Multi-party Chat Using the Message Session Relay Protocol (MSRP)
draft-ietf-simple-chat-03

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on May 3, 2009.

Copyright Notice

Copyright (C) The IETF Trust (2008).

Abstract

The Message Session Relay Protocol (MSRP) defines a mechanism for sending instant messages within a peer-to-peer session, negotiated using the Session Initiation Protocol (SIP) and the Session Description Protocol (SDP). This document defines the necessary tools for establishing multi-party chat sessions, or chat rooms,
using MSRP.

Table of Contents

1. Introduction .............................................. 3
2. Terminology ............................................... 4
3. Motivations and Requirements .............................. 5
4. Overview of Operation ..................................... 6
5. Creating, Joining, and Deleting a Chat Room ............... 8
   5.1. Creating a Chat Room ................................ 8
   5.2. Joining a Chat Room ................................ 8
   5.3. Deleting a Chat Room ................................. 9
6. Sending and Receiving Instant Messages .................... 9
   6.1. Regular Messages .................................... 9
   6.2. Private Messages .................................... 10
7. Nicknames .................................................. 12
   7.1. Using Nicknames within a Conference ................. 13
   7.2. Modifying a Nickname ................................ 14
   7.3. Removing a Nickname ................................. 14
   7.4. Nicknames in the Conference Event Package .......... 14
   7.5. Nicknames not supported nor allowed ................ 15
8. The SDP ‘chatroom’ attribute ................................ 15
9. Examples .................................................... 16
   9.1. Joining a chat room ................................ 16
   9.2. Setting up a nickname ................................ 18
   9.3. Sending a regular message to the chat room ....... 20
   9.4. Sending a private message to a participant ....... 21
   9.5. Obtaining an anonymous URI ......................... 23
10. IANA Considerations ......................................... 24
11. Security Considerations .................................... 24
12. Contributors ............................................... 25
13. Acknowledgments ........................................... 25
14. References ................................................ 25
   14.1. Normative References ............................... 25
   14.2. Informative References .............................. 26
Authors’ Addresses .......................................... 26
Intellectual Property and Copyright Statements ............... 28
1. Introduction

The Message Session Relay Protocol (MSRP) [RFC4975] defines a mechanism for sending a series of instant messages within a session. The Session Initiation Protocol (SIP) [RFC3261] in combination with the Session Description Protocol (SDP) [RFC3264] allows for two peers to establish and manage such sessions.

In another application of SIP, a user agent can join in a multi-party conversation called a conference that is hosted by a specialized user agent called a focus [RFC4353]. Such a conference can naturally involve an MSRP session as one of possibly many media components. It is the responsibility of an entity handling the media to relay instant messages received from one participant to the rest of the participants in the conference.

Several such systems already exist in the Internet. Participants in a chat room can be identified with a pseudonym or nickname, and decide whether their real identity is disclosed to other participants. Participants can also use a rich set of features such as the ability to send private instant messages to other participants. They also allow combining instant messaging with other media components, such as voice, video, white boarding, screen sharing, and file transfer.

Similar conferences are already available today with other technologies different than MSRP. For example, Internet Relay Chat (IRC) [RFC2810], Extensible Messaging and Presence Protocol [RFC3920] based chat rooms, and many other proprietary systems provide this kind of functionality. It makes sense to specify equivalent functionality for MSRP-based systems to both provide competitive features as well as enable interworking between the systems.

This document defines requirements, conventions, and extensions for providing private messages and nickname management in centralized conferences with MSRP. This document, however, does not specify functionality that can be used in conference with media different than MSRP. This memo uses the SIP Conferencing Framework [RFC4353] as a design basis. It also aims to be compatible with the Centralized Conferencing Framework [I-D.ietf-xcon-framework]. It is expected that future mechanisms will be developed for providing similar functionality in generic conferences, i.e., where the media is not only restricted to MSRP. The mechanisms described in this document provide a future compatible short-term solution for MSRP centralized conferences.
2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119, BCP 14 [RFC2119], and indicate requirement levels for compliant implementations.

This memo deals with a particular case of tightly coupled SIP conferences where the media exchanged consist of session-based instant messaging. Unless otherwise noted, we use the terminology defined in the SIP Conferencing Framework [RFC4353] applied to the scope of this document. In addition to that terminology, we introduce some new terms:

Nickname: a pseudonym or descriptive name associated to a participant. See Section 7 for details

Multi-party chat: an instance of a tightly coupled conference, in which the media exchanged between the participants consist of (among others) MSRP based instant messages. Also known as a chat room.

Chat Room: a synonym for a multi-party chat

Chat Room URI: a URI that identifies a particular chat room. Since a chat room is a specialized conference of instant messages, in the context of this document, a chat room URI is a synonym of a conference URI.

Sender: the conference participant that originally created an instant message and sent it to the chat room for delivery.

