Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 1, 2014

P. Saint-Andre
Cisco Systems, Inc.
S. Ibarra
AG Projects
S. Loreto
Ericsson

September 28, 2013

Interworking between the Session Initiation Protocol (SIP) and the Extensible Messaging and Presence Protocol (XMPP): Groupchat
draft-ietf-stox-groupchat-01

Abstract

This document defines a bidirectional protocol mapping for the exchange of instant messages in the context of a multiparty chat session among users of the Session Initiation Protocol (SIP) and users of the Extensible Messaging and Presence Protocol (XMPP). Specifically, this document defines a mapping between the SIP-based Message Session Relay Protocol (MSRP) and the XMPP Multi-User Chat (MUC) extension.

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 April 1, 2014.

Table of Contents

1. Introduction

Both the Session Initiation Protocol (SIP) [RFC3261] and the Extensible Messaging and Presence Protocol (XMPP) [RFC6120] can be used for the purpose of multiparty text chat over the Internet. To ensure interworking between these technologies, it is important to define bidirectional protocol mappings.

The architectural assumptions underlying such protocol mappings are provided in [I-D.ietf-stox-core], including mapping of addresses and error conditions. This document specifies mappings for multiparty text chat sessions (often called "groupchat"); specifically, this document defines a mapping between the XMPP Multi-User Chat (MUC) extension [XEP-0045] and SIP-based multiparty chat using Message Session Relay Protocol [RFC4975] as specified in [I-D.ietf-simple-chat].

Both MUC and MSRP contain a large set of features, such as the ability to administer rooms, kick and ban users, reserve a nickname within a room, change room subject, enable room moderation, and destroy the room. This document covers only a basic subset of groupchat features: joining a room, establishing or changing (but not permanently registering) a room nickname, modifying presence information within the room, sending a message to all participants, sending a private message to a single participant, inviting another user to the room, and leaving the room. Future documents might define mappings for additional features beyond this set.

The discussion venue for this document is the mailing list of the STOX WG; visit https://www.ietf.org/mailman/listinfo/stox for subscription information and discussion archives.

2. Terminology

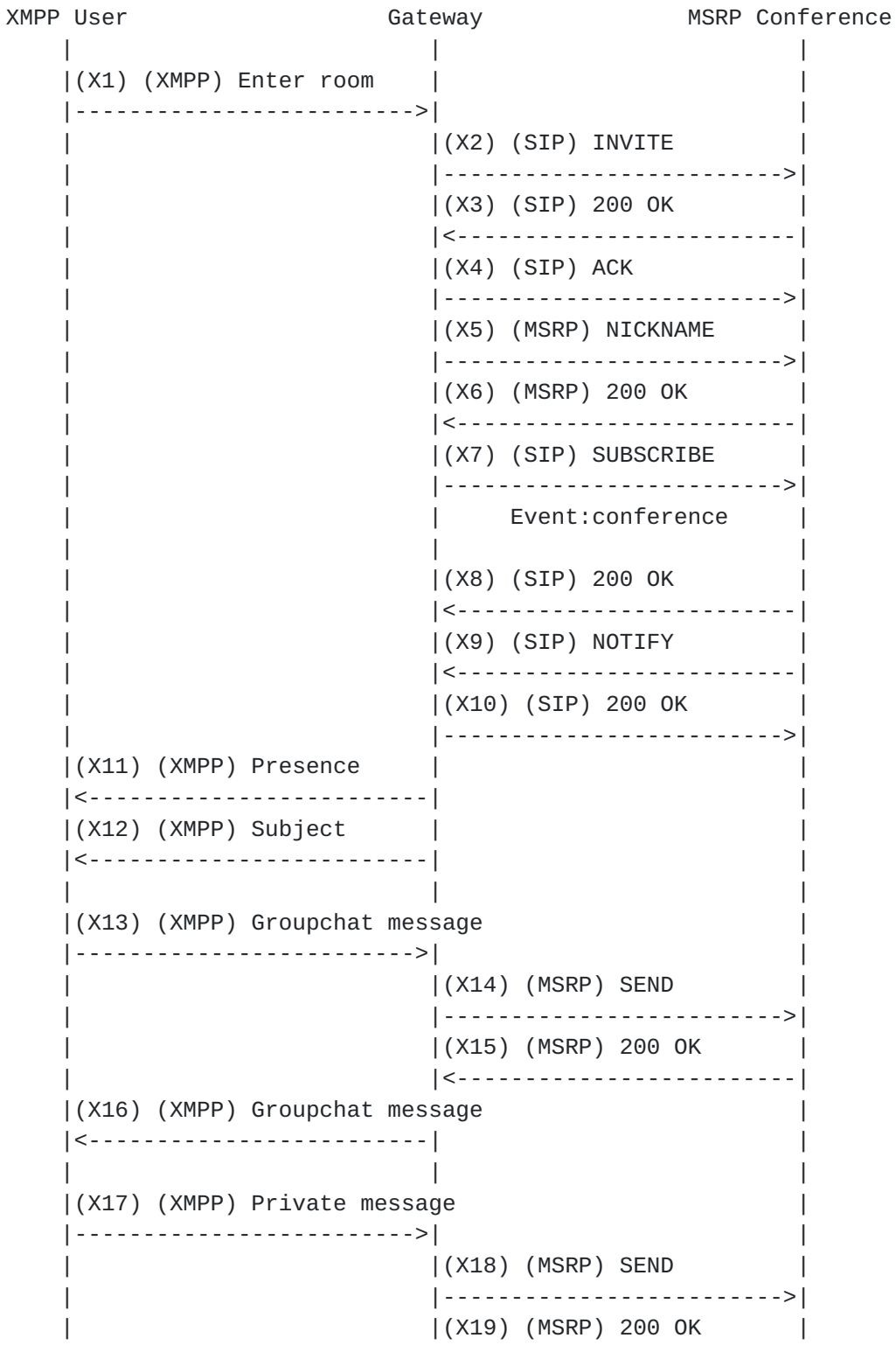
The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

