Multi-party Chat Using the Message Session Relay Protocol (MSRP)
draft-ietf-simple-chat-07

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.

Status of this Memo

This Internet-Draft is submitted in full conformance with the
provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering
Task Force (IETF). Note that other groups may also distribute
working documents as Internet-Drafts. The list of current Internet-
Drafts is at http://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on December 2, 2010.

Copyright Notice

Copyright (c) 2010 IETF Trust and the persons identified as the
document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal
Provisions Relating to IETF Documents
(http://trustee.ietf.org/license-info) in effect on the date of
publication of this document. Please review these documents
carefully, as they describe your rights and restrictions with respect

Table of Contents

1. Introduction ........................................... 4
2. Terminology ............................................ 4
3. Motivations and Requirements ............................ 6
4. Overview of Operation .................................... 7
5. Creating, Joining, and Deleting a Chat Room ............... 10
   5.1. Creating a Chat Room ................................. 10
   5.2. Joining a Chat Room .................................. 10
   5.3. Deleting a Chat Room ................................. 11
6. Sending and Receiving Instant Messages ..................... 12
   6.1. Regular Messages ...................................... 12
   6.2. Private Messages ...................................... 13
   6.3. MSRP reports and responses ............................ 15
7. Nicknames ................................................ 16
   7.1. Using Nicknames within a Conference .................. 16
   7.2. Modifying a Nickname ................................... 17
   7.3. Removing a Nickname ................................... 18
   7.4. Nicknames in the Conference Event Package ............ 18
   7.5. Nicknames not supported nor allowed .................. 18
8. The SDP ‘chatroom’ attribute ................................ 18
9. Examples ............................................... 19
   9.1. Joining a chat room .................................... 19
   9.2. Setting up a nickname .................................. 21
   9.3. Sending a regular message to the chat room ............ 22
   9.4. Sending a private message to a participant ............ 24
   9.5. Chunked private message ............................... 26
   9.6. Nickname in a conference information document ........ 26
10. IANA Considerations ...................................... 27
   10.1. New MSRP Method ...................................... 27
   10.2. New MSRP Header ....................................... 28
   10.3. New MSRP Status Codes .................................. 28
   10.4. New SDP Attribute ...................................... 29
11. Security Considerations ................................... 29
12. Contributors ............................................ 30
13. Acknowledgments .......................................... 30
14. References ............................................... 30
   14.1. Normative References .................................. 30
   14.2. Informative References ............................... 31
Authors’ Addresses .......................................... 31
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 MSRP sessions. 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.

Similar conferences supporting chat rooms are already available today. For example, Internet Relay Chat (IRC) [RFC2810], Extensible Messaging and Presence Protocol [RFC3920] based chat rooms, and many other proprietary systems provide chat room functionality. Specifying equivalent functionality for MSRP-based systems provides competitive features and enables interworking between the systems.

This document defines requirements, conventions, and extensions for providing private messages and nickname management in centralized conferences with MSRP. Participants in a chat room can be identified by a pseudonym, and decide if their real identity is disclosed to other participants. This memo uses the SIP Conferencing Framework [RFC4353] as a design basis. It also aims to be compatible with the A Framework for Centralized Conferencing [RFC5239]. 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 tightly coupled SIP conferences defined in SIP Conferencing Framework [RFC4353] adopting the terminology. 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 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, and is a synonym of a Conference URI defined in RFC 4353 [RFC4353].

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 is a MSRP endpoint. It is a special MSRP endpoint that receives MSRP messages, and delivers them to the other conference participants. The MSRP switch has a similar role to a conference mixer with the exception that the 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 intended for a single participant. A private IM is usually rendered distinctively from the rest of the IMs, indicating that the message was a private communication.

Anonymous URI: a URI concealing the participant’s SIP AOR from the other participants in the conference. The allocation of such a URI is out of scope of this specification. An anonymous URI must be valid for the length of the conference, and will be utilized by the MSRP switch to forward messages to and from anonymous participants.
3. Motivations and Requirements

Although conference frameworks describing many types of conferencing applications already exist, such as the Framework for Centralized Conferencing [RFC5239] 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: A conference participant must be able to determine the identities of the sender and recipient of the received IMs.

REQ-3: 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-4: It must be possible to send a message to a single participant within the conference (i.e., a private instant message).

