XEP-0265: Out-of-Band Stream Data

Dirk Meyer
mailto:dmeyer@tzi.de
xmpp:dmeyer@jabber.org

2009-04-02
Version 0.1

Status	Type	Short Name
Deferred	Standards Track	NOT_YET_ASSIGNED

This specification defines how to send parts of an XML stream over a direct connection between peers. This allows to send large stanzas or binary data without blocking the XML stream for other stanzas.

Legal

Copyright

Permissions

Warranty

Liability

Conformance

Contents

1 Introduction

XMPP uses one XML stream between two peers, either between client and server, between servers, or between clients directly. If a client sends a stanza to a client, either routed over the server or directly, it can not send another stanza at the same time. If an XMPP extension defines very large stanzas, the communication is blocked until this stanza is fully sent to the server. A client or server may also define a maximum stanza size for communication. A future extension to query a remote file server could create such huge stanzas: a directory listing with thousands of photos, each with additional metadata, could produce stanzas of one megabyte or longer. The same is true for TV listings of a remote TV tuner device. Besides text data, a stanza may also contain binary data Base64 encoded. This also increases the stanza size.
This document defines a mechanism to transmit parts of a stanza over a separate end-to-end connection using Jingle. XML data and binary data (without being Base64 encoded) can be transmitted over the external link to keep the actual XML stanza small.

2 Protocol Flow

If a client wants to send a large stanza to a peer, it can remove large elements from the stanza and replace it with a <oob/> element of the 'urn:xmpp:jingle:apps:out-of-band:0' namespace. The definition of "large" depends on the use case and available bandwidth of the stream to the server. It is RECOMMENDED to send stanzas smaller than 4096 Bytes directly because the overhead of the additional stream is too high. If a client knows in advance that it will send or receive several large stanzas or binary data it SHOULD open the out-of-band data stream. One larger chunk may not be worth opening a Jingle session. A client MUST NOT send In-Band Bytestream stanzas out of band because there may be a reason why it is an In-Band Bytestream and not something else such as SOCKS5.
Before a client sends such a stanza to its peer, it MUST open the out of band stream first. It has to initiate a Jingle session and MUST NOT send the stanza it wants to send until the out-of-band stream is open. The following example is based on 'Discovering the Items Associated with a Jabber Entity' Service Discovery (XEP-0030) [1].

Listing 1: Requesting All Items

```
<iq from='hamlet@example.com/denmark'
    to='hamlet@example.com/bot'
    id='hfgte45w'
    type='get'>
  <query xmlns='http://jabber.org/protocol/disco#items'/>
</iq>
```

Normally we would expect the peer to answer with an IQ return stanza with the list of items. But if the listing is very large, the client may decide to send it over an extra stream. Instead

[1] XEP-0030: Service Discovery <https://xmpp.org/extensions/xep-0030.html>.

of sending the IQ result, it opens a Jingle session.

Listing 2: Home Server Initiates Jingle Session

```
<iq from='hamlet@example.com/bot'
    to='hamlet@example.com/denmark'
    id='xn28s7gk'
    type='set'>
  <jingle xmlns='urn:xmpp:jingle:0'>
          action='session-initiate'
          initiator='hamlet@example.com/bot'
          sid='a73sjjvkla37jfea'>
    <content creator='initiator' name='oob'>
      <description xmlns='urn:xmpp:jingle:apps:out-of-band:0'/>
      <transport xmlns='urn:xmpp:jingle:transports:s5b:0'
                 sid='vj3hs98y'
                 mode='tcp'>
        <streamhost
            jid='hamlet@example.com/bot'
            host='192.168.4.1'
            port='5086'/>
      </transport>
      <security xmlns='urn:xmpp:jingle:security:xtls:0'>
        <fingerprint>RomeoX509CertificateHash</fingerprint>
        <method name='x509'/>
      </security>
    </content>
  </jingle>
</iq>
```

The clients open a Jingle session according to Jingle (XEP-0166) [2] with a Jingle SOCKS5 Bytestreams Transport Method (XEP-0260) [3]. For the sake of simplicity this protocol flow is not described here. If the clients opened the Out-of-Band Data Stream, the listing is sent over that stream and a reference is returned in the IQ result of the original request.