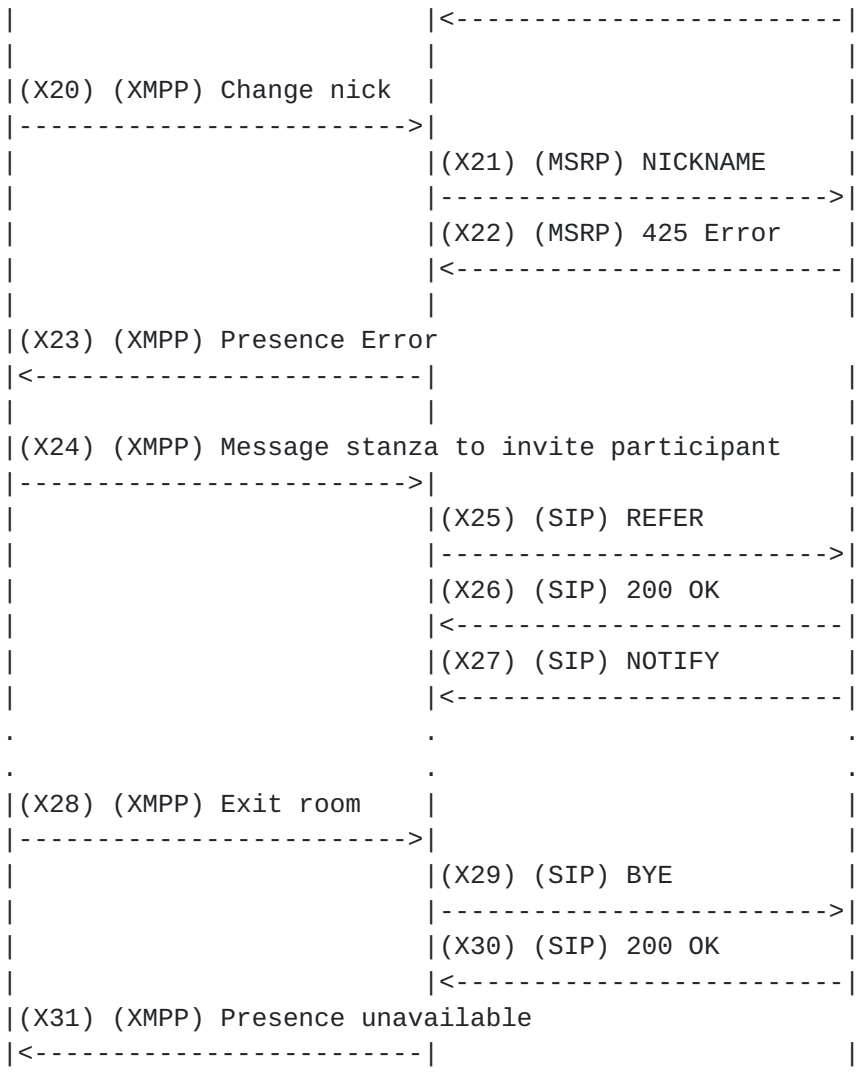
A number of technical terms used here are defined in [RFC3261], [RFC4975], [RFC6120], and [XEP-0045]. The term "JID" is short for "Jabber ID".

3. XMPP MUC to MSRP Multi-party Messaging Session

This section describes how to map an XMPP MUC session to an MSRP

Multi-party Messaging session. The following diagram outlines the overall protocol flow of a sample session, which includes some optional exchanges (such as sending messages, changing nickname, and inviting another user).





Detailed protocol flows and mappings are provided in the following sections.

3.1. Enter Room

As defined in the XMPP Multi-User Chat (MUC) specification [XEP-0045], when an XMPP user (say, "juliet@example.com") wants to join a groupchat room (say, "verona@chat.example.org"), she sends a directed <presence/> stanza [RFC6121] to that chat room. In her request she also specifies the nickname she wants to use within the room (say, "JuliC"); in XMPP this room nickname is the resourcepart of an occupant JID (thus "verona@chat.example.org/JuliC"). The joining client signals its ability to speak the multi-user chat protocol by including in the initial presence stanza an empty <x/> element qualified by the 'http://jabber.org/protocol/muc' namespace.

Example 1: Juliet enters room (X1)

```
|  <presence from='juliet@example.com/balcony'
|            to='verona@chat.example.org/JuliC'>
|    <x xmlns='http='http://jabber.org/protocol/muc'/>
|  </presence>
```

Upon receiving such a presence stanza, the XMPP server to which Juliet has connected needs to determine the identity of the domainpart in the 'to' address, which it does by following the procedures discussed in [I-D.ietf-stox-core]. Here we assume that the XMPP server has determined the domain is serviced by a SIMPLE server, that it contains or has available to it an XMPP-SIMPLE gateway or connection manager (which enables it to speak natively to SIMPLE servers), and that it hands off the presence stanza to the XMPP-SIMPLE gateway.

Because a multi-user chat service accepts the presence stanza shown above as a request to enter a room, the XMPP-to-SIP gateway transforms it into a SIP INVITE request.

Example 2: SIP mapping of room join (X2)

```
|  INVITE sip:verona@chat.example.org SIP/2.0
|  To: <sip:verona@chat.example.org>
|  From: "Juliet" <sip:juliet@example.com>
|  Contact: <sip:juliet@example.com>;gr=balcony
|  Call-ID: 711609sa
|  Content-Type: application/sdp
|  Content-Length: 182
|
|  c=IN IP4 x2s.example.org
|  m=message 7654 TCP/MSRP *
|  a=accept-types:text/cpim
|  a=accept-wrapped-types:text/plain text/html
|  a=path:msrp://x2s.example.com:7654/jshA7weztas;tcp
|  a=chatroom
```

Here the Session Description Protocol offer specifies the MSRP-aware XMPP-to-SIP gateway on the XMPP side as well as other particulars of the session.

There is no direct mapping for the MSRP URIs. In fact MSRP URIs identify a session of instant messages at a particular device; they are ephemeral and have no meaning outside the scope of that session. The authority component of the MSRP URI MUST contain the XMPP-to-SIP gateway hostname or numeric IP address and an explicit port number.

As specified in [I-D.ietf-stox-core], the mapping of XMPP syntax elements to SIP and [RFC4566] syntax elements is as shown in the following table.

Table 1: Message syntax mapping from XMPP to SIP/SDP

XMPP Element or Attribute	SIP Header or SDP Contents
from	From
<thread/>	Call-ID
to (without the /nick)	To

Here we assume that the MSRP conference server accepts the session establishment. It includes the 'isfocus' and other relevant feature tags in the Contact header field of the response. The MSRP conference server also includes an answer session description that acknowledges the choice of media and contains the extensions specified in [I-D.ietf-simple-chat].

Example 3: Chat room accepts session establishment (X3)

```
|  SIP/2.0 200 OK
|  From: <sip:verona@chat.example.org>
|  To: "Juliet" <sip:juliet@example.com>;tag=786
|  Call-ID: 711609sa
|  Contact: <sip:verona@chat.example.org;transport=tcp>;isfocus
|  Content-Type: application/sdp
|  Content-Length: 214
|
|  c=IN IP4 example.org
|  m=message 12763 TCP/MSRP *
|  a=chatroom:nickname private-messages
|  a=accept-types:message/cpim
|  a=accept-wrapped-types:text/plain text/html
|  a=path:msrp://s2x.example.org:12763/kjhd37s2s20w2a;tcp
```

Upon receiving such a response, the SIMPLE server or associated SIP-to-XMPP gateway sends a SIP ACK to the MSRP conference server on behalf of the joining user.

Example 4: Gateway sends ACK to MSRP conference server (X4)

```
|  ACK sip:verona@chat.example.org SIP/2.0
|  To: <sip:verona@chat.example.org>;tag=087js
|  From: "Juliet" <sip:juliet@example.com>;tag=786
|  Call-ID: 711609sa
```

3.2. Set Nickname

If the chat room server accepted the session, the SIMPLE server or associated SIP-to-XMPP gateway MUST set up the nickname as received in the presence stanza (i.e., the resourcepart of the 'to' address, such "JuliC" in "verona@chat.example.org/JuliC"). The nickname is set up using the extension specified in [I-D.ietf-simple-chat].

Example 5: Gateway sets up nickname (X5)

```
|  MSRP a786hjs2 NICKNAME
|  To-Path: msrp://s2x.example.org:12763/kjhd37s2s20w2a;tcp
|  From-Path: msrp://x2s.example.com:7654/jshA7weztas;tcp
|  Use-Nickname: "JuliC"
|  -------a786hjs2
```

The MSRP conference server analyzes the existing allocation of nicknames, accepts the nickname proposal, and answers with a 200 response.

Example 6: MSRP conference accepts nickname proposal (X6)

```
|  MSRP a786hjs2 200 OK
|  To-Path: msrp://x2s.example.com:7654/jshA7weztas;tcp
|  From-Path: msrp://s2x.example.org:12763/kjhd37s2s20w2a;tcp
|  -------a786hjs2
```

This section assumes that the nickname request is successful. The error flow resulting from a nickname conflict is described under Section 3.6.

