



XMPP

XEP-0269: Jingle Early Media

Diana Cionoiu
<mailto:diana@null.ro>
<xmpp:l-fy@jabber.null.ro>

Peter Saint-Andre
<mailto:stpeter@stpeter.im>
<xmpp:stpeter@jabber.org>
<https://stpeter.im/>

2009-05-19
Version 0.1

| Status | Type | Short Name |
|----------|-----------------|------------|
| Deferred | Standards Track | N/A |

This specification describes methods for exchanging early media in the context of Jingle RTP sessions.

Legal

Copyright

This XMPP Extension Protocol is copyright © 1999 – 2024 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 | Concepts | 1 |
| 3 | Scenario | 2 |
| 4 | Security Considerations | 8 |
| 5 | IANA Considerations | 8 |
| 6 | XMPP Registrar Considerations | 8 |

1 Introduction

Jingle (XEP-0166)¹ can be used to initiate and negotiate a wide range of peer-to-peer sessions, including Jingle RTP Sessions (XEP-0167)² (mainly for voice and video) using the Realtime Transport Protocol (RTP; see RFC 3550³). This document describes call flows for so-called "early media" using Jingle.

2 Concepts

The term "early media" refers to media that is exchanged before a responder has definitively accepted a session request generated by an initiator or before end-to-end connectivity has been established (e.g., the media could be generated by an intermediate call manager or media relay). Early media is typically used to send ringing tones and announcements, using either audio streams or Dual Tone Multi-Frequency (DTMF) events.

In Jingle, the exchange of early media is established through use of the "content-add" action. In order to match the usage specified in RFC 3959⁴ and RFC 3960⁵, when adding a content definition for early media the value of the <content/> element's 'disposition' attribute MUST be "early-session" for mapping to a SIP Content-Disposition header value of "early-session". This enables endpoints or intermediate gateways to apply the application server model described in RFC 3960⁶.

An entity that generates a content-add message for early media SHOULD specify the same codecs for both session media and early media (however, it is possible that the entity that generates the early media does not generate the session media, for example in the case of an intermediate gateway or application server; in this case the entity MUST use one of the codecs advertised by the initiator).

Upon receiving a content-add message specifying the use of early media, the initiator's client SHOULD acknowledge the content-add, complete any required transport negotiation, and then send a content-accept (or content-reject) to the sender. When the responder subsequently sends a session-accept message, the acceptance MUST NOT be construed to include the content definition whose disposition is "early-session".

In handling early media and deciding whether to generate local ringing or to play early media received from the responder or an intermediate gateway, the initiator's client SHOULD proceed as follows:

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

²XEP-0167: Jingle RTP Sessions <<https://xmpp.org/extensions/xep-0167.html>>.

³RFC 3550: RTP: A Transport Protocol for Real-Time Applications <<http://tools.ietf.org/html/rfc3550>>.

⁴RFC 3959: The Early Session Disposition Type for the Session Initiation Protocol (SIP) <<http://tools.ietf.org/html/rfc3959>>.

⁵RFC 3960: Early Media and Ringing Tone Generation in the Session Initiation Protocol (SIP) <<http://tools.ietf.org/html/rfc3960>>.

⁶RFC 3960: Early Media and Ringing Tone Generation in the Session Initiation Protocol (SIP) <<http://tools.ietf.org/html/rfc3960>>.

