XEP-0206: XMPP Over BOSH

Ian Paterson
mailto:ian.paterson@clientside.co.uk
xmpp:ian@zoofy.com

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

Lance Stout
mailto:lance@andyet.com
xmpp:lance@lance.im

Winfried Tilanus
mailto:winfried@tilanus.com

2014-04-09
Version 1.4

Status	Type	Short Name
Draft	Standards Track	xbosh

This specification defines how the Bidirectional-streams Over Synchronous HTTP (BOSH) technology can be used to transport XMPP stanzas. The result is an HTTP binding for XMPP communications that is useful in situations where a device or client is unable to maintain a long-lived TCP connection to an XMPP server.

Legal

Copyright

Permissions

Warranty

Liability

Conformance

Contents

1 Introduction

The BOSH (XEP-0124) [1] protocol defines how arbitrary XML elements can be transported efficiently and reliably over HTTP in both directions between a client and server. This document defines some minor extensions to BOSH that enable XMPP streams (as specified in XMPP Core [2]) to be bound to HTTP.

2 <body/> Wrapper Element

If the BOSH <body/> wrapper is not empty, then it SHOULD contain one of the following:

- A complete <stream:features/> element (in which case the BOSH <body/> element MUST include the namespace xmlns:stream='http://etherx.jabber.org/streams').
- A complete element used for SASL negotiation and qualified by the 'urn:ietf:params:xml:ns:xmpp-sasl' namespace.
- One or more complete <message/>, <presence/>, and/or <iq/> elements qualified by the 'jabber:client' namespace.
- A <stream:error/> element (in which case the BOSH <body/> element MUST include the namespace xmlns:stream='http://etherx.jabber.org/streams' and it MUST feature the 'remote-stream-error' terminal error condition), preceded by zero or more complete <message/>, <presence/>, and/or <iq/> elements qualified by the 'jabber:client' namespace.

Note: Many existing XMPP-specific implementations of BOSH clients and connection managers do not specify the namespace of <message/>, <presence/>, or <iq/> elements, since that allows them to forward stanzas without modification (the XMPP <stream:stream/> wrapper element used with TCP typically sets the default namespace to 'jabber:client'). They instead simply assume that the full content of the 'jabber:client' namespace is a subset of the 'http://jabber.org/protocol/httpbind' namespace.
Note: Inclusion of TLS negotiation elements is allowed but is NOT RECOMMENDED. The definition of how TLS might be implemented over BOSH is currently beyond the scope of this document. Instead, channel encryption SHOULD be completed at the HTTP (transport) layer, not the XMPP (application) layer.

3 Session Creation Request

The client SHOULD include a 'version' attribute qualified by the 'urn:xmpp:xbosh' namespace in its session creation request. This attribute corresponds to the 'version' attribute of the

[1] XEP-0124: Bidirectional-streams Over Synchronous HTTP <https://xmpp.org/extensions/xep-0124.html>.
[2] RFC 6120: Extensible Messaging and Presence Protocol (XMPP): Core <http://tools.ietf.org/html/rfc6120>.

XMPP <stream:stream/> element as defined in RFC 6120 [3]. The connection manager SHOULD forward the value to the XMPP server accordingly.

Listing 1: Requesting a session with a version attribute

```
POST /webclient HTTP/1.1
Host: httpcm.jabber.org
Accept-Encoding: gzip, deflate
Content-Type: text/xml; charset=utf-8
Content-Length: 104

<body content='text/xml; charset=utf-8'
      from='user@example.com'
      hold='1'
      rid='1573741820'
      to='example.com'
      route='xmpp:example.com:9999'
      wait='60'
      xml:lang='en'
      xmpp:version='1.0'
      xmlns='http://jabber.org/protocol/httpbind'
      xmlns:xmpp='urn:xmpp:xbosh'/>
```

The connection manager can use the 'from' and 'to' attributes to populate the same attributes on the stream header sent from the connection manager to the XMPP server, or can use them for session management purposes specific to the connection manager implementation.
Note: Unlike the protocol defined in Jabber HTTP Polling (XEP-0025) [4], an opening <stream:stream> tag is not sent to the connection manager (since BOSH <body/> elements MUST not contain partial XML elements). Any XML streams between the connection manager and an XMPP server are the responsibility of the connection manager (and beyond the scope of this document).