Recipient: the destination conference participant(s). This defaults to the full conference participant list, minus the IM Sender.

MSRP switch: a media level entity that receives MSRP messages and delivers them to the other conference participants. An MSRP switch has a similar role to a conference mixer with the exception that an MSRP switch does not actually "mix" together different input media streams; it merely relays the messages between participants.

Private Instant Message: an instant message sent in a chat room whose intended to a single participant. A private IM is usually rendered distinctly from the rest of the IMs, as to indicate that the message was a private communication.
Anonymous URI: a temporary GRUU that can be registered with the conference focus to conceal a participant’s SIP AOR from the other participants in the conference.

3. Motivations and Requirements

Although conference frameworks describing many types of conferencing applications already exist, such as the Framework and Data Model for Centralized Conferencing [I-D.ietf-xcon-framework] and the SIP Conferencing Framework [RFC4353], the exact details of session-based instant messaging conferences are not well-defined at the moment.

To allow interoperable chat implementations, for both conference-aware, and conference-unaware user agents, certain conventions for MSRP conferences need to be defined. It also seems beneficial to provide a set of features that enhance the baseline multi-party MSRP in order to be able to create systems that have functionality on par with existing chat systems, as well as enable building interworking gateways to these existing chat systems.

We define the following requirements:

REQ-1: A basic requirement is the existence of a multi-party conference, where participants can join and leave the conference and get instant messages exchanged to the rest of the participants.

REQ-2: The conference must have the ability to host other media in addition to MSRP, as well as multiple streams of MSRP.

REQ-3: A conference participant must be able to determine the identities of the sender and recipient of the received IMs.

REQ-4: A conference participant must be able to determine the recipient of the received message. For instance, the recipient of the message might be the entire conference or a single participant of the conference (i.e., a private message).

REQ-5: It must be possible to send a message to a single participant within the conference (i.e., a private instant message).

REQ-6: A conference participant may have a nickname or pseudonym associated with their real identity.
REQ-7: It must be possible for a participant to change their nickname during the progress of the conference.

REQ-8: It must be possible that a participant is only known by their nickname and not their real identity to the rest of the conference.

REQ-9: It must be possible for the MSRP switch itself to send IMs to the conference (e.g. message of the day, welcome messages, server is shutting down, etc.)

REQ-10: It must be possible for participants to learn the capabilities support of the features described in this document (and perhaps others).

4. Overview of Operation

In order to set up a conference, one must first be created. Users wishing to host a conference themselves can of course do just that; their user agents simply morph from an ordinary user agent into a special purpose one called a conference focus. Another, commonly used setup is one where a dedicated node in the network functions as a conference focus.

Each chat room has an identity of its own: a SIP URI that participants use to join the conference, e.g. by sending an INVITE request. The conference focus processes the invitations, and as such, maintains SIP dialogs with each participant. In an multi-party chat, or chat room, MSRP is one of the established media streams. Each conference participant establishes an MSRP session with an MSRP switch, which is a special purpose MSRP application. The MSRP switch is similar to a conference mixer in that it handles media sessions with each of the participants and bridges these streams together. However, unlike a conference mixer, the MSRP switch merely relays messages between participants but doesn’t actually mix the streams in any way. The system is illustrated in Figure 1.
Typically conference participants also subscribe to the conference event package [RFC4575] to gather information about the conference roster in the form of conference state notifications. For example, participants can learn about other participants’ identities.

All messages in the chat room use the ‘Message/CPIM’ wrapper content type [RFC3862], so that it is possible to distinguish between private and regular messages. When a participant wants to send an instant message to the conference, it constructs an MSRP SEND request and submits it to the MSRP switch including a regular payload (e.g. a Message/CPIM message that contains a text, html, an image, etc.). The Message/CPIM To header is set to the chat room URI. The switch then fans out the SEND request to all of the other participants using their existing MSRP sessions.

A participant can also send a private instant message addressed to a participants whose identity has been learned, e.g. via a notification from the conference event package [RFC4575]. In this case the sender creates an MSRP SEND request with a Message/CPIM body whose To header contains not the chat room URI but the recipient’s URI. The MSRP switch then forwards the SEND request to the recipient.

We extend the current MSRP negotiation that takes place in SDP
Internet-Draft            Multi-party Chat MSRP             October 2008

[RFC4566] to allow participants to learn whether the chat room supports and is willing to accept (e.g. due to local policy restrictions) certain MSRP functions defined in this memo, such as nicknames or private messaging.

Naturally, when a participant wishes to leave a chat room, it sends a SIP BYE request to the conference focus and disconnects.

5. Creating, Joining, and Deleting a Chat Room

5.1. Creating a Chat Room

Since we consider a chat room a particular type of conference where one of the offered media happens to be MSRP, the methods defined by the SIP Conference Framework [RFC4353] for creating conferences are directly applicable to a chat room.

