=z9hG4bKg4534
      Max-Forwards: 70
      To: <sip:3402934234@conf.example.com>
      From: Alice <sip:alice@atlanta.example.com>;tag=5534562
      Call-ID: 849392fklgl43
      CSeq: 476 REFER
      Contact: <sip:alice@alice.example.com>
      Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER,
       SUBSCRIBE, NOTIFY
      Accept: application/sdp, message/sipfrag
      Refer-To: <sip:bob@biloxi.example.com;method=REFER
                          ?Refer-To=sip:3402934234%40example.com>
      Supported: replaces
      Content-Length: 0


F5    REFER sip:3402934234@conf.example.com SIP/2.0
      Via: SIP/2.0/UDP ms5.conf.example.com;branch=z9hG4bK33445243
      Max-Forwards: 70
      To: <sip:bob@biloxi.example.com>
      From: <sip:3402934234@conf.example.com>;tag=345621412
      Call-ID: 5494204
      CSeq: 4524323 REFER
      Contact: <sip:3402934234@conf.example.com>;isfocus
      Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER,
       SUBSCRIBE, NOTIFY
      Accept: application/sdp, message/sipfrag
      Refer-To: <sip:3402934234@conf.example.com>
      Supported: join, gruu, replaces
      Content-Length: 0


F11   INVITE sip:3402934234@conf.example.com SIP/2.0
      Via: SIP/2.0/UDP client.biloxi.com;branch=z9hG4bKh3887
      Max-Forwards: 70
      To: <sip:3402934234@conf.example.com>
      From: Bob <sip:bob@biloxi.example.com>;tag=32411
      Call-ID: 5d4324fa84b4c76e66710
      CSeq: 764 INVITE
      Contact: <sip:bob@client.biloxi.example.com>
      Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER,
       SUBSCRIBE, NOTIFY
      Allow-Events: dialog
      Accept: application/sdp, message/sipfrag
      Supported: replaces, join
      Content-Type: application/sdp
      Content-Length: ...
```

(SDP not shown)

5.8. Join Header Field: Dialing in to a Conference Using a (3rd Party) Dialog Identifier

Under some circumstances, a participant wanting to join a conference may only know a dialog identifier of one of the legs of the conference. The information may have been learned using the dialog package [18] or some non-SIP means to retrieve this information from another conference participant.

A UA can request to be added to a conference by sending a request to the focus containing a Join [7] header field containing a dialog ID of one leg of the conference (a dialog between another participant and the focus).

There are other scenarios in which a UA can use the Join header for certain conferencing call control scenarios. See [7] for further examples and details.

An example is shown in Figure 8. It is assumed that Alice is a participant of the conference. The dialog identifier between Alice and the focus is abbreviated as A-F and is known by Bob. Bob requests to be added to the conference by sending an INVITE message F1 to the focus containing a Join header that contains the dialog identifier A-F. Bob is added into the conference by the focus.

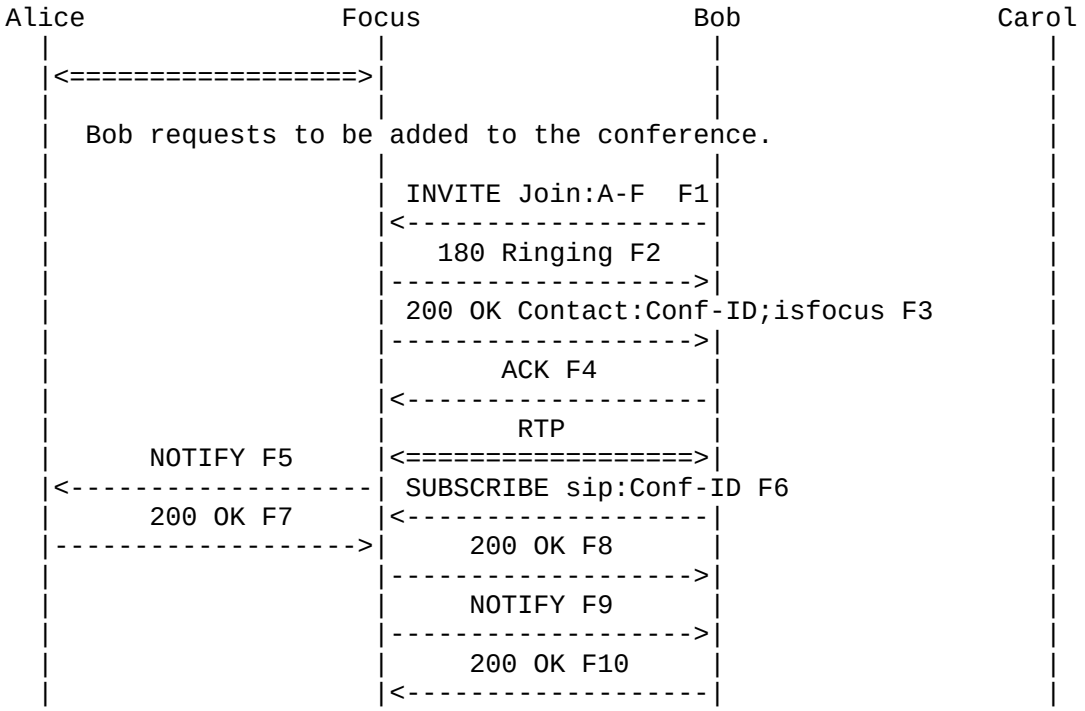


Figure 8. Adding a Participant to an Existing Conference using Join.