4 Session Creation Response

The connection manager SHOULD include a 'version' attribute (qualified by the 'urn:xmpp:xbosh' namespace) and a <stream:features/> element (qualified by the 'http://etherx.jabber.org/streams' namespace) in a response as soon as they are available, either in its session creation response, or (if it has not yet received them from the XMPP server) in any subsequent response.
If the connection manager supports stream restarts, it MUST advertise that fact by including a 'restartlogic' attribute (qualified by the 'urn:xmpp:xbosh' namespace) whose value is set

[3] RFC 6120: Extensible Messaging and Presence Protocol (XMPP): Core <http://tools.ietf.org/html/rfc6120>.
[4] XEP-0025: Jabber HTTP Polling <https://xmpp.org/extensions/xep-0025.html>.

to ”true” [5]. It is STRONGLY RECOMMENDED for all XMPP connection managers to support stream restarts, since they are an integral aspect of stream negotiation in XMPP. However, note that some older BOSH implementations do not explicitly advertise support for stream restarts.
Note: The same procedure applies to the *obsolete* XMPP-specific ’authid’ attribute of the BOSH <body/> element, which contains the value of the XMPP stream ID generated by the XMPP server. This value is needed only by legacy XMPP clients in order to complete digest authentication using the *obsolete* Non-SASL Authentication (XEP-0078) [6] protocol. [7]

Listing 2: Session creation response with stream features

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: 674

<body wait='60'
      inactivity='30'
      polling='5'
      requests='2'
      hold='1'
      from='example.com'
      accept='deflate,gzip'
      sid='SomeSID'
      charsets='ISO_8859-1 ISO-2022-JP'
      xmpp:restartlogic='true'
      xmpp:version='1.0'
      authid='ServerStreamID'
      xmlns='http://jabber.org/protocol/httpbind'
      xmlns:xmpp='urn:xmpp:xbosh'
      xmlns:stream='http://etherx.jabber.org/streams'>
  <stream:features>
    <mechanisms xmlns='urn:ietf:params:xml:ns:xmpp-sasl'>
      <mechanism>SCRAM-SHA-1</mechanism>
      <mechanism>PLAIN</mechanism>
    </mechanisms>
  </stream:features>
</body>
```

If no <stream:features/> element is included in the connection manager’s session creation response, then the client SHOULD send empty request elements until it receives a response containing a <stream:features/> element.

[5] In accordance with Section 3.2.2.1 of XML Schema Part 2: Datatypes, the allowable lexical representations for the xs:boolean datatype are the strings ”0” and ”false” for the concept ’false’ and the strings ”1” and ”true” for the concept ’true’; implementations MUST support both styles of lexical representation.
[6] XEP-0078: Non-SASL Authentication <https://xmpp.org/extensions/xep-0078.html>.
[7] Separate ’sid’ and ’authid’ attributes are required because the connection manager is not necessarily part of a single XMPP server (e.g., it may handle HTTP connections on behalf of multiple XMPP servers).

Listing 3: Subsequent response with stream features

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: 483

<body xmpp:version='1.0'
      authid='ServerStreamID'
      xmlns='http://jabber.org/protocol/httpbind'
      xmlns:xmpp='urn:xmpp:xbosh'
      xmlns:stream='http://etherx.jabber.org/streams'>
  <stream:features>
    <mechanisms xmlns='urn:ietf:params:xml:ns:xmpp-sasl'>
      <mechanism>SCRAM-SHA-1</mechanism>
      <mechanism>PLAIN</mechanism>
    </mechanisms>
  </stream:features>
</body>
```

Note: The client SHOULD ignore any Transport Layer Security (TLS) feature since BOSH channel encryption SHOULD be negotiated at the HTTP layer.
TLS compression (as defined in RFC 6120 [8]) and Stream Compression (as defined in Stream Compression (XEP-0138) [9]) are NOT RECOMMENDED since compression SHOULD be negotiated at the HTTP layer using the 'accept' attribute of the BOSH session creation response. TLS compression and Stream Compression SHOULD NOT be used at the same time as HTTP content encoding.
Note: The 'version' attribute qualified by the 'urn:xmpp:xbosh' namespace SHOULD also be included on the request and response when adding new streams to a session.

5 Authentication and Resource Binding

A success case for authentication and resource binding using the XMPP protocols is shown below. For detailed specification of these protocols (including error cases), refer to RFC 6120 [10]. The server MAY offer the SASL-EXTERNAL method, for example when BOSH is used in conjunction with HTTP authentication or TLS authentication on the HTTP level.

