



XMPP

XEP-0353: Jingle Message Initiation

Philipp Hancke
<mailto:fippo@andyet.com>
<xmpp:fippo@goodadvice.pages.de>

Peter Saint-Andre
<mailto:peter@andyet.net>
<xmpp:stpeter@stpeter.im>
<https://stpeter.im/>

2014-10-02
Version 0.2.0

Status	Type	Short Name
Experimental	Standards Track	jingle-message

This specification provides a way for the initiator of a Jingle session to propose sending an invitation in an XMPP message stanza, thus taking advantage of message delivery semantics instead of sending IQ stanzas to all of the responder's online resources or choosing a particular online resource.

Legal

Copyright

This XMPP Extension Protocol is copyright © 1999 – 2017 by the [XMPP Standards Foundation](#) (XSF).

Permissions

Permission is hereby granted, free of charge, to any person obtaining a copy of this specification (the "Specification"), to make use of the Specification without restriction, including without limitation the rights to implement the Specification in a software program, deploy the Specification in a network service, and copy, modify, merge, publish, translate, distribute, sublicense, or sell copies of the Specification, and to permit persons to whom the Specification is furnished to do so, subject to the condition that the foregoing copyright notice and this permission notice shall be included in all copies or substantial portions of the Specification. Unless separate permission is granted, modified works that are redistributed shall not contain misleading information regarding the authors, title, number, or publisher of the Specification, and shall not claim endorsement of the modified works by the authors, any organization or project to which the authors belong, or the XMPP Standards Foundation.

Warranty

NOTE WELL: This Specification is provided on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE.

Liability

In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall the XMPP Standards Foundation or any author of this Specification be liable for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising from, out of, or in connection with the Specification or the implementation, deployment, or other use of the Specification (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if the XMPP Standards Foundation or such author has been advised of the possibility of such damages.

Conformance

This XMPP Extension Protocol has been contributed in full conformance with the XSF's Intellectual Property Rights Policy (a copy of which can be found at <https://xmpp.org/about/xsf/ipr-policy>) or obtained by writing to XMPP Standards Foundation, P.O. Box 787, Parker, CO 80134 USA).

Contents

1	Introduction	1
2	Requirements	1
3	Use Cases	1
3.1	Indicating Intent to Start a Session	1
3.2	Disavowing Intent to Start a Session	3
3.3	Accepting Intent to Start a Session	3
3.4	Rejecting Intent to Start a Session	4
3.5	Initiating the Jingle Session	5
4	Open Issues	7
5	Acknowledgements	7
6	Security Considerations	7
7	IANA Considerations	7
8	XMPP Registrar Considerations	7
8.1	Protocol Namespaces	7
8.2	Protocol Versioning	8
9	XML Schema	8

1 Introduction

Because [Jingle \(XEP-0166\)](#)¹ uses `<iq/>` stanzas for all interactions between the parties to a session, when sending an invitation the initiator needs to either pick one of the responder's resources (e.g., based on [Entity Capabilities \(XEP-0115\)](#)² information) or send the invitation to all of the responder's resources that support Jingle. The first method is prone to error (e.g., in cases where more than one resource supports Jingle) and the second method requires sending a separate invitation to each resource. Neither of these is ideal. Although [Presence Decloaking \(XEP-0276\)](#)³ proposed a way to overcome the problem, it too has issues (e.g., dependency on a presence service and the need to reveal all supported XMPP features) and in any case has not been widely implemented.

This document proposes an alternative solution: exchanging a `<message/>` stanza before sending the Jingle invitation in an `<iq/>` stanza. (Indeed, in the early discussions leading up to the Jingle protocol the authors considered using `<message/>` stanzas instead of `<iq/>` stanzas, but chose the latter for their deterministic handling semantics.) This method effectively results in a kind of decloaking for Jingle purposes.

2 Requirements

This protocol was designed with the following requirements in mind:

- Allow responder to choose the resource or device on which to take the call.
- Result in "ring-on-all-devices" behavior.
- Not rely on presence.
- Make use of "forking" semantics for message stanzas.
- Allow indication of session content.
- Work with push notifications.

3 Use Cases

3.1 Indicating Intent to Start a Session

In order to prepare for sending a Jingle invitation, the initiator (e.g., Romeo) sends a `<message/>` stanza containing a `<propose/>` element qualified by the 'urn:xmpp:jingle-message:0' namespace. The `<propose/>` element MUST possess an 'id' attribute that will be used for

¹XEP-0166: Jingle <<https://xmpp.org/extensions/xep-0166.html>>.

