XEP-0261: Jingle In-Band Bytestreams Transport Method

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

2011-09-23
Version 1.0

Status	Type	Short Name
Draft	Standards Track	jingle-ibb

This specification defines a Jingle transport method that results in sending data via the In-Band Bytestreams (IBB) protocol defined in XEP-0047. Essentially this transport method reuses XEP-0047 semantics for sending the data and defines native Jingle methods for starting and ending an IBB session.

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 Protocol **1**
- 2.1 Flow . . . 1
- 2.2 Establishing a Bytestream . . . 2
- 2.3 Exchanging Data . . . 4
- 2.4 Managing Multiple IBB Sessions . . . 5
- 2.5 Closing the Bytestream . . . 6

3 Processing Rules and Usage Guidelines **7**

4 Determining Support **7**

5 Security Considerations **8**
- 5.1 Encryption of Media . . . 8
- 5.2 Use of Base64 . . . 8

6 IANA Considerations **8**

7 XMPP Registrar Considerations **8**
- 7.1 Protocol Namespaces . . . 8
- 7.2 Protocol Versioning . . . 8
- 7.3 Jingle Transport Methods . . . 9

8 XML Schema **9**

9 Acknowledgements **10**

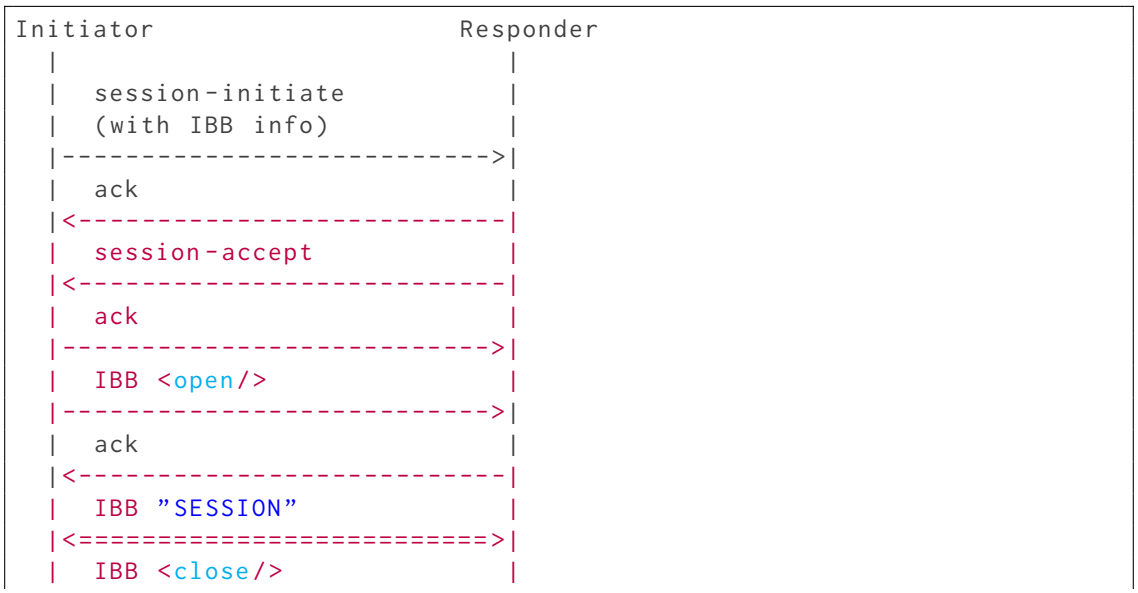
1 Introduction

Jingle (XEP-0166) [1] defines a framework for negotiating and managing data sessions over XMPP. In order to provide a flexible framework, the base Jingle specification defines neither data transport methods nor application formats, leaving that up to separate specifications. The current document defines a transport method for establishing and managing data exchanges between XMPP entities using the existing In-Band Bytestreams (IBB) protocol specified in In-Band Bytestreams (XEP-0047) [2]. This "jingle-ibb" method results in a streaming transport method suitable for use in Jingle application types where packet loss cannot be tolerated (e.g., file transfer); however, because the "jingle-ibb" transport method sends data over the XMPP channel itself (albeit not the Jingle signalling channel), it is intended as a transport of last resort when other streaming transports (e.g., Jingle SOCKS5 Bytestreams Transport Method (XEP-0260) [3]) cannot be negotiated.
The approach taken in this specification is to use the existing IBB mechanisms described in XEP-0047 for transporting the data, and to define Jingle-specific methods only to start and end the in-band bytestream.

2 Protocol

2.1 Flow

The basic flow is as follows.