3.3. Conference Subscription

If the nickname request is successful, the XMPP-to-SIP gateway then formally subscribes to the MSRP conference on behalf of the XMPP user.

Example 7: Gateway subscribes to the MSRP conference (X7)

```
|  SUBSCRIBE sip:verona@chat.example.org SIP/2.0
|  To: <sip:verona@chat.example.org>
|  From: "Juliet" <sip:juliet@example.com>
|  Contact: <sip:juliet@example.com>;gr=balcony
|  Call-ID: hsuKZGkjQewg6IJI8.PD5rRAvbuTYLR-
|  Event: conference
|  Expires: 600
|  Accept: application/conference-info+xml
|  Allow-Events: conference
|  Content-Length: 0
```

The conference server will accept or reject the request based on local policy.

Example 8: MSRP conference server accepts subscription request (X8)

```
|  SIP/2.0 200 OK
|  To: <sip:verona@chat.example.org>
|  From: "Juliet" <sip:juliet@example.com>
|  Call-ID: hsuKZGkjQewg6IJI8.PD5rRAvbuTYLR-
|  Contact: <sip:verona@chat.example.org;transport=tcp>;isfocus
|  Allow: SUBSCRIBE, NOTIFY, PRACK, INVITE, ACK, BYE,
|         CANCEL, UPDATE, MESSAGE, REFER
|  Expires: 600
|  Content-Length: 0
```

3.4. Presence Broadcast

If the MSRP conference service accepts the request to enter a room, the XMPP user expects to receive back presence information from all the existing occupants of the room. So the XMPP-to-SIP gateway MUST subscribe to the Conference Event package [RFC4575] on the MSRP conference server. When the subscription is completed the MSRP conference server sends to the XMPP-to-SIP gateway a NOTIFY containing the presence information of all the existing occupants, represented using the [RFC4575] format.

Example 9: MSRP conference sends presence information (X9)

```
|  NOTIFY sip:verona@chat.example.org SIP/2.0
|  To: "Juliet" <sip:juliet@example.com>;gr=balcony
|  From: <sip:verona@chat.example.org>;tag=a3343df32
|  Call-ID: k3l43id034kserereee
|  Event: conference
|  Subscription-State: active;expires=3600
|  Content-Type: application/conference-info+xml
|  Content-Length: ...
|
|  <conference-info version="0" state="full"
|      entity="sip:3402934234@chat.example.org">
|    <conference-description>
|      <subject>Today in Verona</subject>
|      <conf-uris>
|        <entry>
|          <uri>tel:+18882934234</uri>
|        </entry>
|      </conf-uris>
|    </conference-description>
|    <users>
|      <user entity="sip:verona@chat.example.org;gr=Romeo"
|            state="full">
|        <display-text>Romeo</display-text>
|        <roles>
|          <entry>participant</entry>
|        </roles>
|      </user>
|      <user entity="sip:verona@chat.example.org;gr=Ben"
|            state="full">
|        <display-text>Ben</display-text>
|        <roles>
|          <entry>participant</entry>
|        </roles>
|      </user>
|      <user entity="sip:verona@chat.example.org;gr=JuliC"
|            state="full">
|        <display-text>JuliC</display-text>
|        <roles>
|          <entry>participant</entry>
|        </roles>
|      </user>
|    </users>
|  </conference-info>
```

The following table shows the syntax mapping from the RFC 4575 payload to the XMPP participant list. (Mappings for elements not

mentioned are undefined.)

Table 2: Participant list mapping

RFC 4575 Element	XMPP Element or Attribute
conference-info entity	room JID
conference subject	room subject
user entity	participant bare JID
user display-text / nickname	participant nickname
endpoint entity	participant full JID

Upon receiving such a response, the SIP-to-XMPP gateway MUST send a 200 OK to the MSRP conference server and translate the participant list into a series of XMPP presence stanzas.

Example 10: XMPP mapping of chatroom presence (F11)

```
|  <presence from='verona@chat.example.org/Romeo'
|            to='juliet@example.com/balcony'>
|    <x xmlns='http://jabber.org/protocol/muc#user'>
|      <item affiliation='none' role='participant'/>
|    </x>
|  </presence>

|  <presence from='verona@chat.example.org/Ben'
|            to='juliet@example.com/balcony'>
|    <x xmlns='http://jabber.org/protocol/muc#user'>
|      <item affiliation='none' role='participant'/>
|    </x>
|  </presence>

|  <presence from='verona@chat.example.org/JuliC'
|            to='juliet@example.com/balcony'>
|    <x xmlns='http://jabber.org/protocol/muc#user'>
|      <item affiliation='none' role='participant'/>
|      <status code='110'/>
|    </x>
|  </presence>
```

If the NOTIFY included a subject, the gateway converts that into a separate XMPP message.

Example 11: XMPP mapping of chatroom subject (F12)

```
|  <message from='verona@chat.example.com/mayor'
|           to='juliet@example.com/balcony'
|           id='mbh2vd68'>
|    <subject>Today in Verona</subject>
|  </message>
```

The mapping of SIP and [RFC4575] payload syntax elements to XMPP syntax elements is as shown in the following table. (Mappings for elements not mentioned are undefined.)

Table 3: Message syntax mapping from SIP to XMPP

SIP Header or RFC4575 Contents	XMPP Element or Attribute
<user entity=...>	from
Call-ID	thread
To + / <display-text>	to
roles	role
'none'	affiliation

3.5. Exchange Messages

Once the user has joined the chatroom, the user can exchange an unbounded number of messages, both public and private.

The mapping of XMPP syntax elements to MSRP syntax elements is as shown in the following table. (Mappings for elements not mentioned are undefined.)

Table 4: Message syntax mapping from XMPP Message to MSRP

XMPP Element or Attribute	CPIM Header
to	To
from	From
<body/>	body of the SEND request

3.5.1. Send a Message to All Occupants

When Juliet wants to sends a message to all other occupants in the room, she sends a message of type "groupchat" to the <room@service> itself (in our example, <verona@chat.example.org>).

The following examples show an exchange of a public message.

Example 12: Juliet sends message to all occupants (X13)

```
|  <message from='juliet@example.com/balcony'
|           to='verona@chat.example.org'
|           type='groupchat'
|           id='lzfed24s'>
|        <body>Who knows where Romeo is?</body>
|  </message>
```

Upon receiving such a message, the XMPP-to-SIP gateway MUST translate it into an MSRP SEND message.

Example 13: Gateway maps XMPP message to MSRP (X14)

```
|  MSRP a786hjs2 SEND
|  To-Path: msrp://s2x.example.org:12763/kjhd37s2s20w2a;tcp
|  From-Path: msrp://x2s.example.com:7654/jshA7weztas;tcp
|  Message-ID: 87652491
|  Byte-Range: 1-*/*
|  Content-Type: message/cpim
|
|  To: <sip:verona@chat.example.org>
|  From: "Juliet" <sip:juliet@example.com>
|  DateTime: 2008-10-15T15:02:31-03:00
|  Content-Type: text/plain
|
|  Who knows where Romeo is?
|  -------a786hjs2$
```

Upon receiving the SEND request, if the request either contains a Failure-Report header field value of "yes" or does not contain a Failure-Report header at all, the MSRP conference server MUST immediately generate and send a response.

