Centralized Conferencing Manipulation Protocol (CCMP) Call Flow Examples
draft-barnes-xcon-examples-01.txt

Status of this Memo

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

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

This Internet-Draft will expire on September 11, 2009.

Copyright Notice

Copyright (c) 2009 IETF Trust and the persons identified as the
document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal
Provisions Relating to IETF Documents in effect on the date of
publication of this document (http://trustee.ietf.org/license-info).
Please review these documents carefully, as they describe your rights
and restrictions with respect to this document.
Abstract

This document provides detailed call flows for the scenarios documented in the Centralized Conferencing (XCON) Framework and the XCON Scenarios. The call flows document the use of the interface between a conference control client and a conference control server using the Centralized Conferencing Manipulation Protocol (CCMP). The objective is to provide a base reference for both protocol researchers and developers.

Table of Contents

1. Introduction ............................................. 4
2. Conventions ............................................. 4
3. Terminology ............................................. 4
4. Overview ............................................... 5
5. Conference Creation ..................................... 6
   5.1. Basic Conference Creation .......................... 6
   5.2. Basic Conference Creation for a specific instance of Conference Information .......................... 9
   5.3. Basic Conference Creation - Cloning an existing Conference ............................................. 12
   5.4. Conference Creation using Blueprints .................. 14
6. General Conference scenarios and examples ................. 17
   6.1. Conference Announcements and Recordings ............... 17
   6.2. Monitoring for DTMF .................................. 18
   6.3. Adding a Party ...................................... 19
   6.4. Joining a Conference .................................. 19
   6.5. Muting a Party ...................................... 20
   6.6. Internal Sidebar ...................................... 21
   6.7. External Sidebar ...................................... 23
   6.8. Floor control using sidebars ......................... 24
   6.9. Whispering or Private Messages ....................... 25
   6.10. Observing and Coaching .............................. 26
7. Removing participants and deleting conferences .......... 28
   7.1. Removing a Party ...................................... 28
   7.2. Deleting a Conference ................................. 29
8. Additional Conference Scenarios and Examples ............. 29
   8.1. Chat ................................................. 29
       8.1.1. Basic Chat Operations ............................. 30
       8.1.2. Advanced Operations ............................... 34
9. IANA Considerations ..................................... 37
10. Security Considerations .................................. 37
11. Change Summary ......................................... 38
12. Acknowledgements ...................................... 38
13. References ............................................. 38
    13.1. Normative References ............................... 38
1. Introduction

This document provides detailed call flows for the scenarios documented in the Centralized Conferencing (XCON) Framework [RFC5239] and the XCON Scenarios [RFC4597]. The XCON scenarios describe a broad range of use cases taking advantage of the advanced conferencing capabilities provided by a system realization of the XCON framework. The call flows document the use of the interface between a conference control client and a conference control server using the Centralized Conferencing Manipulation Protocol (CCMP)[I-D.ietf-xcon-ccmp].

Due to the broad range of functionality provided by the XCON Framework and the flexibility of the CCMP messaging, these call flows should not be considered inclusive of all the functionality that can be provided by the XCON Framework and protocol implementations. These flows represent a sample to provide an overview of the feature rich capabilities of the XCON framework and CCMP messaging for protocol developers, software developers and researchers.

2. Conventions

In this document, the key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" are to be interpreted as described in BCP 14, RFC 2119 [RFC2119] and indicate requirement levels for compliant implementations. In this document, these key words are used when describing normative functionality based on the XCON Framework and CCMP.

Note that due to RFC formatting conventions, this document often splits message details whose content would exceed 72 characters. A backslash character marks where this line folding has taken place. This backslash and its trailing CRLF and whitespace would not appear in the actual protocol contents.

3. Terminology

This document uses the same terminology as found in the referenced documents, with the following terms and abbreviations used in the call flows. Also, note that the term "call flows" is used in a very generic sense in this document since the media is not limited to voice. The calls supported by the XCON framework and CCMP can consist of media such as text, voice and video, including multiple media types in a single active conference.
Conferencing and Media Client Client (CMCC): This client may be an integral part of a User Agent Client (UAC) per [RFC3261]. In the flows in this document, the CMCC is logically equivalent to the use of UAC as the client notation in the media control call flows [I-D.miniero-mediactrl-escs].

Conferencing Server (ConfS): In this document, the conferencing server is used interchangeably with the term Application Server (AS) in the Media Control framework [I-D.ietf-mediactrl-architecture] to simplify the call flows. However, these need not be the same entities in an implementation.

Media Server (MS): Media Server. Per its definition in the Media Control Architecture.

4. Overview

This document provides a sampling of detailed call flows that can be implemented based on a system realization of [RFC5239] and implementation of [I-D.ietf-xcon-ccmp]. This is intended to be a simple guide on the use of the conference control protocol between the Conference Server and the Conference Control Client. The objective is to provide an information base reference for protocol developers, software developers and researchers.

This document focuses on the interaction between the Conference (and Media) Control Client and the Conferencing system, specifically the Conference Server. The initial scenarios chosen are based on the ones described in the XCON framework, many of which are based on the advanced conferencing capabilities described in the XCON scenarios. Additional scenarios have been added to provide examples of other real life scenarios that are anticipated to be supported by the framework and to document the conference control that complements the Media Control Call Flows [I-D.miniero-mediactrl-escs] for conferencing.

Rather than repeat the details associated with the media control, this document references the media control call flow examples [I-D.miniero-mediactrl-escs] by Figure title to aid the user in finding the flows in that document. This approach was taken rather than integrating the two documents due to dependencies in completing working group items and because the messages for the two protocols provide fairly discrete operations. In addition, the scenarios for which floor control are used also do not include details of the Binary Floor Control Protocol (BFCP) [RFC4582], but rather refer to that document for further details for clients that also implement BFCP for floor control.
5. Conference Creation

This section provides the details associated with the various ways in which a conference can be created using CCMP and the XCON framework constructs. As previously mentioned the details of the media control and floor control protocols, where applicable, are annotated in the flows without showing all the details. However, for clarification purposes, the first example Section 5.1 provides the details of the media control messaging along with an example of the standard annotation used throughout the remainder of this document. In subsequent flows, only this annotation (identified by lower case letters) is included and the reader is encouraged to either refer back to this first flow or to additional relevant flows in the media control call flow document [I-D.miniero-mediactrl-escs] (e.g., for IVR interactions, etc.).