[1] XEP-0166: Jingle <https://xmpp.org/extensions/xep-0166.html>.
[2] XEP-0047: In-Band Bytestreams <https://xmpp.org/extensions/xep-0047.html>.
[3] XEP-0260: Jingle SOCKS5 Bytestreams Transport Method <https://xmpp.org/extensions/xep-0260.html>.

```
  |--------------------------->|
  |  ack                       |
  |<---------------------------|
  |  session-terminate         |
  |<---------------------------|
  |  ack                       |
  |--------------------------->|
  |                            |
```

This flow is illustrated in the following sections (to prevent confusion these use an ”example” description instead of a real application type).

2.2 Establishing a Bytestream

First the initiator sends a Jingle session-initiate request.

Listing 1: Initiator sends session-initiate

```
<iq from='romeo@montague.lit/orchard'
    id='xn28s7gk'
    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='ex'>
      <description xmlns='urn:xmpp:example'/>
      <transport xmlns='urn:xmpp:jingle:transports:ibb:1'
                 block-size='4096'
                 sid='ch3d9s71'/>
    </content>
  </jingle>
</iq>
```

Note: The default value of the 'stanza' attribute is ”iq”, signifying use of <iq/> stanzas for data exchange; a value of ”message” signifies that <message/> stanzas are to be used for data exchange. See XEP-0047 for further discussion regarding use of these stanza types for data exchange.
The responder immediately acknowledges receipt (but does not yet accept the session).

Listing 2: Responder acknowledges session-initiate

```
<iq from='juliet@capulet.lit/balcony'
    id='xn28s7gk'
    to='romeo@montague.lit/orchard'
    type='result'/>
```

If the offer is acceptable, the responder returns a Jingle session-accept. If the responder wishes to use a smaller block-size, the responder can specify that in the session-accept by returning a different value for the 'block-size' attribute.

Listing 3: Responder definitively accepts the session

```
<iq from='juliet@capulet.lit/balcony'
    id='bsa91h56'
    to='romeo@montague.lit/orchard'
    type='set'>
  <jingle xmlns='urn:xmpp:jingle:1'
          action='session-accept'
          responder='juliet@capulet.lit/balcony'
          sid='a73sjjvkla37jfea'>
    <content creator='initiator' name='ex'>
      <description xmlns='urn:xmpp:example'/>
      <transport xmlns='urn:xmpp:jingle:transports:ibb:1'
                 block-size='2048'
                 sid='ch3d9s71'/>
    </content>
  </jingle>
</iq>
```

The initiator then acknowledges the session-accept.

Listing 4: Initiator acknowledges session-accept

```
<iq from='romeo@montague.lit/orchard'
    id='bsa91h56'
    to='juliet@capulet.lit/balcony'
    type='result'/>
```

In essence, the foregoing Jingle negotiation replaces the <open/> element from XEP-0047. However, to provide consistent layering of Jingle on top of IBB (thus enabling separation of existing IBB code from new Jingle code), the initiator now MUST also send the <open/> element, with the same 'block-size' and 'sid' values as for the Jingle <transport/> element it negotiated with the recipient (i.e., if the recipient sent a modified <transport/> element element containing a different block size, the initiator MUST use the negotiated values). This adds a roundtrip to the negotiation and could be considered a "no-op", but the extra roundtrip is inconsequential given that the parties will be exchanging base64-encoded data in-band.

Listing 5: Initiator sends IBB <open/>

```
<iq from='romeo@montague.net/orchard'
    id='jn3h8g65'
    to='juliet@capulet.com/balcony'
    type='set'>
```

```
  <open xmlns='http://jabber.org/protocol/ibb'
        block-size='2048'
        sid='ch3d9s71'
        stanza='iq'/>
</iq>
```

If no error occurs, the responder returns an IQ-result to the initiator.

Listing 6: Responder accepts IBB <open/>

```
<iq from='juliet@capulet.com/balcony'
    id='jn3h8g65'
    to='romeo@montague.net/orchard'
    type='result'/>
```

However, one of the errors described in XEP-0047 might occur; in particular, if the value of the IBB 'block-size' attribute sent by the initiator in the <open/> element does not match the value of the 'block-size' attribute communicated by the responder in the Jingle session-accept message then the responder SHOULD return a <resource-constraint/> error as described in XEP-0047.

2.3 Exchanging Data

Now the initiator can begin sending IBB packets using an IQ-set for each chunk as described in XEP-0047, where the responder will acknowledge each IQ-set in accordance with XMPP Core [4].

