

SIPPING Working Group
Internet-Draft
Expires: December 27, 2003

A. Johnston
MCI
O. Levin
RADVISION
June 28, 2003

**Session Initiation Protocol Call Control - Conferencing for User
Agents
draft-ietf-sipping-cc-conferencing-01**

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of [Section 10 of RFC2026](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/1id-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on December 27, 2003.

Copyright Notice

Copyright (C) The Internet Society (2003). All Rights Reserved.

Abstract

This document 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 different user agent (UA) types perspective: conference-unaware, conference-aware and focus UAs. The use of URIs in conferencing, OPTIONS for capabilities discovery, and call control using REFER are covered in detail with example call flow diagrams.

Table of Contents

1.	Introduction	3
2.	Usage of the 'isfocus' Feature Parameter	3
2.1	General	3
2.2	Session Establishment	4
2.3	OPTIONS	4
3.	SIP User Agent Conferencing Capability Types	4
3.1	Focus UA	5
3.2	Conference Factory URI	5
3.3	Conference-Unaware UA	5
3.4	Conference-Aware UA	6
4.	SIP Conferencing Primitives	6
4.1	Joining a Conference using the Conference URI - Dial In . .	6
4.2	Adding a Participant by the Focus - Dial Out	7
4.3	Manually Creating a Conference by Dialing into a Conferencing Application	8
4.4	Creating a Conference by a Conference-Unaware UA	10
4.5	Creating a Conference using Ad-Hoc SIP Methods	11
4.6	Requesting the Focus Add a New Resource to a Conference . .	13
4.7	Adding a 3rd Party Using Conference URI	15
4.8	Adding a 3rd Party Using a Dialog Identifier	16
4.9	Changing User Agents within a Conference	17
4.10	Bringing a Point-to-Point Dialog into a Conference	18
4.11	Requesting the Focus Remove a Participant from a Conference	19
4.12	Discovery of Conferencing Capabilities using OPTIONS	20
5.	Security Considerations	21
6.	Contributors	22
7.	Changes since -00	22
	Normative References	22
	Informative References	23
	Authors' Addresses	24
	Intellectual Property and Copyright Statements	25

1. Introduction

This document uses the concepts and definitions from the high level requirements [[9](#)] and the SIP conferencing framework [[10](#)] documents.

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 having different capabilities and who potentially have joined the conference by different means (i.e. dial-in, dial-out, scheduled, and ad-hoc).

The call control and dialog manipulation approach is based on the multiparty framework [[11](#)] document. That document defines the basic approach of service design adopted for SIP which includes:

- Definition of primitives, not services
- Signaling model independent
- Invoker oriented
- Primitives make full use of URIs
- Include authentication, authorization, logging, etc. policies
- 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 service-related header fields), as discussed in [RFC 3087](#) [[12](#)], is fundamental to this approach.

2. Usage of the 'isfocus' Feature Parameter

2.1 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 backwards 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

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 [7] document which includes the definition of the "isfocus" feature parameter.

2.2 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.

2.3 OPTIONS

Currently the only met requirement is: given an opaque URI, being able to recognize whether it belongs to a certain conference (i.e. meaning that it is a conference URI) or not. As with any other OPTIONS request, it 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.

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 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 [\[5\]](#) and indicate so in Allow-Events header fields in requests and responses. A focus MAY include information about the conference in SDP message bodies sent.

A focus SHOULD support the Replaces [\[8\]](#) header field.

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. These are open to the conferencing server implementation policy and include non-automated means (such as IVR), SIP, and a conference policy 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 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 4.5](#).

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 into 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 needs only to 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 [14] document.

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](#).

A conference-aware UA MUST recognize the "isfocus" feature parameter. A conference-aware UA SHOULD support REFER [3], SIP events [4], and the conferencing package [5].

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.

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.

4. SIP Conferencing Primitives

The SIP conferencing call control flows presented in this section are the call control building blocks for various SIP tight conferencing applications as described in the conferencing requirements [9] and framework [10] documents. The major design goal is that the same SIP conferencing primitives would be used by user agents having different conferencing capabilities and comprising different applications.

4.1 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.

If the UA is the first participant of the conference to dial in, it is likely that this INVITE will create 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 SHOULD 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.

Participants leaving the conference send a BYE to the focus. If they have a current subscription to the conference package, they also must send a SUBSCRIBE with an Expires:0 header field to terminate the subscription since the BYE does not automatically terminate the subscription.