²XEP-0115: Entity Capabilities <<https://xmpp.org/extensions/xep-0115.html>>.

³XEP-0276: Presence Decloaking <<https://xmpp.org/extensions/xep-0276.html>>.

the session invitation and MUST contain one <description/> element for each media type associated with the intended session.

Listing 1: Initiator Sends Intent Message

```
<message from='romeo@montague.example/orchard'
  to='juliet@capulet.example'>
  <propose xmlns='urn:xmpp:jingle-message:0' id='a73sjjvkl37jfea'>
    <description xmlns='urn:xmpp:jingle:apps:rtp:1' media='audio' />
  </propose>
</message>
```

The server of the responder (e.g., Juliet) distributes this message stanza to all of Juliet's available resources (and to push resources as appropriate). Those devices might start ringing as a result.

Listing 2: Responder's Server Distributes Intent Message

```
<!--{}- delivered to juliet@capulet.example/desktop {}-->
<message from='romeo@montague.example/orchard'
  to='juliet@capulet.example'>
  <propose xmlns='urn:xmpp:jingle-message:0' id='a73sjjvkl37jfea'>
    <description xmlns='urn:xmpp:jingle:apps:rtp:1' media='audio' />
  </propose>
</message>

<!--{}- delivered to juliet@capulet.example/tablet {}-->
<message from='romeo@montague.example/orchard'
  to='juliet@capulet.example'>
  <propose xmlns='urn:xmpp:jingle-message:0' id='a73sjjvkl37jfea'>
    <description xmlns='urn:xmpp:jingle:apps:rtp:1' media='audio' />
  </propose>
</message>

<!--{}- delivered to juliet@capulet.example/phone {}-->
<message from='romeo@montague.example/orchard'
  to='juliet@capulet.example'>
  <propose xmlns='urn:xmpp:jingle-message:0' id='a73sjjvkl37jfea'>
    <description xmlns='urn:xmpp:jingle:apps:rtp:1' media='audio' />
  </propose>
</message>
```

Consistent with the recommendation for one-to-one chat sessions in Section 5.1 of RFC 6121⁴, the initiator SHOULD also send directed presence to the responder if the two entities do not already share presence information; including Entity Capabilities (XEP-0115) information in this directed presence stanza enables the responder to know the availability of the initiator

⁴RFC 6121: Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence <<http://tools.ietf.org/html/rfc6121>>.

(e.g., in case the message is actually delivered quite a bit later because it is saved to offline storage) and also to know the XMPP features supported by the initiator.

Listing 3: Initiator Sends Directed Presence

```
<presence to='romeo@montague.example/orchard'
          from='juliet@capulet.example'>
  <c xmlns='http://jabber.org/protocol/caps'
    hash='sha-1'
    node='http://psi-im.org'
    ver='q07IKJEyvjvHSyhy//CH0CxmKi8w=' />
</presence>
```

3.2 Disavowing Intent to Start a Session

It can happen that the initiator might want to disavow intent to send a session invitation (e.g., because the initiator has accepted another session). The initiator can do so by sending a message stanza containing a `<retract/>` element specifying the same session ID.

Listing 4: Initiator Sends Stop Message

```
<message from='romeo@montague.example/orchard'
          to='juliet@capulet.example'>
  <retract xmlns='urn:xmpp:jingle-message:0' id='a73sjjvkl37jfea' />
</message>
```

3.3 Accepting Intent to Start a Session

Upon receiving the intent message, the responder's various devices will "ring" and the responder will answer the call on a particular device. Here we assume that since this is an audio-only call, Juliet chooses to take the call on the device associated with her "phone" resource.

As a first step, her "phone" resource informs all of her resources about accepting the call by sending a message to her own bare JID containing an `<accept/>` element specifying the session ID of the original `<propose/>` message.