Listing 7: An IBB packet

```
<iq from='romeo@montague.net/orchard'
    id='ls72b58f'
    to='juliet@capulet.com/balcony'
    type='set'>
  <data xmlns='http://jabber.org/protocol/ibb' seq='0' sid='ch3d9s71'>
    qANQR1DBwU4DX7jmYZnncmUQB/9KuKBddzQH+tZ1ZywKK0yHKnq57kWq+RFtQdCJ
    WpdWpR0uQsuJe7+vh3NWn59/gTc5MDlX8dS9p0ovStmNcyLhxVgmqS8ZKhsblVeu
    IpQ0JgavABqibJolc3BKrVtVV1igKiX/N7Pi8RtY1K18toaMDhdEfhBRzO/XB0+P
    AQhYlRjNacGcslkhXqNjK5Va4tuOAPy2n1Q8UUrHbUd0g+xJ9Bm0G0LZXyvCWyKH
    kuNEHFQiLuCY6Iv0myq6iX6tjuHehZlFSh80b5BVV9tNLwNR5Eqz1klxMhoghJOA
  </data>
</iq>
```

Listing 8: An IBB ack

```
<iq from='juliet@capulet.com/balcony'
```

[4] RFC 6120: Extensible Messaging and Presence Protocol (XMPP): Core <http://tools.ietf.org/html/rfc6120>.

```
    id='ls72b58f'
    to='romeo@montague.net/orchard'
    type='result'/>
```

2.4 Managing Multiple IBB Sessions

As IBB is defined in XEP-0047, there is one session per bytestream (which can be used in both directions). However, because Jingle-IBB provides a management layer on top of IBB, it can be used to run multiple IBB sessions over a single bytestream. This can be done by sending a transport-info message that authorizes an additional session, as shown in the following example (although this example shows the initiator adding a session, the responder could just as well do so).

Listing 9: Initiator adds a session to the bytestream

```
<iq from='romeo@montague.lit/orchard'
    id='znb376s4'
    to='juliet@capulet.lit/balcony'
    type='set'>
  <jingle xmlns='urn:xmpp:jingle:1'
          action='transport-info'
          sid='a73sjjvkla37jfea'>
    <content creator='initiator' name='ex'>
      <transport xmlns='urn:xmpp:jingle:transports:ibb:1'
                 block-size='2048'
                 sid='bt8a71h6'/>
    </content>
  </jingle>
</iq>
```

Here the Jingle Session ID is the same ("a73sjjvkla37jfea") but the new IBB Session ID ("bt8a71h6") is different from the old IBB Session ID that is already in use ("ch3d9s71").
The initiator opens the second session by sending an IBB <open/> element, which the responder acknowledges (not shown).

Listing 10: Initiator sends IBB <open/>

```
<iq from='romeo@montague.net/orchard'
    id='yh3vs613'
    to='juliet@capulet.com/balcony'
    type='set'>
  <open xmlns='http://jabber.org/protocol/ibb'
        block-size='2048'
        sid='bt8a71h6'
        stanza='iq'/>
</iq>
```

The parties can then exchange data over the second session (see XEP-0047).
If a party wishes to close one session within a bytestream, it sends an IBB <close/> element as defined in XEP-0047 specifying the appropriate IBB SessionID.

Listing 11: Ending an IBB session

```
<iq from='romeo@montague.net/orchard'
    id='us71g45j'
    to='juliet@capulet.com/balcony'
    type='set'>
  <close xmlns='http://jabber.org/protocol/ibb'
         sid='bt8a71h6'/>
</iq>
```

The receiving party then acknowledges that the IBB session has been closed by returning an IQ-result.

Listing 12: Success response

```
<iq from='juliet@capulet.com/balcony'
    id='us71g45j'
    to='romeo@montague.net/orchard'
    type='result'/>
```

2.5 Closing the Bytestream

Whenever a party is finished with a particular session within the bytestream, it SHOULD send an IBB <close/> as shown above. This applies to all sessions, including the last one.
To close the bytestream itself (e.g., because the parties have finished using all sessions associated with the bytestream), a party sends a Jingle session-terminate action as defined in XEP-0166.

Listing 13: Initiator terminates the session

```
<iq from='romeo@montague.lit/orchard'
    id='hz81vf48'
    to='juliet@capulet.lit/balcony'
    type='set'>
  <jingle xmlns='urn:xmpp:jingle:1'
          action='session-terminate'
          sid='a73sjjvkla37jfea'>
    <reason><success/></reason>
  </jingle>
</iq>
```

The other party then acknowledges the session-terminate and the Jingle session is finished.

