Group Conference Management with SIP

Thomas C. Schmidt [1]
Matthias Wählisch[2,1]
{t.schmidt,waehlisch}@ieee.org
[1]HAW Hamburg, Department Informatik, Berliner Tor 7, 20099 Hamburg, Germany
Fon: +49 40 42875 - 8157, Fax: +49 40 42875 - 8409
[2]link-lab, Hönower Str. 35, 10318 Berlin, Germany
Fon: +49-30.2023 8871-0 , Fax: +49-30.2023 8871-9

March 1, 2008

Contents

Chapter 1

Group Conference Management with SIP

1.1 Introduction

Voice and video conferencing in the Internet is about to become a lightweight day-to-day application. This trend follows from high bandwidth data connections, which are increasingly available to the public at reasonable prices. There is also some remarkable progress in video/audio compression algorithms, which reduce a media data stream considerably and reconstruct it again to a high quality playout sequence on the receiver site. Furthermore battery powered mobile devices rapidly gain processing and communication performance. They nowadays can host desktop conferencing software, thereby seamlessly using any Internet connectivity available.

Conferencing is known for years from the ITU telephone architecture as a feature-rich centralized service, suffering from its static, complex and expensive nature. Many of the issues in multi-party communication can be solved in a simpler, more generic way in the Internet, where users, groups and devices can be addressed individually and independent of each other. The Session Initialization Protocol (SIP) [32] as presented by the IETF forms a flexible, comprehensive signaling solution and allows for a lightweight deployment of many, but not all of the well-known service features. Since media sessions are only negotiated by, but not tight to SIP control streams, calls may be arranged in a variable, adaptive fashion. This degree of freedom is of particular importance in large conferences, where media transmission and processing can easily rise beyond capacities of single devices.

The original development of SIP has been inspired by connection oriented telephone services, whence its nature derives from a point-to-point model. It is designed as a multi-layered application protocol that interacts between components in a transactional way. Each (asynchronous) request initiates an open transaction state and requires completion by at least one response. Group communication complicates this process significantly. A newly joining member faces an entire group, which requires appropriate addressing *and* transactional state management. Negotiations on media parameters grow complex as common parameter intersections may have to be evaluated for many members. Extensions to perform scalable group session management are not easy to achieve, while schemes refrain from central control.

In multimedia conference scenarios each member commonly operates as receiver and as sender on a group communication layer. In addition, real-time communication such as voice or video over IP places severe QoS requirements: Seamless distribution services need to limit disruptions or delay to less than 100 ms. Jitter disturbances should not to exceed 50 ms. Note that 100 ms is about the duration of a spoken syllable in real-time audio.

As an extensible protocol, SIP is open for the creation of new methods, header fields and protocol semantics. This opportunity is extensively used for conferencing, resulting in a large number of conceptual documents describing standard extensions, best current practices, draft proposals etc., see [28] for a guided overview. The present contribution aims to withstand an enumeration of proposed features, but rather concentrates on core concepts and tries to outline conferencing solutions of common use as well as future directions of promising development.

This chapter will at first illustrate the fundamental issues and SIP concepts for multi-party conversations along the line of examples and characteristic applications. An overview of core concepts and technologies for SIP initiated group conferencing follows in section 1.2. Point-to-point schemes and multicast solutions are covered herein, as well as mobility aspects. Section 1.3 then takes a closer look on SIP standard infrastructure components and their potential to facilitate group conferencing in an uncomplex manner. A detailed discussion of conferences solely managed by peers will be in focus of the subsequent section 1.4. Finally, a summary and conclusions will complete this report on conferencing.

1.1.1 Two Introductory Scenarios

Group communication can be of manifold nature and a variety of quite different views or scenarios have been contributed to the field. For a start at a prime perspective, two well known and established synchronous group communication services from everyday life are considered: The 3-way conference like in ISDN telephone systems on the one hand, and the reception of live broadcast media like in radio and television, with an optional offer of selective feedback channels, on the other. While in the first scenario users act primarily dialog-oriented, most of them are bound to passive reception in the distribution-oriented second setup. In the following a closer inspection will reveal further characteristic differences.

3-way Conference

Often a 3-way conference is initiated spontaneously from a 2-party session. For example, while Charlie and Lucy are in a call, Charlie decides to add Snoopy into the conversation or Snoopy rings him, being aware or unaware of the already established dialog. In either case, Charlie is in the role to act upon Snoopy joining in and thereby turning the session into a conference. As visualized in figure 1.1, parties are in individual, point-to-point contacts and naturally manage conference negotiations and policy operations between peers. Switching the communication context from a 2-party session to a 3-party conference raises two additional duties at Charlie's site. The parallel calls, one with Lucy and the other with Snoopy, need a logical join to form one conference. Also, media data have to be arranged to arrive at all three participants.

The simplest solution for this end-point hosted conference would work without additional signaling, when realized within the conferencing application. Based on user reactions to calls, Charlie's end system could sort the two sessions into a virtual conference, and at the same time start to mix media such that each correspondent receives all information from Charlie within one stream[1]. Lucy and Snoopy could thus participate without logical or technical awareness of the multi-party situation.

There are however drawbacks of this simplistic approach. Obviously all communication relies on the presence of Charlie, his disappearance will terminate both calls. In particular there will be no seamless mechanism to turn the conference into a call between Lucy and Snoopy. Scaling issues may arise from media mixing, which requires transcoding in the absence of a common coding scheme. Without explicit group management, no opportunity is given to negotiate on common codecs, nor are means provided to distribute mixing tasks or redirect media streams in many-party scenarios. Finally, privacy concerns are raised by a solution that allows for an undisclosed third party joining in a conversation – an explicit IETF policy holds off Internet protocols from "wiretapping" [19].

SIP resolves these issues by explicitly defining a conference focus, which is identified by a URI. This URI represents the conference and while it is uniquely created, additional SDP [15] media negotiations are foreseen [27]. The focus forms the central point of control *only* for the SIP conference management, which is free to define media distribution or mixing otherwise. The focus may be altered within an ongoing multi-party conversation, but this will lead to the creation of a new conference instance distinguished by a newly defined conference URI.

In detail, SIP operations for our example will proceed as follows (cf. [13]). After Charlie decided to invite Snoopy into the conversation with Lucy, he generates a conference URI and issues a SIP INVITE to Snoopy using this URI in his Contact field as follows:

```
INVITE sips:snoopy@dog.net SIP/2.0
Via: SIP/2.0/TLS alpha.brown.com:5061;branch=z9hG4bcHlkapff
Max-Forwards: 70
From: Charlie <sips:charlie@brown.com>;tag=4576932
To: Snoopy <sips:snoopy@dog.net>
Call-ID: 777777@alpha.brown.com
CSeq: 1024 INVITE
Contact: <sips:dog-matter@alpha.brown.com>;isfocus
Content-Type: application/sdp
...
```

Even though the contact URI addresses Charlie directly, it is not inherently bound to a conference situation. Charlie needs to add an 'isfocus' feature tag to make the multi-party situation transparent and to mark himself as the focus point. After the session has been successfully established, with SDP offer/answer negotiations aware of Lucy's media capabilities and the conference media distribution policy included [31], Charlie analogously re-invites Lucy into the conference (see section 1.2.2):

```
INVITE sips:lucy@psychic.org SIP/2.0
Via: SIP/2.0/TLS alpha.brown.com:5061;branch=z9hG4bKnashds
Max-Forwards: 70
```

[1] Media mixing is easily achieved for voice, but does likewise work for video with combined pictures or virtual environments with merged update sets.

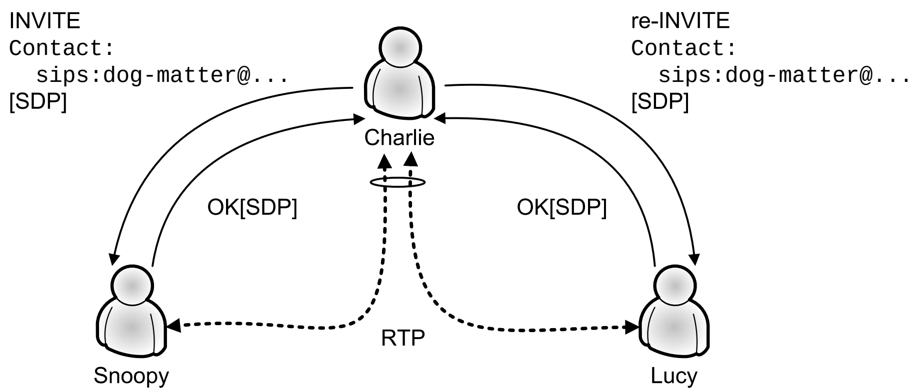


Figure 1.1: A 3-party dialog

```
From: Charlie <sips:charlie@brown.com>;tag=23431
To: Lucy <sips:lucy@psychic.org>;tag=1234567
Call-ID: 888888@alpha.brown.com
CSeq: 1024 INVITE
Contact: <sips:dog-matter@alpha.brown.com>;isfocus
Content-Type: application/sdp
...
```

Note that Lucy can explicitly react upon the group context and – if desired – decline the re-invitation.

If Snoopy wishes to contact Charlie instead, he may be aware or unaware of an already ongoing conference. In the latter case he will just call Charlie, who will reply with his conference contact information or re-invite Snoopy after their 2-party dialog has been established. In the first case, when Snoopy knows about the conference including the identification of one ongoing call, he can explicitly express his will to participate using the Join header field (see section 1.2.2).

This simple, end-point hosted conferencing scheme can be extended to an arbitrary number of participants, only limited by scalability issues. Two extensions beyond the base specification have been in use, the 'isfocus' tag and the Join header. Any conference-unaware user agent implementing only RFC 3261 cannot act as a focus or request to join an ongoing conference, but may nevertheless participate in a regular client role. The presence of the 'isfocus' tag in a Contact header field does not cause interoperability issues, since it will be simply ignored as an unknown header parameter.

Large-scale Conference

Large conferencing instances usually go along with some external occasion or scheduling. This may be a program announcement via a Web page, SAP [16] or printed guides like in IPTV offers or a streamed real-world conference, a meeting organization through personal communication or email like in appointed conference calls, or a well-known community information in contexts like gaming. These application scenarios all have in common not only large scaling requirements, but also the need for an explicit policy management. In cases where feedback is foreseen and distribution does not remain unidirectional, floor control mechanisms must guide party interactions. Media distribution will scale up to millions of users, if IP multicast is used, cf. section 1.2.3. For smaller numbers, powerful multipoint control units (MCUs) may satisfy the demands, or multicast implementations on the application layer will enable an optimized stream replication without infrastructure assistance.

