Articulated Narrowcasting for Privacy and Awareness in Multimedia Conferencing Systems and Design for Implementation Within a SIP Framework

Mohammad Sabbir Alam [*], Michael Cohen [†], and Ashir Ahmed [‡]

[*]R&D Dept., Mobile Technika Inc.
Shinjuku, Tokyo 162-0845
e-mail: sabbir@mobiletechnika.jp

[†]Spatial Media Group, University of Aizu
Aizu-Wakamatsu, Fukushima-ken 965-8580; Japan
e-mail: mcohen@u-aizu.ac.jp

[‡]Dept. of CSCE, Kyushu University
744 Moto'oka, Fukuoka 819-0395; Japan
email: ashir@c.csce.kyushu-u.ac.jp

Abstract

This article proposes a new focus of research for multimedia conferencing systems which allows a participant to flexibly select another participant or a group for media transmission. For example, in a traditional conference system, participants' voices might by default be shared with all others, but one might want to select a subset of the conference members to send his/her media to or receive media from. We review the concept of narrowcasting, a model for limiting such information streams in a multimedia conference, and describe a design to use existing standard protocols (SIP and SDP) for controlling fine-grained narrowcasting sessions.

First presented at the International Conference of Human and Computer 2006, extended and revised for JVRB

Keywords: Narrowcasting, SIP, SDP, conferencing, device capability, media direction control, privacy and awareness.

1 Introduction

Multimedia conferencing has been in the research agenda for many years. A traditional conferencing system over the PSTN (**p**ublic **s**witched **t**elephone **n**etwork) has many features implemented in a centrally controlled conference server. The development of IP technology has brought new media (e.g. video) into conferencing systems. H.323 [Pac03] and SIP (Session Initiation Protocol) [RSC$^+$02] [Joh04] are popular protocols for IP-based conferencing systems. SIP, a simpler text-based protocol developed by the IETF (Internet Engineering Task Force), added presence features allowing users to discover the availability of participants and also, with a large extension, to control media transmission and the direction from the endpoints. Although SIP was designed for multimedia conferencing systems, only VOIP applications have yet gained popularity in the industry and received priority in the SIP design community (**w**orking **g**roups). SIPPING [CBP$^+$08] and XCON [JRPJ08] WGs inside the IETF are considering conferencing frameworks. While SIPPING is designing a conferencing framework using SIP, the XCON system is independent of any signaling protocol. Both conferencing models focus only on centralized conferencing systems, where the signaling and media mixing are handled by a central conference server and centralized media mixer.

However, one may want to control media of a particular participant— e.g., participant P_1 wanting to block media from participant P_2 or wanting to receive media streams only from participant P_3. Controlling such media vectors from an endpoint has been a challenging issue. As a simple example, a user's voice might by default be shared with all others in a conference, but a versatile interface would allow a secret to be shared only with some selected subset of the members. Current commercially-available conference systems do not generally support such features.

Our research introduces a flexible multiparty multimedia user-adjustable conference system, including "narrowcasting" functionality, as an application within the SIP framework. A human user wants to distribute attention and availability, and narrowcasting provides a formalization of such presence filters. Narrowcasting systems extend broad- and multicasting systems by allowing media streams to be filtered— for relevancy control, privacy, security, and user interface optimization. As SIP was designed for multimedia session control, narrowcasting attributes can be implemented within the existing SIP framework. In this article, we propose a design for narrowcasting attributes and consider the feasibility of implementing it in a SIP framework.

The rest of this article is structured as follows. Section 2 reviews some background information regarding conferencing. In section 3 is explained our proposal for a SIP-based implementation. Section 4 details the call flow of narrowcasting implementation in SIP. Finally, the conclusion and ideas for future research are presented in section 5.

2 Conventional Conference Architecture and Call Control Limitations

This section discusses a common conference architecture, requirements of a typical conference systems, and limitations of existing systems.

2.1 Architecture

Conventional conferencing systems can be categorized into three different types, depending upon where media streams from participants are mixed.

Centralized Conferencing

A centralized conference [SKBR03] bridge exists in a centralized model. The conference bridge is a conceptually simple device, consisting of a SIP user agent to handle signaling, an RTP mixer to handle the media streams, a conference application layer for authentication, authorization & accounting services, and possibly conference control functions. Participants establish one-to-one media and signaling connections with the bridge. The bridge establishes voice paths between endpoints by collecting input signals and returning summed signals to conferees. Figure 1 illustrates how the media is mixed, (en/de)coded if necessary, and redistributed to participants.

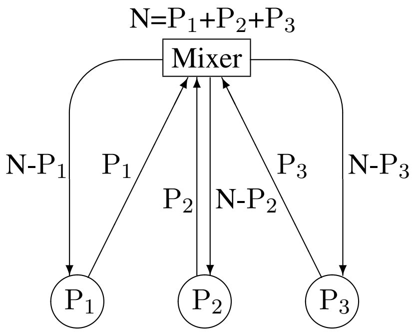


Figure 1: Media Mixing in a Centralized Conferencing

Most current multimedia conferencing systems fall into this category. As permissions are controlled by an administrator (a.k.a. floor controller), end users don't have much access to configuration features.

Decentralized Conferencing

In a decentralized model, signaling control is centralized but media are exchanged between participants without going through a centralized bridge. There is no conference server or central point of control. Decentralized conferencing can be either of two types: full mesh or multicast.

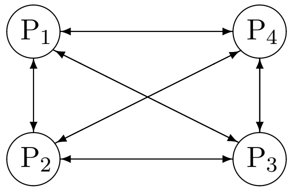


Figure 2: Full Mesh Conferencing

A. Full Mesh Conferencing: A full-duplex media link (Figure 2) can be established between every pair of participants, resulting in a fully-connected mesh. Each endpoint transmits a copy of its stream to the $N - 1$ other endpoints, and receives $N - 1$ streams in return, on separate ports. Each pair of participants can communicate through any mutually supported codec type.

