Centralized Conference Control Protocol (CCCP)
draft-levin-xcon-cccp-00

Status of this Memo

This document is an Internet-Draft and is subject to all provisions of section 3 of RFC 3667. By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she become aware will be disclosed, in accordance with RFC 3668.

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 April 18, 2005.

Copyright Notice

Copyright (C) The Internet Society (2004).

Abstract

This document defines a new client-server Centralized Conferencing Control Protocol (CCCP) for manipulating a conference behavior by either a conference participant or otherwise authorized entity that implements a CCCP client. By implementing a CCCP server, a conference focus provides a means for its clients to control the conference policy and the state of the focus and other participants in the conference.
Table of Contents

1. Introduction ........................................ 3
2. Terminology .......................................... 3
3. Background ........................................... 3
4. Data Model ........................................... 4
   4.1 Relationships between Membership and Media Information ... 5
   4.2 Conference Policy .................................. 6
   4.3 Conference State .................................. 6
   4.4 Policy and State Dependencies ...................... 7
5. The Protocol .......................................... 7
   5.1 The Transaction Model .............................. 7
   5.2 XML ................................................ 7
   5.3 Example ............................................ 9
6. IANA Considerations .................................. 11
7. Security Considerations .............................. 11
8. Acknowledgements ..................................... 11
9. References ............................................ 11
   9.1 Normative References ............................... 11
   9.2 Informative References ............................. 11
   Authors’ Addresses ................................... 12
   Intellectual Property and Copyright Statements .......... 13
1. Introduction

General centralized conferencing architecture is described in the SIPPING Conference Framework [5]. The XCON Conference Framework [4] extends and expands upon the SIPPING conference framework architecture to provide a protocol agnostic centralized conferencing system, defining how it maps into the XCON entities and protocols and providing a related data model. The framework documents define the concept of a conference policy and a conference state as the data model for representing all the information about a particular conference instance.

This document defines the protocol for manipulation of this data model (both the policy and the state of a particular conference) in a system built according to the XCON architecture.

This document extends the conference state information (defined in the SIPPING Conference Package [2]) to be reused as the data model for CCCP.

2. Terminology

In this document, the key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT REQUIRED", "MAY", and "OPTIONAL" are to be interpreted as described in BCP 14, RFC 2119 [1] and indicate requirement levels for compliant implementations.

3. Background

Today XCON defines a policy XML schema [8], and relies on data manipulation of the policy document to indirectly influence the conference state. The policy updates can be done anytime, before or during a conference.

CCCP approach doesn't preclude the use of CPCP for the manipulation of the data outside an "active" conference instance. Conference policy changes may also need to be made during a conference, but they are likely to be much less frequent than conference state changes. Consequently, this document recognizes that during a conference, the most common conference control operations involve conference state rather than policy and lays out an architectural approach in which the control interface directly operates on the conference state providing the required expedient effect.

The XCAP Usage for Conference Policy Manipulation [8] approach has no application semantics and requires a document locking property.
Consequently, only one user can edit the policy document (or any
other shared XML document) at a time.

In comparison, the CCCP approach defines a set of semantics (add, get, set, delete, remove) that operate directly on the conference state elements (as provided to the subscribed users in the SIPPING Conference Package [2] notifications). CCCP requests are submitted to the focus and can be prioritized and queued, or even interleaved based on requester’s role and the affected XML element(s) correspondingly. No single lock per document is required, and the CCCP server implemented in the focus, can locally decide on its optimization strategy without relying on any special CCCP clients behavior.

4. Data Model

EDITOR’s NOTE: It is suggested that this section will be incorporated into the XCON Framework document.

The Figure 1 below depicts a Conference Server logical decomposition with an example of SIP mechanisms (SIP Call Control "1st-party" signaling and SUBSCRIBE/NOTIFY as a notification means) in conjunction with the CCCP (as a "3rd-party" control protocol for manipulation of conferencing policy and state) being used to communicate with external entities.
4.1 Relationships between Membership and Media Information

All the information about a particular conference can be modeled as two main pieces of data: the conference policy and the conference state.

In the data model defined in this document there is no strict separation between "the conference (i.e. membership and signaling) data model" and "the media data model". In other words, policy related to media is a part of the overall conference policy and state information about media is a part of the overall conference state. This meets the requirement in many conference control operations to enable synchronized access to the conference policy as a whole, the conference state as a whole, and for getting notifications about changes in any of them as a whole.

