SIP Conference Policy Formalism

Ivaylo I. Atanasov, Pavlina H. Koleva and Evelina N. Pencheva

Abstract - **The Session Initiation Protocol (SIP) supports the initiation, modification, and termination of multimedia conferences with multiple participants. The paper presents a formal description of a SIP conference policy based on the framework for conferencing with SIP. The requirements for tightly coupled conferences and requirements for floor control are considered. The formal description includes conference information about settings, conference time, participants, and rules for floor control. The formal description may be used in implementation of conference policy server.**

Keywords –**Session Initiation Protocol, multimedia conference, conference policy**

I. Introduction

The Session Initiation Protocol (SIP) supports establishment, maintenance and termination of multimedia sessions including multi-party conferences. SIP supports various multi-party conferencing models [1] including loosely coupled model and tightly coupled model. A loosely coupled conference is a conference without coordinated signaling relationships amongst participants. Loosely coupled conferences frequently use multicast for distribution of conference memberships. A tightly coupled conference is a conference in which a single user agent, referred to as focus, maintains the dialog with each participant. The focus plays the role of the centralized manager of the conference, and is addressed by a conference URI. Conference policy is a complete set of rules governing particular conference. The rules can be simple, such as an access list that defines the set of allowed participants in a conference. The rules can also be complex, specifying time-of-day-based rules on participation, conditional on the presence of other participants, etc. There is no restriction on the type of rules that can be encapsulated in a conference policy. The conference policy can be manipulated using web applications or voice applications. It can also be manipulated with non-SIP-specific standard or proprietary protocols. The conference policy server interfaces policy control protocol to the conference policy. Some of the aspects of Conference Policy Control Protocol are described in [2]. The logical function of the conference policy server is to store and manipulate the conference policy. The implementation of the conference policy server requires formal description of conference policy.

In this paper we present an approach to formal description of conference policy based on the framework for conferencing with SIP.

We consider the high-level requirements for tightly coupled SIP conferencing defined in [3] and SIP conference event package defined in [4]. The floor control policy is a part of conference policy and the requirements for floor control protocol are defined in [5]. We examine the specifications of Binary Floor Control Protocol (BFCP) [6] in floor control policy formalism. To formalize the media processing in tightly coupled conferences we use the media policy manipulation model defined in [7]. The formal description is in Augment Backus-Naur Form (ABNF) [8].

II. Formal Description of SIP Conferences

In SIP, a conference is an instance of a multi-party conversation. Within the context of the paper, a conference is always a tightly coupled conference with centralized control. The focus is a SIP user agent that maintains a SIP signaling relationship with each participant in the conference.

The SIP conference has conference owner who is a privileged user controlling the conference. The conference owner possesses special privileges to create floors, to assign and to deassign floor chairs and does not have to be a member in the conference. Typically, the conference owner specifies the conference policy and is identified by SIP Uniform Resource Identifier (URI).

Conference information includes informative parameters which may be helpful in describing the purpose of a conference, e.g. for search purposes or for providing host contact information. The conference information must specify the language used. The conference may have a subject which describes the current topic in a conference. The free text and the keywords provide additional textual information about the conference. This information can be made available to potential conference participants by non-SIP means. Examples of usage could be searching for a conference based on some keywords. The web page points to a URI where information about the conference can be found.

The conference identifier is a URI, usually a SIP URI that identifies the focus of a conference. The conference may occur for a limited period of time (i.e. bounded), or the conference may be unbounded (i.e. it does not have a specified end time). Bounded conferences may occur multiple times (e.g. on weekly basis). The information related to conference time and lifetime contains one or more conference occurrence elements, each defining the time information of a single conference occurrence. Multiple conference occurrence elements may be used if a conference is active at multiple irregularly spaced times. For each occurrence, the start-time specifies when a conference starts, and the stop-time specifies the time a conference stops. As to [2], if the start-time is not present, it indicates that the conference starts immediately. If the stop-time is set to zero, then conference occurrence is not

Ivaylo I. Atanasov, Pavlina H. Koleva and Evelina N. Pencheva are with the Faculty of Telecommunications, Technical University of Sofia, Kliment Ohridski 8, 1000 Sofia, Bulgaria, E-mail: iia@tu-sofia.bg, p_koleva@tu-sofia.bg, enp@tu-sofia.bg

bounded, i.e. permanent, though it will not become active until the start-time.