Once a chat room is created, it is identified by a SIP URI, like any other conference.

5.2. Joining a Chat Room

Participants usually join the conference by sending an INVITE request to the conference URI. As long as the conference policy allows, the INVITE request is accepted by the focus and the user is brought into the conference. Participants are aware that the peer is a focus due to the presence of the "isfocus" feature tag [RFC3840] in the Contact header field of the 200-class response to the INVITE request. Participants are also aware that the mixer is an MSRP switch due to the presence of an additional ‘message’ media type and either TCP/MSRP or TCP/TLS/MSRP as the protocol field in the SDP [RFC4566] media-line.

The conference focus of a chat room MUST include support for a Message/CPIM [RFC3862] top-level wrapper for the MSRP messages by setting the ‘accept-types’ MSRP media line attribute in the SDP offer or answer to include ‘Message/CPIM’.

Note that the ‘Message/CPIM’ wrapper is used to carry the sender information that, otherwise, it will not be available to the recipient. Additionally, ‘Message/CPIM’ wrapper carries the recipient information (e.g. To and Cc: headers).

If a participant wants to remain anonymous to the rest of the participants in the conference, the participant’s UA can register or acquire by other means a temporary GRUU with the conference focus. The procedure SHOULD follow the recommendation of draft-ietf-sip-gruu
The temporary GRUU can be used in the From and To header in the 'Message/CPIM' wrapper concealing the participant’s SIP AOR from the other participants in the conference.

The conference focus of a chat room MUST learn the chat room capabilities of each participant that joins the chat room, and MUST inform the MSRP switch of such support. This is to prevent that the MSRP switch distributes private messages to participants who do not support private messaging.

5.3. Deleting a Chat Room

As with creating a conference, the methods defined by the SIP Conference Framework [RFC4353] for deleting a conference are directly applicable to a chat room.

Deleting a chat room is an action that heavily depends on the policy of the chat room. The policy can determine that the chat room is deleted when the creator leaves the conference, or with any out of band mechanism.

6. Sending and Receiving Instant Messages

6.1. Regular Messages

This section describes the conventions used to send and receive instant messages that are addressed to all the participants in the chat room. These are sent over a regular MSRP SEND request that contains a Message/CPIM wrapper [RFC3862] that in turn contains the desired payload (e.g. text, image, video-clip, etc.).

When a chat room participant wishes to send an instant message to all the other participants in the chat room, he constructs an MSRP SEND request that MUST contain a top-level wrapper of type ‘Message/CPIM’ [RFC3862]. The actual instant message payload inside ‘Message/CPIM’ MAY be of any type negotiated in the SDP ‘accepted-types’ attribute according to the MSRP rules.

The sender SHOULD populate the From header of the Message/CPIM wrapper with a proper identity by which the user is recognized in the conference. Identities that can be used (among others) are:

- A SIP URI [RFC3261] representing the participant’s address-of-record
- A tel URI [RFC3966] representing the participant’s telephone number
An MSRP switch that receives a SEND request from a participant SHOULD first verify that the From header field of the Message/CPIM wrapper is correctly populated with a valid URI. The valid URI can be the SIP AOR of the participant, or a temporary GRUU registered with the focus associated with an anonymous participant. If the URI included in the From header field of the Message/CPIM wrapper is not valid (e.g., because it does not "belong" to the user), then the MSRP switch MUST generate a 403 response and MUST NOT forward the SEND request to any of the participants. Otherwise, the MSRP switch SHOULD generate a 200 response according to the MSRP rules for response generation.

Then the MSRP switch should inspect the To header field of the Message/CPIM wrapper. If the To header field contains the chat room URI, the MSRP switch can generate a copy of the SEND request to each of the participants in the conference except the sender. The MSRP switch MUST NOT modify any of the bodies included in the received SEND request. Note that the MSRP switch does not need to wait for the reception of the complete MSRP chunk or MSRP message before it starts the distribution to the rest of the participants. Instead, once the MSRP switch has received the headers of the Message/CPIM body it SHOULD start the distribution process.

An MSRP endpoint that receives a SEND request from an MSRP switch containing a Message/CPIM wrapper SHOULD first inspect the To header field of the Message/CPIM body. If the To header field is set to the chat room URI, then it is a regular message that has been distributed to all the participants in the conference. Then the MSRP endpoint SHOULD inspect the From header field of the Message/CPIM body to identify the sender. The From header field will include a URI that identifies the sender. The endpoint might have also received further identity information through a subscription to the SIP conference event package [RFC4575].

### 6.2. Private Messages

This section describes the conventions used to send and receive private instant messages, i.e., instant messages that are addressed to one participant of the chat room rather to all of them. A chat room can signal support for private messages using the chatroom-attribute (see Section 8 for details).
When a chat room participant wishes to send a private instant message to a participant in the chat room, he constructs an MSRP SEND request that MUST contain a top-level wrapper of type `Message/CPIM` [RFC3862]. The actual instant message payload inside `Message/CPIM` MAY be of any type negotiated in the SDP `accepted-types` attribute according to the MSRP rules (e.g. text, image, video-clip etc.)