B. Multicast Conferencing: In a multicast conference, participants join a session by subscribing to a conference multicast address. This address might be advertised by one of the participants or a central server, or distributed to conferees prior to a conference. Each participant transmits a single copy of his stream to the conference multicast address, receiving $N - 1$ streams in return. From a receiver perspective, nothing changes from the full mesh arrangement except that the streams arrive on a single port. Multicast conferences can scale up to millions of users and do not really require any SIP signaling. However, native multicast is not yet widely available.

2.2 Requirements of a Flexible Conferencing System

To implement a flexible end-to-end conferencing system, the following considerations apply:

General requirements: A conference control framework should be scalable, extensible, generic, reliable, and secure. The scalability requirement means that the conference control framework must support reasonably large, geographically distributed, conferences. Moreover, it should be extensibly modular so that new components can be easily added or existing components changed. The conference control framework must also be generic so that it is not tied to any particular application. While conference control protocols are likely to consume significantly less bandwidth than media streams, some care needs to be taken for large conferences. Since the conference description and policy information can be massive, incremental updates are preferred to having to resend entire descriptions after each change. Similarly, changes in participant lists should be distributed as additions and removals. Also, not all participants care about the same level of detail; for example, some may only be interested when new members join or leave, but not when a participant adds herself to a floor queue. The importance of reliability and security is obvious.

Session establishment: A mechanism is required to establish connections among multiple participants, to manipulate and describe media "mixing" or "topology" for multiple media types (audio, video, text, position data, etc.). SIP is a good candidate for this purpose. Technical challenges involve flexibly defining the media and its transmission using the SDP (Session Description Protocol) [HJ98].

Network resource management: Network resources are an important factor determining the communication quality of a conference, or "QoS" (**q**uality **o**f **s**ervice). Conferencing on a best-effort internet is an on-going challenge. Large delay or jitter irritates participants and degrades conference quality. Considering network characteristics and available bandwidth, proper encoding/decoding schemes must be deployed.

Policy: A user rights database specifies the privileges of potential participants. User rights lists might include information about who can authorize the admission or expulsion of participants and who can act on floor control requests. Such functions are often combined into the role of a moderator, but a flexible system should allow them to be distributed among a set of participants.

Security and privacy: Unwelcome participants are excluded, so no unauthorized party may intrude upon or eavesdrop in a conference. A mechanism for membership and authorization control is required. The policy may describe which users are pre-authorized to join ("white list"), are explicitly forbidden from joining ("black list" or "block list"), or may join but in listen-only ("lurk") mode. Since internet-based signaling protocols offer a variety of authentication mechanisms, a policy might also define at what strength each participant must authenticate. Unauthenticated users may be rejected or relegated to audience status.

Role-based policy: Conflicts may occur when participants with different priority levels try to set individual policies. For an example, a parent should be able to listen to children's audio streams, but children should not be allowed to mute or deafen a parent. A mechanism for articulating role paradigms for users is required, applicable in situations where central policies are administered.

Privacy and awareness: During a conference, a participant may want to share his media with only a subset of the conference, as in a private sub-caucus. Typically, other session participants would not be notified of such changes in media relationships. SIP and SIMPLE (SIP for Instant Messaging and Presence Leveraging Extensions) [SKPJ08] allow each participant to monitor the availability and other presence information regarding other participants. However, a participant may want to allow only particular participants to be notified of his presence. More importantly, a participant may want only some selected participants to share his media during the conference. This requirement applies to receiving media streams too.

Floor control requirements: Floor control should support different policies, such as moderator-controlled or first-come/first-served. A moderator-controlled policy is relatively easy to implement, but needs to be able to handle disconnection of the moderator. Automated queuing policies may cause starvation if one user holds the floor indefinitely. Time limits and renewable floor permissions are solutions to prevent filibusters and indefinite blocking.

2.3 Related Research

Over the years, there have been many studies in the area of conference control [KSW02] [SNS01]. Most earlier works discuss only floor control aspects of conference control. Standardization efforts have met with limited success. H.323, developed by the ITUT, has several problems, including scalability issues due to insufficient T.124 database replication protocol and its limitation to binary ASN.1 format (not text-based) protocol. SIP, in contrast, is a text-based protocol which can easily interact with other internet protocols. SIP is a signaling protocol for creating, modifying, and terminating multimedia sessions between multiple participants. Conferencing is possible using standard SIP methods [RSC+02], allowing users to join and leave conferences and allowing invitation of other participants. However, SIP by itself does not offer configurable conference policies, participant access lists, floor control, or user privilege levels. The SIPPING (Session Initiation Protocol Project Investigation) [CBP+08] WG is chartered to develop requirements for extensions to SIP needed for multi-party applications. XCON, working closely with SIPPING, focuses on development of a standardized suite of protocols for tightly coupled multimedia conferences [JRPJ08].

A limitation of traditional conferencing systems is that a participant (not a conference administrator) can not control other participants' displays. Current conferencing systems generally do not have capability to select a subset of the conference participants to whom his media are sent or from whom streams are received. In this article, we introduce narrowcasting attributes to implement media restriction features within a SIP framework.

3 Enhancement of Conferencing System: Narrowcasting

In this section, we describe the feature set for narrowcasting in SIP-based conferences. In our group's earlier publications [FCDK05] [ACA05] [FAD+06], we introduced the concept of narrowcasting attributes, described functions to apply these features in a standard conferencing model (recapitulated in Figure 6), and proposed how features could be implemented using standard SIP methods and headers defined in RFC 3261 [RSC+02]. Advantages of such a deployment include the convenience that no new methods or header extensions would be required to implement the features.