Listing 14: Responder acknowledges session-terminate

```
<iq from='juliet@capulet.lit/balcony'
    id='hz81vf48'
    to='romeo@montague.lit/orchard'
    type='result'/>
```

3 Processing Rules and Usage Guidelines

The same processing rules and usage guidelines defined in XEP-0047 apply to the Jingle IBB Transport Method.

4 Determining Support

To advertise its support for the Jingle In-Band Bytestreams Transport Method, when replying to Service Discovery (XEP-0030) [5] information requests an entity MUST return URNs for any version of this protocol that the entity supports -- e.g., "urn:xmpp:jingle:transports:ibb:1" for this version (see Namespace Versioning regarding the possibility of incrementing the version number).

Listing 15: Service discovery information request

```
<iq from='romeo@montague.lit/orchard'
    id='uw72g176'
    to='juliet@capulet.lit/balcony'
    type='get'>
  <query xmlns='http://jabber.org/protocol/disco#info'/>
</iq>
```

Listing 16: Service discovery information response

```
<iq from='juliet@capulet.lit/balcony'
    id='uw72g176'
    to='romeo@montague.lit/orchard'
    type='result'>
  <query xmlns='http://jabber.org/protocol/disco#info'>
    <feature var='urn:xmpp:jingle:1'/>
    <feature var='urn:xmpp:jingle:transports:ibb:1'/>
  </query>
</iq>
```

In order for an application to determine whether an entity supports this protocol, where possible it SHOULD use the dynamic, presence-based profile of service discovery defined in Entity Capabilities (XEP-0115) [6]. However, if an application has not received entity capabilities

[5] XEP-0030: Service Discovery <https://xmpp.org/extensions/xep-0030.html>.
[6] XEP-0115: Entity Capabilities <https://xmpp.org/extensions/xep-0115.html>.

information from an entity, it SHOULD use explicit service discovery instead.

5 Security Considerations

5.1 Encryption of Media

This specification, like XEP-0047 before it, does not directly support end-to-end encryption of the media sent over the transport.

5.2 Use of Base64

See XEP-0047 for security considerations related to the use of Base64.

6 IANA Considerations

This document requires no interaction with the Internet Assigned Numbers Authority (IANA) [7].

7 XMPP Registrar Considerations

7.1 Protocol Namespaces

The XMPP Registrar [8] includes 'urn:xmpp:jingle:transports:ibb:1' in its registry of protocol namespaces at <https://xmpp.org/registrar/namespaces.html>, as described in Section 4 of XMPP Registrar Function (XEP-0053) [9].

7.2 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.

[7] The 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/>.

[8] The 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/>.

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

7.3 Jingle Transport Methods

The XMPP Registrar includes "jingle-ibb" in its registry of Jingle transport methods at <https://xmpp.org/registrar/jingle-transports.html>. The registry submission is as follows.

```
<transport>
  <name>ibb</name>
  <desc>
    A method for data exchange over In-Band Bytestreams.
  </desc>
  <type>streaming</type>
  <doc>XEP-0261</doc>
</transport>
```

8 XML Schema

```
<?xml version='1.0' encoding='UTF-8'?>

<xs:schema
    xmlns:xs='http://www.w3.org/2001/XMLSchema'
    targetNamespace='urn:xmpp:jingle:transports:ibb:1'
    xmlns='urn:xmpp:jingle:transports:ibb:1'
    elementFormDefault='qualified'>

  <xs:annotation>
    <xs:documentation>
      The protocol documented by this schema is defined in
      XEP-0261: http://www.xmpp.org/extensions/xep-0261.html
    </xs:documentation>
  </xs:annotation>

   <xs:element name='transport'>
     <xs:complexType>
      <xs:simpleContent>
        <xs:extension base='empty'>
          <xs:attribute name='block-size'
                        type='xs:short'
                        use='required'/>
          <xs:attribute name='sid'
                        type='xs:string'
                        use='required'/>
          <xs:attribute name='stanza'
                        use='optional'
                        default='iq'>
            <xs:simpleType>
              <xs:restriction base='xs:NCName'>
```

```
                <xs:enumeration value='iq'/>
                <xs:enumeration value='message'/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>

  <xs:simpleType name='empty'>
    <xs:restriction base='xs:string'>
      <xs:enumeration value=''/>
    </xs:restriction>
  </xs:simpleType>

</xs:schema>
```

9 Acknowledgements

Thanks to Paul Aurich, Fabio Forno, and Marcus Lundblad for their feedback.