

Internet Engineering Task Force
Internet Draft
[draft-koskelainen-sipping-conf-requirements-00.txt](#)
April 29, 2002
Expires: October 2002

Koskelainen, Schulzrinne, Wu
Nokia, Columbia U.

Additional Requirements to Conferencing

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>

To view the list Internet-Draft Shadow Directories, see <http://www.ietf.org/shadow.html>.

Abstract

This document focuses on advanced media-independent conferencing and conference control requirements, especially on user management related requirements. Media control and floor control are out of scope.

1 Introduction

This document is an input document to the IETF SIPPING WG. This document complements other conferencing requirements such as [[1](#)] and focuses more on advanced user management issues. Floor control and media control are totally out of scope.

This document focuses only on centralized (star-based) conferences in

which the server hosts the conference. Scalable solutions to support large conferences are out of scope. Later standardization may define how star-based conference topology can be extended to support really large conferences. Work on scalable solutions can be found e.g. in [2], which defines a mechanism that can be used to distribute the conference across several conference servers using hierarchical, distributed server tree (DST) or single source multicast.

Management functions identified in this document may be utilized for other services as well. For example, creating a group list at a server (without intension to do conferencing) may utilize same operations, such as creating a group list and adding new users into it.

1.1 Conventions of This Document

In this document, the key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALLNOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" are to be interpreted as described in [RFC 2119](#) [3].

2 Overview

We define conference as a set of members that exchange one or more media streams or share some other state, such as an application. Conferencing includes two main parts. First part enables participants to join and leave a conference. Signaling protocols, such as SIP, can generally support these operations if conferences are addressed like regular endpoints. The second conferencing part, conference control, can be considered as consisting three components:

- conference management (creating, modifying and deleting conferences as a whole),
- user management (adding and deleting conference participants and modifying their properties), and
- resource contention management, also known as floor control.

The main function of floor control is to provide safe access to shared resources. Controlling permissions to talk is the most typical example of floor control. However, as said earlier, floor control is out of scope.

Conferences are addressed like regular endpoints so conference has an identifier such as sip:conf34@server.com. The conference server has a point-to-point relationship with each conference member.

SIP can control application sessions only between one user and the conference server. It cannot change the conference state or the

conference-wide media settings at the server so additional management operations are needed [2].

3 Example Use Cases

Use Case 1: (Conference creation and conference control)

A conference is created at a server, and it is managed during the conference lifetime. Conference may have a lot of properties such as member list, access control list (ACL), user privileges, and media channels.

Use Case 2: (Creating and managing user list without actual conference)

A group name such as sip:myfriends@server.com is created. Creator can add and remove users. This group name can be used e.g. as an alias for multiparty one-shot messaging, or it can be used later for conferencing (no need to create new user list for each conference, instead, user can utilize existing user lists).

4 General Requirements

- * Tight conference control (in contrast to loose conference control)
- * Centralized architecture (in contrast to fully meshed or multicast)
- * Simplicity:
Conferencing framework must balance being powerful enough for most practical scenarios, yet avoid being too complex to be widely accepted and implemented on devices with limited computational resources and user interfaces.
- * Mobile-friendliness
Solution MUST be mobile-friendly, since mobile low-bandwidth devices represent large portion of user base. Bandwidth consumption on the access link is the most important mobile requirement. It means that unnecessary (frequent) communication SHOULD be minimized. For example, pre-defined definitions (access lists) eliminate the need to consult human moderator for every join attempt. Moreover, the notification frequency SHOULD be configurable by the user. Similarly, partial modifications MUST be supported. This means that there is no need to send all parameters if you change only one. In addition, changes in the member lists should be distributed as additions and removals (instead of distributing whole list). Moreover, the solution MUST NOT assume IP multicast since it is not widely available in mobile networks or in residential environments (such as PPPoE).
- * Component based

The solutions MUST be component-based (in contrast to monolithic). This allows clients to implement only relevant components. It also allows more advanced components to be added later (such as floor control), or existing components may be changed.