The sender SHOULD populate the From header of the `Message/CPIM` wrapper with a proper identity by which the user is recognized in the conference as indicated for regular instant messages. Then the sender MUST populate the To header field with the identity of the intended recipient. The identity can be SIP, TEL, and IM URIs typically learned from the information received in notifications of the conference event package [RFC4575].

As for regular messages, an MSRP switch that receives a SEND request from a participant SHOULD first verify that the From header field of the `Message/CPIM` wrapper is correctly populated with a valid URI. If the URI included in the From header field of the `Message/CPIM` wrapper is not valid (e.g, because it does not "belong" to the user), then the MSRP switch MUST generate a 403 response and MUST NOT forward the SEND request to any of the participants. Otherwise, the MSRP switch SHOULD generate a 200 response according to the MSRP rules for response generation.

Then the MSRP switch MUST inspect the To header field of the `Message/CPIM` wrapper. If the To header field of the `Message/CPIM` wrapper does not contain the chat room URI, it must check if it contains a participant's URI associated with a participant. If the URI in the To header can not be resolved (e.g. cased by a mistyped URI or that the recipient has abandoned the chat room), and the Failure-Report header field of the SEND request was either not present in the original request, or had a value of "yes", the MSRP switch MUST generate a REPORT request to the sender. The status header field MUST be set to 427. The new 427 status code indicates a failure to resolve the recipient URI in the To header field. If the recipient is valid, but the recipient does not support private messages, and the Failure-Report header field of the SEND request was either not present in the original request, or had a value of "yes", the MSRP switch MUST send a REPORT request having the status code of 428. The new response 428 indicate that the recipient does not support private messages. In either case the REPORT request MUST include a `Message/CPIM` wrapper, with the original From header field included in the SEND request, and the To header field of the original message. The message MUST not be forwarded to the recipient if above conditions applies. The MSRP switch should search its mapping table to find the MSRP session established towards the recipient. If a match is found the MSRP switch MUST create a SEND request and MUST copy the contents of the
sender’s message to it.

If the original SEND request contained a Success-report header field with the value of "yes" it MUST be added to the SEND request intended for the recipient. If the MSRP switch receives an success report from the recipient of the private message, and the original request had the Success-report header field present with a value of "yes", the MSRP switch MUST create a success REPORT and MUST copy the contents of the recipient’s report to it. The REPORT MUST be sent to the originator of the original SEND request.

An MSRP endpoint that receives a SEND request from an MSRP switch containing a Message/CPIM wrapper SHOULD first inspect the To header field of the Message/CPIM body. If the To header field is not set to the chat room URI, then it is a private message. Then the MSRP endpoint SHOULD inspect the From header field of the Message/CPIM body to identify the sender. The From header field will include a URI that identifies the sender. The endpoint might have also received further identity information through a subscription to the SIP conference event package [RFC4575].

It is possible that a participant, identified by a SIP Address of Record, joins a conference of instant messages from two or more different SIP UAs. It is RECOMMENDED that the an MSRP switch can map a participant or anonymous URI for two or more MSRP sessions. If the policy of the server allows for this, the MSRP switch MUST copy all messages intended for the recipient through each MSRP session.

7. Nicknames

A common characteristic of existing chat room services is that participants have the ability to identify themselves with a nickname to the rest of the participants of the conference. It is used for easy reference of participants in the chat room, and can also provide anonymous participants with a meaningful descriptive name.

Nicknames are a useful construct in many use cases, of which MSRP chat is but one example. Nicknames are an alternate form of identity, associated with a URI of which the participant is known to the focus. It is not a ‘display-name’, but it is used somewhat like a display name. A main difference is that a nickname is unique inside a chat room to allow an unambiguous reference to a participant in the chat. Nicknames may be long lived, or may be temporary. Users also need to reserve a nickname prior to its utilization.

This memo specifies the nickname as a string. The nickname string MUST be unambiguous within the scope of the chat room (conference
instance). This scope is similar to having a nickname unique inside a chat room from Extensible Messaging and Presence Protocol [RFC3920]. The chat room may have policies associated with nicknames. It may not accept nickname strings at all, or a it may provide a wider unambiguous scope like a domain or server, similar to Internet Relay Chat (IRC) [RFC2810].

7.1. Using Nicknames within a Conference

This memo provides a mechanism to reserve a nickname for a participant for as long as the participants is logged into the chat room. The mechanism is based on a NICKNAME MSRP method (see below) and a new "Use-Nickname" header. Note that other mechanisms may exists (for example, a web page reservation system), although they are outside the scope of this document.

