Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 2, 2014

P. Saint-Andre
Cisco Systems, Inc.
S. Ibarra
AG Projects
E. Ivov
Jitsi

July 1, 2013

Interworking between the Session Initiation Protocol (SIP) and the Extensible Messaging and Presence Protocol (XMPP): Media Sessions
draft-ietf-stox-media-00

Abstract

This document defines a bi-directional protocol mapping for use by gateways that enable the exchange of media signalling messages between systems that implement the Jingle extensions to the Extensible Messaging and Presence Protocol (XMPP) and those that implement the Session Initiation Protocol (SIP).

Table of Contents

1. Introduction

The Session Initiation Protocol [RFC3261] is a widely-deployed technology for the management of media sessions (such as voice calls) over the Internet. SIP itself provides a signalling channel (typically via the User Datagram Protocol [RFC768]), over which two or more parties can exchange messages for the purpose of negotiating a media session that uses a dedicated media channel such as the Real-time Transport Protocol [RFC3550].

The Extensible Messaging and Presence Protocol [RFC6120] also provides a signalling channel, typically via the Transmission Control Protocol [RFC793]. Given the significant differences between XMPP and SIP, it is difficult to combine the two technologies in a single user agent. Therefore, developers wishing to add media session capabilities to XMPP clients have defined an XMPP-specific negotiation protocol called Jingle [XEP-0166].

However, Jingle has been designed to easily map to SIP for communication through gateways or other transformation mechanisms. Therefore, consistent with existing specifications for mapping between SIP and XMPP (see [I-D.ietf-stox-core] and other specifications in that "series"), this document describes a bi-directional protocol mapping for use by gateways that enable the exchange of media signalling messages between systems that implement SIP and those that implement the XMPP Jingle extensions.

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], [RFC6120], [XEP-0166], and [XEP-0167]. The term "JID" is short for "Jabber Identifier".

3. Jingle to SIP

3.1. Overview

As mentioned, Jingle was designed in part to enable straightforward protocol mapping between XMPP and SIP. However, given the significantly different technology assumptions underlying XMPP and SIP, Jingle is naturally different from SIP in several important respects:

o Base SIP messages and headers use a plaintext format similar in some ways to the Hypertext Transport Protocol [RFC2616], whereas Jingle messages are pure XML. Mappings between SIP headers and Jingle message syntax are provided below.
o The SIP payloads defining session semantics use the Session Description Protocol [RFC4566], whereas the equivalent Jingle payloads are defined as XML child elements of the Jingle <content/> element. However, the Jingle specifications defining such child elements specify mappings to SDP for all Jingle syntax, making the mapping relatively straightforward.
o The SIP signalling channel is transported over UDP, whereas the signalling channel for Jingle is XMPP over TCP. Mapping between the transport layers typically happens within a gateway using techniques below the application level, and therefore is not addressed in this specification.

3.2. Syntax Mappings

3.2.1. Generic Jingle Syntax

Jingle is designed in a modular fashion, so that session description data is generally carried in a payload within the generic Jingle elements, i.e., the <jingle/> element and its <content/> child. The following example illustrates this structure, where the XMPP stanza is a request to initiate an audio session using RTP over a raw UDP transport.

```
<iq from='romeo@example.net/v3rsch1kk3l1jk'
    id='ne91v36s'
    to='juliet@example.com/t3hr0zny'
    type='set'>
  <jingle xmlns='urn:xmpp:jingle:1'
          action='session-initiate'
          initiator='romeo@example.net/v3rsch1kk3l1jk'
          sid='a73sjjvkla37jfea'>
    <content creator='initiator'
             media='audio'
             name='this-is-the-audio-content'
             senders='both'>
      <description xmlns='urn:xmpp:jingle:app:rtp:1'>
        <payload-type id='96' name='speex' clockrate='16000'/>
        <payload-type id='97' name='speex' clockrate='8000'/>
        <payload-type id='18' name='G729'/>
        <payload-type channels='2'
                      clockrate='16000'
                      id='103'
                      name='L16'/>
        <payload-type id='98' name='x-ISAC' clockrate='8000'/>
      </description>
      <transport xmlns='urn:xmpp:jingle:transport:raw-udp'>
        <candidate ip='10.1.1.104' port='13540' generation='0'/>
      </transport>
    </content>
  </jingle>
</iq>
```

In the foregoing example, the syntax and semantics of the <jingle/> and <content/> elements are defined in [XEP-0166], the syntax and semantics of the <description/> element are defined in [XEP-0167], and the syntax and semantics of the <transport/> element are defined in [XEP-0177]. Other <description/> elements are defined in specifications for the appropriate application types (see for example [XEP-0167]) and other <transport/> elements are defined in the specifications for appropriate transport methods (see for example [XEP-0176], which defines an XMPP profile of [RFC5245]).