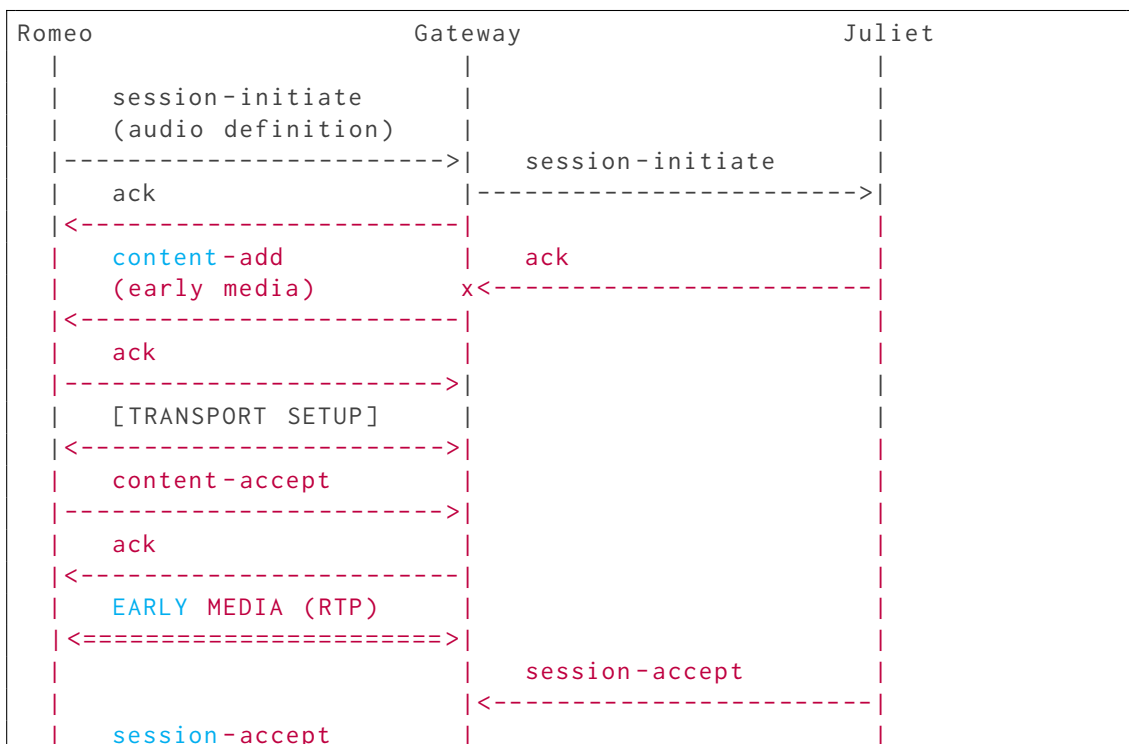
1. If no ringing notification is received via a session-info event containing a <ringing/> condition, do not generate local ringing.
2. If a ringing notification is received and no early media is received, generate local ringing.
3. If a ringing notification is received but early media is received, play the early media and do not generate local media.
4. Once the responder has accepted the session and the session data (as opposed to early session data) has begun to flow, stop local ringing or stop playing early media.

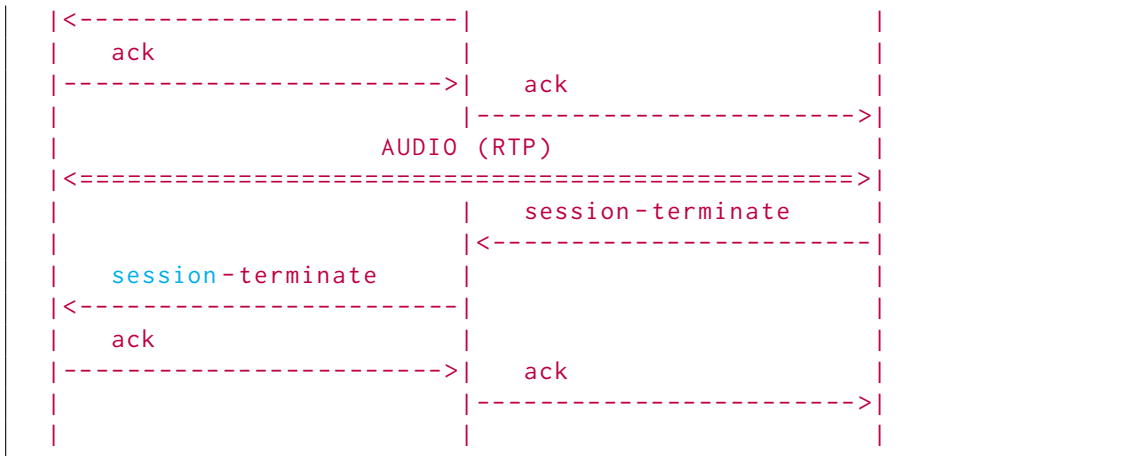
For examples of early media, see the Jingle Audio via RTP with Early Media section of this document.