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

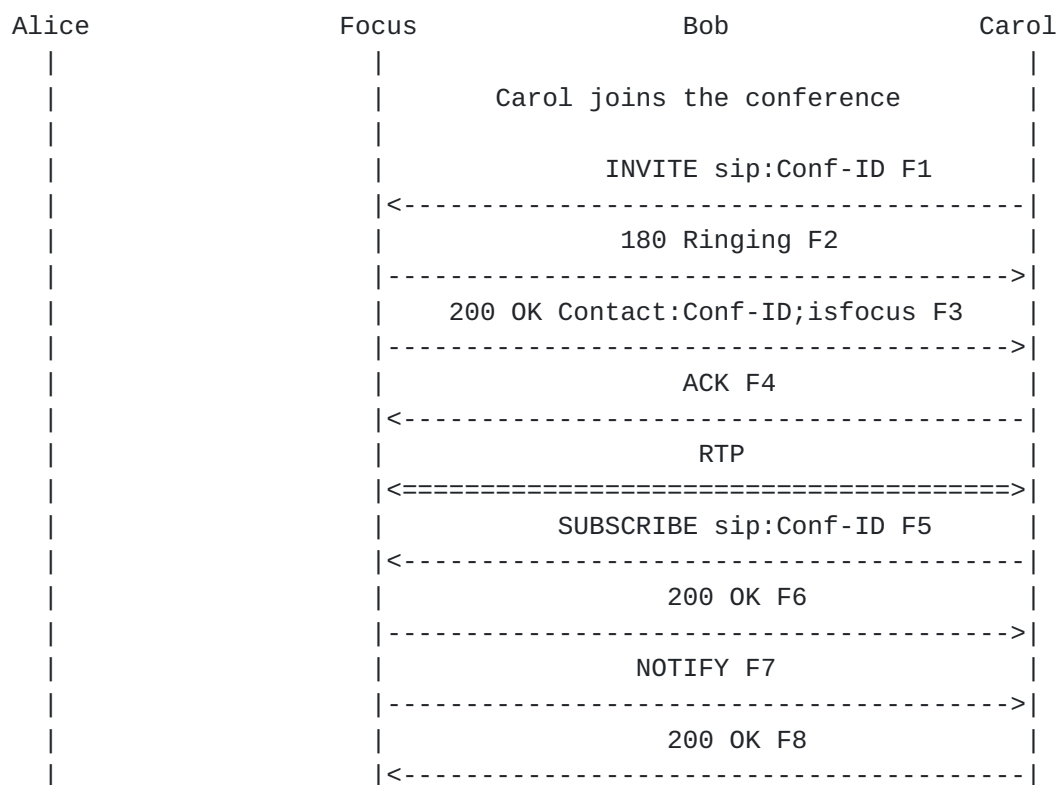


Figure 1. A Participant Joins a Conference using the Conference URI.

[4.2](#) 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.

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.

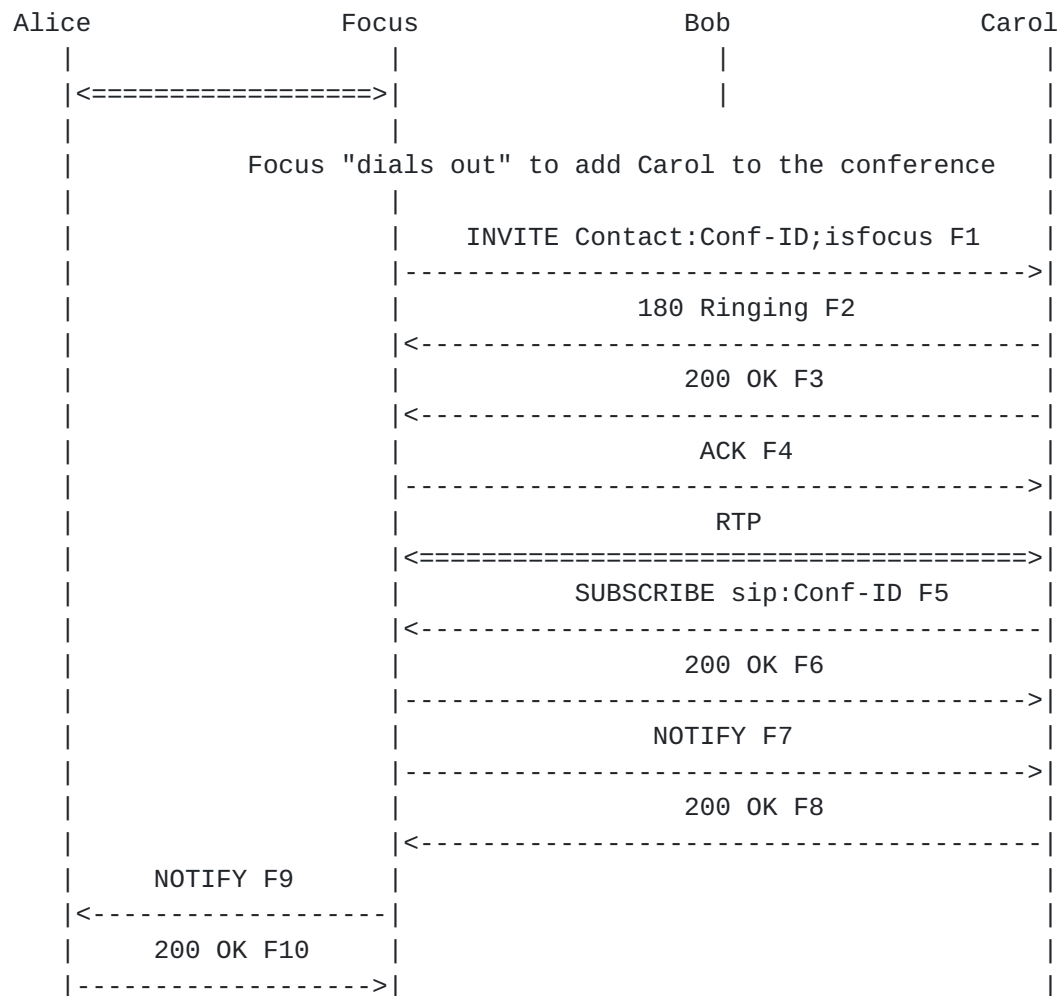


Figure 2. A Focus "dials out" to Add a Participant to the Conference.