The concept of a "template" is discussed in XCON Media Control [9] in strictly media terms. However, a conference template can be thought
of as "pre-state" belonging to a conference policy. It contains just enough information to define what the initial state of the conference will be when it is created. While the conference state will be highly dynamic, the conference template (as a part of the larger conference policy) is likely to be relatively static.

Today, the template contains basic information about the conference mixing capabilities, the conference media controls, sliders, radio boxes, etc. but also the participant’s roles, which is more general than just media-related.

On the other hand, the XCON Media Control [9] document uses "Media Policy" terminology when referring to the concept called in this document the "Conference State".

In conclusion, for advanced conference manipulations (e.g. media layouts) extensions to both the policy (e.g. to templates) and the state schemas will be needed. Additionally, CCCP may need to be extended for manipulating the policy schema with its templates.

4.2 Conference Policy

Conference policy is a set of parameters and rules (e.g. maximum number of participants, needs chair-person supervision or not, password protected or not, duration, a way of media mixing, etc.) that are defined at the onset of a conference and MAY be modified during the conference.

The Conference policy would typically be derived from the system’s default template or templates. On a particular conference onset, the default parameters and rules can be overridden and/or expanded. The XCON CPCP [7] contains an example of a conference policy schema.

4.3 Conference State

Conference state is the set of information describing the conference in progress. This includes participants’ information (such as dialog identifiers), media sessions in progress, the current loudest speaker, the current chair, etc. The basic XML schema is defined the SIPPING Conference Package [2].

CPCP is used to directly affect the conference state and expands it for this purpose. Changes in the conference state also occur as a result of participants’ state changes and learned by the focus through the call signaling channel with each of the participants.
Changes in the conference state are reported to conferencing participants and otherwise authorized party using well-defined event
mechanisms subject to each interested party privileges. For example, for SIP entities it is the Event Notification mechanism defined in RFC 3265 [1].

4.4 Policy and State Dependencies

Changes in the Conference Policy can automatically cause changes in the Conference State, while changes in the conference state never change conference policy.

There are many examples of how a change in conference policy can change the state - changing the end time of a conference causes the conference to end early, reducing the maximum number of participants could cause some to be booted.

A change in conference state never changes the conference policy because by definition the conference policy must authorize any change in state. If a request fails due to a policy, this will be indicated in the response reason. The user may then attempt to change the policy, and then retry the state change request.

For example, a user may request a participant to be added to a conference. The focus must check the policy to see if the user’s role and/or URI allow the user to participate in the conference. For example, the policy might say that only members of example.com may join the conference.

5. The Protocol

5.1 The Transaction Model

CCCP is defined as a set of transactions issued over a reliable transport protocol. A transaction consists of a Request carrying the required information in an XML body and a corresponding Response carrying an appropriate XML body.

EDITOR’s NOTE: TBD the requirements from the transport protocols fitting CCCP.

5.2 XML

This document expands the XML schema defined in SIPPING Conference Package [2] as defined in this section below.
<xs:element name="conference-request" type="conference-request-type"/>

<!-- CONFERENCE RESPONSE ELEMENT -->
<xs:element name="conference-response" type="conference-response-type"/>

<!-- CONFERENCE REQUEST TYPE -->
<xs:complexType name="conference-request-type">
  <xs:sequence>
    <xs:element name="content" type="conference-type"/>
    <xs:element name="extended" type="extended-type" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="request-id" type="xs:string"/>
  <xs:anyAttribute/>
</xs:complexType>

<!-- CONFERENCE RESPONSE TYPE -->
<xs:complexType name="conference-response-type">
  <xs:sequence>
    <xs:element name="content" type="conference-type" minOccurs="0"/>
    <xs:element name="status" type="response-status-type"/>
    <xs:element name="extended" type="extended-type" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="request-id" type="xs:string"/>
  <xs:anyAttribute/>
</xs:complexType>

<!-- RESPONSE STATUS TYPE -->
<xs:complexType name="response-status-type">
  <xs:sequence>
    <xs:element name="code" type="response-code-type"/>  
    <xs:element name="reason" type="xs:string" minOccurs="0"/>  
    <xs:element name="extended" type="extended-type" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute/>
</xs:complexType>