```
F1   INVITE sip:3402934234@conf.example.com SIP/2.0
     Via: SIP/2.0/UDP client.biloxi.com;branch=z9hG4bKh3832
     Max-Forwards: 70
     To: <sip:3402934234@conf.example.com>
     From: Bob <sip:bob@biloxi.example.com>;tag=32411
     Call-ID: d432fa84b4c76e66710
     CSeq: 8 INVITE
     Contact: <sip:bob@client.biloxi.example.com>
     Join: 3434034-293553453;to-tag=fdj3l34;from-tag=12f331
     Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER,
      SUBSCRIBE, NOTIFY
     Allow-Events: dialog
     Accept: application/sdp, message/sipfrag
     Supported: replaces, join
     Content-Type: application/sdp
     Content-Length: ...

     (SDP not shown)
```

5.9. Replaces Header Field: Switching User Agents within a Conference

Participants in a conference may want to change the user agent (i.e., the endpoint or the device) with which they participate in the conference. This could be done by simply sending a BYE from one user agent to leave the conference and an INVITE from the other user agent to rejoin. However, the SIP Replaces [6] primitive is perfectly suited to this operation.

An example is shown in Figure 9. It is assumed that Alice is a participant of the conference using user agent #1. The dialog identifier between Alice's user agent #1 and the focus is abbreviated as A-F. Alice switches to user agent #2 and sends an INVITE message F1 to the focus containing a Replaces header that contains the dialog identifier A-F. Note that this dialog identifier could be learned through some non-SIP mechanism, or by use of SUBSCRIBE/NOTIFY and the dialog event package [18]. Alice's user agent #2 is added into the conference by the focus. The focus sends a BYE to user agent #1. User agent #1 then automatically terminates the subscription by sending a SUBSCRIBE with Expires:0 to terminate the subscription. Note that the participant list (roster) has not necessarily changed during this scenario, unless detailed information about Alice user agents (i.e. endpoints) is included in the conference state notifications. For a full discussion of conference package notifications, refer to [9].

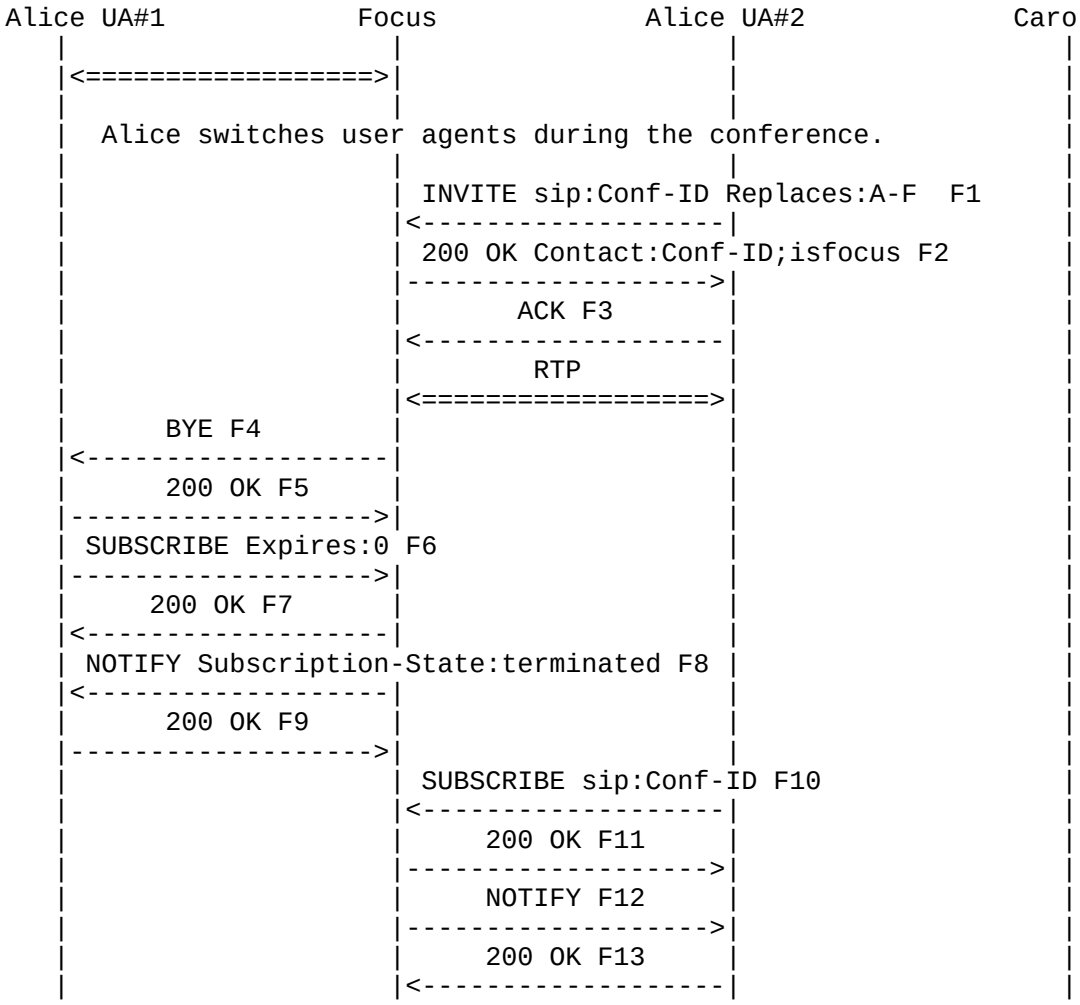


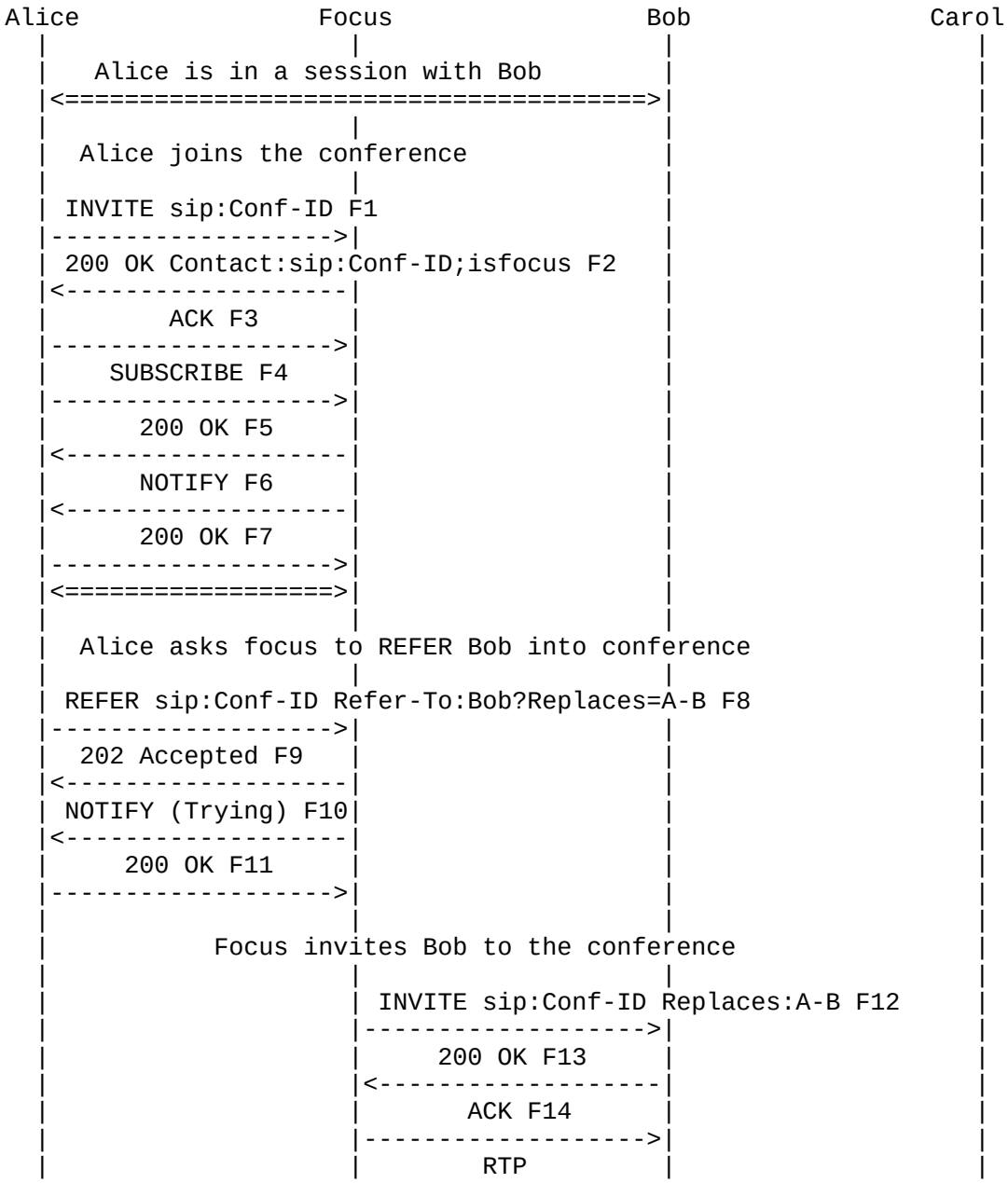
Figure 9. Switching a User Agent within a Conference.