A conference participant who has established an MSRP session with an MSRP switch, where the MSRP switch has indicated the support and availability of nicknames with the 'nicknames' token in the 'chatroom' SDP attribute, MAY send a NICKNAME request to the MSRP switch. The NICKNAME request MUST include a new Use-Nickname header that contains the nickname string that the participant wants to reserve.

An MSRP switch that receives a NICKNAME request containing a nickname in the Use-Nickname header field SHOULD first verify whether the policy of the chat room allows the nickname functionality. If is not allowed, the MSRP switch MUST answer with a 501 response.

If the policy of the chat room allows the usage of nicknames, the MSRP switch SHOULD validate that the SIP AOR is entitled to reserve the nickname. The participant’s authenticated identity can be derived after a successful HTTP Digest Authentication, included in a trusted SIP P-Asserted-Identity header field, included in a valid SIP Identity header field, or derived from any other present or future SIP authentication mechanism. Once the MSRP switch has validated that the participant is entitled to reserve the nickname, the MSRP switch answers to the MSRP NICKNAME request with a 200 response.

The reservation of a nickname can fail, e.g. if the NICKNAME request contains a malformed or non-existent Use-Nickname header field, or if the same nickname has already been reserved by another participant in the conference. The validation can also fail where the SIP AOR is not entitled to reserve the nickname. In any of these cases the MSRP switch MUST answer with a newly defined 423 response. The semantics of the 423 response are: "Nickname usage failed; the nickname is not allocated to this user".
As indicated earlier, this specification defines a new MSRP header field: "Use-Nickname". The Use-Nickname header field carries a nickname string, and SHOULD be included in the NICKNAME requests.

The syntax of the NICKNAME method and the "Use-Nickname" header field is built upon the MSRP formal syntax [RFC4975]

\[
\text{ext-method} =/ \text{NICKNAMEm} \\
\text{NICKNAMEm} = \%x4E.49.43.4B.4E.41.4D.45 ; \text{NICKNAME in caps} \\
\text{ext-header} =/ \text{Use-Nickname} \\
; \text{ext-header is specified in RFC 4975} \\
\text{Use-Nickname} = "\text{Use-Nickname" ":" nickname} \\
\text{nickname} = \text{quoted-string}
\]

7.2. Modifying a Nickname

Typically participants will reserve a nickname as soon as they join the chat room. But it is also possible for participants to modify their own nicknames and replace them it a new one at any time during the duration of the MSRP session. Modification of the nickname is not different from the initial reservation and usage of a nickname, thus the NICKNAME method is used as described in Section 7.1.

If a NICKNAME request that attempts to modify the current nickname of the user for some reason fails, the current nickname stays in effect. A new nickname comes into effect and the old one is released only after a NICKNAME request is accepted with a 200 response.

7.3. Removing a Nickname

If the participant no longer wants to be known by a nickname in the conference, the participant can follow the method described in Section 7.2. The nickname element of the Use-Nickname header MUST be set to an empty quoted string.

7.4. Nicknames in the Conference Event Package

Typically the conference focus acts as a notifier of the SIP conference event package [RFC4575]. The conference focus MAY notify subscribers of the nickname reserved by a given participant. We define an extension to the conference event package to include nicknames. The extension adds a \langle nickname-text \rangle attribute to the \langle user \rangle containing the nickname string.
TO BE DONE: include a formal definition of the <nickname> extension to the conference event package.

7.5. Nicknames not supported nor allowed

The participants SHOULD be notified of the URIs associated with the other participants of the conference even if nicknames are provided. The entity attribute in event notification framework being an SIP AOR or anonymous URI. A client not supporting the extensions of this memo will not render nicknames and can therefore cannot be referred to using nickname inside the chat room. The same would apply where a chat room do not allow nicknames to be used.

8. The SDP ‘chatroom’ attribute

There are a handful of use cases where a participant would like to learn the chat room capabilities supported by the MSRP switch and the chat room. For example, a participant would like to learn if the MSRP switch supports private messaging, otherwise, the participant may send what he believes is a private instant message addressed to a participant, but since the MSRP switch does not support the functions specified in this memo, the message gets eventually distributed to all the participants of the chat room.

The reverse case also exists. A participant, say Alice, whose user agent does not support the extensions defined by this document joins the chat room. The MSRP switch learns that Alice application does not support private messaging nor nicknames. If another participant, say Bob, sends a private message to Alice, the MSRP switch does not distribute it to Alice, because Alice is not able to differentiate it from a regular message sent to the whole roster. Further more, if Alice replied to this message, she would do it to the whole roster. Because of this, the MSRP switch keeps also track of users who do not support the extensions defined in this document.

In another scenario, the policy of a chat room may indicate that certain functions are not allowed. For example, the policy may indicate that nicknames or private messages are not allowed.