3.1 Narrowcasting Concept

Figure 3: Media Restriction (Narrowcasting Attribute)

Figure 3 shows a famous Japanese sculpture which is good example of narrowcasting attributes. Three monkeys: *Mizaru* (the monkey with eyes covered), *Iwazaru* (mouth covered), and *Kikazaru* (ears blocked) manifest the notion of limiting media vectors. Mizaru can not see (but can hear and speak); Iwazaru can not speak (but can see and hear); Kikazaru can not hear (but can speak and see).

In analogy to broad-, multi-, and any-casting, narrowcasting is a technique for limiting and focusing information streams, either sources or sinks (receivers). We employ the paradigm of multiple simultaneous chatspaces, each with several or many conversants and across which one has "multipresence," permitted designation of multiple instances of one's "self." The audio windows narrowcasting predicate calculus [Coh00] is an formalization for such a permission scheme. In Table 1, narrowcasting audio attributes are listed and their characteristics explained. This article proposes deployment of these attributes within a SIP framework.

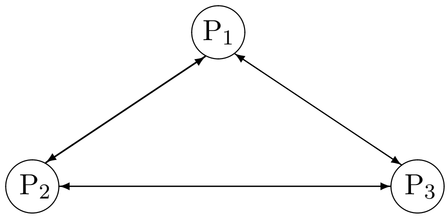


Figure 4: A Three-Party Conference Model

Figure 4 shows the initial state of conference in which three participants— P_1, P_2 and P_3— can talk to and hear each other. In other words, all the participants are in a fully connected media relationship. Our design will allow each user to send or receive data streams to/from a specific set of recipients in a session. For easier understanding, we consider only audio streams in this article. However, this design applies equally well to other media types, including video, text, and data (geographic location, for example).

3.1.1 Source Functions: `Mute` and `Select`

A "mute" function is available in present-day conference systems. However, in most cases, a participant mutes herself by connecting the other conversant to "music on hold." On-hold parties hear to the music, but no voice media is transmitted. In our definition, a user can explicitly mute another party.

In Table 2(a), three participants participate in a conference in which P_2 has been `muted` by P_1. This means P_1 doesn't want to hear P_2, but only P_3. Specifically,

- P_1 has a simplex (one-way) relationship with P_2, $P_1 \rightarrow P_2$.
- P_1 has a duplex (two-way) media relationship with P_3, $P_1 \leftrightarrow P_3$.
- P_2 has a duplex media relationship with P_3, $P_2 \leftrightarrow P_3$.

As a result,

- When P_1 speaks, both P_2 and P_3 will hear.

Attributes	Description
`Mute`	blocks the media stream coming from a source. In Table 2(a,b), P_1 mutes P_2, i.e. P_1 blocks the media coming from P_2. As a result, P_1 does not hear P_2. However, P_2 can still hear P_1.
`Select`	limits the projected sound to particular sources. In Table 2(c,d), P_1 selects P_2, i.e. P_1 focuses on media coming from P_2. As a result, P_1 can listen only to P_2's voice; P_1 can not hear other participants.
`Deafen`	blocks media streams going to a sink. In Table 2(e,f), P_1 deafens P_2, i.e. P_1 blocks media going towards P_2. As a result, P_2 can not hear P_1. The relationship between P_1 and other participants remains the same.
`Attend`	limits received sound to particular sinks. In Table 2(g,h), P_1 attends P_2, i.e. media from P_1 can go only to P_2. As a result, only P_2 can hear P_1 but others can't.

Table 1: Proposed Audio Narrowcasting Attributes

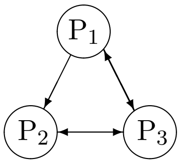
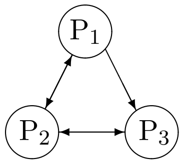
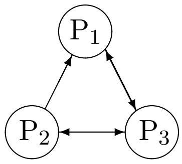
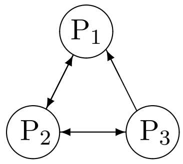
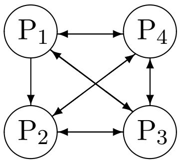
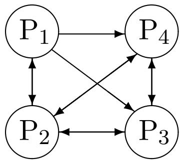
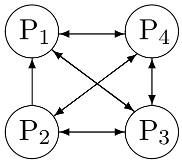
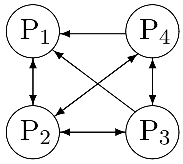
Control	`Mute`	`Select`	`Deafen`	`Attend`
$P_1 \rightarrow P_2$	 (a)	 (c)	 (e)	 (g)
	 (b)	 (d)	 (f)	 (h)
Situation	A participant wants to block media from a specific participant. In this scenario, P_1 `mutes` P_2.	A participant wants to receive media only from a particular participant. In this scenario, P_1 `selects` P_2.	A participant wants to block media to specific participant(s). In this scenario, P_1 `deafens` P_2.	A participant wants to send media to a specific participant. In this scenario, P_1 `attends` P_2.
Result	P_1 has only send-only relationship with P_2. Other media vectors remain the same.	Only $P_1 \leftrightarrow P_2$ remains same. Other participants have receive-only media relationship with P_1.	P_1 has a receive-only media relationship with P_2. Others remain the same.	Media from P_1 only goes to P_2. Others only send to P_1 but cannot receive media from P_1.

Table 2: Narrowcasting Models

- When P_2 speaks, only P_3 will hear (and NOT P_1).
- When P_3 speaks, both P_1 and P_2 will hear.

