XEP-0047: In-Band Bytestreams

Justin Karneges
mailto:justin@affinix.com
xmpp:justin@andbit.net

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

Status	Type	Short Name
Final	Standards Track	ibb

This specification defines an XMPP protocol extension that enables any two entities to establish a one-to-one bytestream between themselves, where the data is broken down into smaller chunks and transported in-band over XMPP.

Contents

1 Introduction

This document describes In-Band Bytestreams (IBB), an XMPP protocol extension that enables two entities to establish a virtual bytestream over which they can exchange Base64-encoded chunks of data over XMPP itself. Because IBB provides a generic bytestream, its usage is open-ended. To date it has been used as a fallback method for sending files (see SI File Transfer [1] and Jingle File Transfer [2]) when out-of-band methods such as SOCKS5 Bytestreams [3] are not available. However, IBB could also be useful for any kind of relatively low-bandwidth activity, such as games, shell sessions, or encrypted text.

2 Protocol

IBB as specified in this document defines two protocol aspects:

1. How to set up and tear down an IBB session using <open/> and <close/> elements sent within IQ stanzas.

2. How to send chunks of IBB data using IQ (or message) stanzas containing <data/> elements.

Other methods can be used for setup and teardown, such as Jingle [4] as specified in Jingle In-Band Bytestreams Transport Method [5].

2.1 Creating a Bytestream

In order to set up an in-band bytestream, the initiator sends an IQ stanza of type "set" containing an <open/> element qualified by the 'http://jabber.org/protocol/ibb' namespace. This element possesses three attributes:

- The REQUIRED 'block-size' attribute defines the maximum size in bytes of each data chunk (which MUST NOT be greater than 65535).

- The REQUIRED 'sid' attribute defines a unique session ID for this IBB session (which MUST match the NMTOKEN datatype).

- The OPTIONAL 'stanza' attribute defines whether the data will be sent using IQ stanzas or message stanzas (the default value is "iq")

---

[1]XEP-0096: SI File Transfer <http://xmpp.org/extensions/xep-0096.html>.
[2]XEP-0234: Jingle File Transfer <http://xmpp.org/extensions/xep-0234.html>.
[3]XEP-0065: SOCKS5 Bytestreams <http://xmpp.org/extensions/xep-0065.html>.
[4]XEP-0166: Jingle <http://xmpp.org/extensions/xep-0166.html>.
[5]XEP-0261: Jingle In-Band Bytestreams Transport Method <http://xmpp.org/extensions/xep-0261.html>.

Note: It is RECOMMENDED to send IBB data using IQ stanzas instead of message stanzas because IQ stanzas provide feedback to the sender regarding delivery to the recipient).

Listing 1: Initiator requests session

```
<iq from='romeo@montague.net/orchard'
    id='jn3h8g65'
    to='juliet@capulet.com/balcony'
    type='set'>
  <open xmlns='http://jabber.org/protocol/ibb'
        block-size='4096'
        sid='i781hf64'
        stanza='iq'/>
</iq>
```

If the responder wishes to proceed with the IBB session, it returns an IQ-result to the initiator.

Listing 2: Responder accepts session

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

If the responder does not support IBB, it returns a <service-unavailable/> or <feature-not-implemented/> error.

Listing 3: Responder does not support IBB

```
<iq from='juliet@capulet.com/balcony'
    id='jn3h8g65'
    to='romeo@montague.net/orchard'
    type='error'>
  <error type='cancel'>
    <service-unavailable xmlns='urn:ietf:params:xml:ns:xmpp-stanzas'/>
  </error>
</iq>
```

If the responder supports IBB but would prefer a smaller block-size, it returns a <resource-constraint/> error.

Listing 4: Responder prefers smaller chunks

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

```
      <resource-constraint xmlns='urn:ietf:params:xml:ns:xmpp-stanzas'/>
  </error>
</iq>
```

If the responder supports IBB but does not wish to proceed with the session, it returns a <not-acceptable/> error.

Listing 5: Responder does not wish to proceed

```
<iq from='juliet@capulet.com/balcony'
    id='jn3h8g65'
    to='romeo@montague.net/orchard'
    type='error'>
  <error type='cancel'>
    <not-acceptable xmlns='urn:ietf:params:xml:ns:xmpp-stanzas'/>
  </error>
</iq>
```

2.2 Sending Data

If the responder informs the initiator that it wishes to proceed with the session, the initiator can begin to send data over the bytestream (in addition, because the bytestream is bidirectional, the responder can also send data; see the Bidirectionality section of this document for details).

Each chunk of data is contained in a <data/> element qualified by the 'http://jabber.org/protocol/ibb' namespace. The data element SHOULD be sent in an IQ stanza to enable proper tracking and throttling, but instead MAY be sent in a message stanza. The data to be sent, prior to base64-encoding and prior to any wrapping in XML, MUST NOT be larger than the 'block-size' determined in the bytestream negotiation.