The call signaling interactions for clients to join or be added to a conference are also not shown, but rather annotated in a manner similar to those for the media control messaging. The annotations for the call signaling are on the left side of the conferencing server vertical bar and those for the media control messaging are on the right side.

The term conferencing server (ConfS) is shown in the diagrams in this document as opposed to the more generic application server (AS) in the media control flow document [I-D.miniero-mediactrl-escs]. Also, note that the term conferencing and media control client (CMCC) in these call flows may be an integral part of a User Agent Client (UAC) per [RFC3261], which is the client notation used in the media control call flow document. However, in the context of XCON, the conferencing and media control client can be independent of the call signaling client.

5.1. Basic Conference Creation

The simplest manner in which a conference can be created is accomplished by the client sending a "confRequest" message with the "create" operation as the only parameter to the conference server. This results in the creation of a default conference, with an XCON-URI in the form of the "confObjID" parameter, the XCON-UserID in the form of the "confUserID" parameter and the data for the conference object in the "confInfo" parameter all returned in the "confResponse" message. This example also adds the user that invoked the conference upon creation (i.e., "method" attribute is set to "dial out" for this client based on the particular conferencing systems default), thus the call signaling interactions to add the CMCC1 to the conference are completed prior to returning the confResponse message. Note, that depending upon the conferencing system, this default conference
could be specific to the client requesting the conference and thus may be different for the initiator than other participants (e.g., IVR interactions in this case which are not shown). The media interactions are also handled just prior to sending the "confResponse" message.

The specific data for the conference object is returned in the "confResponse" message in the "confInfo" parameter. This allows the client (with the appropriate authorization) to manipulate this data and add additional participants to the conference, as well as change the data during the conference. In addition, the client may distribute the conferencing information to other participants allowing them to join. The details of such are provided in additional flows.

Clients that are not XCON-aware may join the conference using a specific signaling interface such as SIP, using the signaling interface to the conference focus as described in [RFC4579]. However, these details are not shown in the message flows. The message flows in this document identify the point in the message flows at which this signaling occurs via the lower case letter items (i.e., (a)...(x)) along with the appropriate text for the processing done by the focus.
<table>
<thead>
<tr>
<th>CMCC1</th>
<th>CMCC2</th>
<th>CMCCx</th>
<th>CONFS</th>
<th>MS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. confRequest</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------------------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(a) Create +---</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Conf</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Object</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| | & IDs +-->
| | A1. CONTROL | |
| | +----------->| |
| | (create conf)--- (b) | |
| | | create | |
| | | conf and | |
| | | A2. 200 OK |<+ its ID | |
| | | <=+ create conf | |
| | | (confid=Y) | |

| (2) confResponse(create, success, | |
| confObjID, confUserID | |
| conf-info) | |
| (c) Focus +--- | |
| | sets up | |
| | signaling | |
| | to CMCC1 +-->
| | B1. CONTROL | |
| | +----------->| |
| | (join CMCC1 | |
| | <+confY) | |
| | +----------+ | |
| | (d) join CMCC1 & | |
| | B2. 200 OK |<+ conf Y | |
| | <=+ conf Y | |
| | | <=+ conf Y | |

<<<<<<<<<<<>><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><<
### 5.2. Basic Conference Creation for a specific instance of Conference Information

A conference can also be created by the client sending a "confRequest" message with the "create" operation, along with the desired data in the form of the "confInfo" parameter for the conference to be created. An example where this approach might be applicable would be in the case where a conference user might need to use a different conferencing system than is typically used (e.g., one that is more geographically appropriate for some participants that
perhaps do not support advanced conferencing functionality). If the specific conferencing system can support that specific type of conference (capabilities, etc.), then the request results in the creation of a conference. In this success case, an XCON-URI in the form of the "confObjID" parameter and the XCON-UserID in the form of the "confUserID" parameter are returned in the "confResponse" message. The "confInfo" is not returned unless changes have been made, in which case the "responseCode" is "modified". This example also activates the conference upon creation, thus the call signaling interactions to add the CMCC to the conference are completed prior to returning the confResponse message. The media interactions handled when the confResponse message is sent.

This example also activates the conference upon creation (i.e., "method" attribute is set to "dial out" for this client based on the particular conferencing systems default), thus the call signaling interactions to add the CMCC to the conference are completed prior to returning the confResponse message. Note, that depending upon the conferencing system, this default conference could be specific to the client requesting the conference and thus may be different for the initiator than other participants (e.g., IVR interactions in this case which are not shown. The media interactions are also handled just prior to sending the "confResponse" message.

```
CMCC "Alice" CMCC "Bob" CMCCx CONFS
            |               |           |           |
            |               |           |           |
            |(1)confRequest |           |           |
            |------------------------|          |
            |(confInfo)             |           |
            |               |           |           |
            |               |           |           |
            |               |           |           |
            |               |           |           |
            |               |           |           |
            |(2) confResponse     |           |
            |------------------------|          |
            |( create,success, confObjID |
            | confUserID, confInfo) |           |--+ (b) MS
            |               |           |        | creates
            |               |           |        | conf and
            |               |           |        | & IDs +-->|
            |               |           |        | <-+ its ID
            |               |           |        | (confid=Y)
            |               |           |        | (c) Focus +-->
            |               |           |        | sets up +-->
            |               |           |        | signaling +-->
            |               |           |        | to Alice +-->|
```
Alice is mixed in the conference

Bob is mixed too

***All parties connected to conf Y***
5.3. Basic Conference Creation - Cloning an existing Conference

A client can also create another conference by cloning an existing conference, such as an active conference or conference reservation. In this example, the client sends a "confRequest" message with the "create" operation, along with a specific "confObjID", from which a new conference is to be created by cloning an existing conference.