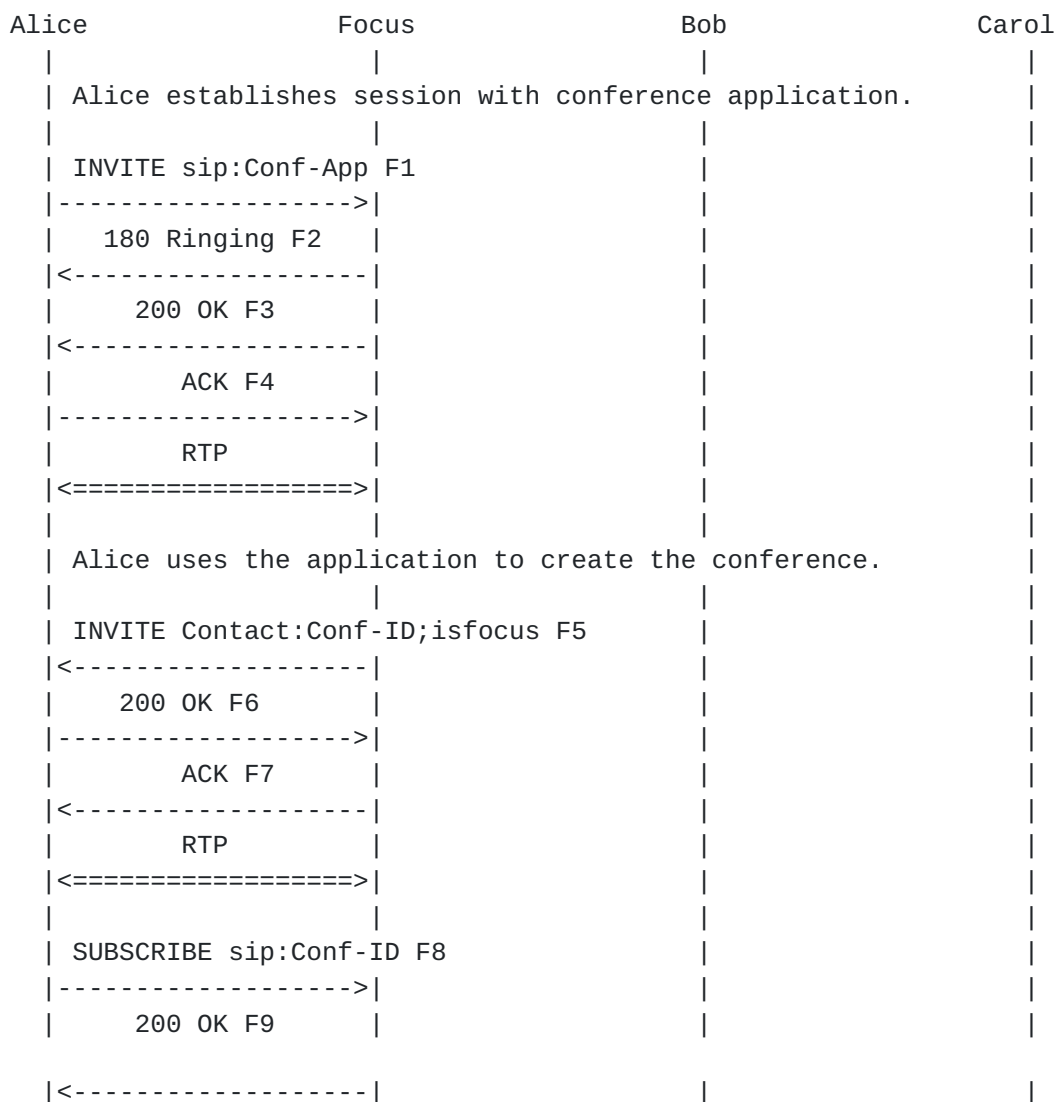
4.3 Manually Creating a Conference by Dialing into a Conferencing Application

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. At this point the conference

server MUST re-INVITE the user with the conference URI in Contact with the "isfocus" feature parameter.

Alternatively, the additional information MAY be provided by the user during an early dialog established. This could be accomplished by a 183 Session Progress response sent by the conferencing application. After the conference is created, the conference URI MUST then be returned in a Contact in the 200 OK.

An example call flow is shown in Figure 3. In this example, Alice uses a conference application which 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.



	NOTIFY F10					
	<-----					
	200 OK F11					
	----->					

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

[4.4](#) Creating a Conference by a Conference-Unaware UA

It is a requirement that a user (human) be able to use a conference-unaware UA to create and add participants to a conference.

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 [[16](#)] 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](#) 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 4. 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).

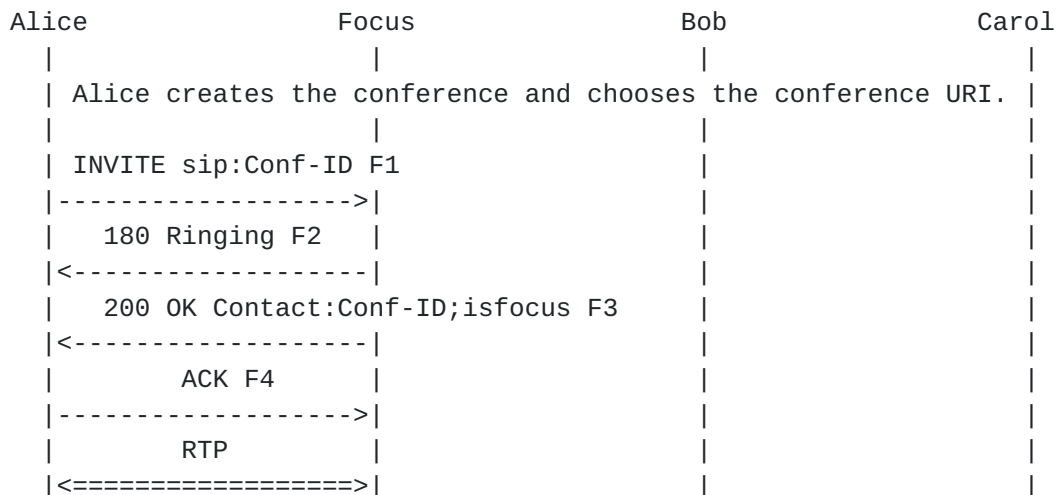


Figure 4. A Conferencing Unaware Participant Creates a Conference

4.5 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 2.4](#)) is used to automatically create the conference in this example.

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 4.3](#) is that no user intervention (IVR, web page form, etc.) is required to create the conference.