REQ-5: A conference participant may have a nickname or pseudonym associated with their real identity.

REQ-6: It must be possible for a participant to change their nickname during the progress of the conference.

REQ-7: It must be possible that a participant is only known by an anonymous identity and not their real identity to the rest of the conference.
REQ-8: It must be possible for the MSRP switch to originate IMs to the conference as an owner or administrator (e.g. message of the day, welcome messages, server is shutting down, etc.)

REQ-9: It must be possible for the conference participants to learn the chat room capabilities described in this document.

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 Agent (UA) simply morphs from an ordinary UA into a special purpose one called a Focus UA. Another, commonly used setup is one where a dedicated node in the network functions as a Focus UA.

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 a multi-party chat, or chat room, MSRP is one of the established media streams. Each conference participant establishes an MSRP session with the MSRP switch, which is a special purpose MSRP application. The MSRP sessions can be relayed by one or more MSRP relays, which are specified in RFC 4976 [RFC4976]. This is illustrated in Figure 1
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 forwards messages between participants but doesn’t actually mix the streams in any way. The system is illustrated in Figure 2.
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 participant 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. This specification supports the sending of private messages to one and only one recipient. However, if the recipient is logged from
different endpoints, the MSRP switch will distribute the private message to each endpoint the recipient is logged.

We extend the current MSRP negotiation that takes place in SDP [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 Focus UA and terminates the SIP dialog with the focus and MSRP sessions with the MSRP switch.

This document assumes that each chat room is allocated its own SIP URI. A user joining a chat room sends an INVITE request to that SIP URI, and as a result, a new MSRP session is established between the user and the MSRP switch. It is assumed that an MSRP session is mapped to a chat room. If a user wants to join a second chat room, he creates a different INVITE request, through a different SIP dialog, which leads to the creation of a second MSRP session between the user and the MSRP switch. Notice that these two MSRP sessions can still be multiplexed over the same TCP connection as per regular MSRP procedures. However, each chat room is associated to a unique MSRP session and a unique SIP dialog.

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 having MSRP media, 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.

The MSRP switch needs to be aware of the URIs of the participant (SIP, Tel, or IM URIs) in order to validate messages sent from this participant prior to their forwarding. This information is known to
the focus of the conference. Therefore an interface between the focus and the MSRP switch is assumed. However, the interface between the focus and the MSRP switch is outside the scope of this document.

Conference aware participants will detect 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. Conference unaware participants will not notice it is a focus, and can not apply the additional mechanisms defined in this document. Participants are also aware that the mixer is an MSRP switch due to the presence of an ‘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 must provide an anonymous URI to the conference focus. The URI will be used in the From and To headers in the ‘Message/CPIM’ wrapper, and can be learned by the other participants of the conference. Notice that in order for the anonymity mechanism to work, the anonymous URI must not reveal the participant’s SIP AOR. The mechanism for acquiring an anonymous URI is outside the scope of this specification.

The conference focus of a chat room MUST learn the chat room capabilities of each participant that joins the chat room. The conference focus must inform the MSRP switch of such support in order to prevent the MSRP switch from distributing private messages to participants who do not support private messaging. The recipient would not be able to render the message as private, and any potential reply would be sent to the whole chat room.

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. The MSRP switch will terminate the MSRP sessions with all the participants.

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, it constructs an MSRP SEND request according to the procedures specified in RFC 4975 [RFC4975].
The sender MAY choose the desired MSRP report model (e.g., populate
the Success-Report and Failure-Report MSRP header fields).

The SEND request MUST contain a top-level wrapper of type 'Message/
CPIM' according to RFC 3862 [RFC3862]. The actual instant message
payload MUST be included as payload of the 'Message/CPIM' wrapper and
MAY be of any type negotiated in the SDP 'accept-types' attribute
according to the MSRP rules.

On sending a regular message the sender MUST populate the To header
of the Message/CPIM wrapper with the URI of the chat room. 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 IM URI [RFC3860] representing the participant’s instant
  messaging address
- An Anonymous URI representing the participant’s anonymous address

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 of a participant. This
imposes a requirement for the focus of the conference to inform the
MSRP switch the URIs which the participant is known, in order for the MSRP switch to validate messages. Section 6.3 provides further information with the actions to be taken in case this validation fails.