* Asynchronous commands

The execution of commands may take a long time, especially if human intervention is required so asynchronous commands MUST be supported.

* Asynchronous notifications

There MUST be a support for asynchronous notifications (e.g. network-initiated notifications without client pull).

* Media-independency:

The solution MUST be media-independent since in many cases there is no (RTP) media at all.

5 Component Specific Requirements

5.1 Conference Management requirements

Conference management may include several optional sub-components, but it has also a few mandatory top-level requirements:

* Conference creation: This creates a conference at a server. It may have parameters which describe basic conference-wide properties such as conference name and subject. Conference may be created for one-time use only, or as a permanent conference or group (conference name remains reserved).

* Conference termination: This terminates the conference. The conference name may still be left to the server so same conference name and parameters may be utilized for later conferences.

* Modification of conference parameters: This modifies conference parameters (e.g. adds video channel).

* Fetching conference description from the server: User may fetch the conference description even before joining the conference.

* Finding out the status of the conference: Solution MUST NOT be based on RTCP.

* Search conferences at the server (e.g. search all conferences which are public and subject line or keywords includes word "soccer")

* User SHOULD be able to define the level of details it want to receive as conference notifications or announcements (e.g. user may

not want to receive informational notifications at all)

It must be possible to create a simple conference using these top-level functions. More advanced features can be provided by optional sub-components.

5.2 User Management component

This is an optional component. It must enable at least the following operations or features:

*** Managing user right levels:**

Manage user right levels, such as audience, guru, moderator. It must be possible to create and delete a user level, and add (remove) a user to (from) a user level. Default user levels may be defined (e.g. all new users are audience).

*** Managing privileges for user levels:**

This sets the user level privileges in a conference and defines what is allowed for each user level. Managing privileges can be an arbitrarily complex, expressible only in a full programming language. However, the majority of cases can be handled by defining a few simple privileges that are directly tied to conference management and giving users rights to those privileges. The privileges are at least invite users, expel members, make real-time authorizations (such as accept or deny a join attempt), manage access control list (ACL), manage media sessions (e.g. change a user to receive only mode, or add video channel into a conference), and manage general parameters such as subject, authentication, visibility or start-time. Typically, moderator user level has all privileges.

*** Managing ACL:**

Manage ACL and sets allow/block definitions for a list of users. Wildcards must be supported (e.g. block sip:*@spam.com). Add/remove operations must be supported.

*** Expelling users:**

This expels list of users from the conference.

*** Mass-invitation:**

Invites a list of users into the conference

*** Real-time authorization operations:**

For unexpected decisions during the conference. For example, "allow user X to join the conference".

*** Finding out conference members:** It SHOULD be possible to find out the identities of current conference members at any given time. Must

not assume RTCP.

* Anonymous participation: It must be possible to participate anonymously to the conference (e.g. by creating an anonymous user name into the server)

5.3 Other components

The solution must allow other sub-components to be introduced later into the framework (such as SIP/SOAP floor control [[4](#)]).

5.4 Group Operations

* Group name creation, modifications, termination:
Create/Modify/Terminate a group name which can be later used e.g. for conferencing or one-shots. In many cases, this group name includes just a list of users and is used only for one-shot messaging. (this overlaps partly with conference creation..)

6 Notifications

Following notifications or announcements are required:

- Occasional user related notifications, such as "Bob and Tom have joined", or "Dear moderator, user X wants to join the conference. Accept or Deny?", or other general conference notifications.
- result for the asynchronous command (e.g. moderator decided - after thinking for a while - not to allow Bob to join).

7 Denial-of-Service attacks

DoS attacks are possible. For example, a large number of join attempts may be created. Human moderator should not be automatically consulted about each join attempt. Instead, access lists are checked first. Rate-control must be applied in the server so notifications are not sent too often. Moreover, this often allows the server to include several notification into one message (e.g. 3 join attempts are reported in one notification).