Example 14: MSRP conference returns 200 OK (X15)

```
|  MSRP d93kswow 200 OK
|  To-Path: msrp://x2s.example.com:7654/jshA7weztas;tcp
|  From-Path: msrp://s2x.example.org:12763/kjhd37s2s20w2a;tcp
|  -------d93kswow$
```

Since an XMPP MUC room could be moderated and an XMPP user cannot be sure whether her message has been accepted or not without receiving it back from the server, [XEP-0045] states that the sender needs to receive the same message it has generated. So in this scenario the XMPP-to-SIP gateway has to reflect the message back to the sender.

This prodedure only applies to XMPP endpoints.

Example 15: Gateway reflects message to XMPP user (X16)

```
|  <message from='verona@chat.example.org/JuliC'
|           to='verona@chat.example.org'
|           type='groupchat'
|           id='ix51z73m'>
|        <body>Who knows where Romeo is?</body>
|  </message>
```

3.5.2. Send a Private Message

Since each occupant has a unique JID, Juliet can send a "private message" to a selected occupant through the service by sending a message to the user's occupant JID. The XMPP message type SHOULD be "chat" and MUST NOT be "groupchat", but MAY be left unspecified.

If the XMPP-to-SIP gateway has support for private messaging it MUST advertise that fact by adding a "private-messages" value to the a=chatroom SDP attribute it sends to the MSRP conference server, as specified in [I-D.ietf-simple-chat].

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

The following examples show an exchange of a private message.

Example 16: Juliet sends private message (X17)

```
|  <message from='juliet@example.com/balcony'
|           to='verona@chat.example.org/Romeo'
|           type='chat'
|           id='6sfln45q'>
|        <body>O Romeo, Romeo! wherefore art thou Romeo?</body>
|  </message>
```

Upon receiving such a message, the XMPP-to-SIP gateway MUST translate it into an MSRP SEND message.

Example 17: Gateway maps private message from XMPP to MSRP (X18)

```
|  MSRP a786hjs2 SEND
|  To-Path: msrp://s2x.example.org:12763/kjhd37s2s20w2a;tcp
|  From-Path: msrp://x2s.example.com:7654/jshA7weztas;tcp
|  Message-ID: 87652491
|  Byte-Range: 1-*/*
|  Content-Type: message/cpim
|
|  To: <sip:verona@chat.example.org>;gr=Romeo
|  From: <sip:juliet@example.org>;gr=balcony
|  DateTime: 2008-10-15T15:02:31-03:00
|  Content-Type: text/plain
|
|  O Romeo, Romeo! wherefore art thou Romeo?
|  -------a786hjs2$
```

After acknowledging the message by sending a SIP 200 OK (step X19, not shown), the MSRP conference server is responsible for sending the message to the intended recipient (not shown in the protocol flow). When doing so, it MUST modify the "From" header to the sender's address within the chatroom.

Example 18: MSRP conference sends private message to SIP user

```
|  MSRP a786hjs2 SEND
|  To-Path: msrp://s2x.example.org:12763/kjhd37s2s20w2a;tcp
|  From-Path: msrp://x2s.example.com:7654/jshA7weztas;tcp
|  Message-ID: 87652491
|  Byte-Range: 1-*/*
|  Content-Type: message/cpim
|
|  To: <sip:romeo@example.com>
|  From: <sip:verona@chat.example.com>;gr=JuliC
|  DateTime: 2008-10-15T15:02:31-03:00
|  Content-Type: text/plain
|
|  O Romeo, Romeo! wherefore art thou Romeo?
|  -------a786hjs2$
```

3.6. Change Nickname

The XMPP user might want to change her nickname. She can do so by sending an updated presence stanza to the room, containing a new nickname.

Example 19: Juliet changes her nickname (X20)

```
|  <presence from='juliet@example.com/balcony'
|            to='verona@chat.example.org/CapuletGirl'/>
```

So far we have assumed that the requested nickname did not conflict with any existing nicknames. The following text describes the handling of a nickname conflict.

The MSRP conference server analyzes the existing allocation of nicknames, and detects that the nickname proposal is already provided to another participant. In this case the MSRP conference server answers with a 425 response.

Example 20: MSRP conference does not accept nickname proposal (X22)

```
|  MSRP a786hjs2 425 Nickname usage failed
|  To-Path: msrp://x2s.example.com:7654/jshA7weztas;tcp
|  From-Path: msrp://s2x.example.org:12763/kjhd37s2s20w2a;tcp
|  -------a786hjs2
```

Upon receiving such a response, the SIP-to-XMPP gateway SHOULD translate it into an XMPP presence stanza of type "error" specifying a <conflict/> error condition (which implies that the XMPP client will then need to choose another nickname and repeat the process of joining).

Example 21: Conflict error for nickname (X23)

```
|  <presence from='verona@chat.example.org/JuliC'
|            to='juliet@example.com/balcony'
|            type='error'>
|    <x xmlns='http='http://jabber.org/protocol/muc'/>
|    <error type='cancel' by='verona@chat.example.org'>
|      <conflict xmlns='urn:ietf:params:xml:ns:xmpp-stanzas'/>
|    </error>
|  </presence>
```

Alternatively, the gateway might generate a new nickname request on behalf of the XMPP user, thus shielding the XMPP client from handling the conflict error.

3.7. Invite Another User to a Room

In XMPP there are two methods for inviting another user to a room: direct invitations [XEP-0249] (sent directly from the user's real JID outside the room to the invitee's real JID) and mediated invitations (sent through the room from the user's occupant JID to the invitee's

JID). In this document we cover mediated invitations only.

For example, if Juliet decides to invite Benvolio to the room, she sends a message stanza with an invite and Benvolio's JID (which could be his real JID or an occupant JID in another room).

Example 22: Juliet invites Benvolio to the room (X24)

```
|  <message from='juliet@example.com/balcony'
|           id='nzd143v8'
|           to='verona@chat.example.org'>
|    <x xmlns='http://jabber.org/protocol/muc#user'>
|      <invite to='benvolio@example.com'/>
|    </x>
|  </message>
```

The SIP-to-XMPP gateway then sends a SIP REFER request to the MSRP conference server indicating who needs to be invited in the Refer-To header, as per [RFC4579] (sec 5.5)

Example 23: SIP mapping of invite (X25)

```
|  REFER sip:verona@chat.example.com SIP/2.0
|  To: <sip:verona@chat.example.com>
|  From: "Juliet" <sip:juliet@example.com>;tag=5534562
|  Call-ID: 849392fklgl43
|  Contact: <sip:juliet@example.com>;gr=balcony
|  Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, SUBSCRIBE, NOTIFY
|  Accept: message/sipfrag
|  Refer-To: <sip:benvolio@example.com>
|  Supported: replaces
|  Content-Length: 0
```

The MSRP conference server then acknowledges the SIP REFER request with a 200 OK response (step X25, not shown).

The progress of the invitation will be tracked by the received NOTIFY requests as per [RFC3515].

Example 24: Progress notification for invitation (X27)

```
|  NOTIFY sip:juliet@example.com SIP/2.0
|  To: <sip:juliet@example.com>;tag=5534562
|  From: <sip:verona@chat.example.com>;tag=18747389
|  Call-ID: 849392fklgl43
|  Max-Forwards: 70
|  Event: refer
|  Subscription-State: active;expires=60
|  Contact: <sip:verona@chat.example.com;transport=tcp>;isfocus
|  Content-Type: message/sipfrag;version=2.0
|  Content-Length: ...
```

3.8. Exit Room

If Juliet decides to exit the chatroom, her client sends a directed presence stanza of type "unavailable" to the occupant JID she is currently using in the room (here <verona@chat.example.org/JuliC>).