In order to provide the user with a good chat room experience, we define a new ‘chatroom’ SDP attribute. The ‘chatroom’ attribute is a media-level attribute that MAY be included in conjunction with and MSRP media stream (i.e., when an m= line in SDP indicates "TCP/MSRP" or "TCP/TLS/MSRP"). The ‘chatroom’ attribute indicates the intersection of support and chat room local policy allowance for a number of functions specified in this document. Specifically, we provide the means for indicating support to use nicknames and private
messaging.

The 'chatroom' SDP attribute has the following syntax:

```
chatroom = chatroom-label "::" chat-token *(SP chat-token)
chatroom-label = "chatroom"
chat-token = (nicknames-token | private-msg-token | token)
nicknames-token = "nicknames"
private-msg-token = "private-messages"
```

A conference focus that includes the 'nicknames' token in the session description is signaling that the MSRP switch supports and the chat room allows to use the procedures specified in Section 7. A conference focus that includes the 'private-messages' in the SDP description is signaling that the MSRP switch supports and the chat room allows to use the procedures specified in Section 6.2.

Example of the 'chatroom' attribute for an MSRP media stream that indicates the acceptance of nicknames and private messages:

```
a=chatroom:nickname private-messages
```

9. Examples

9.1. Joining a chat room

Figure 5 presents a flow diagram where Alice joins a chat room by sending an INVITE request. This INVITE request contains a session description that includes the chatroom extensions defined in this document.

```
Alice               Conference focus
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) (SIP) INVITE</td>
<td>----------------------------</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>(2) (SIP) 200 OK</td>
<td>&lt;-----------------------------</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>(3) (SIP) ACK</td>
<td>----------------------------</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------------------------</td>
</tr>
</tbody>
</table>
```

Figure 5: Flow diagram of a user joining a chat room

F1: Alice constructs an SDP description that includes an MSRP media stream. She also indicates her support for the chatroom extensions defined in this document. She sends the INVITE request to the chat room server.
INVITE sip:chatroom22@chat.example.com SIP/2.0
Via: SIP/2.0/TCP client.atlanta.example.com:5060;branch=z9hG4bK74bf9
Max-Forwards: 70
From: Alice <sip:alice@atlanta.example.com>;tag=9fxced76sl
To: Chatroom 22 <sip:chatroom22@chat.example.com>
Call-ID: 3848276298220188511@atlanta.example.com
CSeq: 1 INVITE
Contact: <sip:alice@client.atlanta.example.com;transport=tcp>
Content-Type: application/sdp
Content-Length: [length]

v=0
c=IN IP4 atlanta.example.com
m=message 7654 TCP/MSRP *
a=accept-types:message/cpim text/plain text/html
a=path:msrp://client.atlanta.example.com:7654/jshA7weztas;tcp
a=chatroom:nickname private-messages

Figure 6: INVITE request containing an SDP offer with chat room extensions

F2: The chat room server accepts the session establishment. It includes the 'isfocus' and other relevant feature tags in the Contact header field of the response. The chat room server also builds an SDP answer that also forces the reception of messages wrapped in message/cpim envelops. It also includes the chatroom attribute with the allowed extensions.
Figure 7: 200 (OK) response including chat room extensions

F3: The session established is acknowledged (details not shown).

9.2. Setting up a nickname

Figure 8 shows an example of Alice setting up a nickname using the conference as provider. Her first proposal is not accepted because the proposed nickname is already in use. Her second proposal is accepted.

Alice | MSRP switch
---|---
(1) (MSRP) NICKNAME | 
| (2) (MSRP) 423 |
| (3) (MSRP) NICKNAME |
| (4) (MSRP) 200 |

Figure 8: Flow diagram of a user setting up her nickname
F1: Alice sends an MSRP NICKNAME request that contains her proposed nicknames in the Set-Nickname header field.

MSRP d93kswow NICKNAME
To-Path: msrp://chat.example.com:12763/kjhd37s2s20w2a;tcp
From-Path: msrp://client.atlanta.example.com:7654/jshA7weztas;tcp
Use-Nickname: "Alice the great"
-------d93kswow$

Figure 9: MSRP NICKNAME request with an initial nickname proposal

F2: The MSRP switch analyzes the existing allocation of nicknames and detects that the nickname "Alice is great" is already provided to another participant by the conference. The MSRP switch answers with a 423 response.

MSRP d93kswow 423 Nickname usage failed
To-Path: msrp://client.atlanta.example.com:7654/jshA7weztas;tcp
From-Path: msrp://chat.example.com:12763/kjhd37s2s20w2a;tcp
-------d93kswow$

Figure 10: MSRP 423 response

F3: Alice receives the response. She proposes a new nickname in a second NICKNAME request.

MSRP 09swk2d NICKNAME
To-Path: msrp://chat.example.com:12763/kjhd37s2s20w2a;tcp
From-Path: msrp://client.atlanta.example.com:7654/jshA7weztas;tcp
Use-Nickname: "Alice in wonderland"
-------09swk2d$

Figure 11: MSRP NICKNAME request with a second nickname proposal

F4: The MSRP switch accepts the nickname proposal and answers with a 200 response.