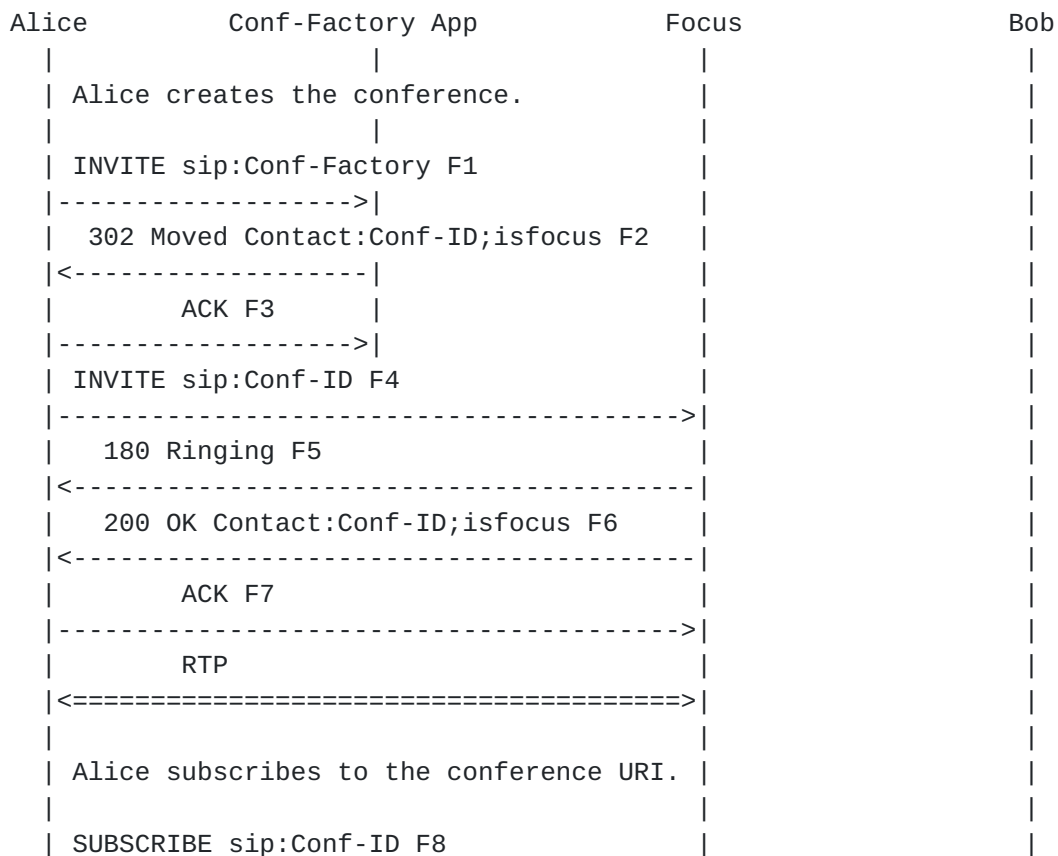
The SIP URI of the conference factory can be provisioned in the UA (as in a "create new conference" button on a SIP phone) or can be

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.

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 5. 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 using a 302 Moved Temporarily response to the focus. Note that with proxy recursion, 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.




```

|----->|
| 200 OK F9 |
|<-----|
| NOTIFY F10 |
|<-----|
| 200 OK F11 |
|----->|

```

Figure 5. Creation of a Conference using SIP Ad-Hoc Methods.

[4.6](#) Requesting the Focus Add a New Resource to a Conference

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 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 dependant both on the nature of the new resource (as expressed by its URI) and the own focus policies regarding IM, central vs. distributed real time media processing, etc.

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 6. 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.