<!-- RESPONSE CODE TYPE -->
<xs:simpleType name="response-code-type">
</xs:simpleType>
<xs:restriction base="xs:string">
  <xs:enumeration value="success"/>
  <xs:enumeration value="pending"/>
  <xs:enumeration value="failure"/>
</xs:restriction>
</xs:simpleType>

<xs:complexType name="operator-type">
  <xs:sequence>
    <xs:element name="code" type="operator-code-type"/>
    <xs:element name="to-entity" type="xs:string" minOccurs="0"/>
    <xs:element name="from-entity" type="xs:string" minOccurs="0"/>
    <xs:element name="extended" type="extended-type" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute/>
</xs:complexType>

<xs:complexType name="operator-code-type">
  <xs:restriction base="xs:string">
    <xs:enumeration value="remove"/>
    <xs:enumeration value="add"/>
    <xs:enumeration value="get"/>
    <xs:enumeration value="set"/>
    <xs:enumeration value="move"/>
  </xs:restriction>
</xs:simpleType>

5.3 Example

<conference-request request-id="8797">
  <content entity="sips:conf233@example.com">
    <user entity="sip:bob@example.com">
      <operator><code>add</code></operator>
      <display-text>Bob Hoskins</display-text>
      <endpoint entity="sip:bob@pc4.example.com">
        <display-text>Bob’s Laptop</display-text>
        <joining-method>dialed-out</joining-method>
        <media entity="1">Levin & Kimchi
        <![CDATA[Expires April 18, 2005]]>
      </media>
    </endpoint>
  </user>
</content>
</conference-request>

Example

<conference-request request-id="9119">
  <content entity="sip:007@example.com">
    <user entity="sip:007@example.com">
      <operator><code>add</code></operator>
      <display-text>Bob Hoskins</display-text>
      <endpoint entity="sip:007@pc4.example.com">
        <display-text>Bob’s Laptop</display-text>
        <joining-method>dialed-out</joining-method>
        <media entity="1">Levin & Kimchi
        <![CDATA[Expires April 18, 2005]]>
      </media>
    </endpoint>
  </user>
</content>
</conference-request>
<conference-response request-id="8797">
  <status><code>pending</code></status>
</conference-response>

<!-- REQUEST IS PENDING -->

<conference-response request-id="8797">
  <content entity="sips:conf233@example.com">
    <user entity="sip:bob@example.com">
      <display-text>Bob Hoskins</display-text>
      <endpoint entity="sip:bob@pc4.example.com">
        <display-text>Bob’s Laptop</display-text>
        <state>connected</state>
        <joining-method>dialed-out</joining-method>
        <joining-info>
          <when>2005-03-04T20:00:00Z</when>
          <by>sip:mike@example.com</by>
        </joining-info>
      </endpoint>
      <media entity="1">
        <display-text>main audio</display-text>
        <proto>audio</proto>
        <ssrc>432424</ssrc>
        <label>34567</label>
        <state>active</state>
        <call>
          <sip>
            <display-text>full info</display-text>
            <dialog-id>hajh8980vhsb78</dialog-id>
            <from-tag>vav738dvbs</from-tag>
            <to-tag>8954jgjj8432</to-tag>
          </sip>
        </call>
      </media>
    </user>
  </content>
</conference-response>

<!-- SUCCESFUL RESPONSE -->
6. IANA Considerations

None.

7. Security Considerations

Will be completed in the next version.

8. Acknowledgements

Authors would like to thank Mary Barnes and Alan Johnston for providing helpful inputs.

9. References

9.1 Normative References


9.2 Informative References


Initiation Protocol",
draft-ietf-sipping-conferencing-framework-02 (work in progress),

draft-ietf-sipping-conferencing-requirements-01 (work in progress), October 2004.


Authors’ Addresses

Orit Levin
Microsoft Corporation
One Microsoft Way
Redmond, WA  98052
USA
EMail: oritl@microsoft.com

Gur Kimchi
Microsoft Corporation
One Microsoft Way
Redmond, WA  98052
USA
EMail: gkimchi@microsoft.com
Intellectual Property Statement

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

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

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

Disclaimer of Validity

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

Copyright Statement

Copyright (C) The Internet Society (2004). This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

Acknowledgment

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