The conference policy contains a set of parameters and rules (e.g., maximum number of participants, needs chair-person supervision or not, password protected or not, duration or a way of media mixing) that are defined at the onset of a conference. The conference may have a default policy or a policy specified by the conference owner. The conference may be chaired or free. In a chaired conference the application or one of the participants acting as chair has special privileges, e.g. can control the video distribution. If the conference is chaired the conference policy must define the floor control policy and the media control policy.

The conference policy may define a list of users who may dial-in the conference and become participants. Following the conference policy the focus may invite to the conference other user agents defined in the dial-out list.

A participant in a conference is any SIP user agent that has a dialog with the focus. This SIP user agent can be a PC application, a SIP hardphone, or a PSTN gateway. A conference-unaware participant is a participant that is not aware that it is actually in a conference. Conference-unaware participants have access to a good deal of functionality. They can join and leave conferences using SIP, and obtain more advanced features through stimulus signaling, as discussed in [1]. However, if the participant wishes to explicitly control aspects of the conference using functional signaling protocols, the participant must be conference-aware.

A conference-aware participant is a participant in a conference that has learned, through automated means that it is in a conference. A conference-aware participant can use the conference notification service.

The media type specifies the media that are allowed to be used by the participants. E.g. this can be used to limit the conference to audio only, even when all participants support video. The conference participant may be allowed to use one or more media types. The user participation may be time of day based or conditional on presence of other participants.

Fig.1 shows the approach to formal description of SIP conference policy.

The privilege information is mandatory for a user and the next section considers possible privileges.

III. Privileges in a Conference

Advanced privilege models can be applied in conferencing context. SIP events framework defines general mechanisms for subscribing to, and receiving notifications of, events within SIP networks. It introduces the notion of a package, which is a specific "instantiation" of the events framework for a well-defined set of events. The conference event package [5] allows a user to subscribe to a conference. The focus has sufficient information about the state of the conference to inform subscribers about it.

The conference state is a virtual data base describing the conference in progress. This includes different conference aspects: participants' information, media sessions in progress, the current loudest speaker, the current chair, etc.

```
conference = "conference=(" conference-owner conference-policy
        conference-info conference-settings ")"
conference-owner = "conf-own=" SP URI CRLF
conference-info =language SP [subject] SP [free-text] SP
        [keywords] SP [web-page] CRLF
conference-settings =conference-ID SP conference-time CRLF
conference-ID = SIP-URI
conference-time =1*conference-occurrence
conference-occurrence =start-time SP stop-time SP
conference-policy = default-policy / specified-policy
specified-policy ="(" max-participants SP chaired SP password
        CRLF [dial-in-list CRLF]  [dial-out-list CRLF]  [floor-control-
        policy CRLF] [media-control-policy] ")" CRLF
chaired = "chaired=" right
right = "true" / "false"
password = "password=" string
dial-in-list = "dial-in=(" 1*(conference-aware SP) ")"
dial-out-list = "dial-out=(" 1*(conference-unaware SP / conference-
        aware SP) ")"
conference-unaware = "conf-unaware=(" SIP-URI  ")" SP media-
        allowed ")" SP participation SP privileges ")"
conference-aware = "conf-aware=(" SIP-URI SP media-allowed SP
        participation CRLF [isfocus] conference-ID SP password
        CRLF privileges CRLF ")"
isfocus = "isfocus "
media-allowed  = "media-allowed=(" 1*(media-type SP )")"
media-type = "audio" / "video" / "application" / "data" / "control" /
        "message" / "text"
participation = "participation=" "default" / ([time-based] SP
        [conditional] )
time-based = "time-based=(" start-time stop-time ")"
conditional = "condition=" 1*user-URI “)”
SP = " "
```