Listing 5: One of Responder's Resources Accepts the Call

```
<message from='juliet@capulet.example/phone'
          to='juliet@capulet.example'>
  <accept xmlns='urn:xmpp:jingle-message:0' id='a73sjjvkl37jfea' />
</message>
```

Juliet's server broadcasts this accept message to all of her available resources (as described in RFC 6121), which stop ringing:

Listing 6: Responder's Server Delivers Accept Message

```

<!--{}- delivered to juliet@capulet.example/desktop -{}-->
<message from='juliet@capulet.example/phone'
  to='juliet@capulet.example'>
  <accept xmlns='urn:xmpp:jingle-message:0' id='a73sjjvkla37jfea' />
</message>

<!--{}- delivered to juliet@capulet.example/tablet -{}-->
<message from='juliet@capulet.example/phone'
  to='juliet@capulet.example'>
  <accept xmlns='urn:xmpp:jingle-message:0' id='a73sjjvkla37jfea' />
</message>

<!--{}- delivered to juliet@capulet.example/phone -{}-->
<message from='juliet@capulet.example/phone'
  to='juliet@capulet.example'>
  <accept xmlns='urn:xmpp:jingle-message:0' id='a73sjjvkla37jfea' />
</message>

```

Next, the device from which Juliet accepted the call tells Romeo to proceed with the session (via a message stanza containing a `<proceed/>` element), and also sends directed presence for the reasons described above.

Listing 7: Responder Sends Directed Presence and Start Message

```

<message from='juliet@capulet.example/phone'
  to='romeo@montague.example/orchard'>
  <proceed xmlns='urn:xmpp:jingle-message:0' id='a73sjjvkla37jfea' />
</message>

<presence from='juliet@capulet.example/phone'
  to='romeo@montague.example/orchard'>
  <c xmlns='http://jabber.org/protocol/caps'
    hash='sha-1'
    node='http://code.google.com/p/exodus'
    ver='QgayPKawpkPSDYmwT/WM94uAlu0=' />
</presence>

```

3.4 Rejecting Intent to Start a Session

Instead of accepting the call, the responder might want to ignore the call and tell all of her devices to stop ringing (e.g., perhaps because Romeo is getting to be a bit of a nuisance). She does this by rejecting the call on one of her devices and having that device tell all of the other devices to stop ringing, in the form of a message to her own bare JID containing an `<reject/>` element specifying the session ID of the original `<propose/>` message.

Listing 8: One of Responder's Resources Rejects the Call

```
<message from='juliet@capulet.example/tablet'
        to='juliet@capulet.example'>
  <reject xmlns='urn:xmpp:jingle-message:0' id='a73sjjvkl37jfea' />
</message>
```

Juliet's server broadcasts this reject message to all of her available resources (as described in RFC 6121), which stop ringing:

Listing 9: Responder's Server Delivers Reject Message

```
<!--{}- delivered to juliet@capulet.example/desktop -{}-->
<message from='juliet@capulet.example/tablet'
        to='juliet@capulet.example'>
  <reject xmlns='urn:xmpp:jingle-message:0' id='a73sjjvkl37jfea' />
</message>

<!--{}- delivered to juliet@capulet.example/tablet -{}-->
<message from='juliet@capulet.example/tablet'
        to='juliet@capulet.example'>
  <reject xmlns='urn:xmpp:jingle-message:0' id='a73sjjvkl37jfea' />
</message>

<!--{}- delivered to juliet@capulet.example/phone -{}-->
<message from='juliet@capulet.example/tablet'
        to='juliet@capulet.example'>
  <reject xmlns='urn:xmpp:jingle-message:0' id='a73sjjvkl37jfea' />
</message>
```

Next, the responder MAY want to decline the call explicitly, in the form of a message to the sender's full JID containing a <reject/> element specifying the session ID of the original <propose/> message.

Listing 10: Responder Rejects the Call Explicitly to the Sender

```
<message from='juliet@capulet.example/tablet'
        to='romeo@montague.example/orchard'>
  <reject xmlns='urn:xmpp:jingle-message:0' id='a73sjjvkl37jfea' />
</message>
```

3.5 Initiating the Jingle Session

Now Romeo actually initiates the session (segue to Jingle itself).