Listing 6: Sending data in an IQ stanza

```
<iq from='romeo@montague.net/orchard'
    id='kr91n475'
    to='juliet@capulet.com/balcony'
    type='set'>
  <data xmlns='http://jabber.org/protocol/ibb' seq='0' sid='i781hf64'>
    qANQR1DBwU4DX7jmYZnncmUQB/9KuKBddzQH+tZ1ZywKK0yHKnq57kWq+RFtQdCJ
    WpdWpR0uQsuJe7+vh3NWn59/gTc5MDlX8dS9p0ovStmNcyLhxVgmqS8ZKhsblVeu
    IpQ0JgavABqibJolc3BKrVtVV1igKiX/N7Pi8RtY1K18toaMDhdEfhBRzO/XB0+P
    AQhYlRjNacGcslkhXqNjK5Va4tuOAPy2n1Q8UUrHbUd0g+xJ9Bm0G0LZXyvCWyKH
    kuNEHFQiLuCY6Iv0myq6iX6tjuHehZlFSh80b5BVV9tNLwNR5Eqz1klxMhoghJOA
  </data>
</iq>
```

Each chunk of data is included as the XML character data of the <data/> element after being encoded as Base64 as specified in Section 4 of RFC 4648 [6]. Each block MUST be a valid base64 block with padding at the end if needed.

The <data/> element MUST possess a 'seq' attribute; this is a 16-bit unsigned integer that acts as a counter for data chunks sent in a particular direction within this session. The 'seq' value starts at 0 (zero) for each sender and MUST be incremented for each packet sent by that entity. Thus, the second chunk sent has a 'seq' value of 1, the third chunk has a 'seq' value of 2, and so on. The counter loops at maximum, so that after value 65535 (2^{15} - 1) the 'seq' MUST start again at 0.

The <data/> element MUST also possess a 'sid' attribute that ties the data chunk to this particular IBB session.

In the case of IQ stanzas, if the packet can be processed then the recipient MUST reply with an IQ stanza of type "result".

Listing 7: Acknowledging data received via IQ

```
<iq from='juliet@capulet.com/balcony'
    id='kr91n475'
    to='romeo@montague.net/orchard'
    type='result'/>
```

The sender of a data chunk need not wait for these acknowledgements before sending further stanzas. However, it is RECOMMENDED that the sender does wait in order to minimize the potential for rate-limiting penalties or throttling.

It is possible that delivery of the stanza might fail, in which case the sender's or recipient's server shall return an appropriate error:

1. Because the recipient has gone offline, the recipient's server returns a <recipient-unavailable/> error with a type of 'wait'.

2. Because a server-to-server link has gone down, the sender's server returns a <remote-server-timeout/> error with a type of 'wait'.

Upon receiving an error related to delivery of a <message/> or <iq/> stanza, the sender SHOULD temporarily suspend the stream but either (1) retry sending at a later time or (2) renegotiate the higher-level session (if any) such as a session controlled via Jingle File Transfer [7].

It is also possible that there is a problem with the data packet itself, in which case the recipient shall return an appropriate error:

1. Because the session ID is unknown, the recipient returns an <item-not-found/> error with a type of 'cancel'.

---

[6]RFC 4648: The Base16, Base32, and Base64 Data Encodings <http://tools.ietf.org/html/rfc4648>.
[7]XEP-0234: Jingle File Transfer <http://xmpp.org/extensions/xep-0234.html>.

2. Because the sequence number has already been used, the recipient returns an <unexpected-request/> error with a type of 'cancel'.

3. Because the data is not formatted in accordance with Section 4 of RFC 4648, the recipient returns a <bad-request/> error with a type of 'cancel'.

Upon receiving an error related to the data packet, the sender MUST close the bytestream as described under Closing the Bytestream.

Data packets MUST be processed in the order they are received. If an out-of-sequence packet is received for a particular direction within a bytestream (determined by checking the 'seq' attribute), then this indicates that a packet has been lost. The recipient MUST NOT process the data of such an out-of-sequence packet, nor any that follow it within the same bytestream; instead, the recipient MUST close the bytestream as described in the next section.

2.3 Closing the Bytestream

To close the bytestream, either party sends an IQ-set containing a <close/> element.

Listing 8: Closing the bytestream

```
<iq from='romeo@montague.net/orchard'
    id='us71g45j'
    to='juliet@capulet.com/balcony'
    type='set'>
  <close xmlns='http://jabber.org/protocol/ibb' sid='i781hf64'/>
</iq>
```

The receiving party then acknowledges that the bytestream has been closed by returning an IQ-result (however, the receiving party might wait until it has had a chance to send any remaining data in the other direction, if the bytestream is bidirectional; in this case, the party that sent the original <close/> element SHOULD wait to receive the IQ response from the receiving party before considering the bytestream to be closed).

Listing 9: Success response

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

It is possible that the recipient of the close notification does not know about the bytestream, in which case it would return an <item-not-found/> error.

Listing 10: Recipient does not know about the IBB session

```
<iq type='error'
```

5

```
      from='juliet@capulet.com/balcony'
      to='romeo@montague.net/orchard'
      id='us71g45j'>
  <error type='cancel'>
      <item-not-found xmlns='urn:ietf:params:xml:ns:xmpp-stanzas'/>
  </error>
</iq>
```

In either case, both parties MUST consider the bytestream to be closed.