An example of how a client can create a conference based on a blueprint is provided in Section 5.4. The manner by which a client in this example might learn about a conference reservation or active conferences is similar to the first step in the blueprint example, with the exception of specifying querying for different types of conference objects supported by the specific conferencing system. For example, in this example, the client clones a conference reservation, thus the client would include the appropriate "confObjState" parameter. [Note: we don't currently have this parameter in the XML schema in CCMP, BUT this functionality is documented in the text.]

If the conferencing system can support a new instance of the specific type of conference(capabilities, etc.), then the request results in the creation of a conference, with an XCON-URI in the form of a new value in the "confObjID" parameter to reflect the newly cloned conference object returned in the "confResponse" message. The "confInfo" is not returned unless there had been changes, in which case the "responseCode" is "modified". This example also activates the conference upon creation, thus the call signaling interactions to add the CMCC to the conference are completed prior to returning the confResponse message. The media interactions handled when the confResponse message is sent.
Figure 6: Create Basic Conference - Clone

1. "Alice" sends a confRequest message to clone a conference based on an existing conference reservation. "Alice" indicates this conference should be a "child" of the parent conference represented by the "confObjID" in the request.

2. Upon receipt of the confRequest message containing a "create" operation and "confObjID", the conferencing system ensures that the "confObjID" received is valid. The conferencing system determines the appropriate read/write access of any users to be added to a conference based on this "confObjID" (using membership, roles, etc.). The conferencing system uses the received "confObjID" to clone a conference reservation. The conferencing system also reserves or allocates a new "confObjID" to be used for the cloned conference object. Any subsequent protocol requests from any of the members of the conference. The conferencing system maintains the mapping between this conference ID and the parent conference object ID associated with the reservation through the conference instance.

(CCMP Messaging details not available yet).
5.4. Conference Creation using Blueprints

Figure 8 provides an example of one client "Alice" determining the conference blueprints available for a particular conferencing system and creating a conference based on the desired blueprint.
Figure 8: Client Creation of Conference using Blueprints

1. "Alice" first sends an "optionsRequest" message to the conferencing system identified by the conference server discovery
process (details TBD). Upon receipt of the "optionsRequest", the conferencing system would first authenticate "Alice" (and allocate a conference user identifier, if necessary) and then ensure that "Alice" has the appropriate authority based on system policies to receive any blueprints supported by that system. Any blueprints that "Alice" is authorized to use are returned in a "optionsResponse" message in the "blueprints" attribute, along with the "confUserID" parameter.

2. Upon receipt of the "optionsResponse" containing the blueprints, "Alice" determines which blueprint to use for the conference to be created. "Alice" creates a conference object based on the blueprint (i.e., clones) and modifies applicable fields, such as membership list and start time.

3. "Alice" then sends a "confRequest" with a "create" operation to the conferencing system to create a conference reservation based upon the updated blueprint, including the appropriate "blueprintName" and associated "confObjID".

   Note: This conference is created as independent of the parent (blueprint), but there are no hard and fast requirements as to whether conference from blueprints are always independent or whether the conferences cloned from conference reservations or active conferences are also children. The protocol is flexible enough to allow all the variations, thus any limitations would be specific to a conferencing system.

Upon receipt of the "confRequest" message with a "create" operation and an "action" to "reserve" a conference based upon the blueprint in the request, the conferencing system ensures that the blueprint received is a valid blueprint (i.e. the values of the various field are within range). [Note: we don't currently have this "action" field defined for the "confRequest" message.] The conferencing system determines the appropriate read/write access of any users to be added to a conference based on this blueprint (using membership, roles, etc.). The conferencing system uses the received blueprint to clone a conference reservation. The conferencing system also reserves or allocates a conference ID to be used for any subsequent CCMP requests from any of the members of the conference. The conferencing system maintains the mapping between this conference ID and the "confObjID" associated with the reservation through the conference instance.

4. The conferencing server then sends a "confResponse" message including the "confObjID" associated with the reserved conference. Upon receipt of the "confResponse" message, "Alice" can now create an active conference using that reservation or create additional
reservations based upon the existing reservation.

5. In this example, "Alice" has reserved a meetme conference bridge. Thus, "Alice" provides the conference information, including the necessary "confObjID", to desired participants. Note, that this interface is entirely outside the scope of the XCON framework, protocols and this document. When the first participant, "Alice" in this example, then requests to be added to the conference by sending a "userRequest" ....

6. Upon receipt of the "userRequest" message, the conference is activated and the focus is created. The focus is associated with the "confObjID" received in the request. Any participants that have the authority to manipulate the conference would receive the "confObjID" in any responses. The conference server then sends "userResponse" message....

(CCM Messaging details not available yet).

Figure 9: Create Conference (Blueprint) Detailed Messaging

6. General Conference scenarios and examples

The following scenarios are based on those documented in the XCON framework. The examples assume that a conference has already been correctly established, with media, if applicable, per one of the examples in Section 5.

6.1. Conference Announcements and Recordings

In this example, as shown in Figure 10 "Alice" is joining "Bob"'s conference that requires that she first enter a pass code. After successfully entering the passcode, an announcement prompts "Alice to speak her name so it can be recorded. When "Alice" is added to the active conference, the recording is played back to all the existing participants.

(Figure not available yet).

Figure 10: Recording and Announcements
1. Upon receipt of the userRequest from "Alice" to be added to "Bob's" conference, the conferencing system maps the identifier received in the request to the conference object representing "Bob's" active conference. The conferencing system determines that a password is required for this specific conference, thus an announcement asking "Alice" to enter the password is provided to "Alice". Once "Alice" enters the password, it is validated against the policies associated with "Bob's" active conference. The conferencing system then connects to a server which prompts and records "Alice's" name. The conferencing system must also determine whether "Alice" is already a user of this conferencing system or whether she is a new user.

2. "Alice" is a new user for this conferencing system, so a conference user identifier is created for "Alice". Based upon the addressing information provided by "Alice", the call signaling to add "Alice" to the conference is instigated through the Focus. In addition, "Alice" is sent a userResponse message which includes the confUserID assigned by the conferencing system for "Alice". This would allow "Alice" to later perform operations on the conference (if she were to have the appropriate policies), including registering for event notifications associated with the conference.