5.10. Replaces Header Field: Transferring a Point-to-Point Session into a Conference

This call flow shows how a point-to-point call can be transferred to a conference call involving an external focus.

Alice and Bob have an established session with a dialog identifier A-B. Alice joins the conference with the focus by sending an INVITE to the Conference URI. Alice then sends a REFER request to the focus to send an INVITE request to the other participant. Alice includes an escaped Replaces header field in the URI included in the Refer-To

header field. Bob receives the INVITE from the focus and matches the dialog in the Replaces header field with the dialog with Alice. As a result, Bob accepts the INVITE, joins the conference, and sends a BYE to Alice to tear down their point-to-point dialog.



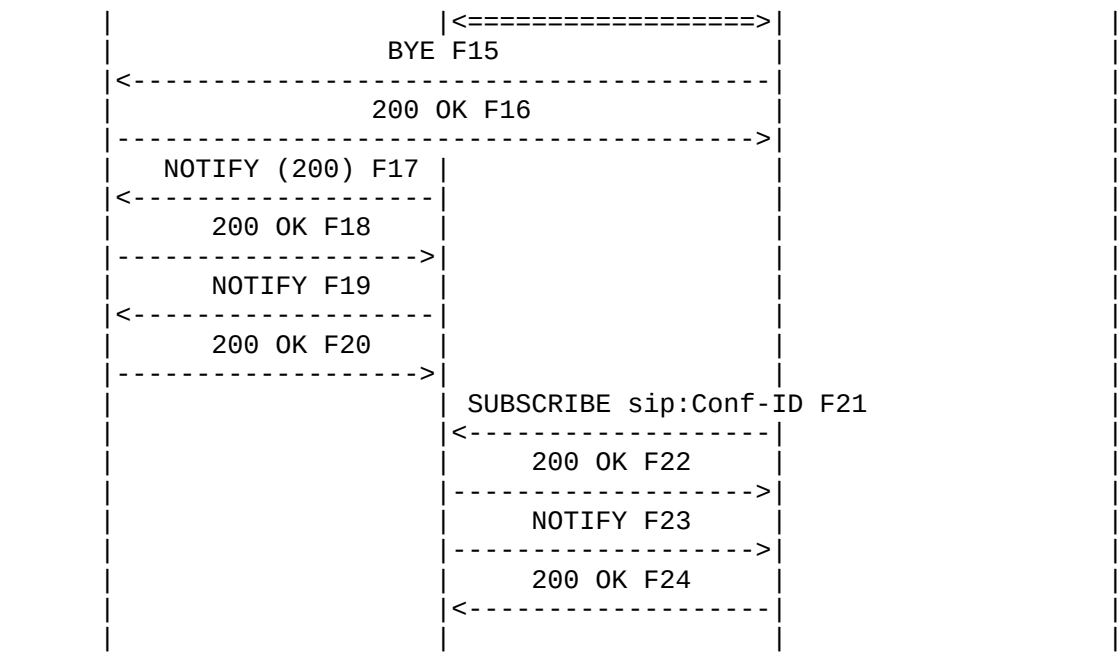


Figure 10. Transitioning a Point to Point Session into a Conference.

5.11. REFER with BYE: Requesting That the Focus Remove a Participant from a Conference

To request that the focus remove a participant from the specified conference, a properly authorized SIP UA (typically the conference owner) can send a REFER to the conference URI with a Refer-To containing the URI of the participant and with the method set to BYE. The requestor does not need to know the dialog information about the dialog between the focus and the participant who will be removed -- the focus knows this information and fills it when it generates the BYE request.

An example call flow is shown in Figure 11. It is assumed that Alice and Carol are already participants of the conference and that Alice is authorized to remove members from the conference. Alice sends a REFER to the conference URI with a Refer-To header containing a URI of the form sip:carol@chicago.example.com;method=BYE.