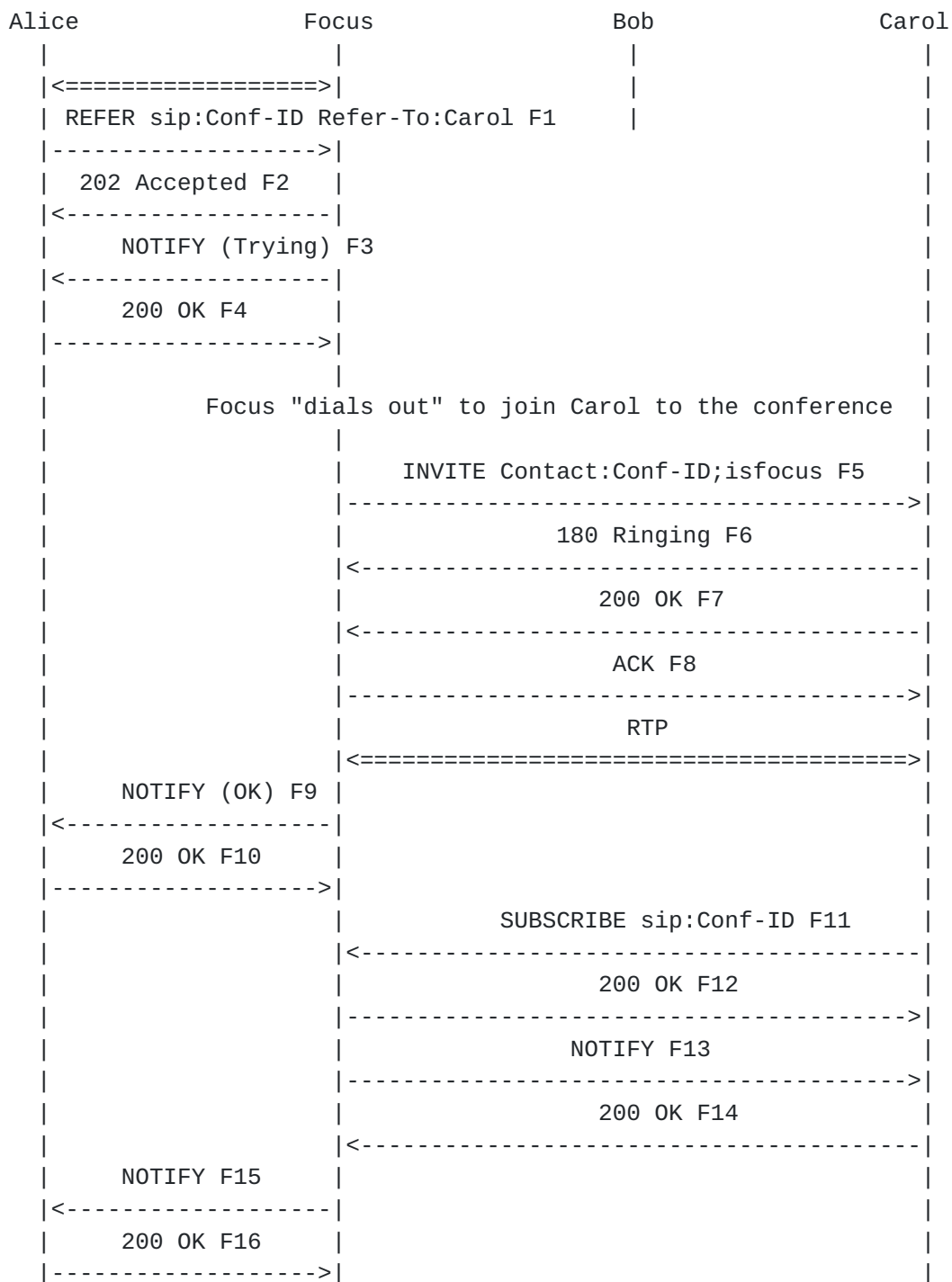


Figure 6. Participant Requests Focus add a Participant to the Conference.

4.7 Adding a 3rd Party Using Conference URI

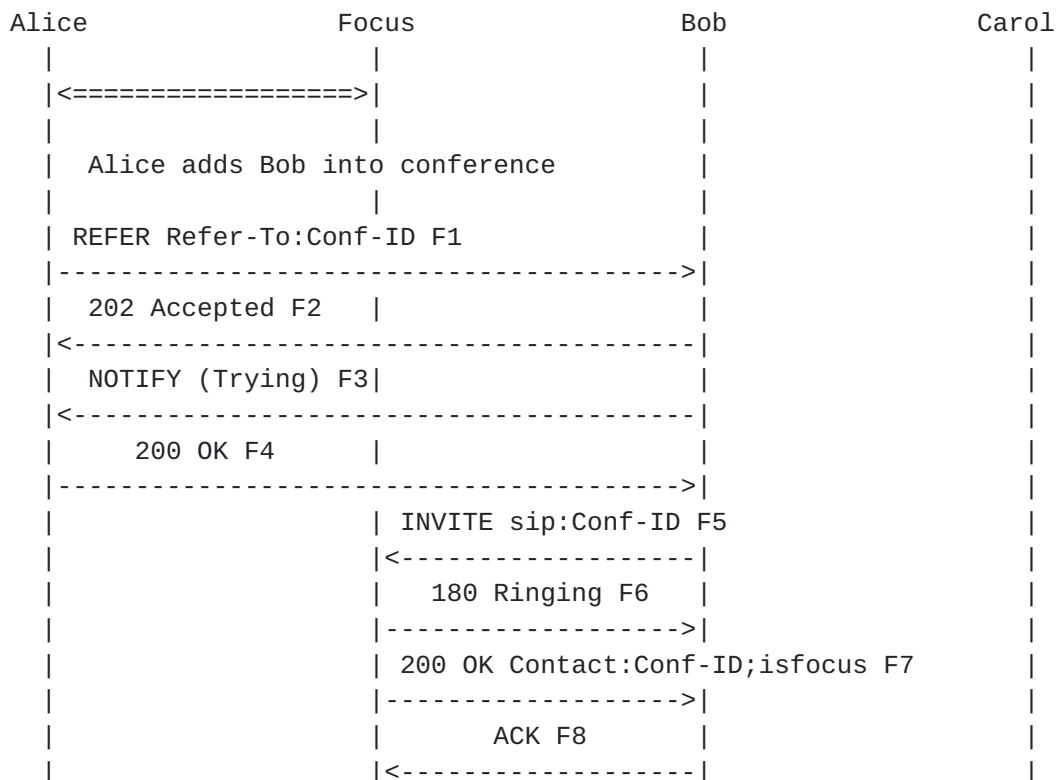
A participant wishing to add a new participant will request this participant to send an INVITE to the conference URI. This can be done using a non-SIP means (such as passing or publishing the conference URI in an email, IM, or web page). If a non-SIP means is used, then the flow and requirements are identical to [Section 4.1](#).

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 4.1](#). 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 7. 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 [\[16\]](#).



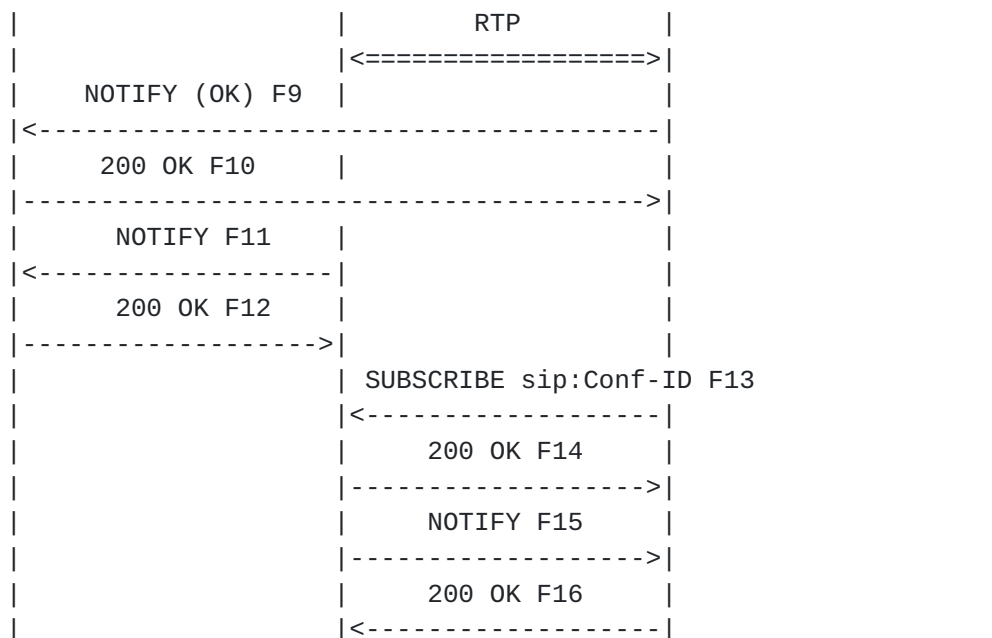


Figure 7. Adding a Participant to an Existing Conference.