MSRP 09swk2d 200 OK
To-Path: msrp://client.atlanta.example.com:7654/jshA7weztas;tcp
From-Path: msrp://chat.example.com:12763/kjhd37s2s20w2a;tcp
-------09swk2d$

Figure 12: MSRP NICKNAME request
9.3. Sending a regular message to the chat room

Figure 13 depicts a flow diagram where Alice is sending a regular message addressed to the chat room. The MSRP switch distributes the message to the rest of the participants.

<table>
<thead>
<tr>
<th>Alice</th>
<th>MSRP switch</th>
<th>Bob</th>
<th>Charlie</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) (MSRP) SEND</td>
<td>--------------</td>
<td>--------------</td>
<td>--------------</td>
</tr>
<tr>
<td>(2) (MSRP) 200</td>
<td>--------------</td>
<td>--------------</td>
<td>--------------</td>
</tr>
<tr>
<td>&lt;-----------------&gt;</td>
<td>(3) (MSRP) SEND</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(4) (MSRP) SEND</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(5) (MSRP) 200 OK</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;-------------------&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(6) (MSRP) 200 OK</td>
<td></td>
</tr>
</tbody>
</table>

Figure 13: Sending a regular message to the chat room

F1: Alice builds a text message and wraps it in a CPIM message. She addresses the CPIM message to the chat room. She encloses the result in an MSRP SEND request and sends it to the MSRP switch via the existing TCP connection.

MSRP 3490visdm SEND
To-Path: msrp://chat.example.com:12763/kjhd37s2s20w2a;tcp
From-Path: msrp://client.atlanta.example.com:7654/jshA7weztas;tcp
Message-ID: 99s9s2
Byte-Range: 1-/*
Content-Type: message/cpim

To: <sip:chatroom22@chat.example.com;transport=tcp>
From: <sip:alice@atlanta.example.com>
DateTime: 2007-03-02T15:02:31-03:00
Content-Type: text/plain

Hello guys, how are you today?
-------3490visdm$

Figure 14: Instant message addressed to all participants in the chat room

F2: The MSRP switch acknowledges the reception of the SEND request with a 200 (OK) response.
MSRP 3490visdm 200 OK
To-Path: msrp://client.atlanta.example.com:7654/jshA7weztas;tcp
From-Path: msrp://chat.example.com:12763/kjhd37s2s20w2a;tcp
Message-ID: 99s9s2
Byte-Range: 1-/*/*
-------3490visdm$

Figure 15: 200 (OK) response

F3: The MSRP switch creates a new MSRP SEND request that contains the received message/cpim body and sends it to Bob.

MSRP 490ej23 SEND
To-Path: msrp://client.biloxi.example.com:4923/49dufdje2;tcp
From-Path: msrp://chat.example.com:5678/jofofo3;tcp
Message-ID: 304sse2
Byte-Range: 1-/*/*
Content-Type: message/cpim

To: <sip:chatroom22@chat.example.com;transport=tcp>
From: <sip:alice@atlanta.example.com>
DateTime: 2007-03-02T15:02:31-03:00
Content-Type: text/plain

Hello guys, how are you today?
-------490ej23$

Figure 16: Instant message sent to all participants

The rest of the message flows are analogous to the previous. They are not shown here.

9.4. Sending a private message to a participant

Figure 17 depicts a flow diagram where Alice is sending a private message addressed to Bob’s SIP AOR. The MSRP switch distributes the message only to Bob.

Alice                     MSRP switch                     Bob                     Charlie
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) (MSRP) SEND</td>
<td>(3) (MSRP) SEND</td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------------------</td>
<td>------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(2) (MSRP) 200</td>
<td>(4) (MSRP) SEND</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;------------------------</td>
<td>------------------------</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Figure 17: Sending a private message to Bob

F1: Alice builds a text message and wraps it in a CPIM message. She addresses the CPIM message to the Bob’s nickname, which she learned from a notification in the conference event package. She encloses the result in an MSRP SEND request and sends it to the MSRP switch via the existing TCP connection.

MSRP 6959ssdf SEND
To-Path: msrp://chat.example.com:12763/kjhd37s2s20w2a;tcp
From-Path: msrp://client.atlanta.example.com:7654/jshA7wezatas;tcp
Message-ID: okj3kw
Byte-Range: 1-/*/*
Content-Type: message/cpim

To: <sip:Bob%20the%20great@example.com>
From: <sip:Alice%20in%20wonderland@example.com>
DateTime: 2007-03-02T15:02:31-03:00
Content-Type: text/plain

Hello Bob.
------6959ssdf$

Figure 18: Private instant message addressed to one participant

F2: The MSRP switch acknowledges the reception of the SEND request with a 200 (OK) response.