Listing 4: SASL authentication step 1

```
POST /webclient HTTP/1.1
Host: httpcm.example.com
Accept-Encoding: gzip, deflate
Content-Type: text/xml; charset=utf-8
Content-Length: 231
```

[8] RFC 6120: Extensible Messaging and Presence Protocol (XMPP): Core <http://tools.ietf.org/html/rfc6120>.
[9] XEP-0138: Stream Compression <https://xmpp.org/extensions/xep-0138.html>.
[10] RFC 6120: Extensible Messaging and Presence Protocol (XMPP): Core <http://tools.ietf.org/html/rfc6120>.

```
<body rid='1573741821'
      sid='SomeSID'
      xmlns='http://jabber.org/protocol/httpbind'>
  <auth xmlns='urn:ietf:params:xml:ns:xmpp-sasl'
        mechanism='SCRAM-SHA-1'>
     biwsbj1qdWxpZXQscj1vTXNUQUF3QUFBQU1BQUFBTlAwVEFBQUFBQUJQVTBBQQ==
   </auth>
</body>
```

Listing 5: SASL authentication step 2

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: 294

<body xmlns='http://jabber.org/protocol/httpbind'>
  <challenge xmlns='urn:ietf:params:xml:ns:xmpp-sasl'>
    cj1vTXNUQUF3QUFBQU1BQUFBTlAwVEFBQUFBQUJQVTBBQWUxMjQ2OTViLTY5Y
    TktNGRlNi05YzMwLWI1MWIzODA4YzU5ZSxzPU5qaGtZVE0wTURndE5HWTBaaT
    AwTmpkbUxUa3hNbVV0TkRsbU5UTm1ORE5rTURNeixpPTQwOTY=
  </challenge>
</body>
```

Listing 6: SASL authentication step 3

```
POST /webclient HTTP/1.1
Host: httpcm.example.com
Accept-Encoding: gzip, deflate
Content-Type: text/xml; charset=utf-8
Content-Length: 295

<body rid='1573741822'
      sid='SomeSID'
      xmlns='http://jabber.org/protocol/httpbind'>
  <response xmlns='urn:ietf:params:xml:ns:xmpp-sasl'>
    Yz1iaXdzLHI9b01zVEFBd0FBQUFNQUFBQU5QMFRBQUFBQUFCUFUwQUFlMTI0N
    jk1Yi02OWE5LTRkZTYtOWMzMC1iNTFiMzgwOGM1OWUscD1VQTU3dE0vU3ZwQV
    RCa0gyRlhzMFdEWHZKWXc9
  </response>
</body>
```

Listing 7: SASL authentication step 4

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: 149

<body xmlns='http://jabber.org/protocol/httpbind'>
  <success xmlns='urn:ietf:params:xml:ns:xmpp-sasl'>
```

```
    dj1wTk5ERlZFUXh1WHhDb1NFaVc4R0VaKzFSU289
  </success>
</body>
```

Upon receiving the <success/> element, the client MUST then ask the connection manager to restart the stream by sending a ”restart request” that is structured as follows:

- The BOSH <body/> element MUST include a boolean ’restart’ attribute (qualified by the ’urn:xmpp:xbosh’ namespace) whose value is set to ”true” [11].
- The BOSH <body/> element SHOULD include the ’to’ attribute.
- The BOSH <body/> element SHOULD include the ’xml:lang’ attribute.
- The BOSH <body/> element SHOULD be empty (i.e., not contain an XML stanza). However, if the client includes an XML stanza in the body, the connection manager SHOULD ignore it. [12]

When SASL-EXTERNAL is used in combination with BOSH the BOSH <body/> element SHOULD include the ’from’ attribute upon stream restart. This because constrained clients can not always know what credentials were used to authenticate on the HTTP level. The server MUST try to associate the provided ’from’ with the credentials that were provided on the other level. The following example illustrates the format for a restart request.

Listing 8: Restart request

```
POST /webclient HTTP/1.1
Content-Type: text/xml; charset=utf-8
Content-Length: 240

<body rid='1573741824'
      sid='SomeSID'
      to='example.com'
      xml:lang='en'
      xmpp:restart='true'
      xmlns='http://jabber.org/protocol/httpbind'
      xmlns:xmpp='urn:xmpp:xbosh'/>
```

Upon receiving a restart request, the connection manager MUST consider the previous stream with the XMPP server to be closed. It MUST then initiate a new stream by sending an opening <stream:stream> tag over the same TCP connection to the XMPP server. If the connection

[11] In accordance with Section 3.2.2.1 of XML Schema Part 2: Datatypes, the allowable lexical representations for the xs:boolean datatype are the strings ”0” and ”false” for the concept ’false’ and the strings ”1” and ”true” for the concept ’true’; implementations MUST support both styles of lexical representation.