4.8 Adding a 3rd Party Using a 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 and the conference factory URI, instead of the conference URI. The information may have been learned using the dialog package [17] or some non-SIP means to retrieve this information from a conference participant.

A UA can request to be added to a conference by sending a request to the focus containing a Join [6] header field containing a dialog ID of one leg of the conference (a dialog between a participant and the focus).

There are other scenarios in which a UA can use the Join header for certain conferencing call control scenarios. See [6] 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 which 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 [17]. Bob is added into the conference by the

focus.

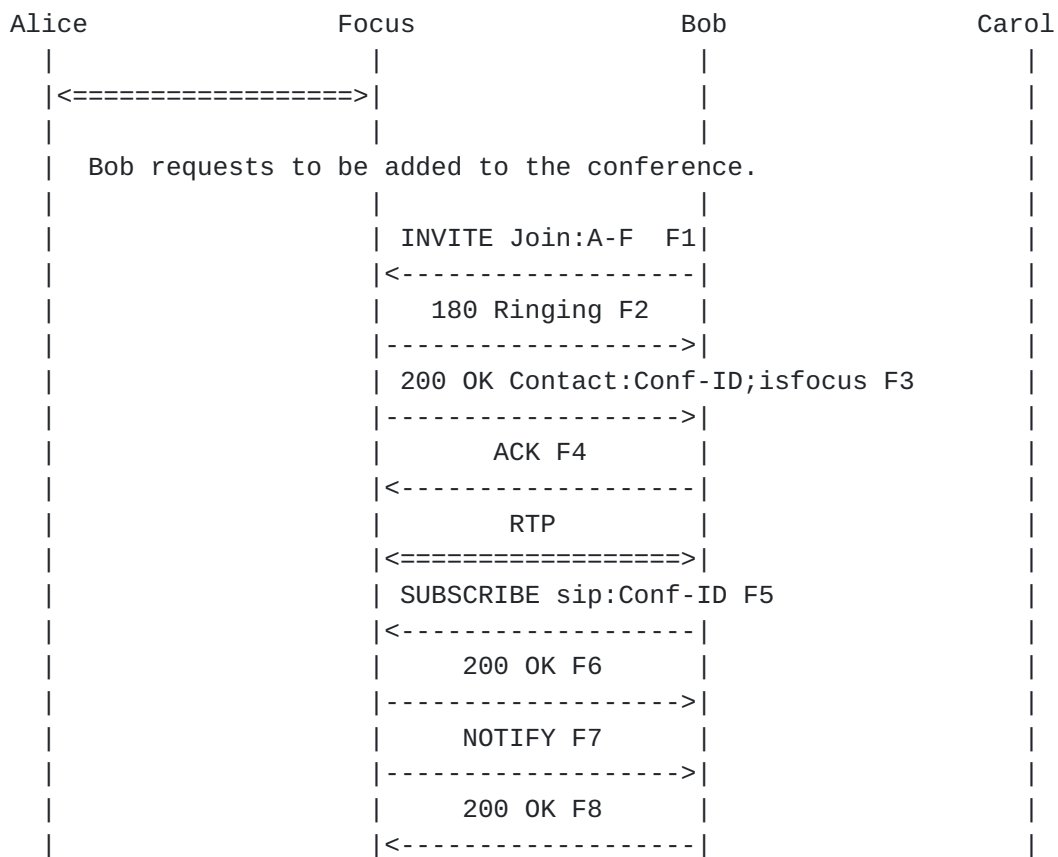


Figure 8. Adding a Participant to an Existing Conference using Join.

4.9 Changing User Agents within a Conference

A participant in a conference may want to change the user agent with which they participate in the conference. While 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 which 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 [17]. 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 as the participant list (roster) has not changed during this scenario, no NOTIFYs are sent by the focus to subscribers to the participant list.