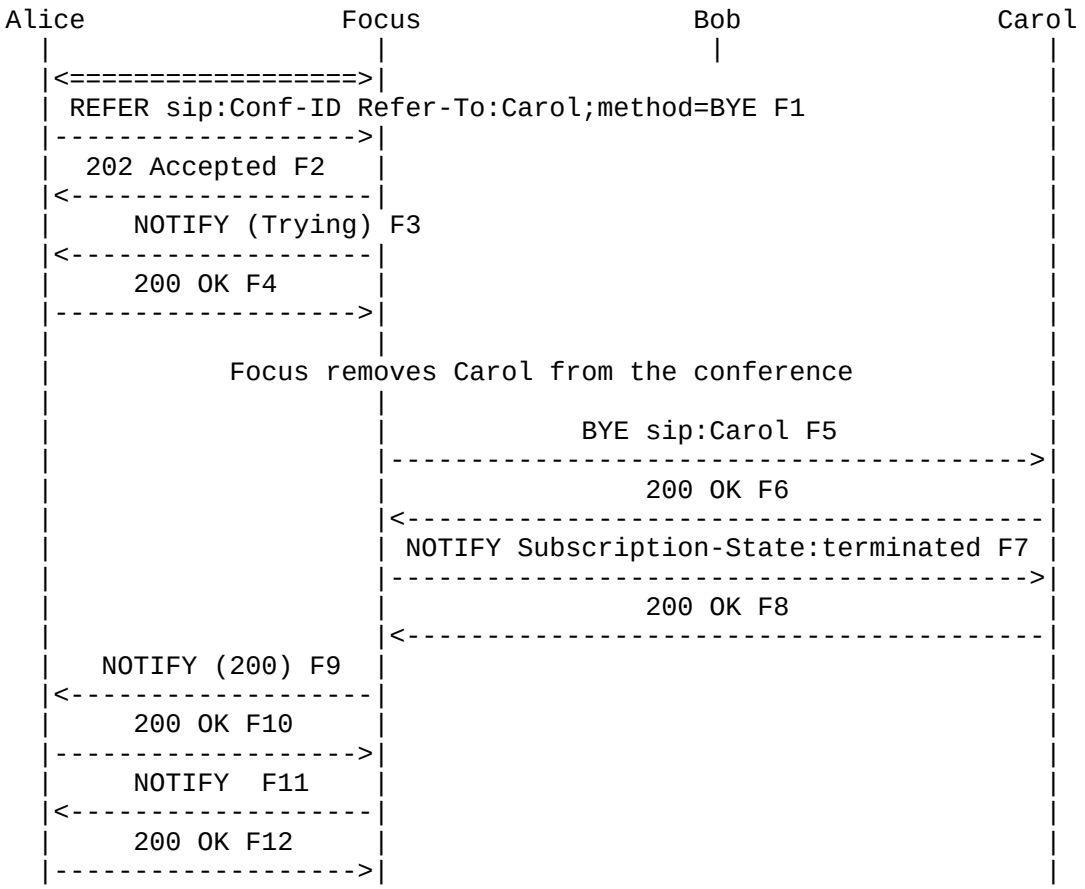


Figure 11. Participant Requests That the Focus Remove a Participant from the Conference.

```
 F1   REFER sip:3402934234@conf.example.com SIP/2.0
      Via: SIP/2.0/UDP client.atlanta.example.com;branch=z9hG4bKg4534
      Max-Forwards: 70
      To: <sip:3402934234@conf.example.com>
      From: Alice <sip:alice@atlanta.example.com>;tag=5534562
      Call-ID: 849392fklgl43
      CSeq: 476 REFER
      Contact: <sip:alice@alice.example.com>
      Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER,
       SUBSCRIBE, NOTIFY
      Accept: application/sdp, message/sipfrag
      Refer-To: <sip:carol@chicago.example.com;method=BYE>
      Supported: replaces
      Content-Length: 0
```

```
F5    BYE sip:carol@client.chicago.example.com SIP/2.0
      Via: SIP/2.0/UDP ms5.conf.example.com;branch=z9hG4bK343gf4
      Max-Forwards: 70
      From: <sip:3402934234@conf.example.com>;tag=5393k2312
      To: Carol <sip:carol@chicago.example.com>;tag=32331
      Call-ID: d432fa84b4c76e66710
      CSeq: 78654 BYE
      Content-Length: 0
```

5.12. Deleting a Conference

The default conference policy for conferences created using the Conference Factory URI is that the conference is deleted when the creator departs.

Figure 12 shows this call flow in which the creator Alice departs causing the conference to be deleted. Note that the order of sending BYEs and final NOTIFYs is not important.