3. Once the call signaling indicates that "Alice" has been successfully added to the specific conference, per updates to the state, and depending upon the policies, other participants (e.g., "Bob") are notified of the addition of "Alice" to the conference via the conference notification service and an announcement is provided to all the participants indicating that "Alice" has joined the conference.

(CCMP Messaging details not available yet).

Figure 11: Announcement Messaging Details

6.2. Monitoring for DTMF

The conferencing system also needs the capability to monitor for DTMF from each individual participant. This would typically be used to enter the identifier and/or access code for joining a specific conference.

An example of DTMF monitoring, within the context of the framework elements, is shown in Figure 10.
6.3. Adding a Party

Figure 12 provides an example of one client "Alice" impacting the state of another client "Bob". This example assumes an established conference. In this example, "Alice" wants to add "Bob" to the conference.

To do.

Figure 12: Client Manipulation of Conference - Add a party

1. Upon receipt of the Conference Control Protocol request to "add" a party ("Bob") in the specific conference as identified by the conference object ID, the conferencing system ensures that "Alice" has the appropriate authority based on the policies associated with that specific conference object to perform the operation. The conferencing system must also determine whether "Bob" is already a user of this conferencing system or whether he is a new user.

2. If "Bob" is a new user for this conferencing system, a Conference User Identifier is created for Bob. Based upon the addressing information provided for "Bob" by "Alice", the call signaling to add "Bob" to the conference is instigated through the Focus.

3. Once the call signaling indicates that "Bob" has been successfully added to the specific conference, per updates to the state, and depending upon the policies, other participants (including "Bob") may be notified of the addition of "Bob" to the conference via the Conference Notification Service.

(CCMP Messaging details not available yet).

Figure 13: Add Party Message Details

6.4. Joining a Conference

Figure 14 provides an example of one client "Duck" joining an active conference with "Alice", "Bob" and "Claire" as participants. Using SIP as a call control protocol such as SIP, "Duck" joins the conference without using any CCMP messaging since the required interactions are specific to the conferencing system via a trigger from the focus upon receipt of the SIP message. The conferencing system does the following to join "Duck" to the active conference:
adds "Duck" as a user, authorizes "Duck" to join the conference, modifies the appropriate conference data, and provides the notifications to the participants that have registered for such.

To do.

Figure 14: Client Joining an Active Conference

1. Upon receipt of the SIP request to "join" a party ("Duck") to the specific conference as identified by the Focus. The conferencing system determines the appropriate conference object ID. The conferencing system then determines whether "Bob" is already a user of this conferencing system or whether he is a new user. If "Bob" is a new user for this conferencing system, a Conference User Identifier is created for Bob. Based upon the addressing information provided for "Bob" by "Alice", the call signaling to add "Bob" to the conference is instigated through the Focus.

2. Once the call signaling indicates that "Duck" has been successfully added to the specific conference, per updates to the state, and depending upon the policies, other participants (including "Duck") may be notified of the addition of "Duck" to the conference via the Conference Notification Service.

(CCMP Messaging details not available yet).

Figure 15: Join Message Details

6.5. Muting a Party

This section provides an example of the muting of a party in an active conference. The unmuting would involve the identical CCMP message flow. Although, in the case that floor control is involved, whether or not a particular conference client can unmute themselves must be considered by the conferencing system.

Figure 16 provides an example of one client "Alice" impacting the media state of another client "Bob". This example assumes an established conference. In this example, the client, "Alice" whose Role is "moderator" of the conference, wants to mute "Bob" on a medium-size multi-party conference, as his device is not muted (and he's obviously not listening to the call) and background noise in his office environment is disruptive to the conference.
1. Upon receipt of the Conference Control Protocol request to "mute" a party ("Bob") in the specific conference as identified by the conference object ID, the Conference Server ensures that "Alice" has the appropriate authority based on the policies associated with that specific conference object to perform the operation. "Bob"'s status is marked as "recvonly" and the conference object is updated to reflect that "Bob"'s media is not to be "mixed" with the conference media. In case the Conference Server relies on a remote Media Server for its multimedia functionality, it subsequently changes "Bob"'s media profile accordingly by means of the related protocol interaction with the MS. An example describing a possible way of dealing with such a situation using the Media Server Control architecture is described in [I-D.miniero-mediactrl-escs], at "Simple Bridging: Framework Transactions (2)".

2...x. Depending upon the policies, other participants (including "Bob") may be notified of this change via the Conference Notification Service.

(CCMP Messaging details not available yet).

6.6. Internal Sidebar

Figure 18 provides an example of one client "Alice" involved in active conference with "Bob" and "Carol". "Alice" wants to create a sidebar to have a side discussion with "Bob" while still viewing the video associated with the main conference. Alternatively, the audio from the main conference could be maintained at a reduced volume. "Alice" initiates the sidebar by sending a request to the conferencing system to create a conference reservation based upon the active conference object. "Alice" and "Bob" would remain on the roster of the main conference, such that other participants could be aware of their participation in the main conference, while an internal-sidebar conference is occurring.
(To be added).

Figure 18: Client Creation of a Sidebar Conference

1. Upon receipt of the Conference Control Protocol request to "reserve" a new sidebar conference, based upon the active conference received in the request, the conferencing system uses the received active conference to clone a conference reservation for the sidebar. The sidebar reservation is NOT independent of the active conference (i.e., parent). The conferencing system also reserves or allocates a conference ID to be used for any subsequent protocol requests from any of the members of the conference.

2. Upon receipt of the conference control protocol response to reserve the conference, "Alice" can now create an active conference using that reservation or create additional reservations based upon the existing reservations. In this example, "Alice" wants only "Bob" to be involved in the sidebar, thus she manipulates the membership. "Alice" also only wants the video from the original conference and wants the audio to be restricted to the participants in the sidebar. Alternatively, "Alice" could manipulate the media values to receive the audio from the main conference at a reduced volume. "Alice" sends a conference control protocol request to update the information in the reservation and to create an active conference.