Equivalently for this simple example, P_3 might be `select`ed by P_1. The connectivity matrix of the situation shown in Table 2(a) can be portrayed as

$$\begin{array}{cccc} & P_1 & P_2 & P_3 \\ P_1 & \times & 1 & 1 \\ P_2 & 0 & \times & 1 \\ P_3 & 1 & 1 & \times \end{array}$$

representable in matrix form as

$$\begin{bmatrix} \times & 1 & 1 \\ 0 & \times & 1 \\ 1 & 1 & \times \end{bmatrix}.$$

where entry c_{ij} of the matrix represents connectivity of source i to sink j, and the main diagonal is populated by "don't care"s.

A scenario with four participants in a session is shown in Table 2(d). Here P_2 is `select`ed by P_1, so P_1 can hear only P_2 but not others. Other participants can hear as usual. The connectivity of Table 2(d) is represented as

$$\begin{bmatrix} \times & 1 & 1 & 1 \\ 1 & \times & 1 & 1 \\ 0 & 1 & \times & 1 \\ 0 & 1 & 1 & \times \end{bmatrix}.$$

3.1.2 Sink Functions: `Deafen` and `Attend`

Remote `deafen` is also available in full-functioned conferencing systems as "Listen-only mode." In most cases, only an end-user or administrator may invoke this feature. In our definition, any user can control the media sent to or received from another.

In Table 2(e), P_2 is `deafend` by P_1. This means P_1 doesn't want to send his voice to P_2 to hear. Specifically,

- P_1 has a simplex media relationship with P_2, $P_1 \leftarrow P_2$.

- P_1 has a duplex media relationship with P_3.
- P_2 has a duplex media relationship with P_3.

In this case:

- When P_1 speaks, P_3 will hear, but P_2 won't.
- When P_2 speaks, both P_1 and P_3 will hear.
- When P_3 speaks, both P_1 and P_2 will hear.

Equivalently, P_3 might be `attend`ed by P_1, so that only P_3 can hear P_1. P_1 could still hear all other streams. The connectivity matrix for Table 2(e) is

$$\begin{bmatrix} \times & 0 & 1 \\ 1 & \times & 1 \\ 1 & 1 & \times \end{bmatrix}.$$

In Table 2(h), P_2 is `attend`ed by P_1. As a result only P_2 can hear from P_1. The connectivity matrix of this situation is

$$\begin{bmatrix} \times & 1 & 0 & 0 \\ 1 & \times & 1 & 1 \\ 1 & 1 & \times & 1 \\ 1 & 1 & 1 & \times \end{bmatrix}.$$

For egalitarian models with flat hierarchies, there is an asymmetry regarding both `mute/select` and `deafen/attend`: audibility of a source with respect to a sink is treated as a revocable privilege and a forsakable right. A sink can by default hear collocated sources, adjustable by narrowcasting commands. For example, if P_2 `attends` P_1 but P_1 has `mute`d P_2, P_1 won't hear P_2. Further policy extensions will extend the permissions of such a protocol, including the ability to force audibility by overriding a source's `mute` or sink's `deafen` (which a parent might invoke when telechiding a distracted child: "How dare you mute me‽"). Consideration of such role-based issues will be the focus of future research.

3.2 SIP for Multimedia Conferencing

Peers in a SIP session are called user agents, and can function in the following roles:

User-Agent Client (UAC) A client application that initiates a SIP request.

User-Agent Server (UAS) A server application that contacts the user when a SIP request is received and returns a response on behalf of the user.

A SIP end-point is capable of functioning as both a UAC and a UAS, but typically functions as only one or the other per session, depending upon the user agent that initiated the request.

SIP makes use of elements called proxy servers to help route requests to users' current locations, authenticate and authorize users for services, implement provider call-routing policies, and provide features to users. SIP also provides a registration function that allows users to upload their current locations (IP addresses) for use by proxy servers.

3.3 Session Establishment in SIP

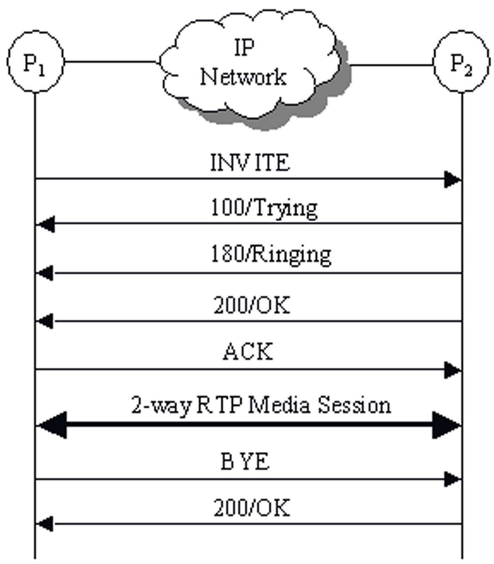


Figure 5: Call Flow of a Typical, Simple SIP Session

A typical hand-shaking exchange is shown in Figure 5, P_1 sending an INVITE request with media capabilities to P_2. A 100/TRYING and a 180/RINGING message confirm that P_2 is being alerted. A 200/OK message (which might also contain the final session description message body, whose significance will be explained later) is sent once P_2 accepts the INVITE, notifying that a connection has been made. Upon receiving the 200/OK from P_2, P_1 sends an ACK, usually triggered by a human user. A two-party duplex session is established at this point. The delay between the 180/RINGING and 200/OK messages depends upon after how many rings the user accepts the call. Participants wishing to leave a session send a BYE request within the session dialog [ACA04].