Listing 11: Initiation

```
<iq from='romeo@montague.example/orchard'
```



```
id='ih28sx61'  
to='juliet@capulet.example/phone'  
type='set'>  
<jingle xmlns='urn:xmpp:jingle:1'  
  action='session-initiate'  
  initiator='romeo@montague.example/orchard'  
  sid='a73sjjvkl37jfea'>  
  <content creator='initiator' name='voice'>  
    <description xmlns='urn:xmpp:jingle:apps:rtp:1' media='audio'>  
      <payload-type id='96' name='speex' clockrate='16000' />  
      <payload-type id='97' name='speex' clockrate='8000' />  
      <payload-type id='18' name='G729' />  
      <payload-type id='0' name='PCMU' />  
      <payload-type id='103' name='L16' clockrate='16000' channels='  
        2' />  
      <payload-type id='98' name='x-ISAC' clockrate='8000' />  
    </description>  
    <transport xmlns='urn:xmpp:jingle:transports:ice-udp:1'  
      pwd='asd88fgpdd777uzjYhagZg'  
      ufrag='8hhy'>  
      <candidate component='1'  
        foundation='1'  
        generation='0'  
        id='el0747fg11'  
        ip='10.0.1.1'  
        network='1'  
        port='8998'  
        priority='2130706431'  
        protocol='udp'  
        type='host' />  
      <candidate component='1'  
        foundation='2'  
        generation='0'  
        id='y3s2b30v3r'  
        ip='192.0.2.3'  
        network='1'  
        port='45664'  
        priority='1694498815'  
        protocol='udp'  
        rel-addr='10.0.1.1'  
        rel-port='8998'  
        type='srflx' />  
    </transport>  
  </content>  
</jingle>  
</iq>
```

4 Open Issues

The following issues remain to be closed:

- Specify how this works (if at all) with resource locking.
- Specify if and how to use messages of type 'headline'.

5 Acknowledgements

Thanks to Lance Stout for his feedback.

6 Security Considerations

Because exchanging messages with other entities is effectively is a presence leak, an XMPP client that implements the receiving side of this specification MUST disable sending of proceed messages by default and MUST enable the feature only as a result of explicit user confirmation. Such confirmation can be provided per request, by whitelisting requests received from Jingle initiators in the responder's contact list, or through some other suitable means as long as sending proceed messages does not occur by default.

7 IANA Considerations

This document requires no interaction with the [Internet Assigned Numbers Authority \(IANA\)](#)⁵.

8 XMPP Registrar Considerations

8.1 Protocol Namespaces

This specification defines the following XML namespace:

- urn:xmpp:jingle:jingle-message:0

⁵The Internet Assigned Numbers Authority (IANA) is the central coordinator for the assignment of unique parameter values for Internet protocols, such as port numbers and URI schemes. For further information, see <http://www.iana.org/>.

The XMPP Registrar ⁶ includes the foregoing namespace to the registry located at <https://xmpp.org/registrar/namespaces.html>, as described in Section 4 of XMPP Registrar Function (XEP-0053) ⁷.

8.2 Protocol Versioning

If the protocol defined in this specification undergoes a revision that is not fully backwards-compatible with an older version, the XMPP Registrar shall increment the protocol version number found at the end of the XML namespaces defined herein, as described in Section 4 of XEP-0053.

9 XML Schema

```
<?xml version='1.0' encoding='UTF-8'?>
<xs:schema
  xmlns:xs='http://www.w3.org/2001/XMLSchema'
  xmlns:xml='http://www.w3.org/XML/1998/namespace'
  targetNamespace='urn:xmpp:jingle-message:0'
  xmlns='urn:xmpp:jingle-message:0'
  elementFormDefault='qualified'>

  <xs:element name='accept'>
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base='empty'>
          <xs:attribute name='id' type='xs:string' use='required' />
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>

  <xs:element name='proceed'>
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base='empty'>
          <xs:attribute name='id' type='xs:string' use='required' />
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

⁶The XMPP Registrar maintains a list of reserved protocol namespaces as well as registries of parameters used in the context of XMPP extension protocols approved by the XMPP Standards Foundation. For further information, see <https://xmpp.org/registrar/>.

⁷XEP-0053: XMPP Registrar Function <https://xmpp.org/extensions/xep-0053.html>.

```
</xs:element>

<xs:element name='propose'>
  <xs:complexType>
    <xs:sequence>
      <xs:any namespace='##other' minOccurs='1' maxOccurs='unbounded'
        />
    </xs:sequence>
    <xs:attribute name='id' type='xs:string' use='required' />
  </xs:complexType>
</xs:element>

<xs:element name='reject'>
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base='empty'>
        <xs:attribute name='id' type='xs:string' use='required' />
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>

<xs:element name='retract'>
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base='empty'>
        <xs:attribute name='id' type='xs:string' use='required' />
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>

</xs:schema>
```