

HyBi Working Group	J. Moffitt, Ed.	
Internet-Draft		
Intended status: Standards Track	E. Cestari	
Expires: June 11, 2011	ProcessOne	
	December 8, 2010	

[TOC](#)

## **An XMPP Sub-protocol for WebSocket draft-moffitt-xmpp-over-websocket-00**

### **Abstract**

This document defines a binding for the XMPP protocol over a WebSocket transport layer. A WebSocket binding for XMPP provides higher performance than the current HTTP binding for XMPP.

### **Status of this Memo**

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on June 11, 2011.

### **Copyright Notice**

Copyright (c) 2010 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

- [1.](#) Introduction
- [2.](#) Terminology
- [3.](#) XMPP Sub-Protocol
  - [3.1.](#) Handshake
  - [3.2.](#) Messages
  - [3.3.](#) XMPP Stream Setup
  - [3.4.](#) Stream Errors
  - [3.5.](#) Closing the Connection
  - [3.6.](#) Stanzas
  - [3.7.](#) Stream Restarts
  - [3.8.](#) Pings and Keepalives
  - [3.9.](#) TLS
- [4.](#) Examples
- [5.](#) IANA Considerations
- [6.](#) Security Considerations
- [7.](#) Informative References
- [§](#) Authors' Addresses

---

## 1. Introduction

[TOC](#)

Applications using XMPP (see [\[RFC3920\]](#) (Saint-Andre, P., Ed., "Extensible Messaging and Presence Protocol (XMPP): Core," October 2004.) and [\[RFC3921\]](#) (Saint-Andre, P., Ed., "Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence," October 2004.)) on the Web currently make use of BOSH (see [\[XEP-0124\]](#) (Paterson, I., Smith, D., Saint-Andre, P., and J. Moffitt, "Bidirectional-streams Over Synchronous HTTP (BOSH)," July 2010.) and [\[XEP-0206\]](#) (Paterson, I. and P. Saint-Andre, "XMPP Over BOSH," July 2010.)), an XMPP binding to HTTP. BOSH is based on the HTTP long polling technique, and it suffers from high transport overhead compared to XMPP's native binding to TCP.

It would be much better in most circumstances to avoid tunneling XMPP over HTTP long polled connections and instead use the XMPP protocol directly. However, the APIs and sandbox that browsers have provided do not allow this. The WebSocket protocol [\[I-D.ietf-hybi-thewebsocketprotocol\]](#) (Fette, I., "The WebSocket protocol," October 2010.) now exists to solve these kinds of problems. The WebSocket protocol is a bi-directional protocol that provides a simple message-based framing layer over raw sockets and allows for more robust and efficient communication in web applications. This document defines a binding of the XMPP protocol over the WebSocket protocol transport. It makes using XMPP within web applications simpler and more efficient.

---

## 2. Terminology

[TOC](#)

The basic unit of framing in the WebSocket protocol is called a message. In XMPP, the basic unit is the stanza, which is a subset of the first-level children of each document in an XMPP stream (see Section 9 of [\[RFC3920\] \(Saint-Andre, P., Ed., "Extensible Messaging and Presence Protocol \(XMPP\): Core," October 2004.\)](#)). XMPP also has a concept of messages, which are stanzas whose top-level element name is message. In this document, the word "message" will mean a WebSocket message, not an XMPP message stanza (see [Section 3.2 \(Messages\)](#)).

---

## 3. XMPP Sub-Protocol

[TOC](#)

### 3.1. Handshake

[TOC](#)

The xmpp sub-protocol is used to transport XMPP over a WebSocket connection. The client and server agree to this protocol during the WebSocket handshake (see Section 1.3 of [\[I-D.ietf-hybi-thewebsocketprotocol\] \(Fette, I., "The WebSocket protocol," October 2010.\)](#)).

During the WebSocket handshake, the client MUST include the |Sec-WebSocket-Protocol| header in its handshake, and the value |xmpp| must be included in the list of protocols. The reply from the server MUST also contain |xmpp| in its own |Sec-WebSocket-Protocol| header in order for an XMPP sub-protocol connection to be established.

Once the handshake is complete, WebSocket messages sent or received will conform to the protocol defined in the rest of this document.

---

### 3.2. Messages

[TOC](#)

Data frame messages in the XMPP sub-protocol MUST be of the text type and contain UTF-8 encoded data. The close control frame's contents are specified in [Section 3.5 \(Closing the Connection\)](#). Control frames other than close are not restricted.

Unless noted in text, the word "message" will mean a WebSocket message containing a text data frame.

---

### 3.3. XMPP Stream Setup

[TOC](#)

The first message sent after the handshake is complete MUST be an XMPP opening stream tag as defined in XMPP [\[RFC3920\] \(Saint-Andre, P., Ed., "Extensible Messaging and Presence Protocol \(XMPP\): Core," October 2004.\)](#) or an XML text declaration (see Section 4.3.1 of [\[W3C.REC-xml-20081126\] \(Sperberg-McQueen, C., Yergeau, F., Maler, E., Paoli, J., and T. Bray, "Extensible Markup Language \(XML\) 1.0 \(Fifth Edition\)," November 2008.\)](#)) followed by an XMPP opening stream tag. The stream tag MUST NOT be closed (i.e. the closing `</stream:stream>` tag should not appear in the message) as it is the start of the client's outgoing XML. The `'<'` character of the tag or text declaration MUST be the first character of the text payload.

The server MUST respond with a message containing an error (see [Section 3.4 \(Stream Errors\)](#)), its own opening stream tag, or an XML text declaration followed by an opening stream tag.

