XEP-0160: Best Practices for Handling Offline Messages

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

2006-01-24
Version 1.0

Status	Type	Short Name
Active	Informational	msgoffline

This document specifies best practices to be followed by Jabber/XMPP servers in handling messages sent to recipients who are offline.

Legal

Copyright

Permissions

Warranty

Liability

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 <http://xmpp.org/about-xmpp/xsf/xsf-ipr-policy/> or obtained by writing to XMPP Standards Foundation, 1899 Wynkoop Street, Suite 600, Denver, CO 80202 USA).

Contents

1 Introduction 1

2 Process Flow 1

3 Handling of Message Types 3

4 Service Discovery 3

5 Security Considerations 4

6 IANA Considerations 4

7 XMPP Registrar Considerations 4

1 Introduction

XMPP Core [1] and XMPP IM [2] specify general rules for handling XML stanzas, but explicitly do not address how to handle message stanzas sent to recipients (e.g., IM users or other nodes) that are offline, except to say that a server MUST return a <service-unavailable/> error if offline message storage or message forwarding is not enabled (see RFC 6121). This document fills the gap by specifying best practices for storage and delivery of so-called "offline messages".

2 Process Flow

The RECOMMENDED process flow is as follows:

1. Sender generates XMPP message stanza [3] for delivery to a recipient such as an IM user or other node, where the 'to' address is of the form <node@domain> or <node@domain/resource> (see RFC 6121 for rules regarding server handling of such XMPP message stanzas).
2. Recipient's server determines that the intended recipient has no available resources that have specified non-negative presence priority. [4]
3. Recipient's server determines that if the server can store offline messages on behalf of the intended recipient; if not (e.g., because the recipient's offline message queue is full), the server returns a <service-unavailable/> error to the sender.
4. Recipient's server does not return a <service-unavailable/> error but instead stores the message stanza for later delivery.
5. When the recipient next sends non-negative available presence to the server, the server delivers the message to the resource that has sent that presence. (Alternatively, the server may support Flexible Offline Message Retrieval [5], although that functionality is not described herein.)

This flow is described more fully below.
First, the sender (in this example, romeo@montague.net) sends a message to an intended recipient (juliet@capulet.com).

[1] RFC 6120: Extensible Messaging and Presence Protocol (XMPP): Core <http://tools.ietf.org/html/rfc6120>.
[2] RFC 6121: Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence <http://tools.ietf.org/html/rfc6121>.
[3] This document does not discuss IQ or presence stanzas, handling of which is described in RFC 6120 and RFC 6121.
[4] As specified in RFC 6120, available resources that have specified a negative presence priority shall never receive message stanzas addressed to <node@domain>.
[5] XEP-0013: Flexible Offline Message Retrieval <http://xmpp.org/extensions/xep-0013.html>.

Listing 1: Sender Generates Message to Recipient

```
<message from='romeo@montague.net/orchard' to='juliet@capulet.com'>
  <body>
    O blessed, blessed night! I am afeard.
    Being in night, all this is but a dream,
    Too flattering-sweet to be substantial.
  </body>
</message>
```

Next, the recipient's server determines if there are any available resources that have sent non-negative presence priority. If there are, the server immediately delivers the message stanza to the resource that it determines to be most available (based on its own algorithm).
Next, the recipient's server determines if offline messages can be stored on behalf of the intended recipient. If not (e.g., because the recipient's offline message queue is full), the server returns a <service-unavailable/> error to the sender. If so, the server stores the message for later delivery.
Now the recipient authenticates with the server and sends initial presence (with a non-negative priority) to the server.

Listing 2: Recipient Becomes Available

```
<presence from='juliet@capulet.com/balcony'>
  <priority>1</priority>
</presence>
```

The recipient's server now delivers the offline message to that resource (it is RECOMMENDED for the server to add a Delayed Delivery [6] extension to indicate that the message was stored offline).

Listing 3: Recipient's Server Delivers Message

```
<message from='romeo@montague.net/orchard' to='juliet@capulet.com'>
  <body>
    O blessed, blessed night! I am afeard.
    Being in night, all this is but a dream,
    Too flattering-sweet to be substantial.
  </body>
  <delay xmlns='urn:xmpp:delay'
     from='capulet.com'
     stamp='2002-09-10T23:08:25Z'>Offline Storage</delay>
</message>
```

[6] XEP-0203: Delayed Delivery <http://xmpp.org/extensions/xep-0203.html>.

3 Handling of Message Types

Message stanzas SHOULD be handled by a server as follows (based on the values of the 'type' attribute specified in RFC 6121):

- normal -- Messages with a 'type' attribute whose value is "normal" (or messages with no 'type' attribute) SHOULD be stored offline.
- chat -- Messages with a 'type' attribute whose value is "chat" SHOULD be stored offline, with the exception of messages that contain only Chat State Notifications [7] content (such messages SHOULD NOT be stored offline).
- groupchat -- Messages with a 'type' attribute whose value is "groupchat" SHOULD NOT be stored offline, since by definition a user without online presence cannot be in a Multi-User Chat [8] room.
- headline -- Messages with a 'type' attribute whose value is "headline" SHOULD NOT be stored offline, since such messages are usually time-sensitive.
- error -- Messages with a 'type' attribute whose value is "error" SHOULD NOT be stored offline, although a server MAY store Advanced Message Processing [9] error messages offline.

4 Service Discovery

If a server supports offline message handling as described herein, it SHOULD return a "msgoffline" feature in response to Service Discovery [10] information requests:

Listing 4: Recipient Queries Server About Support

```
<iq from='juliet@capulet.com/chamber' to='capulet.com'>
  <query xmlns='http://jabber.org/disco#info'/>
</iq>
```

[7] XEP-0085: Chat State Notifications <http://xmpp.org/extensions/xep-0085.html>.
[8] XEP-0045: Multi-User Chat <http://xmpp.org/extensions/xep-0045.html>.
[9] XEP-0079: Advanced Message Processing <http://xmpp.org/extensions/xep-0079.html>.
[10] XEP-0030: Service Discovery <http://xmpp.org/extensions/xep-0030.html>.

Listing 5: Server Returns Information About Support

```
<iq from='capulet.com' to='juliet@capulet.com/chamber'>
  <query xmlns='http://jabber.org/disco#info'>
    ...
    <feature var='msgoffline'/>
  </query>
</iq>
```

5 Security Considerations

A message stored offline may not be readable by the recipient if the message was encrypted using a session-based encryption method such as Encrypted Session Negotiation [11] or if the key used in object encryption is revoked after the message was sent but before it is read.
In certain countries, offline storage of message stanzas may introduce legal requirements or privacy vulnerabilities that do not apply to messages that are delivered immediately and never stored on an intermediate server.
See XEP-0203 for security considerations regarding the inclusion and processing of delayed delivery notations.

6 IANA Considerations

This document requires no interaction with the Internet Assigned Numbers Authority (IANA) [12].

7 XMPP Registrar Considerations

The XMPP Registrar [13] includes ”msgoffline” in its registry of service discovery features (see <http://xmpp.org/registrar/disco-features.html>).

[11] XEP-0116: Encrypted Session Negotiation <http://xmpp.org/extensions/xep-0116.html>.

[12] 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/>.

[13] 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/>.