If the MSRP switch receives a message containing several To header fields in the Message/CPIM wrapper the MSRP switch MUST reject the MSRP SEND request with a 403 response, as per procedures in RFC 4975 [RFC4975].

Then the MSRP switch should inspect the To header field of the Message/CPIM wrapper. If the To header field of the Message/CPIM wrapper contains the chat room URI and there are no other To header fields, 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 the content received in the SEND request. However, the MSRP switch MAY re-chunk any of the outbound MSRP SEND requests.

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. Having the Message/CPIM header only in the first chunk, the MSRP switch MUST track the Message-Id until the last chunk of the message has been distributed.

An MSRP endpoint that receives a SEND request from the 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, it should render it as 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 the chat room, it follows the same procedures for
creating a SEND request as for regular messages (Section 6.1). The only difference is that the MSRP endpoint MUST populate a single To header of the Message/CPIM 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 (i.e., the URI is a participant of this chat room). Section 6.3 provides further information with the actions to be taken in case this validation fails.

If the MSRP switch receives a message containing several To header fields in the Message/CPIM wrapper the MSRP switch MUST reject the MSRP SEND request with a 403 response, as per procedures in RFC 4975 [RFC4975].

Then the MSRP switch MUST verify that the To header of the Message/CPIM wrapper is a participant of the chat room. If this To header field does not contain the URI of a participant of the chat room or if the To header field cannot be resolved (e.g., caused by a mistyped URI), the MSRP switch MUST reject the request with a 404 response. This new 404 status code indicates a failure to resolve the recipient URI in the To header field of the Message/CPIM wrapper.

Notice the importance of the From and To headers in the Message/CPIM wrapper. If an intermediary modifies these values, the MSRP switch might not be able to identify the source or intended destination of the message, resulting in a rejection of the message.

Finally, the MSRP switch MUST verify that the recipient supports private messages. If the recipient does not support private messages, the MSRP switch MUST reject the request with a 428 response. This new response 428 indicate that the recipient does not support private messages. Any potential REPORT request that the MSRP switch sends to the sender MUST include a Message/CPIM wrapper containing the original From header field included in the SEND request and the To header field of the original Message/CPIM wrapper. The MSRP switch MUST NOT forward private messages to a recipient that does not support private messaging.

If successful, the MSRP switch should search its mapping table to find the MSRP sessions 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.
An MSRP endpoint that receives a SEND request from the MSRP switch does the same validations as for regular messages (Section 6.1). If the To header field is different from the chat room URI, the MSRP endpoints know that it is a private message. It should render who it is from based on the From header of the Message/CPIM wrapper.

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

6.3. MSRP reports and responses

This section discusses the common procedures for regular and private messages with respect to MSRP reports and responses. Any particular procedure affecting only regular messages or only private messages is discussed in the previous Section 6.1 or Section 6.2, respectively.

MSRP switches MUST follow the success report and failure report handling described in section 7 of RFC 4975 [RFC4975], complemented with the procedures described in this section. The MSRP switch MUST act as an MSRP endpoint receiver of the request according to section 5.3 of RFC 4975 [RFC4975].

If the MSRP switch receives an MSRP SEND request that does not contain a Message/CPIM wrapper, the MSRP switch MUST reject the request with a 415 response (specified in RFC 4975 [RFC4975]).

If the MSRP switch receives an MSRP SEND request where 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 sender of the message or is not a valid participant of the chat room), the MSRP switch MUST reject the request with a 403 response. In non-error cases, the MSRP switch MUST construct responses according to section 7.2 of RFC 4975 [RFC4975].

On receiving a SEND request, the MSRP switch MAY use any report model in the copies of the regular SEND request intended for the recipients, but any received reports MUST NOT be forwarded to the originator of the original SEND request. This could lead to having the sender receiving multiple reports for a single MSRP request.
7. Nicknames

A common characteristic of existing chat room services is that participants have the ability to present 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.

A nickname is a useful construct in many use cases, of which MSRP chat is but one example. It is associated with a URI of which the participant is known to the focus. It is a user selectable appearance of which the participant wants to be known to the other participants. 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 participant 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 exist (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 the 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. MSRP NICKNAME requests MUST NOT include Success-Report or Failure-Report header fields.