SIP signaling can be transported on either TCP or UDP; a standard SIP entity must support both types

[RSC+02]. For realizing narrowcasting attributes over SIP, a client will follow the guideline of RFC 3261 Section 18: If a request is within 200 bytes of the path MTU (**m**aximum **t**ransmission **u**nit), or if it is larger than 1300 bytes, or the path MTU is unknown, the request must be sent using an RFC 2914 congestion-controlled transport protocol, such as TCP.

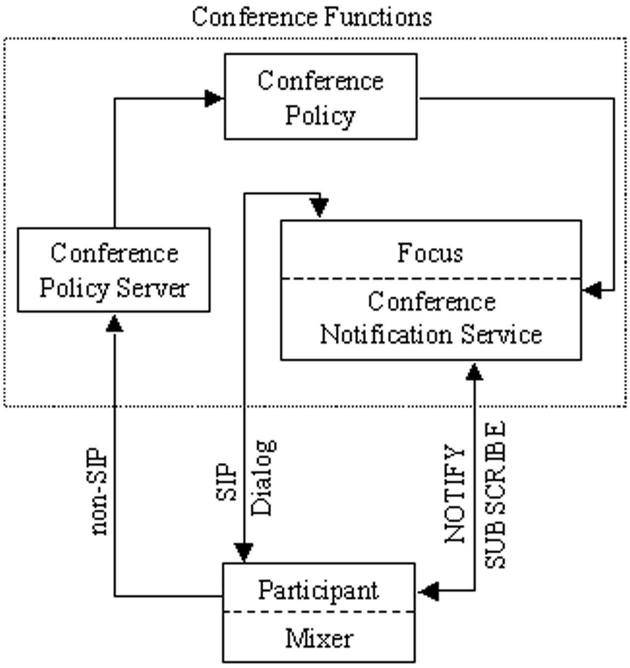


Figure 6: Conferencing Model

3.4 The Conferencing Model

Narrowcasting attributes can be implemented in both centralized and decentralized conferences. This article focuses on a decentralized conference architecture, for which the media is mixed at each end-point. Figure 6 illustrates components of the conferencing system and their roles. We have extended the model being proposed by the IETF with narrowcasting attributes.

Focus: The focus is a SIP user agent addressed by a conference URI (**u**niform **r**esource **i**dentifier). It handles SIP signaling between participants in a conference. The focus establishes media exchange among participants in a conference, and also implements conference policies. Its logical role is in analogy to that of a controller in a centrally signaling, distributed media architecture.

Participants: User agents are identified by a URI, communicating with each other after having been connected through the focus.

Conference notification service: The focus can act logically as a notifier [Roa02], accepting subscriptions to the conference and notifying subscribers about changes to that state. The state includes the state maintained by the focus itself, the conference policy, and the media policy.

Conference policy server: A conference policy server stores and manipulates rules using an XCAP (**Ex**tensible Markup Language **C**onfiguration **A**ccess **P**rotocol) [Ros07] database associated with participation in a conference. These rules include directives on the lifespan of the conference, who can and cannot join it, who can override the media policy, definitions of roles available in the conference, and the responsibilities associated with those roles.

Conference policy: The complete set of rules governing a particular conference is interpreted and enforced by the conference policy server.

4 Design for Implementation of Narrowcasting Attributes in SIP

Implementation of narrowcasting attributes inside SIP can be implemented by modifying only the generator of the SDP message body. Section 3.3 described session establishment in SIP, where SDP is used to indicate media capabilities and destination addresses.

Media negotiation is part of the INVITE/200/ACK sequence to establish a SIP session between two endpoints. SIP itself doesn't provide media negotiation, but it enables media negotiation between user agents using SDP. Each participant sends information via SDP in either an INVITE or in an ACK about her terminal's media capabilities and the transport address at which she wishes to receive RTP packets. In the SDP body attached to the SIP header, the user agents specify the media type, codec, IP address, and port number for each media stream. In the message body of the 200/OK response to the INVITE, the server sends the transport address to which the participant should send his accepted media capabilities RTP packets. Our implementation in SIP [ACA07] will use the narrowcasting attributes `mute`, `select`, `deafen`, and `attend`, along with the media capabilities in the INVITE/200/ACK sequence in the SDP bodies.

Figure 4 showed multiparty voice communication between P_1, P_2, and P_3. Considering the participants' media flow, we propose the protocol elaborated below.

In our design we consider the existing standard media session and send a re-INVITE by modifying the SDP body.

