                  Requirements for Floor Control Protocols

Status of This Memo

   This memo provides information for the Internet community. It does
   not specify an Internet standard of any kind. Distribution of this
   memo is unlimited.

Copyright Notice

Abstract

   Floor control is a means to manage joint or exclusive access to
   shared resources in a (multiparty) conferencing environment.
   Thereby, floor control complements other functions -- such as
   conference and media session setup, conference policy manipulation,
   and media control -- that are realized by other protocols. This
   document defines the requirements for a floor control protocol for
   multiparty conferences in the context of an existing framework.

Table of Contents

1. Introduction

Conference applications often have shared resources such as the right
to talk, input access to a limited-bandwidth video channel, or a
pointer or input focus in a shared application.

In many cases, it is desirable to be able to control who can provide
input (send/write/control, depending on the application) to the
shared resource.

Floor control enables applications or users to gain safe and mutually
exclusive or non-exclusive input access to the shared object or
resource. The floor is an individual temporary access or
manipulation permission for a specific shared resource (or group of
resources) [6].

Floor control is an optional feature for conferencing applications.
SIP [2] conferencing applications may also decide not to support this
feature at all. Two-party applications may use floor control outside
conferencing, although the usefulness of this kind of scenario is
limited. Floor control may be used together with the conference
policy control protocol (CPCP) [7], or it may be used as an
independent stand-alone protocol, e.g., with SIP but without CPCP.

Floor control has been studied extensively over the years (e.g., [8],
[6], and [5]); therefore, earlier work can be leveraged here.

The present document describes the requirements for a floor control
protocol. As a requirements specification, the document makes no
assumptions about the later implementation of the respective

requirements as parts of one or more protocols or about the entities
implementing them and their roles.

This document may be used in conjunction with other documents, such
as the conferencing framework document [3]. In particular, when
speaking about a floor control server, this entity may be identical
to or co-located with the focus or a conference policy server defined
in the framework document, while participants and floor chairs
referred to in this specification may be regular participants as
introduced in the conferencing framework document. In this
specification, the term "floor control protocol" is used in an
abstract sense and may ultimately be mapped to any of the existing
conference control or other signaling protocols (including CPCP and
SIP). However, defining those relationships is left to a concrete
floor control protocol specification.

2. Conventions Used in This Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
"SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this
document are to be interpreted as described in RFC 2119 [1].

3. Terminology

This document uses the definitions from [3].

The following additional definitions apply:

Floor: A permission to access or manipulate a specific shared
resource or set of resources temporarily.

Conference owner: A privileged user who controls the conference,
creates floors, and assigns and deassigns floor chairs. The
conference owner does not have to be a member in a conference.

Floor chair: A user (or an entity) who manages one floor (grants,
denies, or revokes a floor). The floor chair does not have to be a
member in a conference.

Floor control: A mechanism that enables applications or users to gain
safe and mutually exclusive or non-exclusive input access to the
shared object or resource.

Floor control server: A logical entity that maintains the state of
the floor(s) including which floors exists, who the floor chairs are,
who holds a floor, etc. Requests to manipulate a floor are directed
at the floor control server.

   Floor request set: A logical data structure holding all requests for
   a particular floor at a given point in time.

   Floor holder set: A logical data structure identifying all
   participants who currently hold the floor.

4. Model

   The model for floor control is composed of three logical entities: a
   single floor control server, one or more floor chairs (moderators),
   and any number of regular conference participants.

   A floor control protocol is used to convey the floor control messages
   among the floor chairs (moderators) of the conference, the floor
   control server, and the participants of the conference. A
   centralized architecture is assumed in which all messages go via one
   point, the floor control server. Processing (granting or rejecting)
   floor control requests is done by the one or more floor chairs or by
   the server itself, depending on the policy.

   Floor requests from the participants are received by the floor
   control server and kept (at the level of the floor control protocol)
   in a floor request set (i.e., are not ordered in any particular
   fashion). The current floor holders are reflected in a current floor
   holder set. Floor chairs are capable of manipulating both sets to
   grant, revoke, reject, and pass the floor (for example).

   The order in which requests are processed, whether they are granted
   or rejected, and how many participants obtain a floor simultaneously
   are determined by a higher-layer application operating on these sets
   and are not confined by the floor control protocol.