The 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
not allowed, the MSRP switch must reject the request with a 501 response, as per RFC 4975 [RFC4975].

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 MUST answer the NICKNAME request with a 200 response as per regular MSRP procedures.

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 sender of the message is not entitled to reserve the nickname. In any of these cases the MSRP switch MUST answer the NICKNAME request with a 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]

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

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 <nickname> child element to the <user> element containing the nickname string.

The following element is to be added to the sequence of the USER-TYPE in the XML schema in conference event package [RFC4575]

```xml
<xs:element name="nickname" type="xs:string" minOccurs="0" />
```

7.5. Nicknames not supported nor allowed

The participants of the conference are identified by the SIP, TEL and IM URI's typically learned from the information received in notifications of the conference event package [RFC4575]. If nicknames are not supported nor allowed, the participant list of the conference will be less presentable.

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:

```plaintext
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 of 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 of 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:

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

9. Examples

9.1. Joining a chat room

Figure 3 presents a flow diagram where Alice joins a chat room by sending an INVITE request. This INVITE request contains a session
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
o=alice 2890844526 2890844526 IN IP4 client.atlanta.example.com
s=-
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
```

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 envelopes. It also includes the the chatroom attribute with the allowed extensions.
9.2. Setting up a nickname

Figure 4 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
<p>| |
|                        |</p>
<table>
<thead>
<tr>
<th>F1: (MSRP) NICKNAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>F2: (MSRP) 423</td>
</tr>
<tr>
<td>&lt;----------------------</td>
</tr>
<tr>
<td>F3: (MSRP) NICKNAME</td>
</tr>
<tr>
<td>-----------------------</td>
</tr>
<tr>
<td>F4: (MSRP) 200</td>
</tr>
<tr>
<td>&lt;----------------------</td>
</tr>
</tbody>
</table>
```

Figure 4: 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$

F2: The MSRP switch analyzes the existing allocation of nicknames and
detects that the nickname "Alice the 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$

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$

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$

9.3. Sending a regular message to the chat room

Figure 5 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.
Figure 5: 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: 2009-03-02T15:02:31-03:00
Content-Type: text/plain

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

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
-------3490visdm$

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

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: 2009-03-02T15:02:31-03:00
Content-Type: text/plain

Hello guys, how are you today?
-------3490visdm$
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: 2009-03-02T15:02:31-03:00
Content-Type: text/plain

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

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 6 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
|                      |                        |
| F1: (MSRP) SEND     |                        |
|----------------------|---------------------->
| F2: (MSRP) 200       |                        |
|<---------------------|---------------------->

Figure 6: 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 URI, 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/jshA7weztas;tcp
Message-ID: okj3kw
Byte-Range: 1-*/*
Content-Type: message/cpim

To: <sip:bob@example.com>
From: <sip:alice@example.com>
DateTime: 2009-03-02T15:02:31-03:00
Content-Type: text/plain

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

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/jshA7weztas;tcp
From-Path: msrp://chat.example.com:12763/kjhd37s2s20w2a;tcp
Message-ID: okj3kw
-------6959ssdfm$

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:4923/49dufdje2;tcp
From-Path: msrp://chat.example.com:5678/jofofo3;tcp
Message-ID: d9fghe982
Byte-Range: 1-*/*
Content-Type: message/cpim

To: <sip:bob@example.com>
From: <sip:alice@atlanta.example.com>
DateTime: 2009-03-02T15:02:31-03:00
Content-Type: text/plain

Hello Bob.
-------9v9s2$
9.5. Chuncked private message

The MSRP message below depicts an example of the private message in Section 9.4 split in two chunks. The MSRP switch must wait for the complete set of CPIM headers before distributing the messages.