4.1 Mute

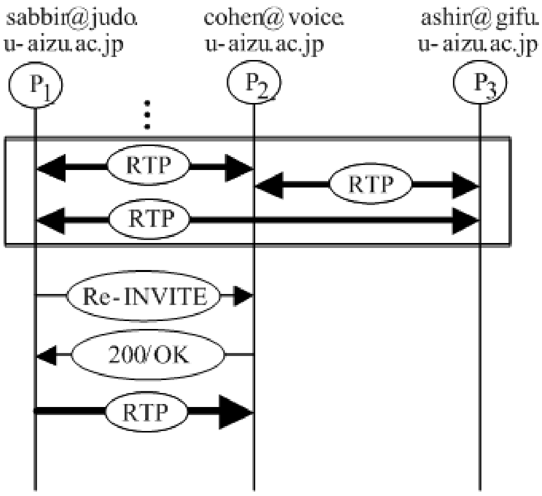


Figure 7: Mute Call Flow

Figure 7 illustrates a scenario in which P_1, P_2, and P_3 are in an RTP media session. If P_1 wants to mute P_2, P_1 sends a re-INVITE to P_2 with a modified SDP attribute, a=sendonly. P_2 then responds with 200/OK including a=recvonly along with other SDP attributes. As the negotiation determines to only send media from P_1 to P_2, a one-way RTP connection is established ($P_1 \rightarrow P_2$). Thus is P_1 muted by P_2. The status of other participants (i.e., P_3 in this example) remains unchanged. An example of the re-INVITE/OK handshake in Figure 7 is shown below, where the first block of each log is the SIP header and the second block is the SDP body.

```
INVITE sip:cohen@voice.u-aizu.ac.jp
SIP/2.0
Via: SIP/2.0/UDP 123.456.789.101
From: sabbir <sip:sabbir@judo.u-aizu.ac.jp>
To: cohen <sip:cohen@voice.u-aizu.ac.jp>
Call-ID:627802096@judo.u-aizu.ac.jp
CSeq: 1 INVITE
Contact:<sip:sabbir@123.456.789.101>
Content-type: application/sdp
Content-Length: 110

v=0
o=sabbir 2345 3345 IN IP4 judo.u-aizu.ac.jp
c=IN IP4 123.456.789.101
m=audio 2410 RTP/AVP 0
a=sendonly
```

The 200/OK sequence looks like

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 123.456.789.101
From: sabbir<sip:sabbir@judo.u-aizu.ac.jp>
To: cohen <sip:cohen@voice.u-aizu.ac.jp>;
tag=659882290
Call-ID:627802096@1judo.u-aizu.ac.jp
CSeq: 1 INVITE
Contact:<sip:cohen@123.456.789.102>
Content-type: application/sdp
Content-Length: 110

v=0
o=sabbir 2345 3345 IN IP4 voice.u-aizu.ac.jp
c=IN IP4 123.456.789.102
m=audio 2410 RTP/AVP 0
a=recvonly
```

4.2 Deafen

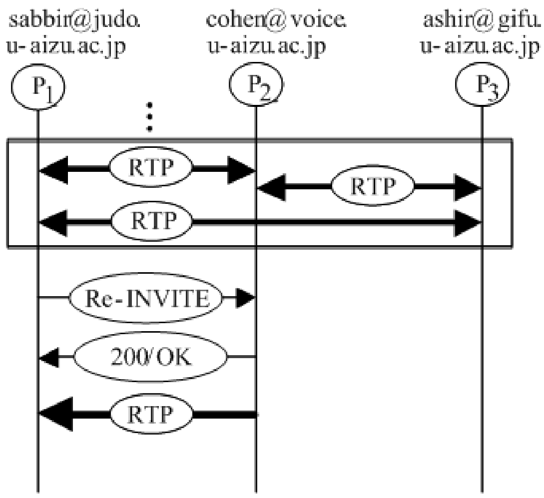


Figure 8: Deafen Call Flow

In order to deafen P_2, P_1 sends a re-INVITE to P_2 with a modified SDP attribute, a=recvonly. P_2 then responds with 200/OK including a=sendonly along with other SDP attributes. As the negotiation determines only to transmit the media from P_2 to P_1, a simplex media connection is established ($P_2 \rightarrow P_1$), thereby deafening P_2 by P_1.

```
INVITE sip:cohen@voice.u-aizu.ac.jp
SIP/2.0
Via: SIP/2.0/UDP 123.456.789.101
From: sabbir <sip:sabbir@judo.u-aizu.ac.jp>
To: cohen <sip:cohen@voice.u-aizu.ac.jp>
Call-ID:627802097@judo.u-aizu.ac.jp
CSeq: 2 INVITE
Contact:<sip:sabbir@123.456.789.101>
Content-type: application/sdp
```

```
Content-Length: 110

v=0
o=sabbir 2345 3345 IN IP4 judo.u-aizu.ac.jp
c=IN IP4 123.456.789.101
m=audio 2410 RTP/AVP 0
a=recvonly
```

The 200/OK sequence looks like

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 123.456.789.101
From: sabbir<sip:sabbir@judo.u-aizu.ac.jp>
To: cohen <sip:cohen@voice.u-aizu.ac.jp>;
tag=659882291
Call-ID:627802097@1judo.u-aizu.ac.jp
CSeq: 2 INVITE
Contact:<sip:cohen@123.456.789.102>
Content-type: application/sdp
Content-Length: 110

v=0
o=sabbir 2345 3345 IN IP4 voice.u-aizu.ac.jp
c=IN IP4 123.456.789.102
m=audio 2410 RTP/AVP 0
a=sendonly
```

4.3 `Select`

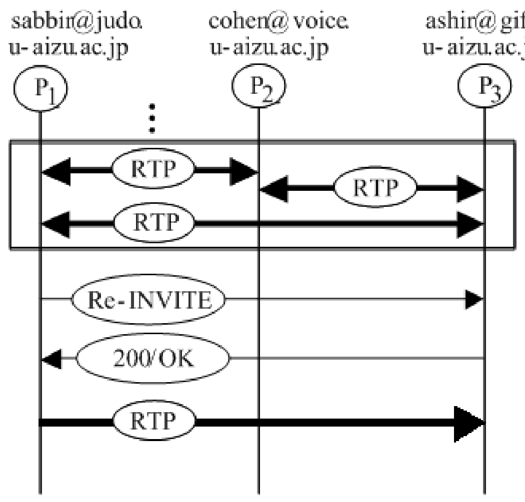


Figure 9: `Select` Call Flow

In order for P_1 to select P_2, P_1 sends a re-INVITE to all other participants except for P_2 with a modified SDP, `a=sendonly`, and other participants in the conference respond with 200/OK with `a=recvonly` along with other SDP attributes. A one-way media connection is established between P_1 and other participants excepting P_2, so P_2 is `selected` by P_1.

```
INVITE sip:ashir@gifu.u-aizu.ac.jp
SIP/2.0
Via: SIP/2.0/UDP 123.456.789.101
From: sabbir <sip:sabbir@judo.u-aizu.ac.jp>
To: ashir <sip:ashir@gifu.u-aizu.ac.jp>
Call-ID:627802098@judo.u-aizu.ac.jp
CSeq: 3 INVITE
Contact:<sip:sabbir@123.456.789.101>
Content-type: application/sdp
Content-Length: 110

v=0
o=sabbir 2345 3345 IN IP4 judo.u-aizu.ac.jp
c=IN IP4 123.456.789.101
m=audio 2410 RTP/AVP 0
a=sendonly
```

A 200/OK from P_3 returned to P_1 confirms the implicit `mute`.