Distribution-oriented large-scale conferences do not necessarily require SIP. Charlie, for example, can subscribe to an open media channel just by joining a multicast group. He may use SIP to inquire on multicast addresses, for authentication and authorisation and for enabling feedback control. Conference control thereby may be fully decoupled from media distribution and reside on a separate entity as displayed in figure 1.2.

Beyond minimal management, SIP supplies a number of optional features, which facilitate interesting interactive enrichments of a basic, reception-oriented application. Charlie could want Snoopy to participate in an ongoing conference. By subscribing to the SIP Event Package for conferencing states, cf. section 1.2.2, he could inquire on Snoopy and discover

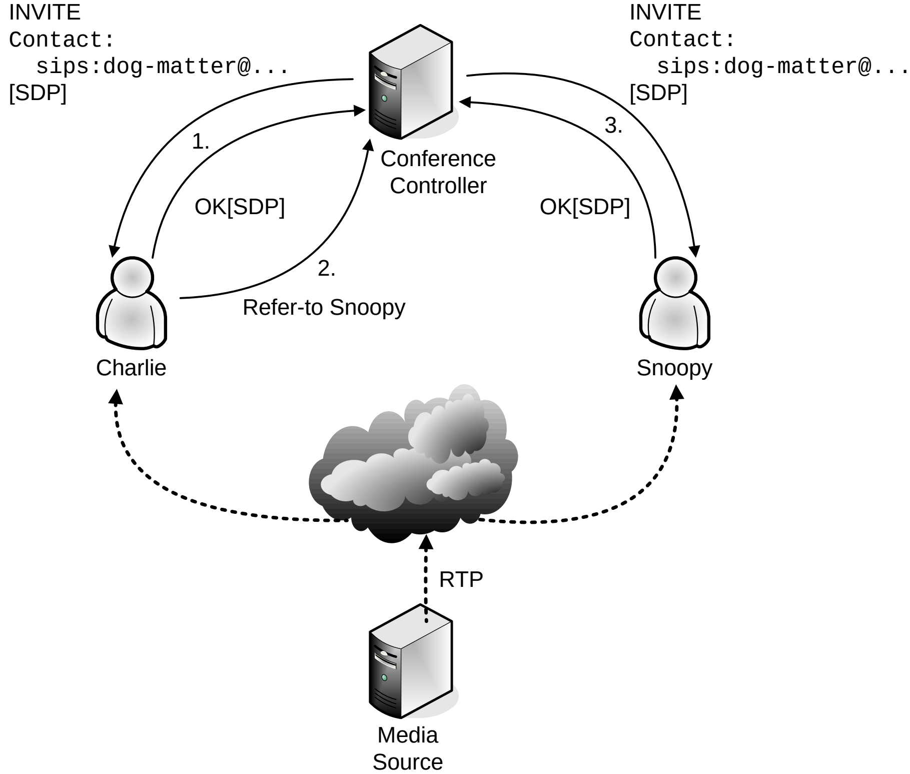


Figure 1.2: A multi-party distribution session

his absence. To invite Snoopy into the conference, either for raising his awareness or for seamlessly integrating his conference-unaware user agent client, he could use the REFER method as described in section 1.2.2. Charlie is also able to submit feedback after registering with the conference, potentially under a basic floor control mechanisms [5] smoothly embedded in SIP addressing.

1.1.2 The Application Domain for SIP-Managed Group Conferencing

Audio Conferences

The primary dedication of the SIP protocol lies in a standardized signaling for transparent Voice over IP services. SIP thereby simultaneously targets at campus solutions in the range of PBXs or the ITU H.323 [21] VoIP architecture, and global call handling based on Internet routing or, possibly in parts, on the public switched telephone network. For the sake of competitiveness, core voice conferencing functionality as offered by the digital telephone infrastructure or available through in-house conference control units must be part of the service spectrum of SIP. Traditional voice conference calls allow for easy media mixing, consist of limited group sizes, and SIP-initiated Internet telephony solutions can easily cope with underlying demands. Obviously, such characteristics likewise hold for simpler or supplementary group applications as chat or whiteboards. However, as a flexible signaling protocol suitable for decentralized relaying architectures, SIP may give rise to a multitude of additional services, unknown from traditional telephone networks.

Videoconferencing over IP

The idea of augmenting voice conferences by video has been around for several decades, but only the flexibility of the Internet generated a noticeable deployment. As compared to audio, video processing places significantly higher demands on end system and network transmission capabilities. The rapid evolution of networks and processors have paved the way for realistic group conferences conducted at standard personal computers, combining about a dozen visual streams of Half-QVGA (240 x 160 pixel @ 15-25 fps) resolution. Thus initial centralized implementations based on H.323 are now superseded by peer-centric personal systems with SIP. Lightweight videoconferencing software today smoothly integrates desktop video of high quality with thin clients on mobile phones [11] like the daViKo system[2].

In concordance with communication capabilities video coding techniques have evolved, as well. The latest standard for video coding H.264/AVC [20], although designed as a generic standard, is predestined for applications like mobile video communications. Besides enhanced compression efficiency, it delivers also network friendly video representation for interactive (video telephony) and non-interactive applications like broadcast, streaming, storage, and video on demand. H.264/AVC provides gains in compression efficiency of up to 50 % over a wide range of bit rates and video resolutions compared to previous standards. While H.264/AVC decoding software has been successfully deployed on handhelds, high computational complexity still challenges current mobile devices, when implemented as pure software encoders; there are however fast hardware implementations available. Next generation codecs

[2] see http://www.daviko.com

like Scalable Video Coding (SVC) are already approved [38]. The main new feature, scalability, addresses schemes for delivery of video to diverse clients over heterogeneous networks, particularly in scenarios where the downstream conditions are not known in advance. The basic idea is that *one* encoded stream can serve networks with varying bandwidths or clients with different display resolutions or systems with different storage resources. This not only enables media mixers or conference bridges to simultaneously serve streams of different resolution without transcoding, but is an obvious advantage in heterogeneous networks prevalent in mobile applications, as well.

Gaming

Multiplayer games fall into the realm of SIP conferencing services for two reasons. At first, interactive multi-party gaming is built on session oriented group communication, which relies on standard signaling relations and largely benefits from additional presence and conference information as provided by SIP. At second, an increasing number of massive multiplayer games offers sidebar voice and chat conferencing services to enrich communication among players.

Sidebar conferences in games can just be treated like any (cascaded) SIP session. A context-aware integration of voice conferencing with multiplayer networked gaming has been presented in [39] along with a SIP-based reference architecture. The authors foresee a tight coupling of game and conference server by SIP means, which lead to an automated conference reconfiguration based on third party call control (cf. section 1.2.1), whenever a switch in the gaming context occurs. For example, if a player changes a room in some game arena, its audio conference will then include the players in the new room and not the old ones. This solution fully complies with the SIP conferencing approach, for which reason the architectural realizations extend from a strictly centralized to a fully distributed model.

Even though today's popular games do not employ SIP standard messaging, this large software market remains open as an appropriate candidate for migrating to SIP, which in turn may facilitate more elaborate, less expensive, distributed architectures for game deployment.

Presence and Instant Messaging

Session dialogs and conferencing are person-oriented services which abstract from specific communication channels and from devices. The way and the instance of favorable information exchange not only depends on content and external demands, but on the individual contexts of each involved party. Compared to the PSTN it is one of the predominant features of the Internet to rigorously preserve this abstraction of user presence over technical entities[3]. Congruously – in taking up earlier, proprietary solutions – SIP fills this paradigm by providing presence information about parties. Dependent on present state, participants may choose to communicate by short messages, email, voice and video, interact within dedicated collaborative environments or not at all.

Presence information not only resolve the physical location of a person or an application, but may indicate the ability and willingness of a user to communicate. Rich presence infor-

[3]Even though this is merely a reformulation of the layered design principle, it is worth noting that at this occasion we benefit from a long-term resistance to layer violations

mation convey the personal and contextual circumstances of a contact, and a caller a priori may judge, whether to pursue a contact, while the callee is busy at the airport and in bad mood. This may be particularly useful in larger or extensive conferences as an unpretentious way of feedback from silent parties.

Presence indicators follow the SIP event state model which is outlined in section 1.2.2. They enrich a community model of "buddies", whose presence is concurrently cultivated and silently enriched by instant message exchange. SIP enables instant messaging in two ways. Short textual news may be submitted in pager mode via the MESSAGE method [7] aside from session establishment. For more extensive IM exchange, the session mode does negotiate media exchange via SDP, where transport in IM sessions is facilitated by the Message Session Relay Protocol (MSRP) [6]. Messages can be exchanged individually or in conferences, realized as client-server or peer-to-peer communication. In particular, IM may be integrated in a voice conferencing infrastructure as a complementary communication format or an orthogonal channel. The latter allows a phoning party to exchange messages with a third person within the same user agent.

Instant messaging is now part of a large number of tremendously popular personal applications. The technical agreement by AOL, IBM, and Microsoft on presence and IM interoperability, SIP for IM and Presence Leveraging Extensions (SIMPLE), has tight this success to SIP.

SIP & the IMS

An alternate approach to SIP conferencing is taken with the ITU-T next generation network (NGN) architecture. Using the Internet technologies, 3G mobile operators and wire phone companies are in the process of building next generation telephone services based on the IP Multimedia Subsystem (IMS) or its wireline emulation TISPAN. SIP has been selected as session signaling protocol as formerly RTP has been chosen in H.323 for media transport. However, this provider-centric, network-controlled and application-aware network architecture takes a completely different perspective.

The IMS accounts for VoIP/VCoIP, presence, messaging, gaming, etc. services which as physical server instances reside on the application server layer. A likewise centralized server instance at this layer will enable conference management. For integration purposes and ease in billing, all these session oriented services reside on a common session layer, whose states are aggregated from SIP message exchanges with application servers. As of 3GPP release 6, Multimedia Broadcast and Multicast Services (MBMS) are foreseen. Anchored at a regional Gateway GPRS Support Node (GGSN), group communication may thus be used to enhance conference data distribution. This strongly debated, complex architecture provides SIP initiated group conferencing services *within* a provider domain. The future will uncover the validity of the IMS business model and reveal, whether people are willing to pay for dedicated conferencing services offered by operators or just use applications available from independent sources.