Fig.1 An approach to SIP conference policy formal description

Each user element has zero or one "status" elements, indicating their status in the conference, zero or one "dialog" elements, indicating their dialog information, and zero or one "media-streams" elements, indicating their media reception information. The status element contains the status of the user in the conference. The following statuses are defined: active - the user is in an active dialog with the focus; departed: the user sent a BYE, thus leaving the conference; booted: the user was sent a BYE by the conference host, booting them out of the conference; failed: the server tried to bring the user into the conference, but its attempt to contact the specific user resulted in a non-200 class final response.

The dialog element is presented from the viewpoint of the focus. The value of the "media-streams" element is an identifier, unique within the conference, which identifies the media stream that a user is connected to. The "media-stream" element also has a mandatory "media-type" attribute which identifies the media type (audio, video, message, text, data, control and application) of the media stream.

In addition to events defined in [7], the conference state needs to provide information about the current speaker, current chair and current focus referred with user URI. The current focus may delegate its focus role to another participant. The conference state has a modular definition and it is possible to access different conference aspects independently.

A user agent may invite another user agent or a list of user agents to the conference. A user agent may remove a participant or all participants from the conference.

The conference policy may define the announcement mechanism. If the announcement application is allowed to play an announcement to all the conference members (for example, to announce a join), it merely sends media to the mixer as would any other participant. Similarly, the announcement application can play an announcement to a specific user by configuring the conference policy so that the media it generates is only heard by the target user. The application then generates the desired announcement, and it will be heard only by the selected recipient.

The conference policy may also define participants' privacy, It may be possible for a conference participant to join the conference "anonymously" that is, the presence is announced but without identity disclosure. A conference participant may join a conference in a "hidden mode" without disclosure of the presence and the identity to the other participants.

Authentication and authorization are another kind of privileges. A participant may authenticate other participants in order to allow them to join the conference. This can be done implicitly by assigning a password to the conference or to each participant. During the conference, the participant may want to give a privilege to another participant. The assigning of privileges may be implicit when requested or explicit by asking the participant to grant a privilege.

A sidebar (sub-conference) appears to the users within the sidebar as a "conference within the conference". It is a conversation amongst a subset of the participants to which the remaining participants are not privy. The conference policy defines the user agent right to create and participate in a sub-conference with one or more participants, to split or to merge sub-conferences.

Fig.2 shows the approach to formal description of privileges in a SIP conference.

IV. FLOOR CONTROL POLICY FORMALISM

The floor is a permission to access or manipulate a specific shared resource or set of resources temporarily. The conference owner creates floors, and assigns and deassigns floor chairs. The floor chair is 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. The floor control policy describes the mechanism that enables applications or users to gain safe and mutually exclusive or non-exclusive input access to the shared object or resource.

Conference participants and floor chairs may be able to get and set floor-related parameters. The conference policy may restrict who may access or alter which parameters. The floor-related parameters defined [6] are described below.