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 123.456.789.101
From: sabbir <sip:sabbir@judo.u-aizu.ac.jp>
To: ashir <sip:ashir@gifu.u-aizu.ac.jp>;
tag=659882292
Call-ID:627802098@1judo.u-aizu.ac.jp
CSeq: 3 INVITE
Contact:<sip:ashir@123.456.789.103>
Content-type: application/sdp
Content-Length: 110

v=0
o=sabbir 2345 3345 IN IP4 sound.u-aizu.ac.jp
c=IN IP4 123.456.789.103
m=audio 2410 RTP/AVP 0
a=recvonly
```

4.4 `Attend`

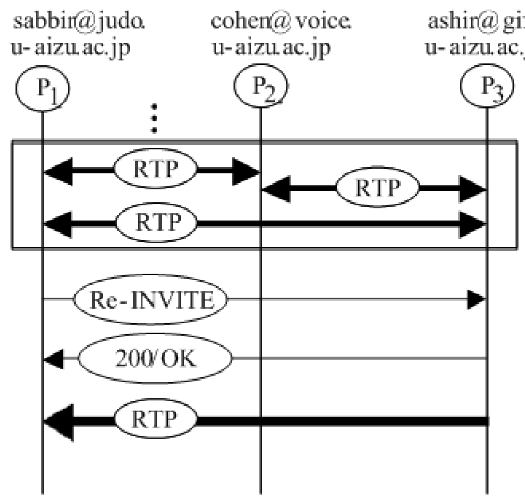


Figure 10: `Attend` Call Flow

As illustrated by Figure 10, P_1 sends a re-INVITE to all other participants (except for P_2) with a modified SDP attribute, `a=recvonly`, who respond with 200/OK including `a=sendonly` along with other SDP attributes. A one-way RTP media connection is thus established with other participants (excepting P_2), so P_2 is `attended` by P_1.

```
INVITE sip:ashir@gifu.u-aizu.ac.jp
SIP/2.0
Via: SIP/2.0/UDP 123.456.789.101
From: sabbir <sip:sabbir@judo.u-aizu.ac.jp>
To: ashir <sip:ashir@gifu.u-aizu.ac.jp>
Call-ID:627802099@judo.u-aizu.ac.jp
CSeq: 4 INVITE
Contact:<sip:sabbir@123.456.789.101>
Content-type: application/sdp
Content-Length: 110

v=0
o=sabbir 2345 3345 IN IP4 judo.u-aizu.ac.jp
c=IN IP4 123.456.789.101
m=audio 2410 RTP/AVP 0
a=recvonly
```

The 200/OK sequence looks like

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 123.456.789.101
From: sabbir <sip:sabbir@judo.u-aizu.ac.jp>
To: ashir <sip:ashir@gifu.u-aizu.ac.jp>;
tag=659882293
Call-ID:627802099@1judo.u-aizu.ac.jp
CSeq: 4 INVITE
Contact:<sip:ashir@123.456.789.104>
Content-type: application/sdp
Content-Length: 110

v=0
o=sabbir 2345 3345 IN IP4 sound.u-aizu.ac.jp
c=IN IP4 123.456.789.104
m=audio 2410 RTP/AVP 0
a=sendonly
```

5 Conclusion and Future Work

In ordinary conversation, participants generally observe turn-taking, as in a CDMA (**c**ollision **d**etection, **m**ultiple **a**ccess) protocol with discretionary backup. That is, an utterance that collides with another will cause one or both of the simultaneous speakers to stop and wait until a break before repeating.

One might wonder what happens to such conversational turn-taking in the presence of asymmetric media filters and the absence of a moderator. Narrowcasting features— like blocklists, side channels, and call-within-a-call— complicate teleconferences, since a deafened conversant might not be aware that another is talking and multiple sources might speak at once. If some avatars in a conference are muted or deafened to some other participants, without formal floor control there is a danger of some “talking on top of” others. In the absence of common floor control, won’t private chats and decentralized control lead to anarchy? Without “traffic signals,” how can collisions be avoided?

In fact, such parallel conversation streams are not a problem. For example, if two participants set up a private side-conference using narrowcasting commands, even though their utterances might collide with others’, they wouldn’t expect or want others to stop conversing. Rather they “listen with one ear” to ongoing conversations while enjoying their own caucus. Listeners can still untangle conversational threads, by context, voice quality, etc. Just as in real social contexts, including informal gatherings like parties, multiple simultaneous speakers are analyzable. Even “linear” conversations like formal meetings might have some subsets of conversants whispering among themselves while a main speaker is talking. Narrowcasting interfaces will be even more useful when extended by spatial audio and attenuation based on mutual virtual position (source projection, sink bearing, and distance), distributing the respective voices across a soundscape.

The status of each participant’s privacy in terms of the media relationship with other participants requires consideration. In this article, we have introduced a design of new features for multimedia conferencing systems. These features could provide enhanced conference functions at the user end “the edge of the network,” rather than at the server. As a result, a conference participant (not an administrator) could easily control media transmission. We also described the design of these features and method of implementation within the standard SIP framework.