At the core Jingle layer, the following mappings are defined.

Jingle	SIP
<jingle/> 'action'	[see next table]
<jingle/> 'initiator'	[no mapping]
<jingle/> 'responder'	[no mapping]
<jingle/> 'sid'	local-part of Call-ID
local-part of 'initiator'	<username> in SDP o= line
<content/> 'creator'	[no mapping]
<content/> 'name'	[no mapping]
<content/> 'profile'	<proto> in SDP m= line
<content/> 'senders' value of both, initiator, or responder	a= line of sendrecv, recvonly, or sendonly

The 'action' attribute of the <jingle/> element has nine allowable values. In general they should be mapped as shown in the following table, with some exceptions as described herein.

Jingle Action	SIP Method
content-accept	INVITE response (1xx)
content-add	INVITE request
content-modify	INVITE request
content-remove	INVITE request
session-accept	INVITE response (1xx or 2xx)
session-info	[varies]
session-initiate	INVITE request
session-terminate	BYE
transport-info	[varies]

3.2.2. Audio Application Format

A Jingle application format for audio exchange via RTP is specified in [XEP-0167]. This application format effectively maps to the "RTP/AVP" profile specified in [RFC3551], where the media type is "audio" and the specific mappings to SDP syntax are provided in [XEP-0167].

3.2.3. Video Application Format

A Jingle application format for video exchange via RTP is specified in [XEP-0167]. This application format effectively maps to the "RTP/AVP" profile specified in [RFC3551], where the media type is "audio" and the specific mappings to SDP syntax are provided in [XEP-0167].

3.2.4. Raw UDP Transport Method

A basic Jingle transport method for exchanging media over UDP is specified in [XEP-0177]. This transport method involves the negotiation of an IP address and port only, and does not provide NAT traversal. The Jingle 'ip' attribute maps to the connection-address parameter of the SDP c= line and the 'port' attribute maps to the port parameter of the SDP m= line.

3.2.5. ICE-UDP Transport Method

A more advanced Jingle transport method for exchanging media over UDP is specified in [XEP-0176]. Under ideal conditions this transport method provides NAT traversal by following the Interactive Connectivity Exchange methodology specified in [RFC5245]. The relevant SDP mappings are provided in [XEP-0176].

3.3. Sample Scenarios

The following sections provide sample scenarios (or "call flows") that illustrate the principles of interworking from Jingle to SIP. These scenarios are not exhaustive.

3.3.1. Basic Voice Chat

The protocol flow for a basic voice chat for which an XMPP user (juliet@example.com) is the iniator and a SIP user (romeo@example.net) is the responder. The voice chat is consummated through a gateway. To simplify the example, the transport method negotiated is "raw user datagram protocol" as specified in [XEP-0177].

```
INITIATOR  ...XMPP...   GATEWAY   ...SIP...    RESPONDER
  |                        |                        |
  | session-initiate       |                        |
  |----------------------->|                        |
  | IQ-result (ack)        |                        |
  |<-----------------------|                        |
  |                        | INVITE                 |
  |                        |----------------------->|
  |                        | 180 Ringing            |
  |                        |<-----------------------|
  | session-info (ringing) |                        |
  |<-----------------------|                        |
  | IQ-result (ack)        |                        |
  |----------------------->|                        |
  |                        | 200 OK                 |
  |                        |<-----------------------|
  | session-accept         |                        |
  |<-----------------------|                        |
  | IQ-result (ack)        |                        |
  |----------------------->|                        |
  |                        | ACK                    |
  |                        |----------------------->|
  |                  MEDIA SESSION                  |
  |<===============================================>|
  |                        | BYE                    |
  |                        |<-----------------------|
  | session-terminate      |                        |
  |<-----------------------|                        |
  | IQ-result (ack)        |                        |
  |----------------------->|                        |
  |                        | 200 OK                 |
  |                        |----------------------->|
  |                        |                        |
```

The packet flow is as follows.