Example 25: Juliet exits room (X28)

```
|  <presence from='juliet@example.com/balcony'
|            to='verona@chat.example.org/JuliC'
|            type='unavailable'/>
```

Upon receiving such a stanza, the XMPP-to-SIP gateway terminates the SIP session by sending a SIP BYE to the MSRP conference server and the MSRP conference server responds with a 200 OK (steps X29 and X30, not shown).

Juliet MAY include a custom exit message in the presence stanza of type "unavailable", in which case it SHOULD be broadcasted to other participants using the methods described above.

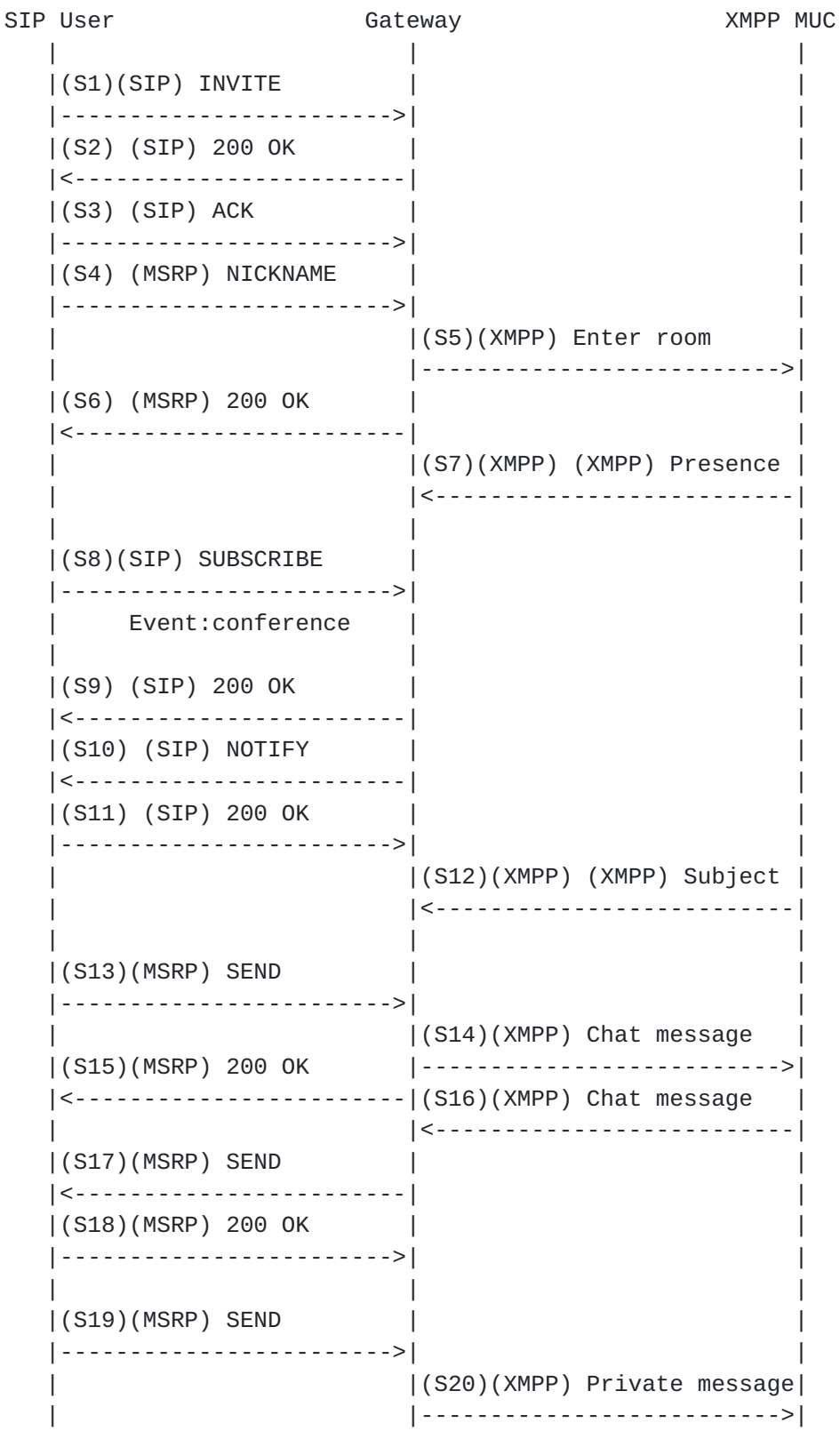
Example 26: Juliet exits the chatroom (F31)

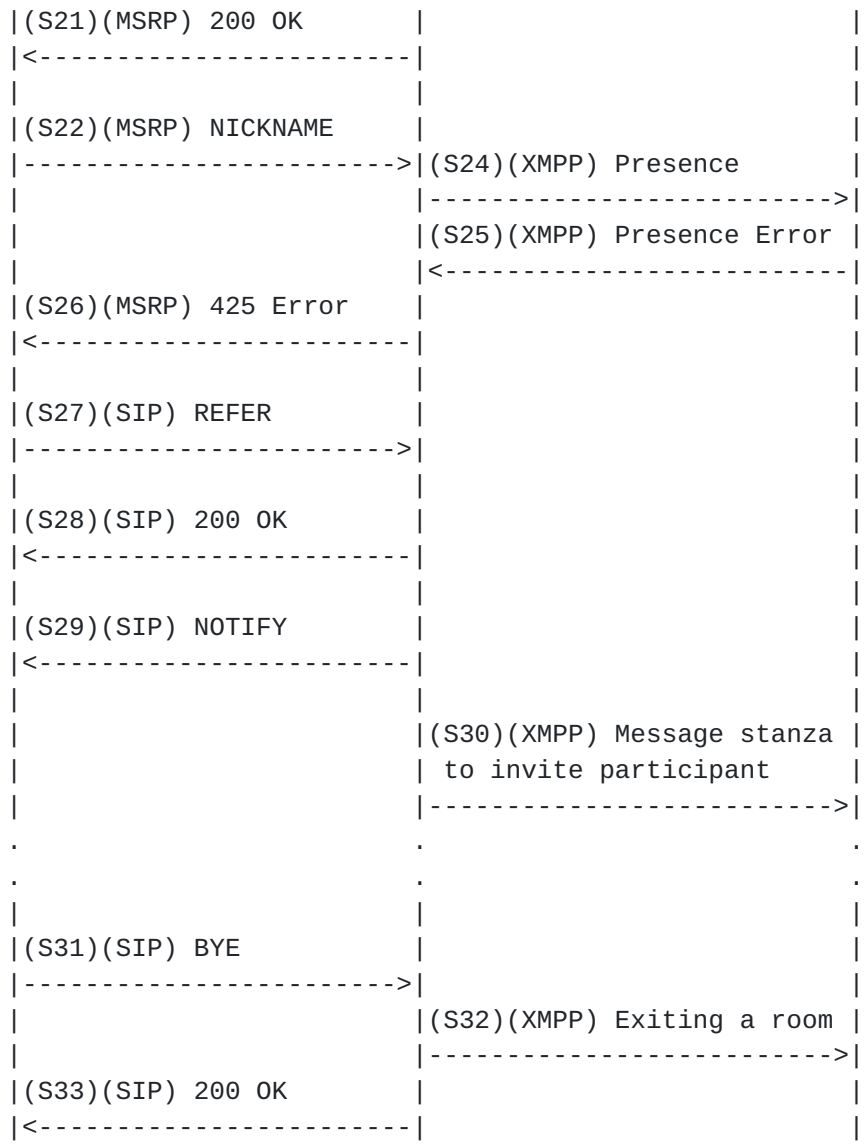
```
|  <presence from='juliet@example.com/balcony'
|            to='verona@chat.example.org/JuliC'
|            type='unavailable'>
|    <status>Time to go!</status>
|  </presence>
```

4. MSRP Multi-party Messaging Session to XMPP MUC

This section describes how to map a Multi-party Instant Message (IM) MSRP session to an XMPP Multi-User Chat (MUC) session. As before, the following diagram outlines the overall protocol flow of a sample

session, which includes some optional exchanges (such as sending messages, changing nickname, and inviting another user).