1.2 Group Conferencing Concepts and Technologies

Conferencing enriches the semantic of basic group communication by providing presence or session logic to the application. Conversely, it widens the meaning of dialog-oriented conversation by augmenting signaling relationships with the multi-party paradigm. Examining both aspects separately, concepts and technologies are around for serving all needs. Group communication is enabled by multicast, session management by SIP, whose base standard [32] already defines a minimal interplay with IP-layer multicast. However, to arrive at conferencing solutions of full functionality and comfort, further efforts of integration are needed. This section provides an overview of the main service models and their technological primitives on the SIP and distribution layer within the framework of group conference management.

1.2.1 Common Service Models for Conferencing

Group conferencing must be considered a generic term for a wide variety of meanings in different contexts. Generally it should be understood as an instance or realization of a multi-party conversation and may consist of one or several SIP sessions, each of which combining the SIP dialogs as required for communication between participants. Such multi-party conversation space may include human and non-human users, e.g., tone generating robots or recording monitors, which may be active and noticeable or hidden. Conferencing may be essentially realized by the three following service models of multi-party communication [27, 25].

Loose Coupling

Within conferences following a loosely coupled model, no signaling relationship is maintained between conference participants. This considerably light-weight approach is exceptionally scalable, easy to implement, but does not foresee any SDP offer/answer dialog. Also, no standard way of authentication, policy execution or conference coordination is provided. There is no single instance of control nor a conference server or focus. Instead participation may be gradually learned by RTCP [37] control streams. Its realization requires an any source multicast distribution layer for media streams. A conference can be entered by simply joining a multicast group, while addresses and media information may be pre-shared by non-SIP means. Alternatively, a SIP party may be invited by any single conference member or issue its INVITE to an ASM group established for SIP signaling, as described in section 1.2.3.

Tight Coupling

Tightly coupled conferences follow a simple, matured communication model and as of today provide the most elaborate services to the users. They rely on a central managing instance, granting a signaling relationship to all participants through a globally routable conference URI (GRUU) [29]. Such conference controller acts as a focus, which governs the conference by applying policy rules and can provide a variety of convenience functions. It may additionally perform media mixing and redistribution, in which case its functionality is equivalent to a conference bridge or multipoint control unit (MCU) known from PSTN and H.323 [21].

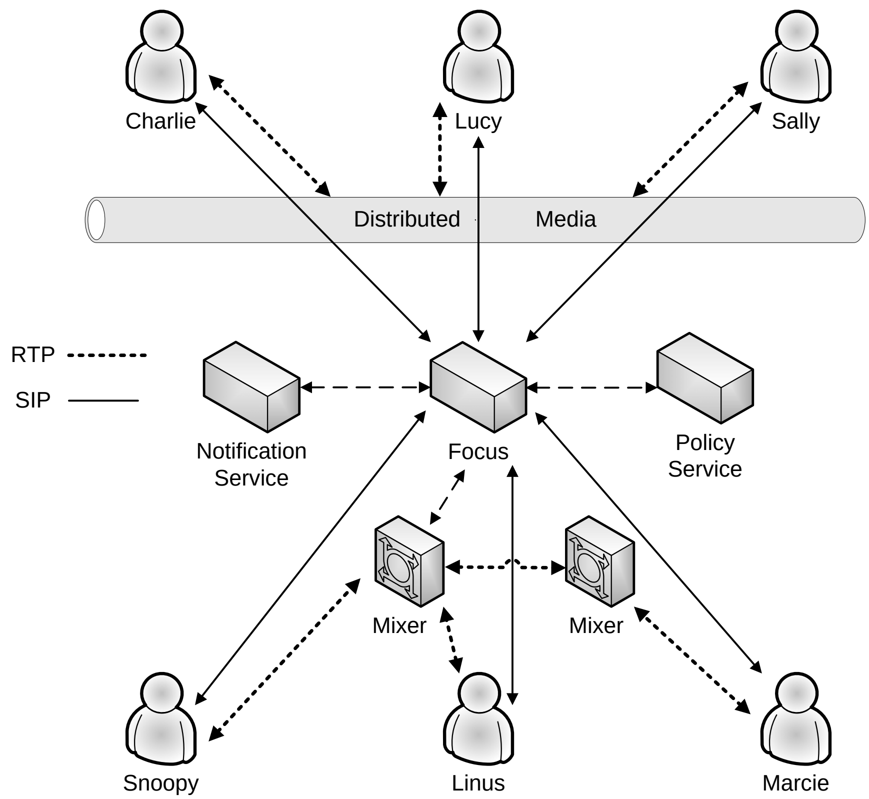


Figure 1.3: General Architecture of a Tightly Coupled Conference

To avoid scalability limits inherent to the ITU architecture, media distribution in a tightly coupled conference is not bound to centralized mixing. The general architecture as defined in [27] is drawn in figure 1.3. The focus, assisted by policy and notification service functions, forms the center of a star topology for SIP session management. Notification and policy service may be part of the conference server or distributed and coupled by non-SIP means. RTP [37] media streams flow in a serverless way, e.g., via multicast, in the upper part of the graph, while a cascade of mixers in the lower part distributes media as instructed by the focus.

The central conference *focus* is responsible for maintaining dialogs with all participants and to ensure that everyone is served with all the media for the conference. Therefore, the conference functions minimally offered are authenticated and authorized session establishments including media negotiations. In the presence of media mixers, it uses its session states and the media policy to appropriately configure one or several mixers. For this purpose the focus interacts with the mixer either through a local interface or, in a distributed scenario, via INVITE / re-INVITE sequences allowing for third party call control (3pcc)[30]. It thereby is fully enabled to (re-)organize media streams at its convenience, e.g., to optimize media distribution according to network and device capabilities. It may follow an efficient strategy of balancing (cascaded) mixers. Additional convenience functions offered by the conference focus most likely are a third party invite (dial-out) for conference-unaware user agents, removal of third parties from the conference and notification services for conference event states. Extended services may be equally included such as floor control, conference announcements and recordings, NAT traversal assistance, user directory, integration and billing functions, etc.

Full Distribution

The fully distributed multi-party model is built upon peer-to-peer signaling relationships between conference members in the sense that

- participants have means to contact each other on an individual basis and negotiate SDP; likewise they may inquire on their presence;
- there is no central point of control nor individual instance of a focus;
- some distributed algorithm is cooperatively performed to achieve appropriate conference coordination and media control.

Distributed peer-to-peer management allows for infrastructureless, instantaneous conference establishment and operation and, if properly designed, may encounter faster and error resilient dialog handling. In contrast to tight coupling, definitions for this conference model are not widely matured, but remain an active area of research and future protocol development.

Most simplistically, such model can be realized in a pairwise unicast or full mesh relationship between n parties. However, due to its n^2 signaling complexity, this approach remains suitable only for small conferences. Improvements are subject to specialized mesh optimizations. Scalability up to very large groups can be achieved by the use of multicast packet distribution, which – in contrast to the loosely coupled model – can follow a source specific model on the IP layer for both, session management and media distribution; see section 1.2.3 for a detailed discussion. User agent clients may also make use of a general multicast service resident on the application layer. Subject to its algorithmic realization, distribution efficiency of application layer multicast commonly increases with a growing number of participants and promotes scalability.

Complementary approaches to distributed conferencing are opened up by structured peer-to-peer technologies. Distributed hash tables (DHTs) offer an efficient, fully distributed persistence layer and can facilitate a decentralized realization of some or all functions of a conference focus. Such a distributed conference coordination scheme raises global routability and NAT traversal issues as major problems to solve. Fundamental work for this purpose is currently chartered in the IETF p2psip working group, complemented by research activities addressed within the IRTF p2prg research group. A detailed discussion on peer-to-peer SIP technologies is given in chapter 8. Beyond signaling, the forwarding capabilities of DHTs give rise to enhanced application layer multicast solutions, which will be reconsidered in more detail in section 1.2.3.

1.2.2 SIP Extensions Used in Conferencing

SIP syntax and semantic require extensions, whenever the support of feature-rich conferencing scenarios is desired. Following SIP design principles, invoker oriented primitives are the common basis to enable a dial-in and dial-out call control for user agents. This simultaneously holds for ad-hoc and scheduled conferences. However, the main design guidelines of SIP extensions limit changes to a minimum and should guarantee seamless backward compatibility with conference-unaware user agents [22]. The subsequently outlined extensions reflect these principles.

Join Header

The Join header [24] expresses the request of the caller to participate in an existing dialog. It is solely used in invitations and designed as the generic operation for a third party to initiate a conference. The user agent receiving a Join normally needs to create a new conference URI, which is then handed to the joining party via a re-INVITE. Conference creation is needless, if the dialog to be joined is already part of a conference. The existing dialog

```
From: Charlie <sips:charlie@brown.com>;tag=7654321
To: Lucy <sips:lucy@psychic.org>;tag=1234567
Call-ID: 333333@alpha.brown.com
```

is identified in the Join header by its Call-ID, the to-tag and the from-tag:

```
INVITE sips:charlie@brown.com SIP/2.0
From: Snoopy <sips:snoopy@dog.net>;tag=95148
To: Charlie <sips:charlie@brown.com>
...
Call-ID: 2772@beta.dog.net
...
Join: 333333@alpha.brown.com; to-tag=1234567; from-tag=7654321
...
```

Note that even though the joining party need not know about an ongoing conference nor a conference URI, it must acquire call information of the existing remote session. These IDs may be learned from non-SIP means or via a SIP Event package [34].

re-INIVITE

The SIP re-INVITE operation [32] is not an independent method, but an iterated INVITE within an ongoing session, i.e., with unaltered Call-ID. It allows for a change of SIP session characteristics and always triggers SDP media negotiations anew. Thus in conferencing situations it can be used to adapt control- and media-session parameters at the same time. Commonly, conference URIs and focus points or multicast groups used for signaling are distributed via re-invitation. Similarly, modified media types and distribution parameters like the definition of new mixers or network and multicast specific parameters are negotiated along the lines. Any failure of a re-INVITE will lead to session continuation with the previously established parameter set. Existing conferences therefore cannot be disturbed or destroyed by inappropriate re-INVITE requests.