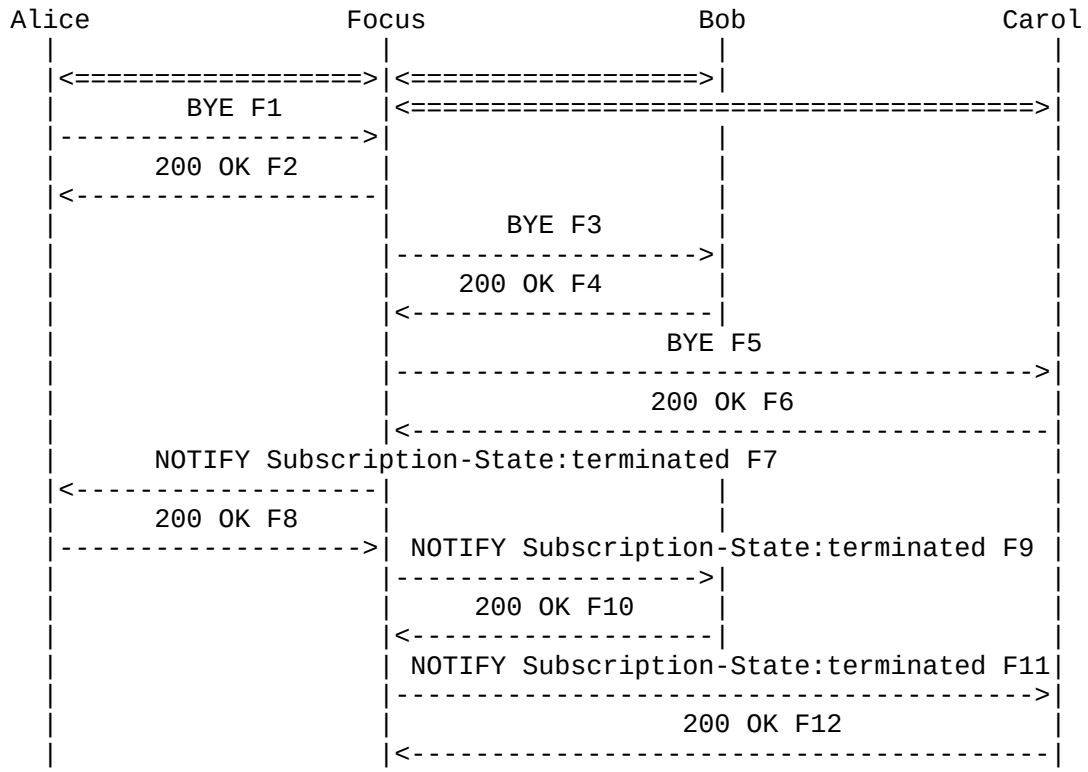


Figure 12. Deleting a Conference.

5.13. Discovery of URI Properties Using OPTIONS

A UA MAY send an OPTIONS request to discover if an opaque URI is a conference URI (resolves to a focus). In addition, the reply to the OPTIONS request can also indicate support for various SIP call control extensions used in this document.

Note that the Allow, Accept, Allow-Events, and Supported header fields should be present in an INVITE from a focus or a 200 OK answer from the focus to an INVITE as a part of a normal dialog establishment process.

An example is shown in Figure 13 where Alice sends an OPTIONS to a URI that resolves to a focus.

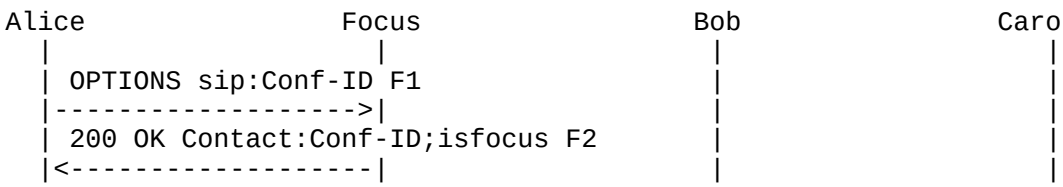


Figure 13. Participant Queries Capabilities of URI of a Focus.

Following is an example of message detail of message F2 in Figure 13. Based on the response, Alice's UA learns that the URI is a conference URI and that the responding UA is focus that supports a number of SIP call control extensions.

The response details are as follows:

```
F2   SIP/2.0 200 OK
     Via: SIP/2.0/UDP pc33.atlanta.example.com;branch=z9hG4bKhjsas87
      ;received=192.0.2.4
     To: <sip:3402934234@conf.example.com>;tag=93810874
     From: Alice <sip:alice@atlanta.example.com>;tag=1928301774
     Call-ID: a84b4c76e66710
     CSeq: 63104 OPTIONS
     Contact: <sip:3402934234@conf.example.com>;isfocus
     Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER,
      SUBSCRIBE, NOTIFY
     Allow-Events: refer, conference
     Accept: application/sdp, message/sipfrag
     Accept-Language: en
     Supported: replaces, join, gruu
     Content-Type: application/sdp
     Content-Length: ...