If the XMPP presence stanza is received before the SIP SUBSCRIBE dialog is established for the "conference" event, then the server SHOULD cache the participant list until the subscription is established and delivered in a SIP NOTIFY request.

4.1. Enter Room (Includes Set Nickname)

When the SIP user ("Romeo") wants to join a groupchat room ("Verona"), he first has to start the SIP session by sending out a SIP INVITE request containing an offered session description that includes an MSRP media line accompanied by a mandatory "path" and "chatroom" attributes. The MSRP media line is also accompanied by an "accept-types" attribute specifing support for a Message/CPIM top level wrapper for the MSRP message.

Example 27: SIP user starts session (S1)

```
|  INVITE sip:verona@chat.example.org SIP/2.0
|  To: <sip:verona@chat.example.org>
|  From: "Romeo" <sip:romeo@example.com>
|  Contact: <sip:romeo@example.com>;gr=orchard
|  Call-ID: 742510no
|  Content-Type: application/sdp
|  Content-Length: 163
|
|  c=IN IP4 s2x.example.net
|  m=message 7313 TCP/MSRP *
|  a=accept-types:message/cpim text/plain text/html
|  a=accept-wrapped-types:text/plain text/html
|  a=path:msrp://s2x.example.net:7313/ansp71weztas;tcp
|  a=chatroom
```

Upon receiving the INVITE, the SIP-to-XMPP gateway needs to determine the identity of the remote domain, which it does by following the procedures discussed in [I-D.ietf-stox-core]. Here we assume that the gateway server has determined that the remote domain is serviced by an XMPP server, that it contains or has available to it a SIP-to-XMPP gateway or connection manager (which enables it to speak natively to XMPP servers), and that it hands off the message to the gateway. If this succeeds, the gateway SHOULD answer successfuly with a SIP 200 OK (S2, not shown).

Implementations MAY wait until the nickname is set with an MSRP NICKNAME chunk before joining the XMPP MUC or MAY choose a temporary nickname (such as the SIP From header display name) and use it to join the room.

Example 28: SIP user acks session (S3)

```
|  SIP/2.0 200 OK
|  To: <sip:verona@chat.example.org>
|  From: "Romeo" <sip:romeo@example.com>
|  Contact: <sip:x2s.example.com;transport=tcp>;isfocus
|  Call-ID: 742510no
|  Content-Type: application/sdp
|
|  c=IN IP4 x2s.example.com
|  m=message 8763 TCP/MSRP *
|  a=accept-types:message/cpim text/plain text/html
|  a=accept-wrapped-types:text/plain text/html
|  a=path:msrp://x2s.example.com:8763/lkjh37s2s20w2a;tcp
|  a=chatroom:nickname private-messages
```

The SIP user then requests a nickname.

Example 29: MSRP user sets up nickname (S4)

```
|  MSRP a786hjs2 NICKNAME
|  To-Path: msrp://s2x.example.net:7313/ansp71weztas;tcp
|  From-Path: msrp://x2s.example.com:8763/lkjh37s2s20w2a;tcp
|  Use-Nickname: "Romeo"
|  -------a786hjs2
```

Upon receiving the MSRP NICKNAME request, the SIP-to-XMPP gateway is responsible for generating an XMPP presence stanza and sending it to the chatroom.

Example 30: Romeo enters XMPP chatroom (S5)

```
|  <presence from='romeo@example.com'
|            to='verona@chat.example.org/Romeo'>
|    <x xmlns='http='http://jabber.org/protocol/muc'/>
|  </presence>
```

If the room does not already contain another user with the requested nickname, the service accepts the access request. Thus if the gateway does not receive any stanza of type "error" specifying a <conflict/> error condition within a reasonable amount of time (e.g., 5 seconds), it MUST answer the MSRP nickname proposal with a 200 OK response (S6).

Example 31: Acknowledgement of join (S6)

```
|  MSRP a786hjs2 200 OK
|  To-Path: msrp://x2s.example.com:8763/lkjh37s2s20w2a;tcp
|  From-Path: msrp://s2x.example.net:7313/ansp71weztas;tcp
|  -------a786hjs2
```

Once the nickname has been set the user subscribes to the conference event.

Example 32: User subscribes to the conference event (S8)

```
|  SUBSCRIBE sip:verona@chat.example.org SIP/2.0
|  To: <sip:verona@chat.example.org>
|  From: "Romeo" <sip:romeo@example.com>
|  Contact: <sip:romeo@example.com>;gr=orchard
|  Call-ID: hsuKZGkjQewg6IJI8.PD5rRAvbuTYLR-
|  Event: conference
|  Expires: 600
|  Accept: application/conference-info+xml
|  Allow-Events: conference
|  Content-Length: 0
```

The gateway will accept or reject the request based on local policy.

Example 33: Gateway accepts subscription (S9)

```
|  SIP/2.0 200 OK
|  To: <sip:verona@chat.example.org>
|  From: "Romeo" <sip:romeo@example.com>
|  Call-ID: hsuKZGkjQewg6IJI8.PD5rRAvbuTYLR-
|  Contact: <sip:verona@chat.example.org;transport=tcp>;isfocus
|  Allow: SUBSCRIBE, NOTIFY, PRACK, INVITE, ACK, BYE,
|         CANCEL, UPDATE, MESSAGE, REFER
|  Expires: 600
|  Content-Length: 0
```

4.2. Presence Broadcast

If the multi-user chat service is able to add the SIP user to the room, it sends presence from all the existing occupants' room JIDs to the new occupant's full JID, including extended presence information about roles in an <x/> element.

Example 34: XMPP service sends presence from existing occupants to new occupant (S7)

```
|  <presence from='verona@chat.example.org/Romeo'
|            to='romeo@example.com'>
|    <x xmlns='http://jabber.org/protocol/muc#user'>
|      <item affiliation='none' role='participant'/>
|      <status='110'/>
|    </x>
|  </presence>

|  <presence from='verona@chat.example.org/Ben'
|            to='romeo@example.com'>
|    <x xmlns='http://jabber.org/protocol/muc#user'>
|      <item affiliation='none' role='participant'/>
|    </x>
|  </presence>

|  <presence from='verona@chat.example.org/JuliC'
|            to='romeo@example.com'>
|    <x xmlns='http://jabber.org/protocol/muc#user'>
|      <item affiliation='none' role='participant'/>
|    </x>
|  </presence>
```

Upon receiving these presence stanzas, if the MSRP conference server has already completed the subscription to the Conference Event package [RFC4575], the XMPP-to-SIP gateway MUST translate them into a SIP NOTIFY request containing the participant list (represented in the conference-info format specified in [RFC4575]).