First the XMPP user sends a Jingle session-initiation request to the SIP user.

```
<iq from='juliet@example.com/t3hr0zny'
    id='hu2s61f4'
    from='romeo@example.net/v3rsch1kk3l1jk'
    type='set'>
  <jingle xmlns='urn:xmpp:jingle:1'
          action='session-initiate'
          initiator='juliet@example.com/t3hr0zny'
          sid='a73sjjvkla37jfea'>
    <content creator='initiator'
             media='audio'
             name='this-is-the-audio-content'>
      <description xmlns='urn:xmpp:jingle:app:rtp:1'>
        <payload-type id='96' name='speex' clockrate='16000'/>
        <payload-type id='97' name='speex' clockrate='8000'/>
        <payload-type id='18' name='G729'/>
      </description>
      <transport xmlns='urn:xmpp:jingle:transport:raw-udp'>
        <candidate ip='192.0.2.101' port='49172' generation='0'/>
      </transport>
    </content>
  </jingle>
</iq>
```

The gateway returns an XMPP IQ-result to the initiator on behalf of the responder.

```
<iq from='juliet@example.com/t3hr0zny'
    id='hu2s61f4'
    to='romeo@example.net/v3rsch1kk3l1jk'
    type='result'/>
```

The gateway transforms the Jingle session-initiate action into a SIP INVITE.

```
INVITE sip:romeo@example.net SIP/2.0
Via: SIP/2.0/TCP client.example.com:5060;branch=z9hG4bK74bf9
Max-Forwards: 70
From: Juliet Capulet <sip:juliet@example.com>;tag=t3hr0zny
To: Romeo Montague <sip:romeo@example.net>
Call-ID: 3848276298220188511@example.com
CSeq: 1 INVITE
Contact: <sip:juliet@client.example.com;transport=tcp>
Content-Type: application/sdp
Content-Length: 184

v=0
o=alice 2890844526 2890844526 IN IP4 client.example.com
s=-
c=IN IP4 192.0.2.101
t=0 0
m=audio 49172 RTP/AVP 0
a=rtpmap:96 SPEEX/16000
a=rtpmap:97 SPEEX/8000
a=rtpmap:18 G729
```

The responder returns a SIP 180 Ringing message.

```
SIP/2.0 180 Ringing
Via: SIP/2.0/TCP client.example.com:5060;branch=z9hG4bK74bf9
 ;received=192.0.2.101
From: Juliet Capulet <sip:juliet@example.com>;tag=t3hr0zny
To: Romeo Montague <sip:romeo@example.net>;tag=v3rsch1kk3l1jk
Call-ID: 3848276298220188511@example.com
CSeq: 1 INVITE
Contact: <sip:romeo@client.example.net;transport=tcp>
Content-Length: 0
```

The gateway transforms the ringing message into XMPP syntax.

```
<iq from='romeo@montague.net/v3rsch1kk3l1jk'
    id='ol3ba71g'
    to='juliet@example.com/t3hr0zny'
    type='set'>
  <jingle xmlns='urn:xmpp:jingle:1'
          action='session-info'
          initiator='juliet@example.com/t3hr0zny'
          sid='a73sjjvkla37jfea'>
    <ringing xmlns='urn:xmpp:jingle:app:rtp:1-info'/>
  </jingle>
</iq>
```

The initiator returns an IQ-result acknowledging receipt of the

ringing message, which is used only by the gateway and not transformed into SIP syntax.

```
<iq from='juliet@example.com/t3hr0zny'
    id='ol3ba71g'
    to='romeo@example.net/v3rsch1kk3l1jk'
    type='result'/>
```

The responder sends a SIP 200 OK to the initiator.

```
SIP/2.0 200 OK
Via: SIP/2.0/TCP client.example.com:5060;branch=z9hG4bK74bf9
 ;received=192.0.2.101
From: Juliet Capulet <sip:juliet@example.com>;tag=t3hr0zny
To: Romeo Montague <sip:romeo@example.net>;tag=v3rsch1kk3l1jk
Call-ID: 3848276298220188511@example.com
CSeq: 1 INVITE
Contact: <sip:romeo@client.example.net;transport=tcp>
Content-Type: application/sdp
Content-Length: 147

v=0
o=romeo 2890844527 2890844527 IN IP4 client.example.net
s=-
c=IN IP4 192.0.2.201
t=0 0
m=audio 3456 RTP/AVP 0
a=rtpmap:97 SPEEX/8000
a=rtpmap:18 G729/8000
```