     v=0
     o=focus431 2890844563 2890842835 IN IP4 ms5.conf.example.com
     s=-
     i=Example Conference Hosted by Example.com
     u=http://conf.example.com/3402934234
     e=3402934234@conf-help.example.com
     p=+18882934234
     c=IN IP4 ms5.conf.example.com
     t=0 0
     m=audio 0 RTP/AVP 0 3 5 7
     m=video 0 RTP/AVP 31 32
```

Useful information from each of these headers is detailed in the next sections.

Allow. The support of methods such as REFER, SUBSCRIBE, and NOTIFY indicates that the user agent supports call control and SIP events.

Accept. The support of bodies such as message/sipfrag [12] indicates support of call control.

Allow-Events. Indicates support of event packages such as refer [4] and conference [9].

Supported. Indicates support of extensions such as replaces, join, and gruu.

Contact. The presence of the 'isfocus' feature parameter in the Contact header indicates that the URI is a conference URI and that the UA is a focus.

6. Security Considerations

This specification defines the interaction between a focus UA and a participant UA in a conferencing application. As a result, the security considerations and mechanisms defined in RFC 3261 [2] apply. However, there are some aspects unique to conferencing that will be discussed here.

A conference often involves the use of substantial network bandwidth and computing resources. As a result, authentication is even more important than in a simple peer-to-peer session. As discussed in the conferencing framework [8], conferences often have policy related to conferencing resources. A focus SHOULD authenticate participants before joining them to a conference and allowing utilization of conferencing resources. Different policies can be applied by a focus to different participants based on the result of authentication.

A participant will be interacting with a number of other participants through the focus. As a result, a participant should authenticate the focus and be sure that the focus used for the conference is trusted. Normal SIP authentication mechanisms are suitable for participant and focus authentication, such as SIP Digest utilizing a shared secret, or certificates, or a secured SIP identity mechanism. In addition, a focus SHOULD support Secure SIP connections so that hop-by-hop mutual authentication and confidentiality provided by TLS can be achieved.

In the SIP dialog between them, a focus utilizes the 'isfocus' feature tag to indicate that the UA is acting as a focus. As such, the SIP header fields such as Contact SHOULD have end to end integrity. A participant and focus SHOULD support an end-to-end integrity mechanism such as S/MIME.

Once a participant has learned that the other UA is a focus, SIP call control operations (such as REFER) can be implemented, or a subscription to the conference package of the focus might be attempted. The security considerations described in RFC 3515 [4] apply to any REFER call control operations. A focus and participant will apply policy to determine which call control operations are allowed.

A focus accepting subscriptions to the conference package must follow the security considerations in RFC 4575 [9]. Since notifications can carry sensitive information, the subscriptions should be authenticated and the notifications delivered with confidentiality and integrity protection. Since a participant is not able to authenticate other participants directly, a participant must rely on the focus to perform this authentication.

A focus MUST support a participant's request for privacy, either through conference policy or as expressed through the signaling. For example, a participant joining a conference and including a Privacy header field [10] must not have identity information revealed to other participants by the focus. If other signaling protocols are used, privacy signaled through them also must be respected.

7. Contributors

We would like to thank Rohan Mahy, Jonathan Rosenberg, Roni Even, Petri Koskelainen, Brian Rosen, Paul Kyzivat, Eric Burger, and others in list discussions.

Thanks to Miguel Garcia for his detailed last-call review and suggestions.

8. References

8.1. Normative References

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

[2] 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.

[3] Roach, A., "Session Initiation Protocol (SIP)-Specific Event Notification", RFC 3265, June 2002.

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

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

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

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

[8] Rosenberg, J., "A Framework for Conferencing with the Session Initiation Protocol (SIP)", RFC 4353, February 2006.

[9] Rosenberg, J., Schulzrinne, H., and O. Levin, "A Session Initiation Protocol (SIP) Event Package for Conference State", RFC 4575, August 2006.

[10] Peterson, J., "A Privacy Mechanism for the Session Initiation Protocol (SIP)", RFC 3323, November 2002.

8.2. Informative References

[11] Campbell, B. and R. Sparks, "Control of Service Context using SIP Request-URI", RFC 3087, April 2001.

[12] Sparks, R., "Internet Media Type message/sipfrag", RFC 3420, November 2002.

[13] Johnston, A., Donovan, S., Sparks, R., Cunningham, C., and K. Summers, "Session Initiation Protocol (SIP) Basic Call Flow Examples", BCP 75, RFC 3665, December 2003.

[14] Levin, O. and R. Even, "High Level Requirements for Tightly Coupled SIP Conferencing", RFC 4245, November 2005.