---

### 3.4. Stream Errors

[TOC](#)

Stream level errors in XMPP are terminal. Should such an error occur, the server MUST send the stream error as a complete element in a message to the client. This message MAY include the closing stream tag if a stream is opened.

If the error occurs during the opening of a stream, the stream error message MUST start with an opening stream tag (see Section 4.7.1 of [\[RFC3920\] \(Saint-Andre, P., Ed., "Extensible Messaging and Presence Protocol \(XMPP\): Core," October 2004.\)](#)) and end with a closing stream tag.

After the stream error and closing stream tag have been sent, the server MUST close the connection as in [Section 3.5 \(Closing the Connection\)](#).

---

### 3.5. Closing the Connection

[TOC](#)

Either the server or the client may close the connection at any time. Before closing the connection, the closing party MUST close the XMPP stream if it has been established. To initiate the close, the closing party MUST send a normal WebSocket close message with an empty body. The connection is considered closed when a matching close message is received (see Section 1.4 of [\[I-D.ietf-hybi-thewebsocketprotocol\] \(Fette, I., "The WebSocket protocol," October 2010.\)](#)).

---

### 3.6. Stanzas

[TOC](#)

Each XMPP stanza MUST be sent in its own message. A stanza MUST NOT be split over multiple messages. All first level children of the <stream:stream> element MUST be treated the same as stanzas (e.g. <stream:features> and <stream:error>).

---

### 3.7. Stream Restarts

[TOC](#)

After successful SASL authentication, an XMPP stream must be restarted. In these cases, as soon as the message is sent (or received) containing the success indication, both the server and client streams are implicitly closed, and new streams must be opened. The client MUST open a new stream as in [Section 3.3 \(XMPP Stream Setup\)](#) and MUST NOT send a closing stream tag.

---

### 3.8. Pings and Keepalives

[TOC](#)

XMPP servers send whitespace pings as keepalives between stanzas, and XMPP clients can do the same thing. These extra whitespace characters are not significant in the protocol. Servers and clients SHOULD use WebSocket ping messages instead for this purpose.

The XMPP Ping extension [\[XEP-0199\] \(Saint-Andre, P., "XMPP Ping," June 2009.\)](#) allows entities to send and respond to ping requests. A client sending a WebSocket ping is equivalent to pinging the WebSocket server, which may also be the XMPP server. When the XMPP server is not also the WebSocket server, a WebSocket ping may be useful to check the health of the intermediary server.

---

### 3.9. TLS

[TOC](#)

TLS cannot be used in The XMPP sub-protocol because the sub-protocol does not allow for raw binary data to be sent. A server MUST NOT advertise TLS as a stream feature (see Section 4.6 of [\[RFC3920\] \(Saint-Andre, P., Ed., "Extensible Messaging and Presence Protocol \(XMPP\): Core," October 2004.\)](#)). A client MUST ignore the TLS feature if it is advertised over WebSocket.

---

[TOC](#)

## 4. Examples

Examples will be added as soon as the WebSocket protocol specification is more stable.

---

## 5. IANA Considerations

[TOC](#)

If a registry is created for WebSocket sub-protocols, the xmpp sub-protocol will be registered.

---

## 6. Security Considerations

[TOC](#)

Since application level TLS cannot be used (see [Section 3.9 \(TLS\)](#)), applications which need to protect the privacy of the XMPP traffic need to do so at the WebSocket level or some other higher level.

---

## 7. Informative References

[TOC](#)

[I-D.ietf-hybi-thewebsocketprotocol]	Fette, I., " <a href="#">The WebSocket protocol</a> ," draft-ietf-hybi-thewebsocketprotocol-03 (work in progress), October 2010 ( <a href="#">TXT</a> ).
[RFC3920]	<a href="#">Saint-Andre, P., Ed.</a> , " <a href="#">Extensible Messaging and Presence Protocol (XMPP): Core</a> ," RFC 3920, October 2004 ( <a href="#">TXT</a> , <a href="#">HTML</a> , <a href="#">XML</a> ).
[RFC3921]	<a href="#">Saint-Andre, P., Ed.</a> , " <a href="#">Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence</a> ," RFC 3921, October 2004 ( <a href="#">TXT</a> , <a href="#">HTML</a> , <a href="#">XML</a> ).
[W3C.REC-xml-20081126]	Sperberg-McQueen, C., Yergeau, F., Maler, E., Paoli, J., and T. Bray, " <a href="#">Extensible Markup Language (XML) 1.0 (Fifth Edition)</a> ," World Wide Web Consortium Recommendation REC-xml-20081126, November 2008 ( <a href="#">HTML</a> ).
[XEP-0124]	<a href="#">Paterson, I.</a> , <a href="#">Smith, D.</a> , <a href="#">Saint-Andre, P.</a> , and <a href="#">J. Moffitt</a> , " <a href="#">Bidirectional-streams Over Synchronous HTTP (BOSH)</a> ," XSF XEP 0124, July 2010.
[XEP-0199]	<a href="#">Saint-Andre, P.</a> , " <a href="#">XMPP Ping</a> ," XSF XEP 0199, June 2009.
[XEP-0206]	<a href="#">Paterson, I.</a> and <a href="#">P. Saint-Andre</a> , " <a href="#">XMPP Over BOSH</a> ," XSF XEP 0206, July 2010.

---

## Authors' Addresses

[TOC](#)

	Jack Moffitt (editor)
Email:	<a href="mailto:jack@metajack.im">jack@metajack.im</a>
	Eric Cestari
	ProcessOne
Email:	<a href="mailto:ecestari@process-one.com">ecestari@process-one.com</a>