REFER Method

The SIP REFER method [40] allows a user agent to request another user agent for accessing a resource it refers to. The URI of the referred resource is given within the Refer-To argument:

```
REFER sips:lucy@psychic.org SIP/2.0
Via: SIP/2.0/TLS alpha.brown.com:5061;branch=z9hG4bKnashds
Max-Forwards: 70
From: Charlie <sips:charlie@brown.com>;tag=23431
To: Lucy <sips:lucy@psychic.org>;tag=1234567
Call-ID: 787878@alpha.brown.com
CSeq: 9380 REFER
Refer-To: <sips:hypnotic-talks@circles.com>
Content-Length: 0
```

A REFER request can be send either inside or outside an existing dialog and also provides mechanisms for to notify the originator of the outcome of his referenced request. Initially created for call transfer, it is used in conferencing to add third parties. Any client may send a REFER request to a partner, asking him to send an INVITE to an established conference URI. Equally, the initiator may send a REFER to the conference focus asking it to invite the partner. The latter way provides the benefit of allowing a client to add a conference-unaware user agent that does not support the REFER method. Analogously, a client may ask the conference focus via REFER to terminate conference membership for a third party.

REFER requests may be cascaded in the following way. A participant, who wishes that a focus or another participant refers a third party into the conference by sending a REFER method, may express this by adding an escaped Refer-to header field within its Refer-to argument:

```
REFER sips:hypnotic-talks@circles.com SIP/2.0
[...]
Refer-To: <sips:lucy@psychic.org;method=REFER?Refer-to=sips:hypnotic-talks%40circles.com>
```

Furthermore the Refer-to URI argument is not limited to SIP, but may point to any valid Internet resource. In referring to non-SIP resource URIs, user agent clients are entitled to exchange input to collateral applications, e.g., as part of collaborative tasks or environments.

INVITE-Contained URI Lists

All mechanisms for ad-hoc conference management described so far define incremental operations for adding or joining single users into a multi-party session. This procedure may be time-consuming for larger conferences and delay conference establishment in an alienating fashion. Current work proposes an extension of SIP operations to include URI lists [4]

conference-info	Identifies this event set by a conference URI and a version.
conference-description	Describes the content of the conference by textual metadata like subject, keywords, additional (service) URIs and available media.
host-info	Contains information about the host of the conference.
conference-state	Expresses the conference state (active, locked) and current user count.
users	This large container block enumerates individual users. A user record consists of a display name, user URI, role and a comprehensive characterization of the employed devices.
sidebars-by-ref	A reference pointer to a sidebar conference, given by its conference URI.
sidebars-by-val	Full representation of a sidebar conference, optionally including all conference state events.

Table 1.1: The conference state event groups

treated like email, where individual delivery can be handled in parallel according to address lists. These lists are encoded in XML, denoted by a 'recipient-list-invite' SIP option-tag, and appended to the regular conference invitation messages. A user agent client in a typical scenario sends an INVITE including a recipient list to a conference server, which then will simultaneously invite all the list members. Note that the conference server does not process the initial INVITE as a nested transaction, but will acknowledge positively, whenever the conference was created. To inquire on actual members, the client needs to actively search user lists provided by the conference state event package described in the following section.

Conferencing Event States

An established conference incorporates a potentially large number of member-dependent states, exceeding those of a regular dialog. To support notification for tightly coupled conferences, an event package for conference states [33] has been defined. Its focus lies in providing membership information, but notifications about additional conference components are foreseen, as well. Typically these data are provided by the conference focus which may learn states while performing its regular conference management tasks. Note that this model allows for cascading conferences, expressed by the sidebar conference elements. An overview of the conference state information is given in table 1.1.

Following the general SIP event notification model, a user agent client will receive conference event state changes from the state agent via the NOTIFY Method after having issued a SUBSCRIBE with reference to the package name 'conference'. The corresponding call flow is displayed in figure 1.4. Event notification is performed incrementally on change except for the initial NOTIFY response to subscription. Due to the expected amount of data in large conferences, a user agent client can receive partial notification about those compound, volatile elements assembled in container groups.

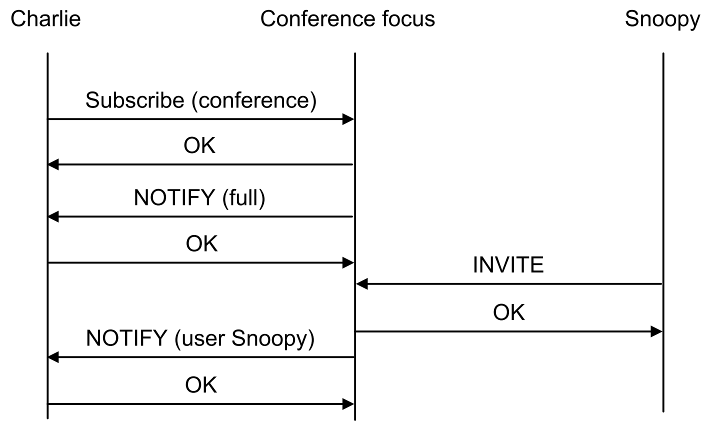


Figure 1.4: Call flow for SIP conference event state subscription and notification

1.2.3 SIP with Multicast

Conferencing requires some basic group communication mechanism for distributing media data to all parties and for mutual signaling, if applicable. The Internet most prominently offers IP layer multicast for this purpose, which is around for about 20 years in the flavor of any source multicast (ASM). The advantages of multicast over unicast-based multi-party communication must be seen in its scalability and lightweight interface at the client side. Deployment of IP multicast has been hesitant over the past decade, though, but tends to emerge in recent times with the remarkable spread of IPTV offers. In the meanwhile, the infrastructure-friendly source specific multicast (SSM) model has been developed, as well as infrastructure-agnostic multicast on the application layer with promising performance potentials. Keeping deployment complexity in mind, it is desirable for any multicast conferencing solution to restrict group communication for signaling and media data to *only one* of the multicast models.

Due to its bandwidth requirements, *transmission of media data* takes the largest advantage of multicast. RTP streams transparently conform to group communication. While SDP session descriptions convey sufficient information for a user agent client to join a multicast group by providing multicast destination addresses within connection type fields on a per session and per media basis [15], SDP offer/answer negotiations [31] only partially fulfill multicast requirements. Supporting only a uniform view of the multicast session among all participants, RFC 3264 requires an SDP answer to match a multicast offer in address, port and directionality. Thus the roles of multicast sender and receiver remain in disguise and neither multicast send/receive capabilities nor SSM specifics may be arranged. Current work in the IETF sipping working group addresses this issue.

Multicast communication may be desired for *SIP signaling*, whenever an elsewhere determined group of receivers are to be contacted or in distributed scenarios with severe scalability constraints. The basic interaction with IP multicast in SIP is defined by the `maddr` parameter in the Via header field. All SIP users of the community `dog.net` listening to the multicast address `239.12.11.10` could thus be contacted using the URI

```
sip:*@dog.net;maddr=239.12.11.10;ttl=3
```

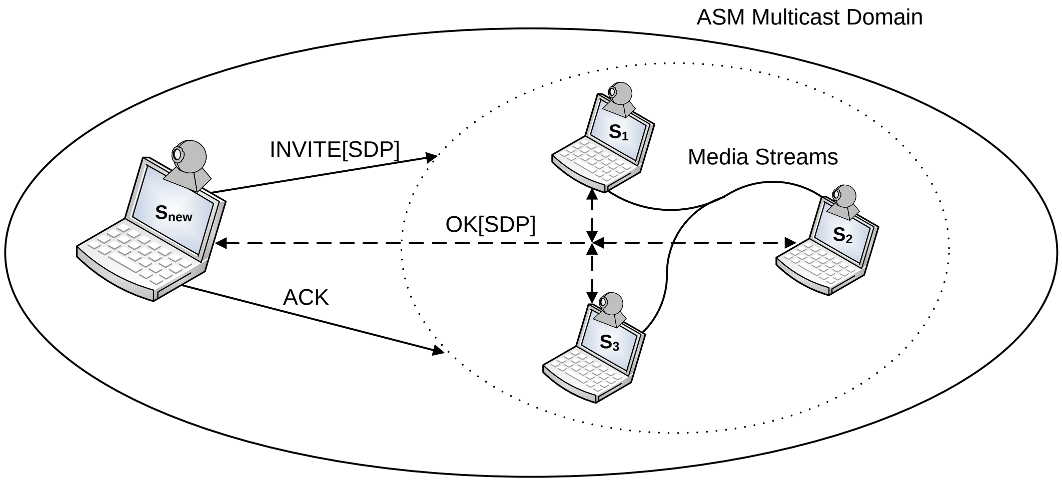


Figure 1.5: SIP call initiation based on ASM – a callee negotiates with a previously defined multicast group

Syntax specification of RFC 3261 suffers from the nit of allowing only IPv4 address arguments with the `ttl` field serving as scoping limiter, while the semantic is bound to ASM use.

Any Source Multicast

Deering's early host group model [12], which enables single packet submission to an unrestricted group of delocalized receivers, is the most general form of group communication. It is also known as any source multicast, since any sender is entitled to issue data to a group of potentially unknown receivers. Receivers simply join to a group address, more specifically a $(*, G)$ channel, and the multicast routing will provide a transparent delivery service. ASM is fundamental to rendezvous processes and service discovery — SIP takes advantage for registrar access, accordingly. A user agent client may send register messages to the well-known "ALL SIP Servers" URI `sip:sip.mcast.net`, mapped to IPv4 address `224.0.1.75`, without being aware of any individual registrar.

For multicast conference management SIP defines only a limited "discovery-like" service, delivering a request to a group of homogeneous servers. A client wishing to initiate or join into a multi-party conference sends its INVITE request to a multicast group by employing the `maddr` attribute in the SIP VIA header. Group members subsequently indicate their presence by responding to the same group (cf. figure 1.5). The transactional nature of SIP dialogs is preserved in the sense that the inviting party interprets the first arriving OK as the regular completion, while further messages are treated as irrelevant iterates. Note that multicast responses require the requestor to subscribe to the same group, and prevent routing of multicast requests via SIP proxy servers. Thus by pure SIP means, a caller cannot issue an INVITE to a remotely located user group, e.g., route to `*@dog.net`, without having established a global multicast connectivity.[4] Suitable for large, loosely coupled and mutually unknown parties, this simple scheme only operates through the use of any source multicast and does not allow for dynamic SDP negotiations.

[4]Such a scenario could only be realized by tunneling the multicast SIP request into the destination domain.