[15] Mahy, R., "A Call Control and Multi-party usage framework for the Session Initiation Protocol (SIP)", Work in Progress, February 2005.

[16] Rosenberg, J., "Obtaining and Using Globally Routable User Agent (UA) URIs (GRUU) in the Session Initiation Protocol (SIP)", Work in Progress, February 2005.

[17] Sparks, R., Johnston, A., and D. Petrie, "Session Initiation Protocol Call Control - Transfer", Work in Progress, April 2005.

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

Appendix A: Creating a Conference by a Conference-Unaware UA

This section discusses how a human user operating a conference-unaware UA can create and add participants to a conference. This method is described as an appendix since it is NOT RECOMMENDED. The scenarios involving creating a conference using ad-hoc or manual means are recommended over this scenario. This scenario is included, however, for completeness.

A user (human) would choose a conference URI according to system rules and insert it into the Request-URI of the INVITE. This same URI is echoed by a focus adhering to certain addressing conventions (discussed below) in the Contact header by the focus. Additional participants could be added by non-SIP means (publication of the chosen conference URI using web pages, email, IM, etc.). Alternatively, the conference-unaware UA could then add other participants to the conference using SIP call control by establishing a session with them, then transferring [17] them to the conference URI. Note that in this scenario only the user (human) is aware of the conferencing application, and the conference-unaware UA only need support RFC 3261 [2] and optionally call transfer.

Making this work does impose certain addressing conventions on a system. As a service/implementation choice, a system could allow the creator of the conference to choose the user portion of the conference URI. However, this requires the URI format to be agreed upon between a user and the system.

For example, a service provider might reserve the domain conf.example.com for all conference URIs. Any URI in the domain of conf.example.com would resolve to the focus. The focus could be configured to interpret an unknown user part in the conf.example.com domain as a request for a conference to be created with the conference URI as the Request-URI. For example, an INVITE sent with a Request-URI of sip:k32934208ds72@conf.example.com could be routed to the focus that would then create the conference. This conference URI should be registered by the newly created focus to become routable as a conference URI within the conf.example.com domain. The returned Contact would look as follows:

```
Contact: <sip:k32934208ds72@conf.example.com>;isfocus
```

Note, however, that this approach relies on conventions adopted between the user (human) and the focus. Also, the approach is not robust against collisions in the conference names. If a second user wishing to create a new conference happened to choose the same user part as an existing conference, the result would be that the second user would be added into the existing conference instead of creating a new one.

As a result, methods of conference creation in which the conference URI is an opaque URI generated by the focus are preferred.

An example call flow is shown in Figure 14. The participant Alice creates the conference URI (using some convention agreed to with the focus domain) and sends an INVITE to that URI which creates the focus. The focus creates the conference and returns the same conference URI in the 200 OK answer to the INVITE (which is ignored by the conference-unaware UA).

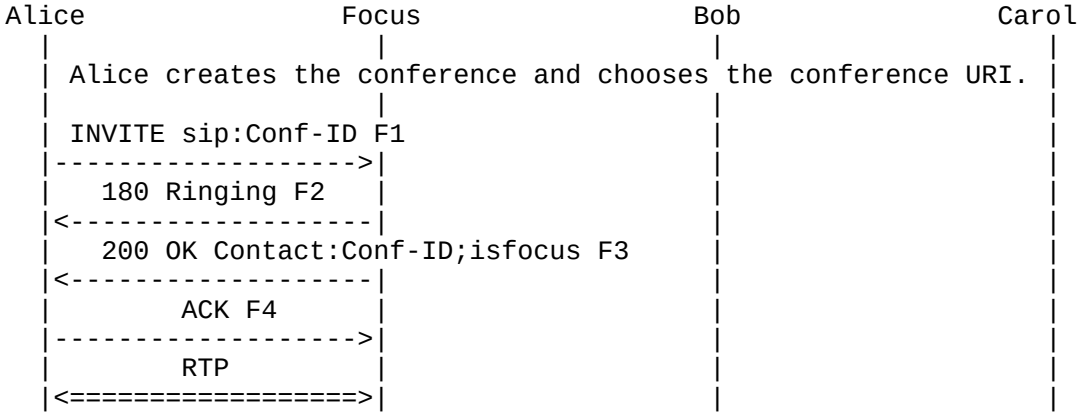


Figure 14. Not Recommended: Conferencing Unaware Participant Creates a Conference

Authors' Addresses

Alan Johnston
Avaya
St. Louis, MO 63102

EMail: alan@sipstation.com

Orit Levin
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052

EMail: oritl@microsoft.com

Full Copyright Statement

Copyright (C) The Internet Society (2006).

This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Acknowledgement

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).