3 Scenario

In this scenario, Romeo initiates a voice chat with Juliet using a transport method of ICE-UDP. There is a gateway between Romeo and Juliet, and the gateway functions as an application server by returning early media to Romeo (perhaps some late medieval hold music or an old-fashioned IVR interaction). To simplify the flow, we have left out any ringing notifications generated by Juliet.

The session flow is as follows.





The protocol flow is as follows, showing only the stanzas sent between Romeo and the gateway (acting on Juliet's behalf).

Listing 1: Initiator sends session-initiate

```

<iq from='romeo@montague.lit/orchard'
  id='ox73bf9h'
  to='juliet@capulet.lit/balcony'
  type='set'>
  <jingle xmlns='urn:xmpp:jingle:1'
    action='session-initiate'
    initiator='romeo@montague.lit/orchard'
    sid='a73sjjvkla37jfea'>
    <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='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>

```

Listing 2: Responder acknowledges session-initiate

```

<iq from='juliet@capulet.lit/balcony'
    id='ox73bf9h'
    to='romeo@montague.lit/orchard'
    type='result' />

```

Now the gateway sends a content-add message to Romeo while waiting for Juliet to pay attention to her telephony interface. It specifies a transport method of Raw UDP because it hosts its own media relay.

Listing 3: Gateway sends content-add on behalf of responder

```

<iq from='juliet@capulet.lit/balcony'
    id='s12f7hb5'
    to='romeo@montague.lit/orchard'
    type='set'>
  <jingle xmlns='urn:xmpp:jingle:1'
        action='content-add'
        initiator='romeo@montague.lit/orchard'
        sid='a73sjjvkla37jfea'>
    <content creator='responder'
            disposition='early-session'
            name='hold_music'
            senders='responder'>
      <description xmlns='urn:xmpp:jingle:apps:rtp:1' media='audio'>
        <payload-type id='18' name='G729' />
      </description>
      <transport xmlns='urn:xmpp:jingle:transports:raw-udp:1'
                pwd='YH75Fviy6338Vbrhrlp8Yh'
                ufrag='9uB6'>

```

```

        <candidate component='1'
                generation='0'
                id='a9j3mnbtu1'
                ip='10.1.1.104'
                port='13540' />
    </transport>
</content>
</jingle>
</iq>

```

Romeo then acknowledges the content-add action.

Listing 4: Initiator acknowledges content-add

```

<iq from='romeo@montague.lit/orchard'
    id='s12f7hb5'
    to='juliet@capulet.lit/balcony'
    type='result' />

```

Because the gateway (on behalf of the responder) specified a transport method of Raw UDP for the early session data, in accordance with the Jingle Raw UDP Transport Method the initiator then SHOULD also send a Raw UDP candidate to the gateway in a transport-info message.

Listing 5: Initiator sends transport-info

```

<iq from='romeo@montague.lit/orchard'
    id='xk2d1a85'
    to='juliet@capulet.lit/balcony'
    type='set'>
  <jingle xmlns='urn:xmpp:jingle:1'
        action='transport-info'
        initiator='romeo@montague.lit/orchard'
        responder='juliet@capulet.lit/balcony'
        sid='a73sjjvkl37jfea'>
    <content creator='initiator' name='voice'>
      <description xmlns='urn:xmpp:jingle:apps:rtp:1' media='audio'>
        <payload-type id='18' name='G729' />
      </description>
      <transport xmlns='urn:xmpp:jingle:transports:raw-udp:1'>
        <candidate component='1'
                generation='0'
                id='z7sdjb01hf'
                ip='208.68.163.214'
                port='9876' />
        <candidate component='2'
                generation='0'
                id='hg92lsn10b'
                ip='208.68.163.214'
                port='9877' />
      </transport>
    </content>
  </jingle>
</iq>

```



```

    </transport>
  </content>
</jingle>
</iq>

```

The gateway would acknowledge receipt of the candidate (not shown here) and both parties would attempt to establish connectivity for early media using the communicated transports. Eventually the initiator would send a content-accept to the gateway.

