XEP-0190: Best Practice for Closing Idle Streams

Carlo von Loesch
mailto:lynX@jabber.getting.psyced.org
xmpp:lynX@ve.symlynX.com

2012-03-06
Version 1.1

Status	Type	Short Name
Obsolete	Informational	N/A

This document specifies a best practice for closing an XML stream that is perceived to be idle.

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

RFC 3920 [1] describes several ways to terminate an XML stream, but does not always make a clear statement about which to use. This can lead to faulty implementations. In particular, closing a stream that has not been in use for a while is very often achieved using a connection-timeout error, then closing the socket. This can lead to loss of data. Therefore this document proposes a practice that will avoid such data loss.
Note: The recommendation described herein has been incorporated into RFC 6120 [2].

2 How to Close an Idle Stream

2.1 Handshake Stream Shutdown

As shown in the basic "session" example in the Simplified Stream Examples (4.8 of RFC 3920), it is a valid transaction to close the outgoing stream by sending

```
</stream:stream>
```

then wait for the other entity to close its stream, like this:

```
</stream:stream>
```

and shut down the underlying TCP connection.
This will ensure that, should the other entity have transmitted any data, it will arrive and be processed before the TCP connection is terminated.
Special care MUST be taken that under no circumstance further packets may be written to the socket after the stream was closed, until the other side shuts down the socket.
On the outgoing TCP connection, an implementation MAY do a read-only shutdown of the socket, as long as the other side will safely be able to send its stream termination token.

2.2 Handshake Failure

In case the other entity fails to close the stream within a reasonable time frame, the entity that started the handshake is entitled to terminate the TCP connection. Since the stream has already been closed, it is correct not to produce an error condition.

---

[1]RFC 3920: Extensible Messaging and Presence Protocol (XMPP): Core <http://tools.ietf.org/html/rfc3920>.
[2]RFC 6120: Extensible Messaging and Presence Protocol (XMPP): Core <http://tools.ietf.org/html/rfc6120>.

3 Implementation Notes

Existing implementations should be updated to use the 'Handshake Stream Shutdown' strategy when shutting down streams that are no longer needed. This strategy is fully backwards-compatible and does not introduce any known communication problems.

4 Security Considerations

This proposal introduces no new security aspects.

5 IANA Considerations

This proposal requires no interaction with the Internet Assigned Numbers Authority (IANA) [3].

6 XMPP Registrar Considerations

This proposal requires no interaction with the XMPP Registrar [4].

---

[3]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/>.

[4]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/>.