3. Upon receipt of the conference control protocol request to update the reservation and to create an active conference for the sidebar, as identified by the conference object ID, the conferencing system ensures that "Alice" has the appropriate authority based on the policies associated with that specific conference object to perform the operation. The conferencing system must also validate the updated information in the reservation, ensuring that a member like "Bob" is already a user of this conferencing system.

4...x. Depending upon the policies, the initiator of the request (i.e., "Alice") and the participants in the sidebar (i.e., "Bob") may be notified of his addition to the sidebar via the conference notification service.

(CCMP Messaging details not available yet).

Figure 19: Internal Sidebar Messaging Details
6.7. External Sidebar

Figure 20 provides an example of one client "Alice" involved in an active conference with "Bob", "Carol", "David" and "Ethel". "Alice" gets an important text message via a whisper from "Bob" that a critical customer needs to talk to "Alice", "Bob" and "Ethel". "Alice" creates a sidebar to have a side discussion with the customer "Fred" including the participants in the current conference with the exception of "Carol" and "David", who remain in the active conference. "Alice" initiates the sidebar by sending a request to the conferencing system to create a conference reservation based upon the active conference object. "Alice", "Bob" and "Ethel" would remain on the roster of the main conference in a hold state. Whether or not the hold state of these participants is visible to other participants depends upon the individual and local policy.

(To be Detailed).

Figure 20: Client Creation of an External Sidebar

1. Upon receipt of the Conference Control Protocol request to "reserve" a new sidebar conference, based upon the active conference received in the request, the conferencing system uses the received active conference to clone a conference reservation for the sidebar. The sidebar reservation is NOT independent of the active conference (i.e., parent). The conferencing system also reserves or allocates a conference ID to be used for any subsequent protocol requests from any of the members of the conference. The conferencing system maintains the mapping between this conference ID and the conference object ID associated with the sidebar reservation through the conference instance.

2. Upon receipt of the conference control protocol response to reserve the conference, "Alice" wants only "Bob" and "Ethel", along with the new participant "Fred" to be involved in the sidebar, thus she manipulates the membership. "Alice" sets the media in the conference-info such that the participants in the sidebar don't receive any media from the main conference.

3. "Alice" sends a conference control protocol request to update the information in the reservation and to create an active conference.

4. Upon receipt of the conference control protocol request to update the reservation and to create an active conference for the sidebar
the conferencing system ensures that "Alice" has the appropriate authority based on the policies associated with that specific conference object to perform the operation. The conferencing system also validates the updated information in the reservation. Since "Fred" is a new user for this conferencing system, a conference user identifier is created for "Fred". Based upon the addressing information provided for "Fred" by "Alice", the call signaling to add "Fred" to the conference is instigated through the Focus.

5...x. Depending upon the policies, the initiator of the request (i.e., "Alice") and the participants in the sidebar (i.e., "Bob" and "Ethel") may be notified of his addition to the sidebar via the conference notification service.

(CCMP Messaging details not available yet).

Figure 21: External Sidebar Messaging Details

### 6.8. Floor control using sidebars

Floor control with sidebars can be used to realize conferencing scenario such as an analyst briefing. In this scenario, the conference call has a panel of speakers who are allowed to talk in the main conference. The other participants are the analysts, who are not allowed to speak unless they have the floor. To request access to the floor, they have to join a new sidebar with the moderator and ask their question. The moderator can also whisper to each analyst what their status/position in the floor control queue, similar to the example in Figure 24. It should be noted that other mechanisms which don't make use of sidebars could be used for floor control such as those detailed in BFCP. [Editor's note: Should we add detailed flows for BFCP to this document and show additional floor control scenarios?

Figure 22 provides an example of the configuration involved for this type of conference. As in the previous sidebar examples, there is the main conference along with a sidebar. "Alice" and "Bob" are the main participants in the conference, with "A1", "A2" and "A3" representing the analysts. The sidebar remains active throughout the conference, with the moderator, "Carol", serving as the chair. As discussed previously, the sidebar conference is NOT independent of the active conference (i.e., parent). The analysts are provided the conference object ID associated with the active sidebar when they join the main conference. The conferencing system also allocates a conference ID to be used for any subsequent manipulations of the
sidebar conference. The conferencing system maintains the mapping between this conference ID and the conference object ID associated with the active sidebar conference through the conference instance. The analysts are permanently muted while in the main conference. The analysts are moved to the sidebar when they wish to speak. Only one analyst is given the floor at a given time. All participants in the main conference receive audio from the sidebar conference, as well as audio provided by the panelists in the main conference.

(To Be added).

Figure 22: Floor Control with sidebars

1. "A1" wishes to ask a question, so he sends a Floor Request message to the floor control server.

2. Upon receipt of the request, the floor control server notifies the moderator, "Carol" of the active sidebar conference, whose serving as the floor chair.

3. Since no other analysts have yet requested the floor, "Carol" indicates to the floor control server that "A1" may be granted the floor.

(CCMP Messaging details not available yet).

Figure 23: Floor Control Messaging Details

6.9. Whispering or Private Messages

The case of private messages can be handled as a sidebar with just two participants, similar to the example in section Section 6.6, but rather than using audio within the sidebar, "Alice" could add an additional text based media stream to the sidebar. The other context, referred to as whisper, in this document refers to situations involving one time media targetted to specific user(s). An example of a whisper would be an announcement injected only to the conference chair or to a new participant joining a conference.

Figure 24 provides an example of one user "Alice" whose chairing a
fixed length conference with "Bob" and "Carol". The configuration is such that only the chair is providing a warning when there is only 10 minutes left in the conference. At that time, "Alice" is moved into a sidebar created by the conferencing system and only "Alice" receives the announcement.

(To Be completed).

Figure 24: Whisper

1. When the conferencing system determines that there is only 10 minutes left in the conference which "Alice" is chairing, the conferencing system directly creates an active sidebar conference, based on the active conference associated with "Alice". This sidebar conference is NOT independent of the active conference (i.e., parent). The conferencing system also allocates a conference ID to be used for any subsequent manipulations of the sidebar conference.

2. Immediately upon creation of the active sidebar conference, the announcement media is provided to "Alice". Depending upon the policies, Alice may be notified of her addition to the sidebar via the conference notification service. "Alice" continues to receive the media from the main conference.