   A floor is associated with one or more media sessions. The
   centralized conference server manages the floors and thus controls
   access to the media sessions. There are two aspects to this: 1) The
   server maintains and distributes consistent state information about
   who has a certain floor at a certain point in time and does so
   following some rule set. This provides all participants with the
   necessary information about who is allowed to speak (for example),
   but relies on a cooperative behavior among all participants. 2) In
   addition, to prevent individuals from ignoring the "hints" given by
   the floor control server, the latter may (e.g., in cooperation with
   other functional entities) enforce compliance with floor status,
   e.g., by blocking media streams from participants not entitled to
   speak. The floor control server controls the floors at least at the
   signaling level. In addition, actively controlling the actual
   (physical) media resources is highly recommended, but beyond the
   scope of this document.

As noted in the introduction, an actual protocol specification
fulfilling the requirements defined in this memo may map the
components of the above model onto the conferencing components
defined in the conferencing framework document. Some of these
aspects are discussed briefly in the next section.

5. Integration with Conferencing

Floor control itself does not support privileges such as creating
floors and appointing floor chairs and handing over chair privileges
to other users (or taking them away). Instead, some external
mechanism, such as conference management (e.g., CPCP or web interface
for policy manipulation) is used for that.

The conference policy (and thus the conference owner or creator)
defines whether floor control is in use or not. Actually enforcing
conference media distribution in line with the respective media's
floor status (e.g., controlling an audio bridge) is beyond the scope
of this document. Floor control itself does not define media
enforcement. It is up to the conference and media policies to define
which media streams may be used in a conference and which ones are
floor controlled.

Typically, the conference owner creates the floor(s) using the
conference policy control protocol (or some other mechanism) and
appoints the floor chair. The conference owner can remove the floor
anytime (so that a media session is not floor-controlled anymore) or
change the floor chair or floor parameters.

The floor chair just controls the access to the floor(s), according
to the conference policy.

A floor control server is a separate logical entity, typically
co-located with focus and/or conference policy server. Therefore,
the floor control server can interact with the focus and conference
policy server and media servers as needed. Communication mechanisms
between the floor control server and other central conferencing
entities are not within the scope of the floor control protocol
requirements described in this document.

Conferences may be cascaded, and hence a single participant in one
conference may represent a second conference (called subconference).
From a floor control perspective, there is no difference between a
participant (identified by its URI) representing a single person or
another (set of) subconference(s).

Note: In the latter case, it is the responsibility of the
subconference to negotiate floor requests internally before passing
on a request to the conference and to assign a floor internally upon
receiving a floor grant. This may be done recursively by employing
the floor control protocol with a different floor control server in
the subconference.

6. Assumptions about a Conference Policy

The floor control protocol is supposed to be used to manage access to
shared resources in the context of a conference. It is up to this
conference -- more precisely, its conference policy [4] -- to define
the rules for the operation of the floor control protocol.
Furthermore, a conference policy control protocol [4] may define
mechanisms that alter those rules during the course of a conference.
This section briefly outlines the assumptions made by a floor control
protocol about the conference policy and means for its modification.

The conference policy is expected to define the rules for floor
control, which implies in particular that it is not the
responsibility of the floor control protocol to establish or
communicate those rules.

In general, it is assumed that the conference policy also defines who
is allowed to create, change, and remove a floor in a conference.

Conference participants and floor chairs should be able to get and
set floor-related parameters. The conference policy may restrict who
may access or alter which parameters. Note that not all parameters
maintained for a floor are also interpreted by the floor control
protocol (e.g., floor policy descriptions may be stored associated
with a floor but may be interpreted by a higher-layer application).
Note also that changes to the floor control policy are outside the
scope of the floor control protocol and are (for example) to be
carried out by a conference policy control protocol.

(For example, it may be useful to see who the floor chair is, what
kind of policy is in use, time limits, number of simultaneous floor
holders, and current floor holder.)

The following requirements on a conference policy related to floor
control are identified in [4]:

REQ-F1: It MUST be possible to define whether floor control is in use
or not.

REQ-F2: It MUST be possible to define the algorithm to be used in
granting the floor. (Note: Examples of algorithms are moderator-
controlled, FCFS, or random.)

Note: It must be possible to use an automated floor policy where the
floor control server decides autonomously about granting and
rejecting floor requests as well as revoking the floor. It must also
be possible to use a chair-controlled floor policy in which the floor
control server notifies the floor chair and waits for the chair to
make a decision. This enables the chair to fully control who has the
floor. The server MAY forward all requests immediately to the floor
chair, or it may do filtering and send only occasional notifications
to the chair.

REQ-F3: It MUST be possible to define how many users can have the
floor at the same time.

REQ-F4: It MUST be possible to have one floor for one or more media
types.

REQ-F5: It MUST be possible to have multiple floors in a conference.