ASM media distribution is easily achieved for homogeneous conference members with uniformly matching session descriptions. Such media parameter settings may be preappointed out of band or learned from a SAP online announcement. However, SAP is rarely used. Once established, the ASM distribution tree serves as a transparent connectivity layer, delivering all media streams to all parties.

Source Specific Multicast

Source specific multicast (SSM) [1, 17], recently released as an initial standard, is considered a promising improvement of group distribution techniques. In contrast to ASM, optimal (S, G) multicast source-rooted trees are constructed immediately from source specific, i.e., (S, G) subscriptions at the client side, without utilizing network flooding or rendezvous points. ASM and SSM can be distinguished from a partitioning of the multicast address space.[5] It is widely believed that simpler and more selective mechanisms for group distribution in SSM will globally disseminate to many users of multicast infrastructure and services. While any source multicast accounts for easy common addressing, SSM presupposes source specific subscriptions. Hence, its use requires a dedicated distribution of source addresses for newly joining session members, which otherwise remain unnoticeable in previously established SSM groups. SIP session initiation in conferencing scenarios could facilitate this requirement [36].

The media parameters in tightly coupled or fully distributed conferences are mutually known to all peers. Despite the limitations of the SDP offer/answer model in the multicast context discussed above, each party will be aware of all source addresses of the correspondents and thus be enabled to issue source specific joins. Without protocol extensions, user agent clients can thus participate in an SSM based media session by simply performing IGMP/MLD [18] (S, G) joins for all peers. A simpler model well suited for medium-size conferences has been introduced in the conferencing framework [27]: Each participant sends its media stream to a central media point or dedicated user agent using unicast. This central point will then redistribute the media using a source specific multicast address. Whenever a new party joins the conference, the focus will perform the necessary third party call control to assure media reception. This scenario, which may lead to triangular traffic detours, is illustrated in figure 1.6.

Up until now, SIP signaling based on SSM has not been defined. Simple possible extensions for SIP over SSM will be discussed in section 1.4.

Application Layer Multicast

Exploiting the novel approach of structured peer–to–peer routing, a collection of group communication services has been developed, with the aim of seamless depoyability as application layer or overlay multicast. Among the most popular approaches are multicast on CAN [26], Bayeux [43], as derived from Tapestry, and Scribe [9] or SplitStream [8], which inherit their distributed indexing from Pastry. Approaches to multicast distribution in the overlay essentially branch in two algorithmic directions. In the first case, distributed hash tables (DHTs) are used to generate a structured sub-overlay of group members, which is then flooded with multicast packets. This mechanism underlies multicast on CAN. In the other, a distribution

[5] In IPv4 the 232/8 range and the prefix FF3x::/32 in IPv6 are designated as source-specific multicast destination addresses.

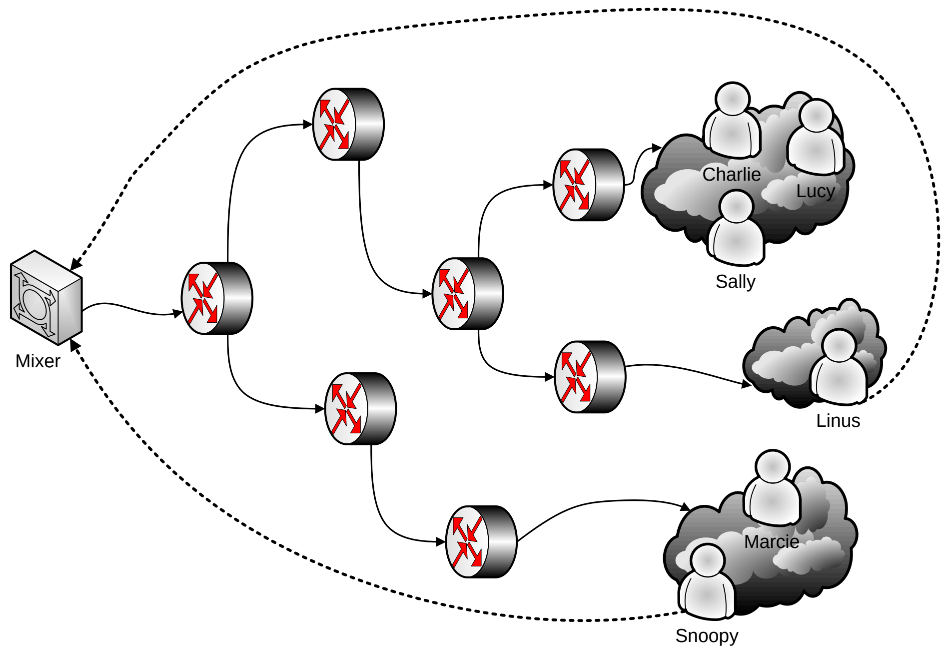


Figure 1.6: Media distribution in a centralized source specific multicast model.

tree is erected within the full overlay, to be used as a shared or source specific tree. The latter schemes are used in Scribe and SplitStream, where a rendezvous node is chosen from group key ownership, or in Bayeux.

The performance of DHT–based multicast has been thoroughly studied in [10] with the comparative focus on tree–based versus flooding approaches built onto CAN and Pastry. The separate construction of mini-overlays per group as needed for a selective flooding incurred significant overhead. In addition, flooding was found to be outperformed by forwarding along trees, where a shared group tree combined with proximity-aware routing as in SCRIBE could minimize the overlay delay penalty down to a factor of *two*. For the sake of completeness we mention that application layer multicast concepts concurrently exist for unstructured peer-to-peer approaches. They operate at lower algorithmic complexity, but show significantly higher efforts in coordinative signaling and thereby admit performance measures far below native multicast.

From a conceptual point of view, a user agent client can transparently take advantage of its multicast access on the application layer by submitting media data or SIP requests to the DHT. Overlay routing will just transport native packets in an application layer tunnel to the corresponding group of receivers. However, standardization work on peer-to-peer SIP has just begun [3, 2] and currently concentrates on distributed user location and NAT traversal. A significant amount of prestandard work is required, until an efficient, globally scalable and robust multicast distribution layer becomes available for wider use. The IRTF p2p and SAM research groups are dedicated to corresponding work.

1.2.4 Mobile Group Members

Mobile end systems today escort people from puberty to sageness. Standard systems such as mobile phones or PDAs, but also specialized gadgets, i.e., gaming stations or dedicated professional equipment, must be increasingly considered as people's primary access devices to information, communication and entertainment. At the same time most of public and private spaces have turned into areas of ubiquitous network operations. Wireless Internet connectivity is offered by a 3GPP or IEEE 802.11 WLAN infrastructure, but emerging 802.16 environments and manet mesh networks have started to complement the established access networks.

Voice calls dominate the mobile market, gradually complemented by selective video services. Mobile telephone operators are on the spot of migrating to entirely IP based fourth-generation networks in accordance with 3GPP release 8 standard track. SIP has been selected to play a dominant role in both, provider-centric architectures based on the IP Multimedia Subsystem (IMS) and provider-independent end-to-end session signaling, the latter being particularly obstructed by NAT and port barriers in the mobile regime. As SIP/IP 2-way dialogs are expected to dominate the mobile world soon, demands for conferencing are likewise foreseeable. Mobile group conferences meet multipoint transmission capabilities in virtually all wireless technologies, e.g., in 802.11 and 802.16, in DVB-H, as well as in 3GPP which defines the Multimedia Broadcast and Multicast Services (MBMS) in release 6. Issues of providing seamless communication services under mobility-related handovers remain though on Internet protocol layers. They will require an explicit handover management, whenever provider or IP subnet addresses change between attachments. Schemes for achieving a seamless mobility management in conferencing vary depending on the communication model, i.e., the use of unicast or multicast.

Unicast-based Mobility

A tightly coupled conference scenario is likely to operate in unicast mode. This does not only hold for SIP signaling relations, but may also apply for media distribution in a central mixing or conference bridge scheme. On handover, any mobile member changing within a given IP subnet may just continue operation on upper layers, any disturbance being bound to disconnection times on the network access layer. In cases where movement occurs between providers or internal subnets, e.g., triggered by a vertical handover between different access technologies, the mobile conference member changes its IP address and needs to procure conference session persistence through additional means. Session continuation can be accomplished by SIP mobility management as described in chapters 7 and 9. Alternatively, transparent handover management may be operated on the transport layer, provided (mobile) SCTP is in use. The most general approach to network layer mobility resides in IPv4/v6 protocol extensions, which have been in focus of IETF work over recent years (cf. [23] for an excellent discussion).

In conferencing situations special care is needed, whenever the mobile node performs internal relaying, as its disconnection will cause data delay or damage for downstream members. Such threat of service degradation is obvious for a peer-managed focus or mixer in tightly coupled conferences, but may result from routing and forwarding functions in distributed settings or application layer multicast, as well.

Mobile Multicast

Multicast mobility management must be assured for mobile user agents, whenever native multicast routing serves for media or signaling distribution. IP multicasting is of particular importance to mobile environments, where users commonly share frequency bands of limited capacities. In general, multicast routing dynamically adapts to session topologies, which may then change under mobility. However, depending on the topology and the protocol in use, routing convergence may be far too slow to support seamless handovers.

In multicast routing the roles of senders and receivers need to be distinguished. Any listener subscribed to a group while in motion, requires delivery branches to pursue its new location; any mobile source requests the entire delivery tree to adapt to its changing positions. Operations should facilitate seamless data flows compliant to real-time requirements and at the same time ensure routing convergence without compromising network functionality, cf. [35] for a detailed discussion. In a conference, multicast routing will be always exposed to listener mobility, while source movement in some schemes may be hidden to the network layer, e.g., by a central static mixer.

Like in the unicast case, SIP may assist multicast mobility on the application layer, even though the undertaking is more complex. Subsequent to handover, a listening user agent client can initiate a unicast data forwarding by third party call control until multicast routing has converged. A moving source may either rely on static entities for packet redistribution or take advantage of application layer tunneling between conference members. Note that a change of its source address will trigger a user agent to issue SIP/SDP updates within immediate signaling relations. Thereafter address information needed for SSM specific rejoins are in place.

In concordance with mobile IP, a natural and efficient way to manage multicast mobility will reside on the network layer. Three approaches to mobile multicast are commonly around:

Bi-directional tunneling guides the mobile node to tunnel all multicast data via its home agent and thereby hide its mobility.

Remote subscription forces the mobile node to re-initiate multicast distribution subsequent to handover by submitting an IGMP join or MLD listener report within the subnet it newly attached to.