The Beneficiary ID identifies uniquely a user within a conference. The Floor ID identifies uniquely a floor within a conference. The Floor Request ID identifies a floor request at the floor control server. The Request Status contains the status of the request. The Supported Attributes contain the types of the attributes that are supported by the floor control server. The Supported Primitives contain the types of the BFCP messages that are supported by the floor control server. The User Display Name contains the encoded name of the user. The User URI contains the user's contact URI, that is, the URI used by the user to set up the resources (e.g., media streams) that are controlled by BFCP. The Extension Attribute refers to attributes that may be defined in the future.

```
privileges = "privileges=(" 1*(privilege CRLF) ")" CRLF
privilege =conference-state / participant-manipulation /
        announcement / privacy / authentication / authorization /
        conference-policy-change / media- control-policy-change /
        floor-control-policy-change / subconference-participation /
        subconference-manipulation / floor-attribute-control
conference-state = "conf-state=(" 0*user-status CRLF 0*dialog-ID
        CRLF 0*(media-streams SP) CRLF current-speaker SP
        current-chair  SP current-focus ")" CRLF
user-status = "user-status=" ("active" / "departed" / "booted" / "failed"
media-stream = "media-stream= " media-ID SP media-type SP
participant-manipulation =add-participant SP remove-participant SP
add-participant = "add-participant=(" *(user-URI SP)  ")"
announcement = "announce-to=" *( user-URI SP) "media-
        announce:" media-type
privacy = "privacy=" (" false" / "anonymous" / "hidden")
authentication = " authentication=" right
authorization = " authorization= " right
conference-policy-change = "conf-policy-change=" right
media-control-policy-change = "media-policy-change=" right
floor-control-policy-change = "floor-policy-change=" right
subconference-manipulation = 1*(sub-conf-operation SP) CRLF
sub-conf-operation = create-subconference / merge-subconferences
        / split-subconferences / move-participant / join-subconference /
        invite-subconference
create-subconference = "create-subconf=:" right
merge-subconferences = "merge-subconf=" right 1*(subconference-
        ID SP)
```

Fig.2 An approach to formal description of participants' privileges

The conference policy defines whether floor control is in use or not. It is possible to define the algorithm to be used in granting the floor. Examples of algorithms are moderator-controlled, first-come-first-served (FCFS), or random. 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 is also 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.

During the conference, the participant may be able to manage whose media is being sent to each participant. For example, the participant may be able to decide that he wants to be a speaker and all the rest to be listeners; he may also specify whose media he wants to receive. Fig.3 shows an approach to formal definition of floor control policy and media control policy.

V. MEDIA CONTROL POLICY FORMALISM

A media topology graph is a loop-free graph which consists of individual media streams, logical groups of media streams, and functions or "operations" performed on those streams [7]. Within the media topology graph, each stream is described by

```
floor-control-policy = "floor-ctrl-policy=(" floor-attribute-control SP
         floor-control-algorithm ")" CRLF
floor-attribute-control = "floor-attr-ctrl=" access alter
access = "access=(" *(floor-attribute SP) ")" CRLF
floor-attribute = "floor-attr=" floor-request-information / floor-ID / floor-
         requested-ID / error-code / error-info / supported-attributes /
         supported-primitives
floor-request-information = "floor-request-info=(" overall-request-
         status  SP 1*(floor-request-status  SP) beneficiary-information
         SP requested-by-information SP priority SP  participant-
         provided-info  SP *(extension-attribute SP) ")" CRLF
overall-request-status = "overall-request-status=(" request-status  SP
         status-info SP 0*(extension-attribute SP) ")" CRLF
floor-request-status: = "floor-request-status=(" request-status SP
         status-info SP 0*(extension-attribute SP) ")" CRLF
beneficiary-information = "beneficiary-info=(" beneficiary-ID SP user-
         display-name SP user-URI ")" CRLF
requested-by-information: = "requested-by=(" user-display-name  SP
         user-URI SP 0*(extension-attribute SP) ")" CRLF
floor-control-algorithm = "floor-ctrl-algorithm=" "moderator-controlled"
         / "FCFS" / "random"
media-topology-graph = "media-topology-graph=("  "(" 1*( media-
         streams SP) ")" CRLF  "(" 1*(media-group SP) ")" CRLF "("
         1*(media-operation SP)  ")"  ")" CRLF
media-stream = "media-stream=" media-type SP direction SP media-
         id CRLF
direction = "dir=" "in" / "out"
media-group = "media-group=(" media-type direction SP bundle SP
         ")" CRLF
bundle = "bundle=" "(" 1*(participant SP) ")" CRLF
media-operation = "media-operation=(" "select=" right SP "combine="
         right SP "mix=" right ")" CRLF
media-control-policy = "media-control-policy = (" *(media-topology-
         graph SP) ")" ")" CRLF
```