3   Bidirectionality

An in-band bytestream is bidirectional. Therefore, either party to the bytestream is allowed to send data. Each sender MUST initialize the 'seq' attribute to zero and increment the 'seq' value by one with each chunk of data it sends. Each recipient MUST track chunks based on the 'seq' values it receives. The 'seq' values in each direction are independent of the values in the other direction. Thus there are two data sequences for each Session ID.

4   Use of Message Stanzas

It is RECOMMENDED to use IQ stanzas when sending data packets. However, an application MAY use message stanzas instead. If message stanzas are used when sending data packets, the sender SHOULD also use Advanced Message Processing [8] or some other stanza flow-control method. For proper tracking of delivery and processing errors related to data packets, the 'id' attribute SHOULD be used with message stanzas.

Listing 11: Sending data in a message stanza

```
<message from='romeo@montague.net/orchard'
         id='dsw71gj3'
         to='juliet@capulet.com/balcony'>
  <data xmlns='http://jabber.org/protocol/ibb' seq='0' sid='i781hf64'>
    qANQR1DBwU4DX7jmYZnncmUQB/9KuKBddzQH+tZ1ZywKK0yHKnq57kWq+RFtQdCJ
    WpdWpR0uQsuJe7+vh3NWn59/gTc5MDlX8dS9p0ovStmNcyLhxVgmqS8ZKhsblVeu
    IpQ0JgavABqibJolc3BKrVtVV1igKiX/N7Pi8RtY1K18toaMDhdEfhBRzO/XB0+P
    AQhYlRjNacGcslkhXqNjK5Va4tuOAPy2n1Q8UUrHbUd0g+xJ9Bm0G0LZXyvCWyKH
    kuNEHFQiLuCY6Iv0myq6iX6tjuHehZlFSh80b5BVV9tNLwNR5Eqz1klxMhoghJOA
  </data>
</message>
```

---

[8] XEP-0079: Advanced Message Processing <http://xmpp.org/extensions/xep-0079.html>.

5 Usage Guidelines

- Generally, in-band bytestreams SHOULD be used only as a last resort. SOCKS5 Bytestreams will almost always be preferable.

- A server MAY rate limit a connection, depending on the size and frequency of data packets.

- A server MAY disconnect a connection that sends overly large packets as defined by its local service policy.

- It is RECOMMENDED to use a 'block-size' of 4096.

6 Security Considerations

The In-Band Bytestreams protocol is as secure as the underlying XMPP transport. The application that uses IBB could have its own security layer, but this is outside of the scope of IBB.

An entity MUST verify any Base64 data received. An implementation MUST reject (not ignore) any characters that are not explicitly allowed by the Base64 alphabet; this helps to guard against creation of a covert channel that could be used to "leak" information. An implementation MUST NOT break on invalid input and MUST reject any sequence of Base64 characters containing the pad ('=') character if that character is included as something other than the last character of the data (e.g., "=AAA" or "BBBB=CCC"); this helps to guard against buffer overflow attacks and other attacks on the implementation. Base64 encoding visually hides otherwise easily recognized information, such as passwords, but does not provide any computational confidentiality. Base64 encoding MUST follow the definition in Section 4 of RFC 4648.

7 IANA Considerations

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

---

[9] 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 XMPP Registrar Considerations

8.1 Protocol Namespaces

The XMPP Registrar [10] includes 'http://jabber.org/protocol/ibb' in its registry of XML namespaces at <http://xmpp.org/registrar/namespaces.html>.

9 XML Schema

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

<xs:schema
    xmlns:xs='http://www.w3.org/2001/XMLSchema'
    targetNamespace='http://jabber.org/protocol/ibb'
    xmlns='http://jabber.org/protocol/ibb'
    elementFormDefault='qualified'>

  <xs:annotation>
    <xs:documentation>
      The protocol documented by this schema is defined in
      XEP-0047: http://www.xmpp.org/extensions/xep-0047.html
    </xs:documentation>
  </xs:annotation>

   <xs:element name='open'>
     <xs:complexType>
      <xs:simpleContent>
        <xs:extension base='empty'>
          <xs:attribute name='block-size' type='xs:unsignedShort' use=
              'required'/>
          <xs:attribute name='sid' type='xs:NMTOKEN' 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>
```

---

[10]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 <http://xmpp.org/registrar/>.

```xml
  <xs:element name='close'>
    <xs:complexType>
     <xs:simpleContent>
       <xs:extension base='empty'>
         <xs:attribute name='sid' type='xs:NMTOKEN' use='required'/>
       </xs:extension>
     </xs:simpleContent>
    </xs:complexType>
  </xs:element>

  <xs:element name='data'>
    <xs:complexType>
     <xs:simpleContent>
       <xs:extension base='xs:string'>
         <xs:attribute name='seq' type='xs:unsignedShort' use='
             required'/>
         <xs:attribute name='sid' type='xs:NMTOKEN' use='required'/>
       </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>
```

10 Acknowledgements