Agent-based solutions attempt to balance between the previous two mechanisms. Static agents typically act as local tunneling proxies, allowing for some inter-agent handover, while the mobile node moves away.

Work on standardizing seamless solutions has just begun, the requirements being discussed in MobOpts research group. For further details and a brief survey on current solutions we refer to [35].

1.3 Infrastructure–Assisted Conferences

Conventional deployments of SIP architectures are accomplished with the help of infrastructural entities, typically a SIP proxy server and registrar, a gateway and additional convenience functions. These components typically reside on network nodes that are well connected and in unrestricted, permanent availability. A manifest strategy to advance service offers lies in augmenting these nodes with lightweight side primitives such as NAT traversal assistance or client mobility management. Likewise a conference control function could reside on a standard SIP server instance. However, media mixing and relaying for large conferences cannot be considered lightweight, but out of range. Multicast distribution can assist in serverless delivery of media streams as discussed in section 1.2.3, and open the floodgates to a scalable conferencing at no additonal cost.

Unfortunately IP multicast is not uniformly available, even though all major router vendors and operating systems offer a wide variety of implementations to support it. While many (walled) domains or enterprise networks operate multicast, group service rollout has been largely limited in public inter-domain scenarios. Application layer multicast based on structured peer-to-peer systems offer group communication in an infrastructure-agnostic fashion. Distributed hash tables (DHT) are reasonably efficient and scale over a wide range of group sizes. However, they do not allow for layer 2 interactions, thus do not facilitate unrestricted scaling in shared end system domains, and experience severe performance degradation when terminal mobility is introduced [14]. These drawbacks may be mitigated by hybrid approaches, where overlay multicast routing only takes place among selected nodes, which are particularly stable and form a virtual infrastructure. The servers of a SIP infrastructure are apt candidates for this, as will be outlined in the following sample architecture.

1.3.1 A Hybrid Architecture for Transparent Group Communication

This section introduces a hybrid architecture for SIP proxy assisted group conferencing, designed to enable global multicast peering at the ISP or enterprise level, and at the same time sustain end system transparency. The basic concept preserves multicast routing and lower layer packet transmission within domains, while bridging the inter-domain gap with the help of a structured overlay network to overcome deployment problems. Its focus originates from a customer network or an ISP domain, where multicast services are locally deployed. Multicast service exchange is then implemented like unicast peering by a gateway function, resident on a SIP proxy. The primary call routing function of a SIP server is thereby augmented with a multicast overlay for conferencing and media distribution. It interconnects the local multicast routing with the distributed peering on the structured overlay.

The new function, Inter-domain Multicast Gateway (IMG), acts as a gateway between the overlay, it is a member of, and the multicast routing at the intra-domain underlay that it resides in, cf. figure 1.7. Such gateway will participate in multicast traffic originating from its residential network, which it will forward into the overlay according to the distributed multicast receiver domains of this group. It will also advertise group membership and receive data according to any subscription from its domain. On the overlay, the IMGs will jointly operate a distributed hash table, hosting an application layer multicast. Depending on specific requirements, any well-known ALM may be executed. The bidirectional shared tree approach introduced in [42] gives rise to fully transparent, mobility-agnostic and proximity-aware multicast and broadcast overlay services.

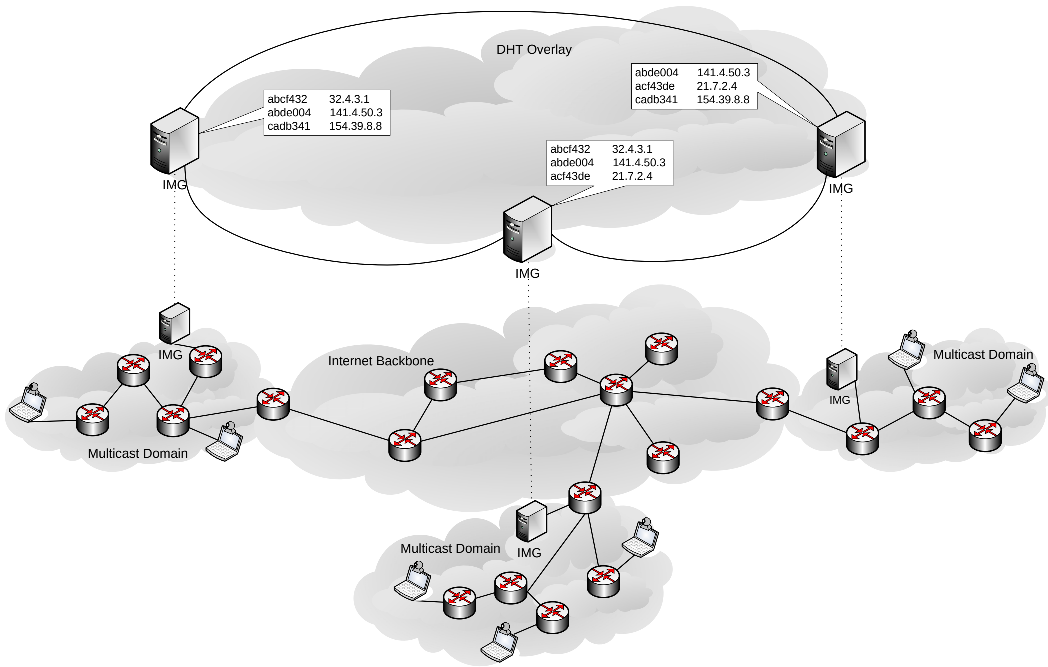


Figure 1.7: A hybrid group communication architecture with SIP inter-domain multicast gateways.

Like a SIP proxy, the IMG function may be positioned anywhere within the multicast domain, and provides a protocol interface to the locally deployed multicast routing. Activation of inter-domain multicast gateway services requires only a small amount of selected information for bootstrapping, i.e., an arbitrary contact member of the structured overlay, authentication and authorization credentials, if applicable. Typically, such initialization is performed at SIP domain peering. The IMG further on remains under the administrative control of the local network operator, who may restrict admission, scoping and QoS characteristics of the group traffic flowing in and out of the intra-domain. Aside from general multicast peering policies, a service provider is thus enabled to implement firewall-type of packet filters at, or co-located with, these multicast gateways. It thereby can easily shape the inter-domain multicast layer according to conferencing application or community specific particularities — without control of the underlying network infrastructure.

This architecture allows for flexibility in several additional ways. A domain operator is enabled to connect to several multicast overlays in parallel, may choose to replicate IMGs for load balancing or redundancy purposes or may transparently take advantage of the fail-safe unicast peering realized by multi-homed network connectivity. Replication operations will be seamlessly empowered by the self-organization capabilities of the DHT overlay. Active coordination between gateway peers may be achieved by SIP or non-SIP means.

1.4 Peer–Managed Conferences

An increasing number of applications aim for simple, flexible, and cost-efficient ad-hoc conferencing functions, which scale appropriately well, but avoid any infrastructure assistance. Such solutions require group session management and media distribution at peers. Commonly implemented as pure software on standard personal devices, user agent peers are exposed to severe restrictions in real-world deployments: Often they are located behind NATs and firewalls with network capacities confined to asymmetric DSL or wireless links. Multicast routing may be unsupported or only available in parts of the network facilities spanned by the conference. Clients may run on handhelds or other truly mobile devices that admit processing resources too scarce to serve for mixing and redistribution purposes. Peers may join or leave a conference in an unpredictable manner, advising other members not to rely on its relaying service. Nevertheless, real-time constraints apply to data processing and packet forwarding, whenever voice, video or interactive elements are the purpose of the conference.

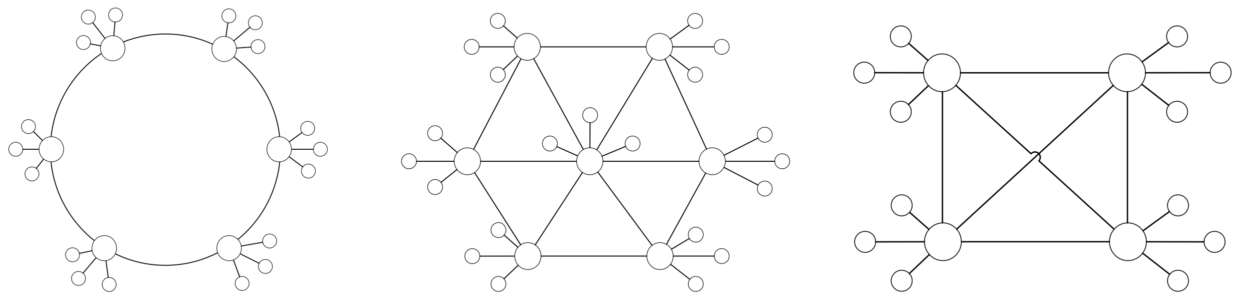
Capacity constraints and resilience to node failures require peer-managed ad-hoc conferences to organize in a distributed multi-party model. As a key component, the heterogenity of clients must be accounted for. Ranges of scalability however may vary at large. From an application point of view, many unmoderated systems are designed for only about one dozen participants, which in particular holds for dialog-oriented video conferences. Other applications, equally built of lightweight peers, may foresee media streams to reach large numbers of receivers. Two examples of SIP initiation, illustrating either side of the coin, are detailed out in the following.

1.4.1 A Simple, Distributed Point-to-Point Model

Peer-to-peer conferencing systems for moderate membership face the grand challenge of realization robust w.r.t. the infrastructure. The role a user agent is able to attain in a distributed scenario needs to be adaptively determined according to constraints of its device and current network attachment. In a simplified scenario, clients may be devided into two groups, distinguished by their ability to act as a focus or not. A focus must hold a GRUU and have access to necessary processing and network resources. This elementary adaptation scheme can be based on individual decisions of user agents and gives rise to a hybrid architecture of super peers, chosen from potential focus nodes, and remaining leaf nodes. Leaf nodes attach to super peers in subordinate position, whereas potential focus nodes may be assigned to be super peers or leaves. Super peers provide global connectivity among each other and NAT traversal assistance to leaves[6], while leaf nodes experience super peers in different roles: A leaf nodes sees its next hop super peer as the conference focus, while the remote super peers act as proxies on the path to the leaves behind.[7] This set-up corresponds to the well known architecture of Gnutella 0.6 and successive hybrid unstructured peer-to-peer systems, cf. [41]. Despite its architectural analogy, a routing layer for real-time group applications should follow a different design.