REQ-F6: It MUST be possible to define whether a floor is moderator-
controlled or not.

REQ-F7: If the floor is moderator-controlled, it MUST be possible to
assign and replace the floor moderator.

7. Floor Control Protocol Requirements

This section covers the requirements on a floor control protocol.
The requirements are grouped as follows: 1) floor control protocol
between participant and server; 2) floor control protocol between
floor chairs and server; 3) floor control server management; and 4)
general protocol requirements.

7.1. Communication between Participant and Server

REQ-PS-1: Participants MUST be able to request (claim) a floor.

REQ-PS-2: It SHOULD be possible for a participant requesting a floor
to give additional information about the request, such as the topic
of the question for an audio floor. Note: In some scenarios, the
floor control server or the floor chair may use this information when
granting the floor to the user, or when manipulating the floor sets
at the server.

REQ-PS-3: It MUST be possible for a participant to modify (e.g., cancel) a previously placed floor request.

REQ-PS-4: It SHOULD be possible for a participant to initiate a floor control operation (e.g., floor request, release) on behalf of another participant (third-party floor control) provided that he is authorized to do so.

REQ-PS-5: A participant MUST be informed that she has been granted the floor.

REQ-PS-6: A participant MUST be informed that his floor request has been rejected.

REQ-PS-7: A participant MUST be informed that the floor was revoked from her.

REQ-PS-8: A participant SHOULD be informed that her floor request is pending and will be processed later.

REQ-PS-9: A floor holder MUST be able to release a floor.

REQ-PS-10: It MUST be possible to notify conference participants of (changes to) the floor holder(s).

REQ-PS-11: It MUST be possible to notify conference participants when a new floor request is being made.

REQ-PS-12: It MUST be possible for a floor requester to request privacy for claiming the floor.

> anonymous: The participants (including the floor chair) cannot see the floor requester's identity. The floor chairs grant the floor based on the claim id and the topic of the claim.
>
> known to the floor chair: Only the floor chair is able to see the floor requester's identity; all other participants do not obtain this information.
>
> public: All the participants can see the floor requester's identity.

REQ-PS-13: It MUST be possible for a participant to request privacy for holding the floor along with a floor request. Note that identity information about the participant may become available to others through different means (e.g., application/media protocols or the media itself such as the voice).

7.2. Communication between Chair and Server

   REQ-CS-1: It MUST be possible to inform the floor chairs, if present,
   about a participant's floor request.

   It SHOULD be possible to convey additional information the
   participant may have provided along with her request.

   It MUST be possible to hide the requesting participant's identity
   from the chair, i.e., not to include this identity information in the
   floor request.

   REQ-CS-2: It MUST be possible to grant a floor to a participant.

   REQ-CS-3: It MUST be possible to reject a participant's floor
   request.

   REQ-CS-4: The floor chair MUST be able to revoke a floor from (one
   of) its current holder(s). Note that the floor chair may also remove
   pending floor requests from the request set (by rejecting them).

   REQ-CS-5: It MUST be possible to notify floor chairs about changes to
   the floor holder(s).

   REQ-CS-6: There SHOULD be operations to manipulate the request set
   available for floor chair(s). Such a request set SHOULD at least
   include creating, maintaining, and re-ordering floor requests in a
   queue and clearing the floor control queue.

   REQ-CS-7: It MUST be possible to hide the identity of a floor chair
   from a subset or all participants of a conference.

   REQ-CS-8: It MUST be possible for a newly assigned floor chair to
   learn (e.g., inquire) about the existing floor request set.

7.3. General Protocol Requirements

   REQ-GEN-1: Bandwidth and terminal limitations SHOULD be taken into
   account in order to ensure that floor control can be efficiently used
   in mobile environments.

   Note that efficient communication by means of minimal-sized messages
   may contradict the desire to express reasons for requesting a floor
   along with other information. Therefore, a floor control protocol
   SHOULD be designed in a way that it allows for expressive as well as
   minimal messaging, as a (negotiable) configuration option and/or
   selectable on a per-message basis.

REQ-GEN-2: The floor control MUST be a reliable client-server
protocol. Hence, it MUST provide a positive response indicating that
a request has been received or an error response if an error has
occurred.

REQ-GEN-3: It MUST be possible for the floor control server to
authenticate participants and chairs.

REQ-GEN-4: It MUST be possible for the participants and chairs to
authenticate the server.

REQ-GEN-5: It MUST be possible to ensure message integrity between
participants and chairs and the floor control server.