Example 35: MSRP mapping of XMPP participant presence (S10)

```
|  NOTIFY sip:romeo@example.com SIP/2.0
|  To: <sip:romeo@example.com>;tag=43524545
|  From: <sip:verona@chat.example.org>;tag=a3343df32
|  Call-ID: k3l43id034kserерff
|  Event: conference
|  Subscription-State: active;expires=3600
|  Content-Type: application/conference-info+xml
|  Content-Length: ...
|
|  <conference-info version="0" state="full"
|      entity="sip:verona@chat.example.org">
|    <conference-description>
|      <subject>Today in Verona</subject>
|      <conf-uris>
|        <entry>
|          <uri>tel:+18882934234</uri>
|          <uri>sip:verona@chat.example.org</uri>
|        </entry>
|      </conf-uris>
|   </conference-description>
|   <users>
|     <user entity="sip:verona@chat.example.org;gr=JuliC"
|           state="full">
|       <display-text>JuliC</display-text>
|       <roles>
|         <entry>participant</entry>
|       </roles>
|     </user>
|     <user entity="sip:verona@chat.example.org;gr=Ben"
|           state="full">
|       <display-text>Ben</display-text>
|       <roles>
|         <entry>participant</entry>
|       </roles>
|   </users>
|  </conference-info>
```

Because the "room roster" is communicated in XMPP by means of multiple presence stanzas (one for each participant) whereas the participant list is communicated in SIP by means of a single conference-info document, the SIP-to-XMPP gateway will need to keep track of the user's SIP URI and the mapping of that URI into an XMPP address; then, based on that mapping, it will need to determine when it has received a complete room roster from the MUC room, i.e., when it has received the in-room presence of the SIP user (which according to XEP-0045 is the last presence stanza received in the initial batch

sent after joining). Once that happens, the SIP-to-XMPP gateway can construct the conference-info document containing the complete participant list and send that to the SIP user.

4.3. Exchange Messages

Once the user has joined the chat room, the user can exchange an unbounded number of messages, both public and private.

The mapping of MSRP syntax elements to XMPP syntax elements SHOULD be as shown in the following table. (Mappings for elements not mentioned are undefined.)

Table 5: Message syntax mapping from MSRP Message to XMPP

CPIM Header	XMPP Element or Attribute
To	to
From	from
body of the SEND request	<body/>

4.3.1. Send a Message to All Occupants

When Romeo wants to send a message to all other occupants in the room, he sends an MSRP SEND request to <room@service> itself (i.e., <verona@chat.example.org> in our example).

The following examples show an exchange of a public message.

Example 36: Romeo sends a message to the chat room (S13)

```
|  MSRP a786hjs2 SEND
|  To-Path: msrp://s2x.example.net:7313/ansp71weztas;tcp
|  From-Path: msrp://x2s.example.com:8763/lkjh37s2s20w2a;tcp
|  Message-ID: 87652492
|  Byte-Range: 1-*/*
|  Content-Type: message/cpim
|
|  To: <sip:verona@chat.example.org>
|  From: "Romeo" <sip:romeo@example.com>;gr=orchard
|  DateTime: 2008-10-15T15:02:31-03:00
|  Content-Type: text/plain
|
|  Romeo is here!
|  -------a786hjs2$
```

Upon receiving the SEND request, if the request either contains a Failure-Report header field value of "yes" or does not contain a Failure-Report header at all, the SIP-to-XMPP gateway MUST immediately translate it into an XMPP message stanza (S14) and then generate and send an MSRP response (S15).

Example 37: XMPP mapping of message (S14)

```
|  <message from='romeo@example.com/orchard'
|           to='verona@chat.example.org'
|           type='groupchat'
|           id='8gbx1g4p'>
|    <body>Romeo is here!</body>
|  </message>
```

Example 38: MSRP response to public message (S15)

```
|  MSRP d93kswow 200 OK
|  To-Path: msrp://x2s.example.com:8763/lkjh37s2s20w2a;tcp
|  From-Path: msrp://s2x.example.net:7313/ansp71weztas;tcp
|  -------d93kswow$
```

Note well that the XMPP MUC room will reflect the sender's message back to all users, including the sender. In MSRP this reflected message is unnecessary. Therefore gateways are advised to maintain a cache and if the same stanza is received within a reasonable amount of time, assume is the reflected message and ignore it.

4.3.2. Send a Private Message

Romeo can send a "private message" to a selected occupant via the chat room service by sending a message to the occupant's room nickname.

The following examples show an exchange of a private message.

Example 39: Romeo sends a private message (S19)

```
|  MSRP a786hjs2 SEND
|  To-Path: msrp://s2x.example.net:7313/ansp71weztas;tcp
|  From-Path: msrp://x2s.example.com:8763/lkjh37s2s20w2a;tcp
|  Message-ID: 87652492
|  Byte-Range: 1-*/*
|  Content-Type: message/cpim
|
|  To: <sip:verona@chat.example.org>;gr=JuliC
|  From: "Romeo" <sip:romeo@example.com>;gr=orchard
|  DateTime: 2008-10-15T15:02:31-03:00
|  Content-Type: text/plain
|
|  I am here!!!
|  -------a786hjs2$
```

The MSRP conference is responsible for transforming the "From" address into an in-room address.

Example 40: MSRP handling of private message

```
|  MSRP a786hjs2 SEND
|  To-Path: msrp://s2x.example.net:7313/ansp71weztas;tcp
|  From-Path: msrp://x2s.example.com:8763/lkjh37s2s20w2a;tcp
|  Message-ID: 87652492
|  Byte-Range: 1-*/*
|  Content-Type: message/cpim
|
|  To: <sip:verona@chat.example.org>;gr=JuliC
|  From: <sip:verona@chat.example.org>;gr=Romeo
|  DateTime: 2008-10-15T15:02:31-03:00
|  Content-Type: text/plain
|
|  I am here!!!
|  -------a786hjs2$
```

Once the MSRP conference sends that message to the gateway, the gateway is responsible for translating it into XMPP syntax.

Example 41: XMPP mapping of private message (S20)

```
|  <message from='verona@chat.example.org/Romeo'
|           to='verona@chat.example.org/JuliC'
|           type='chat'
|           id='rg2ca9k7'/>
|    <body>I am here!!!</body>
|  </message>
```

4.4. Change Nickname

If Romeo decides to change his nickname within the room, he MUST send a new MSRP NICKNAME request. In fact modification of the nickname in MSRP is not different from the initial reservation and usage of a nickname.

Example 42: MSRP user changes nickname (S22)

```
|  MSRP a786hjs2 NICKNAME
|  To-Path: msrp://s2x.example.net:7313/ansp71weztas;tcp
|  From-Path: msrp://x2s.example.com:8763/lkjh37s2s20w2a;tcp
|  Use-Nickname: "montecchi"
|  -------a786hjs2
```

Upon receiving such a message, the SIP-to-XMPP gateway MUST translate it into an XMPP presence stanza.

Example 43: XMPP mapping of nickname change (S24)

```
|  <presence from='romeo@example.com'
|            to='verona@chat.example.org/montecchi'/>
```

The XMPP server will analyze the nickname allocation and determine if the requested nickname is available. In case the nickname is not available or not usable, the server will generate a presence stanza of type "error" specifying a <conflict/> error condition.

Example 44: XMPP conflict error for nickname (S25)

```
|  <presence from='verona@chat.example.org/Romeo'
|            to='romeo@example.com/orchard'
|            type='error'>
|    <x xmlns='http='http://jabber.org/protocol/muc'/>
|    <error type='cancel' by='verona@chat.example.org'>
|      <conflict xmlns='urn:ietf:params:xml:ns:xmpp-stanzas'/>
|    </error>
|  </presence>
```

Upon receiving this stanza, the XMPP-SIP gateway will reply to the NICKNAME request with code 425

Example 45: Gateway translates XMPP nickname conflict to MSRP error code (S26)

```
|  MSRP a786hjs2 425 Nickname usage failed
|  To-Path: msrp://s2x.example.net:7313/ansp71weztas;tcp
|  From-Path: msrp://x2s.example.com:8763/lkjh37s2s20w2a;tcp
```

```
|  -------a786hjs2
```

4.5. Invite Another User to a Room

If a SIP user wants to invite another user to join the conference he will send a REFER request indicating who needs to be invited in the Refer-To header, as per Section 5.5 of [RFC4579].