[6] Super peers are globally addressable nodes with packet relaying function. TURN will be the natural unilateral self-address fixing (UNSAF) protocol to use.

[7] This architecture relies on the presence of at least one globally addressable, sufficiently powerful peer. In scenarios, where this is likely to fail, a common practice of vendors or communities is to permanently deploy a 'silent' relay-peer at some unrestricted place.



(a) Ring (b) Polygon (c) Full mesh

Figure 1.8: Peer-to-peer routing toplogies on the overlay

From the perspective of the conference, super peers form a distributed focus. To keep distribution transparent to leaves, each super peer needs to provide full conferencing service functions, e.g., synchronized policing and event notification, and most likely assistance in media mixing and redistribution. Focus nodes consequently require signaling relationships among each other, established on top of an application layer routing of sufficient performance for a simultaneous distribution of media streams. The design of a routing will admit critical impact on scalability, application performance, as well as forwarding and maintanance load of the super peers. The three characteristic topologies for routing between N super peers as displayed in figure 1.8 explore the problem space: On the one extreme, routing on a ring will minimize neighbor states and forwarding load of each peer, but requires $\mathcal{O}(N)$ hops and thus induces large, varying delays. A full mesh, on the other extreme, places the burden of $N-1$ neighbor states to be fed in replicated forwarding, but guarantees a rigid 3-hop forwarding limit and minimal delays. A polygonal mesh of dimension d keeps replication load constant (but dependent on d), while its corresponding path lengths grow as $\mathcal{O}(\sqrt[d]{N})$. Forwarding on a polygonal mesh will require routing intelligence, which is neither needed on a ring nor in a full mesh topology. As routing paths in conferencing scenarios are equivalent to the signaling relationships, mesh robustness respectively redundancy of the schemes is equivalent to the number of neighbor states at each peer. Alternative routing topologies may be deployed on the basis of DHTs, admitting logarithmic scaling in all measures on the price of a higher algorithmic complexity.

Refocussing on the problem of moderately sized peer-to-peer conferences of simple and robust nature, a favorable routing scheme is easily identified. The full mesh topology outperforms alternative schemes in forwarding efficiency and robustness, while scaling well up to a hundred nodes, provided a significant fraction of unrestricted, high-performance super peers is available. In addition, this scenario will be bound to low complexity, since no routing intelligence beyond standard SIP logic of next hop proxying is required. Full mesh topologies are thus considered here as the favorable approach to mid-size multi-party conversations.

To explore the corresponding conference scenario in detail, consider an ad-hoc join. A client submits an INVITE to any party. It thereby needs to indicate its potential roles in some way.[8] The callee may be a conference focus or leaf node. In the first case, it will be aware of the overall leaf node distribution from the conference event states and will transfer the newly joining party to the least occupied super peer by a REFER, e.g.,

```
REFER sips:lucy@psychic.org SIP/2.0
```

[8] A corresponding client protocol extension has not been specified yet, cf. [3].

```
...
CSeq: 9380 REFER
Refer-To: <sips:hypnotic-talks@vain-focus.circles.com>
Content-Length: 0
```

In the second situation, the contacted leaf node will issue a re-INVITE to attach the new conference member to its focus, which in turn may refer the caller to another focus for the sake of load balancing. Having indicated its ability to serve as a super peer, the newly arrived party may be selected to join the group of focus nodes. This decision is taken by its current super peer and realized via a 3rd party invite issued to the group of all established focus nodes. The elected super peer will thereby establish point-to-point signaling relationships with all correspondents, leading to an immediate formation of the full mesh conference topology. Focus election and leaf node distribution may be conducted in an individual or collective way, following an eager or lazy strategy. Its implementation most likely will depend on the overall environment and the conference persistence of super peers.

Note that media negotiations have been part of the initial arrival steps for each party. Media distribution will naturally follow the paths of the established routing topology, where super peers can act as two-sided mixers: They may combine media streams arriving from their attached leaf nodes before peering them within the focus mesh, but may as well mix media arriving from neighboring super peers for a lean transmission to leaves.

1.4.2 Scalable, Peer-Centric Conferencing Based on SSM

In the presence of source specific multicast at the network layer (cf. section 1.2.3), peer-to-peer conferences can scale to very large numbers. Complying with the fully distributed ad-hoc paradigm, the following scenario is considered throughout this section.

Some party will initiate a conference by contacting one or several peers via unicast addresses as resolved from a SIP URI. Following an initial contact, signaling will then be turned to scalable multicast group communication. Further on new parties will join the conference by either calling or being called by an existing member. Such group conference initiation scheme is currently not covered by a SIP standard, nor is the employment of source specific multicast for group signaling. In order to enable SSM, all dialogs must carefully provision addresses of newly arriving senders to all current group members, which need to adapt source specific subscriptions appropriately; see [36] for further details.

Central to the approach introduced here is the concept of a fully delocalized focus. Conference signaling and management is entirely delegated to the peers participating in the multicast group. This scheme may be equally applied to an undistinguished group of equal participants or to super peers in a hybrid topology as outlined in the previous section. Hybrid architectures may account for SSM accessibility and shield multicast or conference unaware user agents.

In detail, protocol operations of SIP initiated SSM proceed as follows. A caller, wishing to participate in a previously established unicast dialog, will initiate a regular INVITE request to some selected member. Eventually, after the call setup has completed, either party will decide to transfer the established session to group communication. Heading for SSM, it will create a conference URI with a locator consisting of an SSM group address G, e.g.,

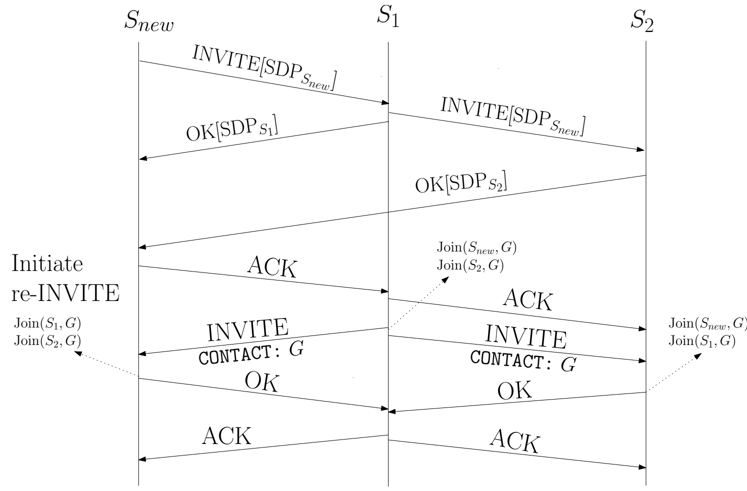


Figure 1.9: Call flow of switching from unicast to SSM within re-INVITE

```
sip:dog-matter@232.116.16.6.
```

Like in unicast conferences, the initiator will submit a re-INVITE announcing its conference URI, i.e., the desired multicast address, in the CONTACT field of the SIP header:

```
INVITE sip:lucy@psychic.org SIP/2.0
...
CSeq: 1024 INVITE
Contact: <sip:dog-matter@232.116.16.6>;isfocus
Content-Type: application/sdp
...
```

In parallel the SIP protocol stack will submit a multicast source specific JOIN to its underlying IGMP/MLD stack, thereby subscribing to the group and the source address of the correspondent peer, both learned from the previous SIP message exchange. Any peer will identify the multicast address in the Contact field, and proceed along the protocol semantic for SSM SIP.

This two-step procedure purposefully decouples application layer session establishment and underlying multicast routing operations. Temporal progress in IP layer multicast routing and SIP transactional timers thereby remain independent for the sake of robustly layered protocol operations. Appropriate media session descriptions for source specific multicast distribution of media streams may or may not be submitted along the re-INVITE request.

In multiparty environments, the straightforward generalization for switching a previously established unicast conference into SSM group communication is shown in figure 1.9. If the callee decides to accept the call from S_{new} it will forward the INVITE to its partner, thereby initiating unicast sessions among the three. Thereafter the callee will turn the conference signaling to multicast by submitting the corresponding re-INVITE procedure.

If a new source S_{new} contacts an established SSM group conference, it will do analogously by inviting some member S. If S decides to accept the caller, it will redistribute its INVITE

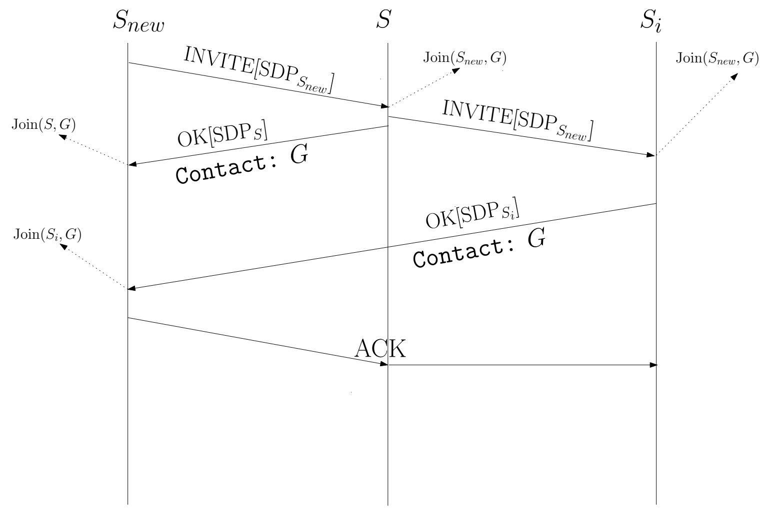


Figure 1.10: Call flow of extending established group sessions to a new party

to the SSM group and acknowledge the initial call by placing the group conference URI in the CONTACT header field. As displayed in figure 1.10, all group members will immediately add S_{new} to their source specific multicast filters. S_{new} subsequently will learn about all group members from (unicast) OK messages as needed for its own multicast subscriptions. Note that call redistribution will remain a point-to-point user transaction of S_{new} at the SIP layer, but a transmission from source S at the network layer and therefore compliant with previous SSM group establishment.

Proceeding along this incremental way, a callee will never be required to redistribute messages to more than one party or group. This scheme thus remains fully scalable and fairly transparent to group sizes. Multicast initiation of media sessions may be led correspondingly.

1.5 Summary & Conclusions

The concepts and mechanisms in this chapter describe how group conferences can be realized with SIP. They equally apply to voice and video, messaging and chat, gaming and collaborative environments, as well as to many other application areas of this lively and growing field.