REQ-GEN-6: It MUST be possible to ensure the privacy of messages
exchanged between participants and chairs and the floor control
server.

[8](#). Security Considerations

Floor control messages are exchanged on one hand between regular
participants and the floor control server and on the other hand
between the floor control server and the floor chair(s).

If enabled, floor control mechanisms are used to control who may
contribute to a conference in arbitrary ways (speak, be seen, write,
etc., as supported by the conferencing applications). It is
important that floor control messages be protected because otherwise
an attacker could prevent participants from being "heard" in the
conference (e.g., in scenarios where silence is considered consent)
or make participants be heard in a conference without their knowledge
(e.g., eavesdropping on the participant's microphone). Such
considerations are particularly relevant when floor control is used
in conjunction with one or more (central) entities (e.g., a media
mixer) controlled by the floor control server to enforce floor
control decisions that may allow an attacker to "mute" a participant
completely.

Communications between a conference participant and the floor control
server are vulnerable to all kinds of masquerading attacks. If an
attacker can spoof the identity of the participant or inject messages
on his behalf, it may generate floor requests (e.g., floor release)
and prevent proper participation of the participant. If an attacker
can inject messages to the participant, it may generate arbitrary
responses and false status information. If an attacker can
impersonate the floor control server, a participant's requests may
never reach the actual floor control server. If an attacker can
intercept either side's messages (and hence become a man in the

middle (MITM)), it may suppress, alter, or inject messages and thus
manipulate a participant's view of the conference floor status as
well as the floor control server's view of a participant.

Similar considerations apply to the communications between the floor
control server and the floor chair(s). If an attacker can intercept
messages from either side, it may defer or prevent responses to floor
control requests (from a particular floor chair). If it can inject
messages (particularly in the direction from the floor chair to the
floor control server), it may steer the assignment of conference
floors. If interception and injection is possible (man-in-the-middle
scenario), an attacker can create an arbitrary image of the
conference for the floor chair. If an attacker can impersonate a
floor chair, it may rule the conference floor assignment (if there is
only a single chair) or disrupt the conference course by means of
arbitrary and potentially conflicting requests/responses/assignments
(if there are multiple floor chairs). In the latter case, the amount
of damage a single attacker can do depends on the floor control
policy.

Finally, attackers may eavesdrop on the floor control communications
and learn which participants are present, how active they are, who
are the floor chairs, etc.

To mitigate the above threats, conference participants, floor control
servers, and floor chairs SHOULD be authenticated upon initial
contact. All floor control messages SHOULD be authenticated and
integrity-protected to prevent third-party intervention and MITM
attacks. Floor control messages SHOULD be encrypted to prevent
eavesdropping.

9. Acknowledgements

The authors would like to thank IETF conferencing design team and
Keith Drage, Marcus Brunner, Sanjoy Sen, Eric Burger, Brian Rosen,
and Nermeen Ismail for their feedback.

10. References

10.1. Normative References

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

[2] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A.,
     Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP:
     Session Initiation Protocol", RFC 3261, June 2002.

10.2. Informative References

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

[4] Koskelainen, P. and H. Khartabil, "Additional Requirements to
     Conferencing", Work in Progress, August 2004.

[5] Koskelainen, P., Schulzrinne, H., and X. Wu, "A SIP-based
     conference control framework", NOSSDAV 2002, Miami Beach,
     May 2002.

[6] Dommel, H. and J. Garcia-Luna-Aceves, "Floor control for
     activity coordination in networked multimedia applications",
     Proc. of 2nd Asian-pacific Conference on Communications APPC,
     Osaka Japan, June 1995.

[7] Koskelainen, P., Khartabil, H., and A. Niemi, "The Conference
     Policy Control Protocol (CPCP)", Work in Progress, October 2004.

[8] Borman, C., Kutscher, D., Ott, J., and D. Trossen, "Simple
     conference control protocol service specification", Work in
     Progress, March 2001.

Authors' Addresses

    Petri Koskelainen
    Nokia
    102 Corporate Park Drive
    White Plains, NY 10604
    USA

    EMail: petri.koskelainen@nokia.com

    Joerg Ott
    Helsinki University of Technology
    Networking Laboratory
    Otakaari 5A
    02150 Espoo
    Finland

    EMail: jo@netlab.hut.fi

    Henning Schulzrinne
    Columbia University
    1214 Amsterdam Avenue
    New York 10027
    USA

    EMail: hgs@cs.columbia.edu

    Xiaotao Wu
    Columbia University
    1214 Amsterdam Avenue
    New York 10027
    USA

    EMail: xiaotaow@cs.columbia.edu