Fig.3 An approach to formal description of floor control policy and media control policy

a media type, direction and at least one identifier. Stream identifiers can be network identifiers or aliases. Network identifiers consist of an address family (IPv4 or IPv6), an IP address, and a port number. Media groups have a media type, a direction and a bundle. Bundles represent a set of individually tagged logical streams. Operators are basic elements that perform simple media operations. They select among media streams, combine streams, or perform other media processing.

VI. Grammar Verification

We have verified the syntax of the formal grammar of conference policy description by the use of available ABNF parser generator [9]. Such a parser generator has at its input the formal conference policy description and generates, if correct ABNF rules are presented, the corresponding code of the parser which follows the grammar rules. When a textual description written according the conference policy grammar is passed to the generated parser, it checks the syntax of the description. Fig.4 gives an example of formal description of conference participant with focus capabilities that has privileges to add and remove two other participants. The focus URI is sip:ivan@tu-sofia.bg and the media allowed is audio and video. The focus participation is protected by password 99ABCD34.

```
conf-aware=(sip:ivan@tu-sofia.bg media-allowed=(audio video)
      participation=default isfocus conf-ID= sip:ivan@tu-sofia.bg
      password=99ABCD34
      privileges=
      (add-participant=(sip:ani@tu-sofia.bg sip:neli@tu-sofia.bg )
      remove-participant= (sip:ani@tu-sofia.bg sip:neli@tu-sofia.bg ) ))
```

Fig.4 An example of formal description of conference focus

VII. Conclusion

The formal description of conference policy grammar may be used in implementation of the Conference Policy server functionality which manages creation and deletion of conferences, authorization, conference longevity, and the media layout or topology.

There are available ABNF parser generators which can be used in accelerating the implementation phase of policy interpretation. The conference policy formal grammar is interpreted by such ABNF parser and the code generated can be complied and linked into the Conference Policy server. The Conference Policy server interprets the conference configuration files and applies a unified policy description grammar structure available for usage to the conference owners. Upon getting decisions about different conference instances, most probably owned by different owners, common policy structure is read, but having different policies' values.

Acknowledgement

The research is partially funded by the project 08107ni-7, “Research on Capabilities for Provisioning of Multimedia Services in Packet Networks”.

References

[1] J. Rosenberg, RFC 4353, “A Framework for Conferencing with the Session Initiation Protocol (SIP)”, 2006

[2] Khartabil H., P. Koskelainen, “Requirements for Conference Policy Control Protocol (CPCP) “, http://tools.ietf.org/html/draft-ietf-xcon-cpcp-reqs-04, 2005

[3] Levin O., R. Even, RFC 4245, “High-Level Requirements for Tightly Coupled SIP Conferencing”, 2005

[4] Rosenberg J., H. Schulzrinne, O. Livin RFC 4575, ”A Session Initiation Protocol (SIP) Event Package for Conference State”, 2006

[5] Koskelainen P., J. Ott, H. Schulzrinne, X. Wu, RFC 4376, “Requirements for Floor Control Protocols”, 2006

[6] Camarillo G., J. Ott, K. Drage, RFC 4582, “The Binary Floor Control Protocol (BFCP)”

[7] Mahy R. N. Ismail, **”**Media Policy Manipulation in the Conference Policy Control Protocol**”**, http://www.softarmor.com /wgdb/docs/draft-mahy-xcon-media-policy-control-00.html

[8] Crocker D., P. Overell, RFC 2234 “Augmented BNF for Syntax Specifications: ABNF”

[9] Lowell D. Thomas, “APG – an ABNF Parser Generator”, http://www.coasttocoastresearch.com/Apg5_0/Apg5_0_Download.htm, 2008