Starting from elementary operations to initiate a multi-party conversation, an overview of basic SIP concepts and service primitives is presented for central and distributed conference management using unicast or multicast. Special focus is donated to the emerging field of mobile user agents which soon are expected to populate daily usage and require seamless integration. Presently available and developing mobility schemes promise to assist roaming conference members at real-time compliant handovers in an end-to-end fashion.

SIP offers widespread potentials for supporting group formation and communication, while the field of optimizing multi-peer sessions is under active current investigation. Comprising peer-to-peer technologies, as well as application layer and source specific multicast, this outline explores the rich solution space of future directions in infrastructure assisted or purely peer-managed conferences. Lightweight solutions adaptive to heterogeneous environments and varying scenarios are discussed. It is the hope of the authors to stimulate an active development of new applications that admit rich functionality and substantial deployment.

References

[1] Supratik Bhattacharyya. An Overview of Source-Specific Multicast (SSM). RFC 3569, IETF, July 2003.

[2] David Bryan, Salman Baset, Marcin Matuszewski, and Henry Sinnreich. P2PSIP Protocol Framework and Requirements. Internet Draft - work in progress 00, IETF, July 2007.

[3] David Bryan, Philip Matthews, Eunsoo Shim, and Dean Willis. Concepts and Terminology for Peer to Peer SIP. Internet Draft - work in progress 01, IETF, November 2007.

[4] Gonzalo Camarillo and Alan Johnston. Conference Establishment Using Request-Contained Lists in the Session Initiation Protocol (SIP). Internet Draft - work in progress 01, IETF, November 2007.

[5] Gonzalo Camarillo, Jörg Ott, and Keith Drage. The Binary Floor Control Protocol (BFCP). RFC 4582, IETF, November 2006.

[6] Ben Campbell, Rohan Mahy, and Cullen Jennings (Eds.). The Message Session Relay Protocol (MSRP). RFC 4975, IETF, September 2007.

[7] Ben Campbell, Jonathan Rosenberg, Henning Schulzrinne andChristian Huitema, and David Gurle. Session Initiation Protocol (SIP) Extension for Instant Messaging. RFC 3428, IETF, December 2002.

[8] Miguel Castro, Peter Druschel, Anne-Marie Kermarrec, Animesh Nandi, Antony I. T. Rowstron, and Atul Singh. SplitStream: High-Bandwidth Content Distribution in Cooperative Environments. In M. Frans Kaashoek and Ion Stoica, editors, *Peer-to-Peer Systems II. Second International Workshop, IPTPS 2003 Berkeley, CA, USA, February 21-22, 2003 Revised Papers*, volume 2735 of *LNCS*, pages 292–303, Berlin Heidelberg, 2003. Springer–Verlag.

[9] Miguel Castro, Peter Druschel, Anne-Marie Kermarrec, and Antony Rowstron. SCRIBE: A large-scale and decentralized application-level multicast infrastructure. *IEEE Journal on Selected Areas in Communications,*, 20(8):100–110, 2002.

[10] Miguel Castro, Michael Jones, Anne-Marie Kermarrec, Antony Rowstron, Marvin Theimer, Helen Wang, and Alec Wolman. An Evaluation of Scalable Application–level Multicast Built Using Peer-to-peer Overlays. In *IEEE Infocom 2003*, volume 2, pages 1510–1520, Piscataway, NJ, USA, March 2003. IEEE Press.

[11] Hans L. Cycon, Thomas C. Schmidt, Gabriel Hege, Matthias Wählisch, Detelv Marpe, Valeri George, and Mark Palkow. An Optimized H.264-based Video Conferencing Software for Mobile Devices. In Antonio Navarro, editor, *ISCE2008 – The 12th IEEE International Symposium on Consumer Electronics*, Piscataway, NJ, USA, April 2008. IEEE, IEEE Press. to appear.

[12] Stephen E. Deering. Host Extensions for IP Multicasting. RFC 1112, IETF, August 1989.

[13] Alan Johnston (Ed.). Session Initiation Protocol Service Examples. Internet Draft - work in progress 13, IETF, July 2007.

[14] Anargyros Garyfalos and Kevin Almeroth. A Flexible Overlay Architecture for Mobile IPv6 Multicast. *IEEE Journal on Selected Areas in Communications*, 23(11):2194–2205, November 2005.

[15] Mark Handley, Van Jacobson, and Colin Perkins. SDP: Session Description Protocol. RFC 4566, IETF, July 2006.

[16] Mark Handley, Colin Perkins, and Edmund Whelan. Session Announcement Protocol. RFC 2974, IETF, October 2000.

[17] H. Holbrook and B. Cain. Source-Specific Multicast for IP. RFC 4607, IETF, August 2006.

[18] Hugh Holbrook, Brad Cain, and Brian Haberman. Using Internet Group Manage-

ment Protocol Version 3 (IGMPv3) and Multicast Listener Discovery Protocol Version 2 (MLDv2) for Source-Specific Multicast. RFC 4604, IETF, August 2006.

[19] IAB and IESG. IETF Policy on Wiretapping. RFC 2804, Internet Engineering Task Force, May 2000.

[20] ITU-T Recommendation H.264 & ISO/IEC 14496-10 AVC. Advanced Video Coding for Generic Audiovisual Services. Technical report, ITU, 2005. Draft Version 3.

[21] ITU-T Recommendation H.323. Infrastructure of audio-visual services - Systems and terminal equipment for audio-visual services: Packet-based multimedia communications systems. Technical report, ITU, 2000. Draft Version 4.

[22] Alan Johnston and Orit Levin. Session Initiation Protocol (SIP) Call Control - Conferencing for User Agents. RFC 4579, IETF, August 2006.

[23] Rajeev Koodli and Charles Perkins. *Mobile Inter-Networking with IPv6*. Wiley-Interscience, Hoboken, NJ., 2007.

[24] Rohan Mahy and Dan Petrie. The Session Initiation Protocol (SIP) "Join" Header. RFC 3911, IETF, October 2004.

[25] Rohan Mahy, Robert Sparks, Jonathan Rosenberg, Dan Petrie, and Alan Johnston. A Call Control and Multi-party framwork for the Session Initiation Protocol (SIP). Internet Draft - work in progress 9, IETF, November 2007.

[26] Sylvia Ratnasamy, Mark Handley, Richard M. Karp, and Scott Shenker. Application-Level Multicast Using Content-Addressable Networks. In Jon Crowcroft and Markus Hofmann, editors, *Networked Group Communication, Third International COST264 Workshop, NGC 2001, London, UK, November 7-9, 2001, Proceedings*, volume 2233 of *LNCS*, pages 14–29, London, UK, 2001. Springer–Verlag.

[27] Jonathan Rosenberg. A Framework for Conferencing with the Session Initiation Protocol (SIP). RFC 4353, IETF, February 2006.

[28] Jonathan Rosenberg. A Hitchhiker's Guide to the Session Initiation Protocol (SIP). Internet Draft - work in progress 04, IETF, November 2007.

[29] Jonathan Rosenberg. Obtaining and Using Globally Routable User Agent (UA) URIs (GRUU) in the Session Initiation Protocol (SIP). Internet Draft - work in progress 15, IETF, October 2007.

[30] Jonathan Rosenberg, Jon Peterson, Henning Schulzrinne, and Gonzalo Camarillo. Best Current Practices for Third Party Call Control (3pcc) in the Session Initiation Protocol (SIP). RFC 3725, IETF, April 2005.

[31] Jonathan Rosenberg and Henning Schulzrinne. An Offer/Answer Model with the Session Description Protocol (SDP). RFC 3264, IETF, June 2002.

[32] Jonathan Rosenberg, Henning Schulzrinne, Gonzalo Camarillo, Alan Johnston, Jon Peterson, Robert Sparks, Mark Handley, and Eve Schooler. SIP: Session Initiation Protocol. RFC 3261, IETF, June 2002.

[33] Jonathan Rosenberg, Henning Schulzrinne, and Orit Levin. A Session Initiation Protocol (SIP) Event Package for Conference State. RFC 4575, IETF, August 2006.

[34] Jonathan Rosenberg, Henning Schulzrinne, and Rohan Mahy. An INVITE-Initiated Dialog Event Package for the Session Initiation Protocol (SIP). RFC 4235, IETF, November 2005.

[35] Thomas C. Schmidt and Matthias Wählisch. Multicast Mobility in MIPv6: Problem Statement and Brief Survey. IRTF Internet Draft – work in progress 02, MobOpts, November 2007.

[36] Thomas C. Schmidt, Matthias Wählisch, Hans L. Cycon, and Mark Palkow. Scalable Mobile Multimedia Group Conferencing based on SIP initiated SSM. In *Proc. of 4th European Conference on Universal Multiservice Networks – ECUMN'2007*, pages 200–209, Washington, DC, USA, February 2007. SEE/EUREL/IEEE, IEEE Computer Society Press.

[37] Henning Schulzrinne, Stephen Casner, Ron Frederick, and Van Jacobson. RTP: A

Transport Protocol for Real-Time Applications. RFC 3550, IETF, July 2003.

[38] Heiko Schwarz, Detlev Marpe, and Thomas Wiegand. Overview of the Scalable Video Coding Extension of the H.264/AVC Standard. *IEEE Transactions on Circuits and Systems for Video Technology)*, 17(9):1103–1120, September 2007.

[39] Aameek Singh and Arup Acharya. Multiplayer networked gaming with the session initiation protocol. *Computer Networks*, 49(1):38–51, November 2005.

[40] Robert Sparks. The Session Initiation Protocol (SIP) Refer Method. RFC 3515, IETF, April 2003.

[41] Ralf Steinmetz and Klaus Wehrle, editors. *Peer–to–Peer Systems and Applications*, volume 3485 of *LNCS*. Springer–Verlag, Berlin Heidelberg, 2005.

[42] Matthias Wählisch and Thomas C. Schmidt. Between Underlay and Overlay: On Deployable, Efficient, Mobility-agnostic Group Communication Services. *Internet Research*, 17(5):519–534, November 2007.

[43] Shelley Q. Zhuang, Ben Y. Zhao, Anthony D. Joseph, Randy H. Katz, and John D. Kubiatowicz. Bayeux: An Architecture for Scalable and Fault-tolerant Wide-Area Data Dissemination. In *Proc. of the 11th International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV 2001)*, pages 11–20, June 2001.

Index