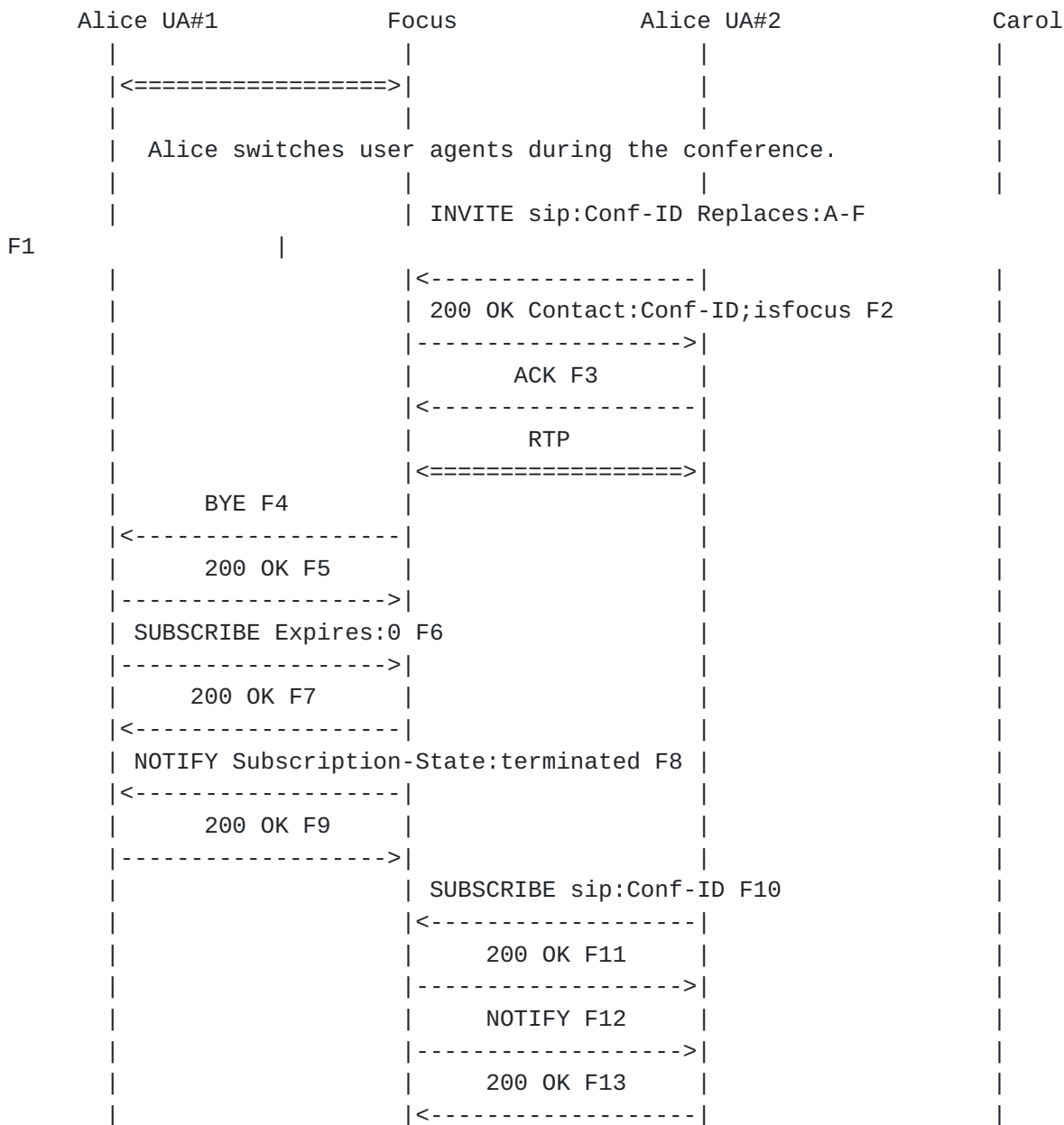


Figure 9. Adding a Participant to an Existing Conference using Join.

[4.10](#) Bringing a Point-to-Point Dialog into a Conference

A focus is capable of bringing an existing point-to-point dialog with another UA to a conference that the focus hosts. The focus would do

it by sending re-INVITE changing the Contact URI to the conference

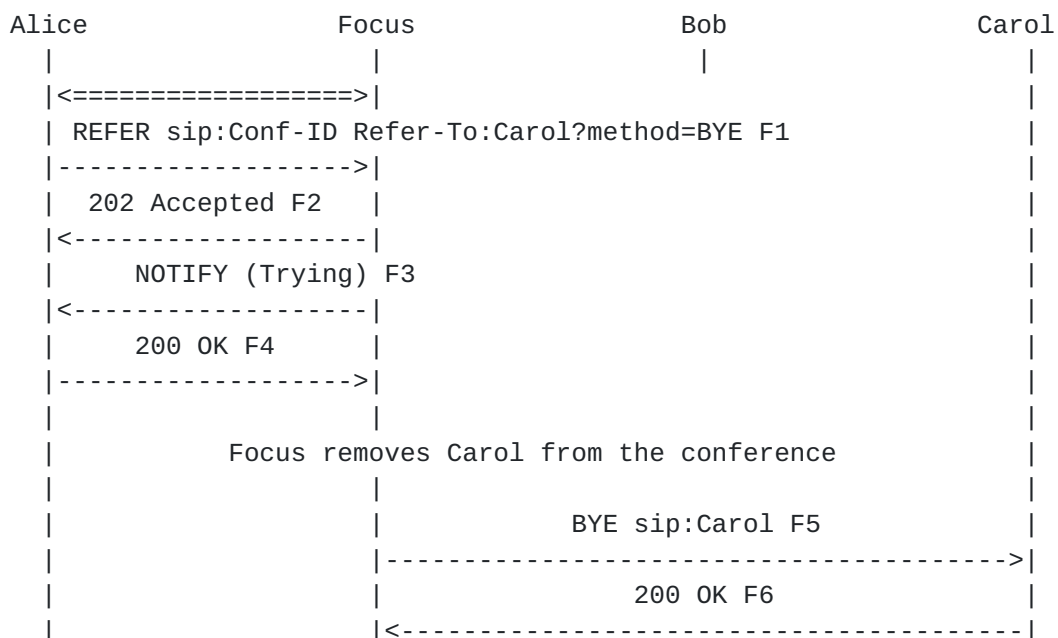
URI with the "isfocus" feature parameter. By doing this, the focus signals to the UA that it becomes a participant of the conference, specified in the Contact header.

Currently, there is no way for a UA, being in an active point-to-point call with a focus, to express by SIP call control means a request to bridge its dialog with a specific conference or to create a new conference and include the dialog in this conference. Instead, a new dialog will need to be created. Even if the UA discovers that the other side has focus capabilities, the UA needs to close the old session and to establish a new session/dialog with the focus.

4.11 Requesting the Focus Remove a Participant from a Conference

To request 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 10. 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>.




```

|                                     | NOTIFY Subscription-State:terminated F7 |
|                                     |----->|
|                                     | 200 OK F8 |
|                                     |<-----|
| NOTIFY (OK) F9 |
|<-----|
| 200 OK F10 |
|----->|
| NOTIFY F11 |
|<-----|
| 200 OK F12 |
|----->|

```

Figure 10. Participant Requests Focus Remove a Participant from the Conference.

[4.12](#) Discovery of Conferencing Capabilities 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 11 where Alice sends an OPTIONS to a URI which resolves to a focus.

Alice	Focus	Bob	Carol
OPTIONS sip:Conf-ID F1			
----->			
200 OK Contact:Conf-ID;isfocus F2			
<-----			

Figure 11. Participant Queries Capabilities of URI which resolves to a Focus.