3. Upon completion of the announcement, "Alice" is removed from the sidebar and the sidebar conference is deleted.

4. "Alice" is notified of her removal from the sidebar via the conference notification service.

(CCMP Messaging details not available yet).

Figure 25: Whisper Messaging Details

6.10. Observing and Coaching

An example of observing and coaching is shown in figure Figure 26. In this example, call center agent "Bob" is involved in a conference with customer "Carol". Since "Bob" is a new agent and "Alice" sees that he has been on the call with "Carol" for longer than normal, she decides to observe the call and coach "Bob" as necessary.
Upon receipt of the Conference Control Protocol request from "Alice" to "reserve" a new sidebar conference, based upon the active conference received in the request, the conferencing system uses the received active conference to clone a conference reservation for the sidebar. The conferencing system also reserves or allocates a conference ID to be used for any subsequent protocol requests from any of the members of the conference. The conferencing system maintains the mapping between this conference ID and the conference object ID associated with the sidebar reservation through the conference instance.

Upon receipt of the conference control protocol response to reserve the conference, "Alice" can now create an active conference using that reservation or create additional reservations based upon the existing reservations. In this example, "Alice" wants only "Bob" to be involved in the sidebar, thus she manipulates the membership. "Alice" also wants the audio to be received by herself and "Bob" from the original conference, but wants any outgoing audio from herself to be restricted to the participants in the sidebar, whereas "Bob's" outgoing audio should go to the main conference, so that both "Alice" and the customer "Carol" hear the same audio from "Bob". "Alice" sends a conference control protocol request to update the information in the reservation and to create an active conference.

Upon receipt of the conference control protocol request to update the reservation and to create an active conference for the sidebar, as identified by the conference object ID, the conferencing system ensures that "Alice" has the appropriate authority based on the policies associated with that specific conference object to perform the operation. Based upon the addressing information provided for "Bob" by "Alice", the call signaling to add "Bob" to the sidebar with the appropriate media characteristics is instigated through the Focus.

"Bob" is notified of his addition to the sidebar via the conference notification service, thus he is aware that "Alice" the supervisor is available for coaching him through this call.

(CCMP Messaging details not available yet).
7. Removing participants and deleting conferences

The following scenarios detail the basic operations associated with removing participants from conferences and entirely deleting conferences. The examples assume that a conference has already been correctly established, with media, if applicable, per one of the examples in Section 5.

7.1. Removing a Party

Figure 28 provides an example of one client "Alice" removing another participant "Bob" from a conference. This example assumes an established conference with "Alice", "Bob", "Claire" and "Duck". In this example, "Alice" wants to remove "Bob" from the conference so that the group can continue in the same conference without "Bob"'s participation.

1. Upon receipt of the confUsersRequest message, with a "change" operation to remove "Bob" from the "allowed-users-list" for the conference identified by the "confObjID" in the request, the conferencing system ensures that "Alice" has the appropriate authority based on the policies associated with that specific conference object to perform the operation.

2. Based upon the addressing and media information in the conference object for "Bob" in the "user" element, the conferencing system instigates the process to remove "Bob" (e.g., the call signaling to remove "Bob" from the conference is instigated through the Focus). In addition, the "conference-info" in the conference object is modified to remove "Bob" from the "users" list.

3. Once the call signaling indicates that "Bob" has been successfully removed from the specific conference, per updates to the state, and depending upon the policies, other participants (including "Bob") may be notified of the removal of "Bob" from the conference via the Conference Notification Service.
7.2. Deleting a Conference

Details to be added.

(Figure not available yet).

Figure 30: Deleting a conference

(Text description to be added).

(CCMP Messaging details not available yet).

Figure 31: Deleting a Conference Messaging Details

8. Additional Conference Scenarios and Examples

The following are additional scenarios making use of the XCON framework and associated protocols. In some cases, these examples make use of some of the building block scenarios detailed in the previous example sections, in which case the appropriate scenario is referenced rather than duplicating details. In addition, in cases where the scenarios make use of other protocols, as in the previous section, the appropriate reference in the form of a title to the specific flow in the appropriate protocol document is included.

8.1. Chat

The chat functionality described in this section of the document allows clients that use the XCON framework and protocols for other media types (e.g. voice/video) to utilize the same conference control mechanisms and conferencing system to establish, update and delete a conference instance associated with an Instant Messaging (IM) chat session, independent of the IM chat protocol. In some cases (e.g., Message Session Relay Protocol (MSRP) chat), this would provide additional capabilities, such as sidebars. This approach also allows the conferencing system to provide a natural interworking point for various IM protocols, the details of the interworking are outside the
scope of this document.

An IM client wishing to join a conference uses standardized centralized conferencing mechanisms for creating and joining a conference, as identified in the previous sections. The request to send an IM to an IM media session is specific to the IM protocol (e.g., MSRP SEND), just as there is specific media control messaging for other types of sessions. An IM client connecting to a conferencing system has a 1:1 relationship with the IM media signaling entity in the conferencing system. This relationship is referred to as an IM session. Further details of the correlation of the IM session identifiers with the XCON session identifiers is provided in [I-D.boulton-xcon-session-chat]. The IM media signaling entity is responsible for distribution of all the messages to the other participants.

As with the other example conferences created, each IM session is logically associated with a specific conference. The conference itself has a specific identifier in the form of the XCON-URI, which is passed in the "confObjID" element in the CCMP messages. This provides the relevant association between IM session and a centralized conference.

An IM client wishing to delete a chat room uses standardized mechanisms for deleting a conference instance, such as those detailed in Section 7.2.

8.1.1. Basic Chat Operations

This section provides details of the realization of the Multi-party IM (chat) within the context of the centralized conferencing framework. A brief discussion and diagrams are provided for creating, joining, and deleting a chat based conference. The discovery of chat rooms available on a specific conferencing system is inherent in the blueprint capability provided by the conferencing system. The objective of this section is to further illustrate the model, mechanisms and protocols presented in the previous sections and also serves to validate that the model, mechanisms and protocols are sufficient to support IM chat.