Example 46: Inviting a XMPP participant (S27)

```
|  REFER sip:verona@chat.example.org SIP/2.0
|  To: <sip:verona@chat.example.org>
|  From: "Romeo" <sip:romeo@example.com>;tag=5534562
|  Call-ID: 849392fklgl43
|  Contact: <sip:romeo@example.com>;gr=orchard
|  Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, SUBSCRIBE, NOTIFY
|  Accept: message/sipfrag
|  Refer-To: <sip:benvolio@example.com>
|  Supported: replaces
|  Content-Length: 0
```

The XMPP-SIP gateway then acknowledges the SIP REFER request with a 200 OK response (step S28, not shown).

The gateway will then send a NOTIFY requests as per [RFC3515] indicating that the invitation is is progress. Since there is no way know the progress of the invitation until the user has joined, implementations are advised to terminate the REFER dialog subscription with the first NOTIFY, with a status code of 100 Trying.

Example 47: Progress notification for invitation (S29)

```
|  NOTIFY sip:romeo@example.com SIP/2.0
|  To: <sip:romeo@example.com>;tag=5534562
|  From: <sip:verona@chat.example.org>;tag=18747389
|  Call-ID: 849392fklgl43
|  Event: refer
|  Subscription-State: terminated;reason=noresource
|  Contact: <sip:verona@chat.example.com;transport=tcp>;isfocus
|  Content-Type: message/sipfrag;version=2.0
|  Content-Length: 20
|
|  SIP/2.0 100 Trying
```

4.6. Exit Room

If Romeo decides to exit the chat room, his client sends a SIP BYE to the <verona@chat.example.org> chat room.

Example 48: Romeo terminates session (S27)

```
|  BYE sip:verona@chat.example.org SIP/2.0
|  Max-Forwards: 70
|  From: "Romeo" <sip:romeo@example.net>;tag=786
|  To: <sip:verona@chat.example.org>;tag=534
|  Call-ID: 742510no
|  Cseq: 1 BYE
|  Content-Length: 0
```

Upon receiving the SIP BYE, the SIP-to-XMPP gateway translates it into a presence stanza (F19) and sends it to the XMPP MUC room service. Then the SIP-to-XMPP gateway responds with a 200 OK to the MSRP user.

Example 49: Romeo exits chatroom (S28)

```
|  <presence from='romeo@example.com'
|            to='verona@chat.example.org/Romeo'
|            type='unavailable'/>
```

5. Handling of Nicknames and Display Names

Fundamental rules for mapping addresses between XMPP and SIP are provided in [I-D.ietf-stox-core]. However, chatrooms include a more specialized, unique identifier for each participant in a room, called a nickname. Implementations are strongly encouraged to apply the rules for preparation and comparison of nicknames specified in [I-D.ietf-precis-nickname].

In addition to nicknames, some groupchat implementations also include display names (which might or might not be different from users' nicknames). A display name need not be unique within the context of a room but instead simply provides a user-friendly name for a participant.

In SIP, the nickname is the value of the XCON 'nickname' attribute of the <user/> element [RFC6501] and the display name is the XML character data of the conference-info <display-text/> element [RFC4575]. In XMPP, the nickname is the value of the resourcepart of the occupant JID [XEP-0045] and the display name is the XML character data of the <nick/> element [XEP-0172].

In practice, the <display-text/> element is treated as canonical in SIP implementations, and the <nick/> element is rarely used in XMPP implementations. Therefore, for display purposes SIP implementations ought to use the <display-text/> element (not the XCON 'nickname' attribute) and XMPP implementations ought to use the resourcepart of the occupant JID (not the character data of the <nick/> element).

If there is a conflict between the SIP nickname and the XMPP nickname, the SIP-to-XMPP or XMPP-to-SIP gateway is responsible for adjusting the nickname to avoid the conflict and for informing the SIP or XMPP client of the unique nickname used to join the chatroom.

6. IANA Considerations

This document requests no actions of the IANA.

7. Security Considerations

The security considerations of [RFC3261], [RFC4975], [RFC6120], [I-D.ietf-stox-core], [I-D.ietf-simple-chat], and [XEP-0045] apply.

8. References

8.1. Normative References

[I-D.ietf-precis-nickname]
Saint-Andre, P., "Preparation and Comparison of Nicknames", draft-ietf-precis-nickname-06 (work in progress), July 2013.

[I-D.ietf-simple-chat]
Niemi, A., Garcia-Martin, M., and G. Sandbakken, "Multi-party Instant Message (IM) Sessions Using the Message Session Relay Protocol (MSRP)", draft-ietf-simple-chat-18 (work in progress), January 2013.

[I-D.ietf-stox-core]
Saint-Andre, P., Houri, A., and J. Hildebrand, "Interworking between the Session Initiation Protocol (SIP) and the Extensible Messaging and Presence Protocol (XMPP): Core", draft-ietf-stox-core-05 (work in progress), September 2013.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[RFC4579] Johnston, A. and O. Levin, "Session Initiation Protocol (SIP) Call Control - Conferencing for User Agents", RFC 4579, August 2006.

[RFC4975] Campbell, B., Mahy, R., and C. Jennings, "The Message Session Relay Protocol (MSRP)", RFC 4975, September 2007.

[RFC6120] Saint-Andre, P., "Extensible Messaging and Presence Protocol (XMPP): Core", RFC 6120, March 2011.

[RFC6121] Saint-Andre, P., "Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence", RFC 6121, March 2011.

[XEP-0045] Saint-Andre, P., "Multi-User Chat", XSF XEP 0045, July 2008.

8.2. Informative References

[RFC3515] Sparks, R., "The Session Initiation Protocol (SIP) Refer Method", RFC 3515, April 2003.

[RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006.

[RFC4575] Rosenberg, J., Schulzrinne, H., and O. Levin, "A Session Initiation Protocol (SIP) Event Package for Conference State", RFC 4575, August 2006.

[RFC6501] Novo, O., Camarillo, G., Morgan, D., and J. Urpalainen, "Conference Information Data Model for Centralized Conferencing (XCON)", RFC 6501, March 2012.

[XEP-0172] Saint-Andre, P. and V. Mercier, "User Nickname", XSF XEP 0172, March 2012.

[XEP-0249] Saint-Andre, P., "Direct MUC Invitations", XSF XEP 0249, September 2011.

Appendix A. Acknowledgements

Special thanks to Fabio Forno for his co-authorship of an early version of this document.

Thanks also to Philipp Hancke for his detailed feedback.

Some text in this document was borrowed from [I-D.ietf-stox-core] and from [XEP-0045].

Authors' Addresses

Peter Saint-Andre
Cisco Systems, Inc.
1899 Wynkoop Street, Suite 600
Denver, CO 80202
USA

Phone: +1-303-308-3282
Email: psaintan@cisco.com

Saul Ibarra Corretge
AG Projects
Dr. Leijdsstraat 92
Haarlem 2021RK
The Netherlands

Email: saul@ag-projects.com

Salvatore Loreto
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

Email: Salvatore.Loreto@ericsson.com