MSRP 7443rules SEND
To-Path: msrp://chat.example.com:12763/kjhd37s2s20w2a;tcp
From-Path: msrp://client.atlanta.example.com:7654/jshA7weztas;tcp
Message-ID: aft4to
Byte-Range: 1-*/174
Content-Type: message/cpim

To: <sip:bob@example.com>
From: <sip:anonymousX@example.com>
-------7443rules$

MSRP 7443rules SEND
To-Path: msrp://chat.example.com:12763/kjhd37s2s20w2a;tcp
From-Path: msrp://client.atlanta.example.com:7654/jshA7weztas;tcp
Message-ID: aft4to
Byte-Range: 68-174/174
Content-Type: message/cpim

DateTime: 2009-03-02T15:02:31-03:00
Content-Type: text/plain

Hello Bob
-------7443rules$

9.6. Nickname in a conference information document

Figure 7 depicts two user elements in a conference information document both having the nickname element with a nickname string.
<?xml version="1.0" encoding="UTF-8"?>
<conference-info
 xmlns="urn:ietf:params:xml:ns:conference-info"
 entity="sip:chatroom22@chat.example.com"
 state="full" version="1">
 <!-- CONFERENCE INFO -->
 <conference-description>
  <subject>MSRP nickname example</subject>
 </conference-description>
 <!-- CONFERENCE STATE -->
 <conference-state>
  <user-count>2</user-count>
 </conference-state>
 <!-- USERS -->
 <users>
  <user entity="sip:alice@atlanta.example.com" state="full">
   <nickname>Depressed Donkey</nickname>
  </user>
  <user entity="sip:bob@example.com" state="full">
   <nickname>Dopey Donkey</nickname>
  </user>
 </users>
</conference-info>

Figure 7: Nickname in a conference information document

10. IANA Considerations

10.1. New MSRP Method

This specification defines a new MSRP method to be added to the
Methods sub-registry of the Message Session Relay Protocol (MSRP)
Parameters registry:

NICKNAME

See section Section 7 for details.
10.2. New MSRP Header

This specification defines a new MSRP header to be added to the Header Field sub-registry of the Message Session Relay Protocol (MSRP) Parameters registry:

Use-Nickname

See Section 7 for details.

10.3. New MSRP Status Codes

This specification defines three new MSRP status codes to be added to the Status-Code sub-registry of the Message Session Relay Protocol (MSRP) parameters registry.

The 404 status code indicates the failure to resolve the recipient URI in the To header field of the Message/CPIM wrapper in the SEND request, e.g., due to an unknown recipient. See Section 6.2 for details.

The 423 response indicates a failure in allocated the requested NICKNAME. This can be caused by a malformed NICKNAME request (e.g., no Use-Nickname header field), an already allocated nickname, or a policy that prevents the sender to use nicknames. See Section 7 for details.

The 428 status code indicates that the recipient of a SEND request does not support private messages. See section Section 6.2 for details.

Table 1 summarizes the IANA registration data with respect to new MSRP status codes:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>404</td>
<td>Failure to resolve recipient’s URI</td>
<td>RFC XXXX</td>
</tr>
<tr>
<td>423</td>
<td>Unable to allocate requested nickname</td>
<td>RFC XXXX</td>
</tr>
<tr>
<td>428</td>
<td>Private messages not supported</td>
<td>RFC XXXX</td>
</tr>
</tbody>
</table>

Table 1: New status codes
10.4. New SDP Attribute

This specification defines a new media-level attribute in the Session Description Protocol (SDP) Parameters registry. The registration data is as follows:

Contact: Miguel Garcia <miguel.a.garcia@ericsson.com>
Phone: +34 91 339 1000
Attribute name: chatroom
Long-form attribute name: Chat Room
Type of attribute: media level only
This attribute is not subject to the charset attribute
Description: This attribute identifiers support and local policy allowance for a number of chatroom related functions
Specification: RFC XXXX

See section Section 8 for details.

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 deem to expose any security threat. Nevertheless, if a participant wants to avoid eavesdropping from non authorized entities, it should send those messages a TLS [RFC5246] transport connection, as allowed by MSRP.

Nicknames will be used to show the appearances of the participants of the conference. A successful take over of a nickname from a participant might lead to private messages to be sent to the wrong destination. The recipient’s URI will be different from the URI associated to the original owner of the nickname, but the sender might not notice this. To avoid take overs the MSRP switch MUST make sure that a nickname is unique inside a chat room. Also the security consideration for any authenticated identity mechanisms used to validate the SIP AOR will apply to this document as well. If a nickname can be reserved if it previously has been used by another
participant in the chat room, is up to the policy of the chat room.

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, Matt Lepinski, Mary Barnes, Ben Campbell, Paul Kyzivat, Adrian Georgescu, and Nancy Greene 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