8 Examples

At least the following parameters should be supported:

- conference name (such as sip:conf34@server.com)
- conference subject (such as "IETF interim SIMPLE WG meeting")
- conference time (e.g. it starts tomorrow 12:00 EST)
- store conference parameters or not for later use
- whether conference is public or not (can it be seen/searched)

- authentication information
- application information (such as conference-wide media codecs)
- list of users to be invited by the server (for mass-invitation)
- user right levels (such as guru, moderator, audience)
- user level access rights (such as guru-members are allowed to expel members)
- ACL

Following is a concrete example of the conference creation, and conference control during the conference.

User A creates a conference at a server with following parameters:

- conference name: sip:conf34@server.com
- subject: Interim Sipping meeting
- authentication: digest, passwd='x243DHEDw'
- media: video channel:H.261., audio channel:PCM..
- user levels: default groups (moderator, audience)
 - moderator list: user A
 - audience list: (default for all new members)
- user level rights:
 - audience: none
 - moderator:expel members, authorize joins, manage ACL, terminate conference)
- invite users: Tom, Alice, John
- ACL: block list=Bob, *.spam.com.

The conference server sends an acknowledgement. It means that the command was accepted without an immediate failure.

The server process the command and makes invitations.

The server sends occasional notifications to the creator, such as "Conference created successfully. Members in a conference: Tom, Alice".

New members try to join the conference. ACL does not include any hint how to act so moderator action is required.

The server waits for a while, and sends a notification to the moderator. The notification includes information about 3 join attempts. The moderator accepts one of them, and denies two of them. The decision is sent to the server in a one message.

9 Conclusions

There are two kinds of operations needed for conference control and

management:

- commands (sent to the server)
- informative notifications aka announcements

10 Authors' Addresses

Petri Koskelainen
Dept. of Computer Science
Columbia University
1214 Amsterdam Avenue, MC 0401
New York, NY 10027
USA
electronic mail: petkos@cs.columbia.edu
electronic mail: petri.koskelainen@nokia.com

Xiaotao Wu
Dept. of Computer Science
Columbia University
1214 Amsterdam Avenue, MC 0401
New York, NY 10027
USA
electronic mail: xiaotaow@cs.columbia.edu

Henning Schulzrinne
Dept. of Computer Science
Columbia University
1214 Amsterdam Avenue, MC 0401
New York, NY 10027
USA
electronic mail: schulzrinne@cs.columbia.edu

11 Bibliography

- [1] O. Levin, A. Zmolek, R. Even, D. Petrie, and P. Koskelainen, "Conferencing requirements for sip based systems," Internet Draft, Internet Engineering Task Force, April 2002. Work in progress.
- [2] P. Koskelainen, H. Schulzrinne, and X. Wu, "A sip-based conference control framework," in The 12nd International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV) , (Miami Beach, Florida), May 2002.
- [3] S. Bradner, "Key words for use in RFCs to indicate requirement levels," [RFC 2119](#), Internet Engineering Task Force, Mar. 1997.
- [4] X. Wu, P. Koskelainen, H. Schulzrinne, and C. Chen, "Use SIP and SOAP for conference floor control," Internet Draft, Internet Engineering Task Force, April 2002. Work in progress.

Full Copyright Statement

Copyright (c) The Internet Society (2002). 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 assigns.

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.

Table of Contents

1	Introduction	1
1.1	Conventions of This Document	2
2	Overview	2
3	Example Use Cases	3
4	General Requirements	3
5	Component Specific Requirements	4
5.1	Conference Management requirements	4
5.2	User Management component	5
5.3	Other components	6
5.4	Group Operations	6
6	Notifications	6
7	Denial-of-Service attacks	6
8	Examples	6

<u>9</u>	Conclusions	<u>7</u>
<u>10</u>	Authors' Addresses	<u>8</u>
<u>11</u>	Bibliography	<u>8</u>