[12] It is known that some connection manager implementations accept an XML stanza in the body of the restart request and send that stanza to the server when the stream is restarted; however there is no guarantee that a connection manager will send the stanza so a client cannot rely on this behavior.

manager receives a <stream:features/> element from the XMPP server, it MUST forward that element to the client:

Listing 9: Result of restart request

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: 221

<body xmlns='http://jabber.org/protocol/httpbind'
      xmlns:stream='http://etherx.jabber.org/streams'>
  <stream:features>
    <bind xmlns='urn:ietf:params:xml:ns:xmpp-bind'/>
  </stream:features>
</body>
```

The client can then complete any mandatory or discretionary stream feature negotiations.

Listing 10: Resource binding request

```
POST /webclient HTTP/1.1
Content-Type: text/xml; charset=utf-8
Content-Length: 240

<body rid='1573741825'
      sid='SomeSID'
      xmlns='http://jabber.org/protocol/httpbind'>
  <iq id='bind_1'
      type='set'
      xmlns='jabber:client'>
    <bind xmlns='urn:ietf:params:xml:ns:xmpp-bind'>
      <resource>httpclient</resource>
    </bind>
  </iq>
</body>
```

Listing 11: Resource binding result

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: 221

<body xmlns='http://jabber.org/protocol/httpbind'>
  <iq id='bind_1'
      type='result'
      xmlns='jabber:client'>
    <bind xmlns='urn:ietf:params:xml:ns:xmpp-bind'>
      <jid>user@example.com/httpclient</jid>
    </bind>
```

```
  </iq>
</body>
```

6 remote-stream-error

The content of the <body/> element is zero or more stanzas followed by a copy of the <stream:error/> element [13] (qualified by the 'http://etherx.jabber.org/streams' namespace) received from the XMPP server:

Listing 12: Remote error

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: 68

<body condition='remote-stream-error'
      type='terminate'
      xmlns='http://jabber.org/protocol/httpbind'
      xmlns:stream='http://etherx.jabber.org/streams'>
  <message from='contact@example.com'
           to='user@example.com'
           xmlns='jabber:client'>
    <body>I said "Hi!"</body>
  </message>
  <stream:error>
    <xml-not-well-formed xmlns='urn:ietf:params:xml:ns:xmpp-streams'/>
    <text xmlns='urn:ietf:params:xml:ns:xmpp-streams'
          xml:lang='en'>
      Some special application diagnostic information!
    </text>
    <escape-your-data xmlns='application-ns'/>
  </stream:error>
</body>
```

7 recipient-unavailable

It is possible that a connection manager will receive a stanza for delivery to a client even though the client connection is no longer active (e.g., before the connection manager is able to inform the XMPP server that the connection has died). In this case, the connection manager would return an error to the XMPP server; it is RECOMMENDED that the connection manager proceed as follows, since the situation is similar to that addressed by point #2 of Section 11.1

[13] Earlier obsolete versions of this protocol specified that the <body/> element should contain only the *content* of the <stream:error/> element.

of RFC 6121 [14]

1. If the delivered stanza was <presence/>, silently drop the stanza and do not return an error to the sender.
2. If the delivered stanza was <iq/>, return a <service-unavailable/> error to the sender.
3. If the delivered stanza was <message/>, return a <recipient-unavailable/> error to the sender.

When an XMPP server receives a <message/> stanza of type "error" containing a <recipient-unavailable/> condition from a connection manager, it SHOULD store the message for later delivery if offline storage is enabled (see Best Practices for Handling Offline Messages (XEP-0160) [15]), otherwise route the error stanza to the sender.

8 Security Considerations

This protocol does not introduce any new security considerations beyond those specified in BOSH and XMPP Core.

9 IANA Considerations

This document requires no interaction with the Internet Assigned Numbers Authority (IANA) [16].

10 XMPP Registrar Considerations

10.1 Protocol Namespaces

The XMPP Registrar [17] includes 'urn:xmpp:xbosh' in its registry of protocol namespaces.

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

[15] XEP-0160: Best Practices for Handling Offline Messages <https://xmpp.org/extensions/xep-0160.html>.

[16] 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/>.

[17] 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/>.

11 XML Schema

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

<xs:schema
    xmlns:xs='http://www.w3.org/2001/XMLSchema'
    targetNamespace='urn:xmpp:xbosh'
    xmlns='urn:xmpp:xbosh'
    attributeFormDefault='qualified'
    elementFormDefault='qualified'>

  <xs:annotation>
    <xs:documentation>
      The protocol documented by this schema is defined in
      XEP-0206: http://www.xmpp.org/extensions/xep-0206.html
    </xs:documentation>
  </xs:annotation>

  <xs:attribute name='restart'
                type='xs:boolean'
                default='false'/>

  <xs:attribute name='restartlogic'
                type='xs:boolean'
                default='false'/>

  <xs:attribute name='version'
                type='xs:string'
                default='1.0'/>

</xs:schema>
```

12 Acknowledgements

Thanks to Dave Cridland, Steffen Larsen, Matt Miller, Jack Moffitt, Stefan Strigler, Mike Taylor, Winfried Tilanus, Ashley Ward, and Matthew Wild for their feedback.
Thanks to Kevin Winters for his assistance with the schema.