Listing 6: Initiator accepts new content definition

```

<iq from='romeo@montague.lit/orchard'
  id='18h4bg8q'
  to='juliet@capulet.lit/balcony'
  type='set'>
  <jingle xmlns='urn:xmpp:jingle:1'
    action='content-accept'
    initiator='romeo@montague.lit/orchard'
    sid='a73sjvkla37jfea'>
    <content creator='responder'
      disposition='early-session'
      name='hold_music'
      senders='responder'>
      <description xmlns='urn:xmpp:jingle:apps:rtp:1' media='audio'>
        <payload-type id='18' name='G729' />
      </description>
      <transport xmlns='urn:xmpp:jingle:transports:raw-udp:1' />
    </content>
  </jingle>
</iq>

```

The gateway then acknowledges the acceptance on behalf of Juliet.

Listing 7: Gateway acknowledges content-accept

```

<iq from='juliet@capulet.lit/balcony'
  id='18h4bg8q'
  to='romeo@montague.lit/orchard'
  type='result' />

```

Because Romeo has attempted to send test media to the gateway as described in [Jingle Raw UDP Transport Method \(XEP-0177\)](#)⁷, he has exposed an IP/port to which the gateway can now send early media via the media relay that it hosts. Eventually, the responder sends a session-accept.

Listing 8: Responder sends session-accept

⁷XEP-0177: Jingle Raw UDP Transport Method <<https://xmpp.org/extensions/xep-0177.html>>.

```

<iq from='juliet@capulet.lit/balcony'
  id='vh3bf781'
  to='romeo@montague.lit/orchard'
  type='set'>
  <jingle xmlns='urn:xmpp:jingle:1'
    action='session-accept'
    initiator='romeo@montague.lit/orchard'
    responder='juliet@capulet.lit/balcony'
    sid='a73sjjvkl37jfea'>
    <content creator='initiator' name='voice'>
      <description xmlns='urn:xmpp:jingle:apps:rtp:1' media='audio'>
        <payload-type id='97' name='speex' clockrate='8000' />
        <payload-type id='18' name='G729' />
      </description>
      <transport xmlns='urn:xmpp:jingle:transports:ice-udp:1'>
        <candidate component='1'
          foundation='1'
          generation='0'
          id='or2ii2syr1'
          ip='192.0.2.1'
          network='0'
          port='3478'
          priority='2130706431'
          protocol='udp'
          type='host' />
      </transport>
    </content>
  </jingle>
</iq>

```

Listing 9: Initiator acknowledges session-accept

```

<iq from='romeo@montague.lit/orchard'
  id='vh3bf781'
  to='juliet@capulet.lit/balcony'
  type='result' />

```

Once end-to-end connectivity is established (which might necessitate the exchange of additional candidates via transport-info messages), the parties begin to exchange media; as a result, Romeo and the gateway terminate the exchange of early media (this does not necessitate exchange of a content-remove message, since the endpoint and the gateway can simply stop sending media).

The endpoints can continue the session as long as desired. Eventually, one of the endpoints terminates the session.

Listing 10: Responder terminates the session

```

<iq from='juliet@capulet.lit/balcony'

```

```
id='gi49vx54'  
to='romeo@montague.lit/orchard'  
type='set'>  
<jingle xmlns='urn:xmpp:jingle:1'  
  action='session-terminate'  
  initiator='romeo@montague.lit/orchard'  
  sid='a73sjjvkl37jfea'>  
  <reason>  
    <success/>  
    <text>Sorry, gotta go!</text>  
  </reason>  
</jingle>  
</iq>
```

The other party then acknowledges termination of the session.

Listing 11: Initiator acknowledges termination

```
<iq from='romeo@montague.lit/orchard'  
  id='gi49vx54'  
  to='juliet@capulet.lit/balcony'  
  type='result' />
```

4 Security Considerations

For security considerations, refer to [Jingle RTP Sessions \(XEP-0167\)](#)⁸.

5 IANA Considerations

This document requires no interaction with the [Internet Assigned Numbers Authority \(IANA\)](#)⁹.

6 XMPP Registrar Considerations

This document requires no interaction with the [XMPP Registrar](#)¹⁰.

⁸XEP-0167: Jingle RTP Sessions <<https://xmpp.org/extensions/xep-0167.html>>.

⁹The 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/>>.

¹⁰The 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/>>.