The gateway transforms the 200 OK into a Jingle session-accept action.

```
<iq from='romeo@example.net/v3rsch1kk3l1jk'
    id='pd1bf839'
    to='juliet@example.com/t3hr0zny'
    type='set'>
  <jingle xmlns='urn:xmpp:jingle:1'
          action='session-accept'
          initiator='juliet@example.com/t3hr0zny'
          responder='romeo@example.net/v3rsch1kk3l1jk'
          sid='a73sjjvkla37jfea'>
    <content creator='initiator'
             media='audio'
             name='this-is-the-audio-content'>
      <description xmlns='urn:xmpp:jingle:app:rtp:1'>
        <payload-type id='97' name='speex' clockrate='8000'/>
        <payload-type id='18' name='G729'/>
        <payload-type id='0' name='PCMU' clockrate='8000'/>
      </description>
      <transport xmlns='urn:xmpp:jingle:transport:raw-udp'>
        <candidate ip='192.0.2.101' port='49172' generation='0'/>
      </transport>
    </content>
  </jingle>
</iq>
```

If the payload types and transport candidate can be successfully used by both parties, then the initiator acknowledges the session-accept action.

```
<iq from='romeo@example.net/v3rsch1kk3l1jk'
    id='pd1bf839'
    to='juliet@example.com/t3hr0zny'
    type='result'/>
```

The parties now begin to exchange media. In this case they would exchange audio using the Speex codec at a clockrate of 8000 since that is the highest-priority codec for the responder (as determined by the XML order of the <payloadtype/> children).

The parties may continue the session as long as desired.

Eventually, one of the parties (in this case the responder) terminates the session.

```
BYE sip:juliet@client.example.com SIP/2.0
Via: SIP/2.0/TCP client.example.net:5060;branch=z9hG4bKnashds7
Max-Forwards: 70
From: Romeo Montague <sip:romeo@example.net>;tag=8321234356
To: Juliet Capulet <sip:juliet@example.com>;tag=9fxced76sl
Call-ID: 3848276298220188511@example.com
CSeq: 1 BYE
Content-Length: 0
```

The gateway transforms the SIP BYE into XMPP syntax.

```
<iq from='romeo@example.net/v3rsch1kk3l1jk'
    id='rv301b47'
    to='juliet@example.com/t3hr0zny'
    type='set'>
  <jingle xmlns='urn:xmpp:jingle:1'
          action='session-terminate'
          initiator='juliet@example.com/t3hr0zny'
          reasoncode='no-error'
          sid='a73sjjvkla37jfea'/>
</iq>
```

The initiator returns an IQ-result acknowledging receipt of the session termination, which is used only by the gateway and not transformed into SIP syntax.

```
<iq from='romeo@example.net/v3rsch1kk3l1jk'
    id='rv301b47'
    to='juliet@example.com/t3hr0zny'
    type='result'/>
```

4. SIP to Jingle

To follow.

5. Security Considerations

Detailed security considerations for session management are given for SIP in [RFC3261] and for XMPP in [XEP-0166] (see also [RFC6120]).

6. IANA Considerations

This document has no actions for the IANA.

7. References

7.1. Normative References

[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-00 (work in progress), July 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.

[RFC3551] Schulzrinne, H. and S. Casner, "RTP Profile for Audio and Video Conferences with Minimal Control", STD 65, RFC 3551, July 2003.

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

[RFC5245] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", RFC 5245, April 2010.

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

[XEP-0166]
Ludwig, S., Beda, J., Saint-Andre, P., McQueen, R., Egan, S., and J. Hildebrand, "Jingle", XSF XEP 0166, June 2007.

[XEP-0167]
Ludwig, S., Saint-Andre, P., Egan, S., and R. McQueen, "Jingle RTP Sessions", XSF XEP 0167, February 2009.

[XEP-0176]
Beda, J., Ludwig, S., Saint-Andre, P., Hildebrand, J., and S. Egan, "Jingle ICE-UDP Transport Method", XSF XEP 0176, February 2009.

[XEP-0177]

Beda, J., Saint-Andre, P., Ludwig, S., Hildebrand, J., and S. Egan, "Jingle Raw UDP Transport", XSF XEP 0177, February 2009.

7.2. Informative References

[RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.

[RFC768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, August 1980.

[RFC793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, September 1981.

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

Emil Ivov
Jitsi
Strasbourg 67000
France

Phone: +33-177-624-330
Email: emcho@jitsi.org