Future challenges include developing an algorithm for role-based policy, and adaptive media-mixing at a centralized media mixer for subscribed users.

References

[ACA04] Mohammad Sabbir Alam, Michael Cohen, and Ashir Ahmed, *A Case Study of VoIP Performance Across Different Networks*, Proc. ICECE: 3rd **I**nt. **C**onf. on **E**lectrical & **C**omputer **E**ngineering (Dhaka), December 2004, ISBN 984-32-1804-4, pp. 295–298.

[ACA05] Mohammad Sabbir Alam, Michael Cohen, and Ashir Ahmed, *Design of Narrowcasting Implementation in* SIP, Proc. HC: Eighth Int. Conf. on **H**uman and **C**omputer (Aizu-Wakamatsu), August 2005, pp. 255–260.

[ACA07] Mohammad Sabbir Alam, Michael Cohen, and Ashir Ahmed, *Narrowcasting— Controlling Media Policy in* SIP *Multimedia Conferencing*, Proc. IEEE CCNC: 4th **C**onsumer **C**ommunications and **N**etworking **C**onf. (Las Vegas), January 2007.

[CBP+08] Gonzalo Camarillo, Mary Barnes, Jon Peterson, Cullen Jennings, and Oscar Novo, *Session Initiation Proposal Investigation (Sipping)*, www.ietf.org/html.charters/sipping-charter.html, 2008, Last Accessed July 11th, 2008.

[Coh00] Michael Cohen, *Exclude and Include for Audio Sources and Sinks: Analogs of mute & solo are deafen & attend*, Presence: Teleoperators and Virtual Environments **9** (2000), no. 1, 84–96, ISSN 1054-7460; `www.u-aizu.ac.jp/~mcohen/welcome/publications/ie1.pdf`.

[FAD+06] Owen Noel Newton Fernando, Kazuya Adachi, Uresh Duminduwardena, Makoto Kawaguchi, and Michael Cohen, *Audio Narrowcasting and Privacy for Multipresent Avatars on Workstations and Mobile Phones*, IEICE Trans. on Information and Systems **E89-D** (2006), no. 1, 73–87, `ietisy.oxfordjournals.org/cgi/content/refs/E89-D/1/73`, ISSN 0916-8532.

[FCDK05] Owen Noel Newton Fernando, Michael Cohen, Uresh Chanaka Dumindawardana, and Makoto Kawaguchi, *Duplex narrowcasting operations for multipresent groupware avatars on mobile devices*, IJWMC: Int. J. of Wireless and Mobile Computing **1** (2005), no. 5, Special Issue on Mobile Multimedia Systems and Applications, ISSN 1741-1084.

[HJ98] M. Handely and V. Jacobson, RFC *2327 – SDP: Session Description Protocol*, www.ietf.org/rfc/rfc2327.txt, 1998, Last Accessed July 11th, 2008.

[Joh04] Alan B. Johnston, SIP*: Understanding the Session Initiation Protocol*, Artech House, 2004, ISBN 1580531687.

[JRPJ08] Alan Johnston, Adam Roach, Jon Peterson, and Cullen Jennings, *Centralized Conferencing (*XCON*)*, www.ietf.org/html.charters/xcon-charter.html, 2008, Last Accessed July 11th, 2008.

[KSW02] Petri Koskelainen, Henning Schulzrinne, and Xiaotao Wu, *A* SIP*-based Conference Control Framework*, NOSSDAV '02: Proc. 12th Int. Wkshp. on Network and Operating Systems Support for Digital Audio and Video (New York, NY), ACM Press, 2002, ISBN 1-58113-512-2, pp. 53–61.

[Pac03] *ITU-T Recommendation H.323— Packet-based Multimedia Communications Systems*, www.itu.int/rec/T-REC-H.323-200307-S/en, 2003, Series H: Audiovisual and multimedia systems, Last Accessed July 11th, 2008.

[Roa02] A. B. Roach, *RFC 3265— Session Initiation Protocol (SIP)— Specific Event Notification*, www.ietf.org/rfc/rfc3265.txt, 2002, Last Accessed July 11th, 2008.

[Ros07] J. Rosenberg, *RFC 4825— The Extensible Markup Language (*XML*) Configuration Access Protocol (*XCAP*)*, www.ietf.org/rfc/rfc4825.txt, 2007, Last Accessed July 11th, 2008.

[RSC+02] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, RFC *3261 – SIP: Session Initiation Protocol*, www.ietf.org/rfc/rfc3261.txt, 2002, Last Accessed July 11th, 2008.

[SKBR03] Paxton J. Smith, Peter Kabal, Maier L. Blostein, and Rafi Rabipour, *Tandem-Free VoIP Conferencing: A Bridge to Next-Generation Networks*, IEEE Communications Magazine (2003), 136–145.

[SKPJ08] Robert Sparks, Hisham Khartabil, Jon Peterson, and Cullen Jennings, SIP *for Instant Messaging and Presence Leveraging Extensions (simple)*, www.ietf.org/html.charters/simple-charter.html, 2008, Last Accessed July 11th, 2008.

[SNS01] Kundan Singh, Gautam Nair, and Henning Schulzrinne, *Centralized Conferencing using* SIP, In Proc. Internet Telephony Workshop (New York), April 2001.

Citation

Mohammad Sabbir Alam, Michael Cohen, and Ashir Ahmed, *Articulated Narrowcasting for Privacy and Awareness in Multimedia Conferencing Systems and Design for Implementation aithin a* SIP *Framework*, Journal of Virtual Reality and Broadcasting, 5(2008), no. 14, Dec. 2008, urn:nbn:de:0009-6-14724, ISSN 1860-2037.