It should be noted that not all entities impacted by the request are shown in the diagram (e.g., Focus), but rather the emphasis is on the new entities introduced by this centralized conferencing framework.

8.1.1.1. Creating a Chat Room

There are different ways to create a conference. A participant can create a conference using call signaling means only, such as SIP, as
detailed in [RFC4579]. For a conferencing client to have more flexibility in defining the characteristics and capabilities of a chat-based conference, a conferencing client would implement a conference control protocol client. By using a conference control protocol, the client can determine the capabilities of a conferencing system and its various resources.

Figure 32 provides an example of one client "Alice" determining the conference blueprints available to support various types of chat rooms for a particular conferencing system and creating a chat-based conference using the desired blueprint.

Details to be added.

Upon receipt of the Conference Control Protocol request for blueprints associated with chat rooms, the conferencing system would first authenticate "Alice" (and allocate a conference user identifier, if necessary) and then ensure that "Alice" has the appropriate authority based on system policies to receive any chat room based blueprints supported by that system. Any blueprints that "Alice" is authorized to use are returned in a response, along with the conference user ID.

Upon receipt of the Conference Control Protocol response containing the blueprints, "Alice" determines which blueprint to use for the conference to be created. "Alice" creates a conference object based on the blueprint (i.e., clones) and modifies applicable fields, such as membership list, topic details, and start time. "Alice" then sends a request to the conferencing system to create a conference reservation based upon the updated blueprint.

Upon receipt of the Conference Control Protocol request to "create" a conference based upon the blueprint in the request, the conferencing system ensures that the blueprint received is a valid blueprint (i.e. the values of the various field are within range). The conferencing system determines the appropriate read/write access of any users to be added to a conference based on this blueprint (using membership, roles, etc.). The conferencing system uses the received blueprint to clone a conference reservation. The conferencing system also reserves or allocates a conference ID to be used for any subsequent protocol requests from any of the members of the conference. The
The conferencing system maintains the mapping between this conference ID and the conference object ID associated with the reservation through the conference instance.

Upon receipt of the conference control protocol response to reserve the conference, "Alice" now creates an active chat room using that reservation. "Alice" provides the conference information, including the necessary conference ID, to desired participants to allow them to join the chat room. "Alice" may also add other users to the chat room. When the first participant, including "Alice", requests to be added to the conference, an active conference and focus are created. The focus is associated with the conference ID received in the request.

(CCMP Messaging details not available yet. Plan is to reference detailed flows in previous sections and add MSRP messaging in the example.)

Figure 33: Chatroom Creation Messaging Details

8.1.1.2. Joining a Chat Room

A participant can join and leave the conference using call signaling means only, such as SIP. However, in order to perform richer conference control a user client can implement a conference control protocol client. By using a conference control protocol, the client can affect its own state and the state of other participants, depending upon policies, which may indirectly affect the state of any of the conference participants.

In the example in section Section 8.1.1.1, "Alice" has reserved a chat room. "Alice" has also already joined the conference and made the chat room active. "Alice" can either add additional participants to the chat room or provide the conference information, including the necessary conference ID, to desired participants and allow them to request to join themselves. Any participants that have the authority to manipulate the conference would receive the conference object identifier of the active conference object in the response to their request to join.

Figure 34 provides an example of "Bob" joining the chat room using the conference ID provided by "Alice" (e.g., in an IM).
Details to be added.

Figure 34: Joining a chat room

Upon receipt of the Conference Control Protocol request to "add" a party ("Bob") in the specific conference as identified by the conference object ID, the conferencing system must determine whether "Bob" is already a user of this conferencing system or whether he is a new user. If "Bob" is a new user for this conferencing system, a Conference User Identifier is created for Bob. The conferencing system must also ensure that "Bob" has the appropriate authority based on the policies associated with that specific conference object to perform the operation.

Once "Bob" has been successfully added to the chat room, a response is sent to "Bob". Depending upon the policies, other participants (including "Bob") may be notified of the addition of "Bob" to the conference via the Conference Notification Service.

(CCMP Messaging details not available yet.
Plan is to reference detailed flows in previous sections as appropriate and add MSRP messaging in the example.)

Figure 35: Chatroom Join Messaging Details

8.1.1.3. Deleting a Chat Room

Depending upon the conferencing system policies and policies specific to the chat room, the creator of the chat would typically be the participant authorized to delete the chat room.

In the example in section Section 8.1.1.1, "Alice" has created a chat room and provided the conference information, including the necessary conference ID, to desired participants and allow them to request to join themselves. "Bob" and others are participants in the chat. Figure 36 provides an example of "Alice" later deleting this same chat room.

Details to be added.

Figure 36: Deleting a chat room
Upon receipt of the Conference Control Protocol request to "delete" the specific chat room as identified by the conference object ID, the conferencing system must determine whether "Alice" has the authority to delete this conference. Since "Alice" is the creator of the conference, the "delete" operation is performed, with the appropriate signaling sent to the participants, including a response to "Alice" indicating that the chat room has been deleted.

One step in the deletion of the chat room may include notifying the participants (including "Bob") that they have been removed via the Conference Notification Service.

(CCMP Messaging details not available yet. Plan is to reference detailed flows in previous sections and add MSRP messaging in the example.)

Figure 37: Chatroom Deletion Messaging Details

8.1.2. Advanced Operations

This section provides details of the realization of advanced chat features, such as sidebars and private messages, within the context of the centralized conferencing framework. As with Section 8.1.1, the objective of this section is to further illustrate the model, mechanisms and protocols presented in the previous sections and also serves to validate that the model, mechanisms and protocols are sufficient to support advance IM chat features.

8.1.2.1. Text Sidebar

The concept of a 'sidebar' in conferencing system is fully described in the Sidebar section and related subsections within the Conferencing Scenarios Realization section of the centralized conferencing framework document [RFC5239]. The creation, manipulation and deletion of sidebars for chat rooms follows the same principles.

A conference object representing a sidebar is created by cloning the parent associated with the existing conference and updating any information specific to the sidebar. A sidebar conference object is implicitly linked to the parent conference object (i.e. it is not an independent object) and is associated with the parent conference object identifier. A conferencing system manages and enforces the parent and appropriate localized restrictions on the sidebar.
conference object (e.g., no members from outside the parent conference instance can join, sidebar conference can not exist if parent conference is terminated, etc.).