MSRP 6959ssdfm 200 OK
To-Path: msrp://client.atlanta.example.com:7654/jshA7wezatas;tcp
From-Path: msrp://chat.example.com:12763/kjhd37s2s20w2a;tcp
Message-ID: okj3kw
Byte-Range: 1-/*/*
------6959ssdfm$

Figure 19: 200 (OK) response

F3: The MSRP switch creates a new MSRP SEND request that contains the received message/cpim body and sends it only to Bob. Bob can distinguish the sender in the From header of the CPIM message. He also identifies this as a private message due to the To CPIM header.
MSRP 9v9s2 SEND
To-Path: msrp://client.biloxi.example.com:4923/49dudfje2;tcp
From-Path: msrp://chat.example.com:5678/jofofo3;tcp
Message-ID: d9fghe982
Byte-Range: 1-/*
Content-Type: message/cpim

To: <sip:Bob%20the%20great@example.com>
From: <sip:alice@atlanta.example.com>
DateTime: 2007-03-02T15:02:31-03:00
Content-Type: text/plain

Hello Bob.
-------9v9s2$

Figure 20: Private instant message sent to Bob

Flow F4 is not shown.

9.5. Obtaining an anonymous URI

Figure 21 presents a flow diagram where Alice registers her SIP AOR with the conference focus. The response will contain a temp-gruu which can be used as an anonymous URI when joining the conference. The temp-gruu is also used to send anonymous MSRP messages to and from the MSRP switch.

Alice               Conference focus
 |                       |
| (1) (SIP) REGISTER    |----------------------->|
| ----------------------| (2) (SIP) 200 OK      |
|                        |<-----------------------|

Figure 21: Flow diagram of registering an anonymous URI

F1: Alice constructs an REGISTER including an instance id in her Contact header defined in draft-ietf-sip-gruu [I-D.ietf-sip-gruu].
REGISTER sip:chatroom22@chat.example.com SIP/2.0
Via: SIP/2.0/TCP client.atlanta.example.com:5060;branch=z9hG4bK74bf9
Max-Forwards: 70
From: Alice <sip:alice@atlanta.example.com>;tag=9fxced76sl
To: Alice <sip:alice@atlanta.example.com>
Supported: gruu
Call-ID: 3848276298220188511@atlanta.example.com
CSeq: 1 REGISTER
Contact: <sip:alice@client.atlanta.example.com;transport=tcp> \
;+sip.instance="<urn:uuid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6>"
Content-Length: 0

Figure 22: REGISTER request containing a Contact header with an instance id

SIP/2.0 200 OK
Via: SIP/2.0/TCP client.atlanta.example.com:5060;branch=z9hG4bK74bf9 \
;received=192.0.2.101
From: Alice <sip:alice@atlanta.example.com>;tag=9fxced76sl
To: Alice <sip:alice@atlanta.example.com>
Call-ID: 3848276298220188511@atlanta.example.com
CSeq: 1 REGISTER
Contact: <sip:alice@client.atlanta..example.com;transport=tcp> \
;pub-gruu="sip:callee@example.com \
;gr=urn:uuid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6" \
;temp-gruu="sip:tgruu.7hatz6cn-098s-anonymous@chat.example.com;gr" \
;+sip.instance="<urn:uuid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6>"
Content-Length: 0

Figure 23: 200 (OK) response including a temp-gruu in the Contact header

10. IANA Considerations

TBD.

11. Security Considerations

This document proposes extensions to the Message Session Relay Protocol [RFC4975]. Therefore, the security considerations of such
document apply to this document as well.

In general, messages sent to a multi-party session based messaging focus are not deemed to expose any security threat. Nevertheless, if a participant wants to avoid eavesdropping from non-authorized entities, it should send those messages a TLS [RFC4346] transport connection, as allowed by MSRP.

12. Contributors

This work would have never been possible without the fruitful discussions in the SIMPLE WG mailing list, specially with Brian Rosen (Neustar) and Paul Kyzivat (Cisco), who provided extensive review and improvements throughout the document.

13. Acknowledgments

The authors want to thank Eva Leppanen, Adamu Haruna, Adam Roach and Matt Leptinski for providing comments.

14. References

14.1. Normative References


14.2. Informative References


Authors’ Addresses

Aki Niemi
Nokia
P.O. Box 407
NOKIA GROUP, FIN 00045
Finland

Phone: +358 50 389 1644
Email: aki.niemi@nokia.com

Miguel A. Garcia-Martin
Nokia Siemens Networks
P.O.Box 6
Nokia Siemens Networks, FIN 02022
Finland

Email: miguel.garcia@nsn.com

Geir A. Sandbakken (editor)
TANDBERG
N-1366 Lysaker
Norway

Phone: +47 67 125 125
Email: geir.sandbakken@tandberg.com
URI:  http://www.tandberg.com