Listing 3: IQ Result with Out-of-Band Stream Data Reference

```
<iq from='hamlet@example.com/bot'
    to='hamlet@example.com/denmark'
    id='hfgte45w'
    type='result'>
  <oob xmlns='urn:xmpp:jingle:apps:out-of-band:0'
       id='hfgte45w-1'
       size='6022'
       hash='sha1+552da749930852c69ae5d2141d3766b1'
       type='text/xml'/>
```

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

```
</iq>
```

The receiver MUST replace the <oob/> element with the content with the given id from the Out-of-Band Data Stream. The size and hash arguments are optional, the type MUST be 'text/xml' since an IQ stanza MUST have one child element. The content send out of band MUST have a valid XML header <?xml version='1.0' ?> with one root element. The XML header is ignored and the root element MUST replace the <oob> element.
It is possible to use out of band data everywhere in the XML stream and not only as first element child of the IQ stanza. The XML schema MUST explicit allow the usage of the <oob/> element; it is not possible to replace any arbitrarily part of a stanza with the <oob/> element. The reason for this restriction is to keep implementations simpler if they do not have to expect out of band data everywhere and to keep the stream compliant to the XML schemas.
If the <oob/> element is not the top level child of an IQ or message stanza, the type attribute does not has to be 'text/xml' and may even be omitted. In that case the content with the given id send out of band MUST be treated as if it was embedded in the XML stream using Base64 encoding. This is useful for sending larger chunks of binary data.

3 Out-of-Band Data Stream

The Out-of-Band Data Stream multiplexes several items into one stream to re-use the stream for several XML elements or binary data without blocking the out-of-band stream with one large item. This avoids negotiating a new Jingle session for each piece of data. The syntax is similar to HTTP 1.1 chunked transfer. Each chunk of data has a one line header with the number of bytes in hex of the data and the content identifier. The last chunk of each piece of content is always a chunk with a length of 0.

Listing 4: Stream Data Definition

```
stream          = *(chunk | last-chunk)
chunk           = chunk-size id CRLF chunk-data CRLF
hexdig-nonzero  = %x31-39 ; "1"-"9"
chunk-size      = hexdig-nonzero *HEXDIG
id              = *(ALPHA | DIGIT)
last-chunk      = 1*("0") id CRLF CRLF
```

In the given example the <oob/> element specifies that the query result is sent out of band and has a size of 6022 bytes. The out of band content requires a valid XML header which adds another 23 bytes. If each chunk has 4096 bytes (0x1000 in hex) the data is split into two chunks (4096 and 1949 bytes). The following data is sent over the out of band stream:

Listing 5: Stream Data with XEP-0030 Items

```
1000 hfgte45w CRLF
<?xml version='1.0' ?>
```

```
<query xmlns='http://jabber.org/protocol/disco#items'>
  ... 4096 Bytes including the XML header ... CRLF
79d hfgte45w CRLF
  ... Last 1949 Bytes ... CRLF
0 hfgte45w CRLF CRLF
```

4 Abort Content Sending

If a client is not not interested in the out of band data (anymore) it MAY abort the sending of a content with a given identifier to save bandwidth. This may happen for large binary data and the client was only interested in the first bytes, e.g. to detect the file type and that it can not decode it.

Listing 6: Abort Content

```
<iq from='hamlet@example.com/denmark'
    to='hamlet@example.com/bot'
    id='hfytewp9'
    type='set'>
  <abort xmlns='urn:xmpp:jingle:apps:out-of-band:0'
         id='hfgte45w'/>
</iq>
```

The peer SHOULD send a last chunk with the length of zero out of band and acknowledge the abortion.

Listing 7: Acknowledge Abort

```
<iq from='hamlet@example.com/bot'
    to='hamlet@example.com/denmark'
    id='hfytewp9'
    type='result'/>
```

5 Security Considerations

The Jingle session SHOULD include TLS as specified in Jingle XTLS [4]. Even if the peers can not verify each others certificates, the leap of faith approach provides at least the same amount of security as if the data where send inside the XML stream.

[4] Extensible Messaging and Presence Protocol (XMPP) End-to-End Encryption Using Transport Layer Security ("XTLS") <http://tools.ietf.org/html/draft-meyer-xmpp-e2e-encryption>.

6 IANA Considerations

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

7 XMPP Registrar Considerations

XMPP Registrar considerations will be provided in a later version of this document.

8 XML Schema

The XML schema will be provided in a later version of this document.

[5] 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/>.