Figure 38 provides an example of one client "Alice" involved in active chat room with "Bob" and "Carol". "Alice" wants to create a sidebar to have a side discussion with "Bob" while still receiving the session based messaging associated with the main chat room. Whether the text is interleaved with the main chat or whether a separate window is created for the sidebar is implementation specific. "Alice" initiates the sidebar by sending a request to the conferencing system to create a conference chat reservation based upon the active chat conference object. "Alice" and "Bob" would remain on the roster of the main conference, such that other participants could be aware of their participation in the main conference, while the text sidebar conference is occurring.

Details to be added.

Figure 38: Client Creation of a Sidebar Conference

Upon receipt of the Conference Control Protocol request to "reserve" a new sidebar chat conference, based upon the active chat conference received in the request, the conferencing system uses the received active chat conference to clone a conference chat reservation for the sidebar. As discussed previously, the sidebar reservation is NOT independent of the active conference (i.e., parent). The conferencing system also reserves or allocates a conference ID to be used for any subsequent protocol requests from any of the members of the conference. The conferencing system maintains the mapping between this conference ID and the conference object ID associated with the sidebar reservation through the conference instance.

Upon receipt of the conference control protocol response to reserve the conference, "Alice" can now create an active chat conference using that reservation or create additional reservations based upon the existing reservations. In this example, "Alice" wants only "Bob" to be involved in the sidebar, thus she manipulates the membership. "Alice" also only wants the text from the original conference, but wants the text within the sidebar to be restricted to the participants in the sidebar. "Alice" sends a conference control protocol request to update the information in the reservation and to create an active conference.
Upon receipt of the conference control protocol request to update the reservation and to create an active chat conference for the sidebar, as identified by the conference object ID, the conferencing system ensures that "Alice" has the appropriate authority based on the policies associated with that specific conference object to perform the operation. The conferencing system must also validate the updated information in the reservation, ensuring that a member like "Bob" is already a user of this conferencing system.

Depending upon the policies, the initiator of the request (i.e., "Alice") and the participants in the sidebar (i.e., "Bob") may be notified of his addition to the sidebar via the conference notification service.

(CCMP Messaging details not available yet. Plan is to reference detailed flows in previous sections.)

Figure 39: Chatroom Sidebar Messaging Details

**8.1.2.2. Private Message**

The case of private messages can be handled as a sidebar with just two participants, identical to the example in section Section 8.1.2.1. The other context, referred to as whisper, in this document refers to situations involving one time media targeted to specific user(s). An example of a whisper would be a text message injected only to the conference chair or to a new participant joining a conference.

Figure 40 provides an example of one user "Alice" who's chairing a fixed length conference with "Bob" and "Carol". The configuration is such that only the chair is providing a warning when there is only 10 minutes left in the conference. At that time, "Alice" is moved into a sidebar created by the conferencing system and only "Alice" receives that text message announcing the 10 minute warning.

Details to be added.

Figure 40: Whisper
When the conferencing system determines that there is only 10 minutes left in the conference which "Alice" is chairing, rather than creating a reservation as was done for the sidebar in Section 8.1.2.1, the conferencing system directly creates an active chat sidebar conference, based on the active chat conference associated with "Alice". As discussed previously, the sidebar conference is NOT independent of the active conference (i.e., parent). The conferencing system also allocates a conference ID to be used for any subsequent manipulations of the sidebar chat conference. The conferencing system maintains the mapping between this conference ID and the conference object ID associated with the active sidebar conference through the conference instance.

Immediately upon creation of the active chat sidebar conference, the text announcement is provided to "Alice". Depending upon the policies, Alice may be notified of her addition to the sidebar via the conference notification service. "Alice" continues to receive the text messages from the main conference.

Upon delivery of the text announcement, "Alice" is removed from the sidebar and the sidebar conference is deleted. Depending upon the policies, "Alice" may be notified of her removal from the sidebar via the conference notification service.

(CCMP Messaging details not available yet. Plan is to reference detailed flows in previous sections.)

Figure 41: Chatroom Sidebar Messaging Details

9. IANA Considerations

This document has no IANA considerations.

10. Security Considerations

The security considerations applicable to the implementation of these call flows is documented in the XCON Framework, with additional security considerations documented in the CCMP document. Where applicable, statements with regards to the necessary security are discussed in particular flows, however, since this is only an informational document, readers are strongly recommended to carefully consider the security considerations defined in the XCON Framework.
and the CCMP document.

11. Change Summary

The following are the major changes between the 00 and the 01 versions of the draft:

- TBD based on WG feedback.

12. Acknowledgements

The detailed content for this document is derived from the prototype work of Lorenzo Miniero, Simon Pietro-Romano, Tobia Castaldi and their colleagues at the University of Napoli.

13. References

13.1. Normative References


13.2. Informative References


draft-ietf-mediactrl-sip-control-framework-10 (work in progress), February 2009.

[I-D.boulton-mmusic-sdp-control-package-attribute]
Boulton, C., "A Session Description Protocol (SDP) Control Package Attribute",

[I-D.boulton-ivr-control-package]
draft-boulton-ivr-control-package-06 (work in progress), February 2008.

[I-D.boulton-conference-control-package]
Boulton, C., Melanchuk, T., McGlashan, S., and A. Shiratzky, "A Conference Control Package for the Media Control Channel Framework",

[I-D.miniero-bfcp-control-package]
draft-miniero-bfcp-control-package-01 (work in progress), July 2008.


[I-D.ietf-simple-chat]
Authors' Addresses

Mary Barnes
Nortel
2201 Lakeside Blvd
Richardson, TX

Email: mary.barnes@nortel.com

Chris Boulton
NS-Technologies

Email: chris@ns-technologies.com

Lorenzo Miniero
University of Napoli
Via Claudio 21
Napoli 80125
Italy

Email: lorenzo.miniero@unina.it

Simon Pietro Romano
University of Napoli
Via Claudio 21
Napoli 80125
Italy

Email: spromano@unina.it