Following is an example message detail of message F2 in Figure 11. 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:

SIP/2.0 200 OK

Johnston & Levin

Expires December 27, 2003

[Page 20]


```
Via: SIP/2.0/UDP pc33.atlanta.com;branch=z9hG4bKhjhs8ass877
    ;received=192.0.2.4
To: <sip:23944852902@chicago.com>;tag=93810874
From: Alice <sip:alice@atlanta.com>;tag=1928301774
Call-ID: a84b4c76e66710
CSeq: 63104 OPTIONS
Contact: <sip:23944852902@chicago.com>;isfocus
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER,
    SUBSCRIBE, NOTIFY
Allow-Events: refer, conference
Accept: application/sdp, application/conference-info+xml,
    message/sipfrag
Accept-Language: en
Supported: replaces
Content-Type: application/sdp
Content-Length: 274
```

(SDP not shown)

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

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

Accept. The support of bodies such as message/sipfrag [13], application/conference-info+xml [5] also indicates support of call control and conferencing.

Allow-Events. The support of event packages such as refer [3], conference [5].

Supported. The support of extensions such as replaces [8].

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.

5. Security Considerations

This document discusses call control for SIP conferencing. Both call control and conferencing have specific security requirements which will be summarized here. Conferences generally have authorization rules about who may or may not join a conference, what type of media may or may not be used, etc. This information is used by the Focus to admit or deny participation in a conference. It is recommended that these types of authorization rules be used to provide security for a SIP conference. For this authorization information to be used,

the focus needs to be able to authenticate potential participants. Normal SIP mechanisms including Digest authentication and certificates can be used. These conference specific security requirements are discussed further in the requirements and framework documents.

For call control security, a user agent must maintain local policy on who is permitted to perform call control operations, initiate REFERS, and replace dialogs. Normal SIP authentication mechanisms are also appropriate here. The specific authentication and authorization schemes are described in the multiparty call control framework document.

6. 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.

7. Changes since -00

- Showed separation between conference factory application and focus by having the application redirect to the newly created focus in the ad-hoc creation scenario.
- Removed inclusion of "isfocus" parameter in Refer-To header field - this may be a useful extension to the REFER mechanism in the future, however.
- Updated reference from Caller Prefs document to the new Capabilities of User Agents document.
- Added scenario of participant changing user agents during a conference.
- Added requirement on focus to support Replaces header field.
- Added discussion about termination of dialog using BYE and subscription using SUBSCRIBE/NOTIFY to flows involving termination of session with the focus.

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] Sparks, R., "The SIP Refer Method", [draft-ietf-sip-refer-07](#) (work in progress), December 2002.
- [4] Roach, A., "Session Initiation Protocol (SIP)-Specific Event Notification", [RFC 3265](#), June 2002.
- [5] Rosenberg, J. and H. Schulzrinne, "A Session Initiation Protocol (SIP) Event Package for Conference State", [draft-ietf-sipping-conference-package-00](#) (work in progress), June 2002.
- [6] Mahy, R. and D. Petrie, "The Session Initiation Protocol (SIP) 'Join' Header", [draft-ietf-sip-join-01](#) (work in progress), March 2003.
- [7] Rosenberg, J., "Indicating User Agent Capabilities in the Session Initiation Protocol (SIP)", [draft-ietf-sip-callee-caps-00](#) (work in progress), June 2003.
- [8] Biggs, B., Dean, R. and R. Mahy, "The Session Initiation Protocol (SIP) 'Replaces' Header", [draft-ietf-sip-replaces-03](#) (work in progress), March 2003.

Informative References

- [9] Levin, O. and R. Even, "High Level Requirements for Tightly Coupled SIP Conferencing", [draft-ietf-sipping-conferencing-requirements-00](#) (work in progress), April 2003.
- [10] Rosenberg, J., "A Framework for Conferencing with the Session Initiation Protocol", [draft-ietf-sipping-conferencing-framework-00](#) (work in progress), May 2003.
- [11] Mahy, R., "A Call Control and Multi-party usage framework for the Session Initiation Protocol (SIP)", [draft-ietf-sipping-cc-framework-02](#) (work in progress), March 2003.
- [12] Campbell, B. and R. Sparks, "Control of Service Context using SIP Request-URI", [RFC 3087](#), April 2001.
- [13] Sparks, R., "Internet Media Type message/sipfrag", [RFC 3420](#), November 2002.

- [14] Johnston, A., "Session Initiation Protocol Basic Call Flow Examples", [draft-ietf-sipping-basic-call-flows-02](#) (work in progress), April 2003.
- [15] Johnston, A. and S. Donovan, "Session Initiation Protocol Service Examples", [draft-ietf-sipping-service-examples-04](#) (work in progress), March 2003.
- [16] Sparks, R. and A. Johnston, "Session Initiation Protocol Call Control - Transfer", [draft-ietf-sipping-cc-transfer-01](#) (work in progress), February 2003.
- [17] Rosenberg, J. and H. Schulzrinne, "An INVITE Initiated Dialog Event Package for the Session Initiation Protocol (SIP", [draft-ietf-sipping-dialog-package-01](#) (work in progress), March 2003.

Authors' Addresses

Alan Johnston
MCI
100 South 4th Street
St. Louis, MO 63102

EMail: alan.johnston@mci.com

Orit Levin
RADVISION
266 Harristown Road
Glen Rock, NJ 75024

EMail: orit@radvision.com

Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any intellectual property 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; neither does it represent that it has made any effort to identify any such rights. Information on the IETF's procedures with respect to rights in standards-track and standards-related documentation can be found in [BCP-11](#). Copies of claims of rights made available for publication 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 implementors or users of this specification can be obtained from the IETF Secretariat.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director.

Full Copyright Statement

Copyright (C) The Internet Society (2003). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assignees.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS 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.

Acknowledgment

Funding for the RFC Editor function is currently provided by the
Internet Society.