XEP-DELIVERY: Reliable Message Delivery

1. Introduction

XMPP has problems with guaranteed message delivery and absence of mandatory unique message identifiers. This protocol is designed to solve both these problems and is necessary for many modern extensions.

1.1. Goals

1. Guarantee the delivery of one and only one message copy from a client to server.
2. Provide a mechanism to resend messages without creating duplicate copies of the same message.
3. Provide the ability to identify and address any message on the server.
4. Provide information about the unique identifier of message to all conversation participants.
5. Provide information about the time when the XMPP server received the message to all conversation participants.

1.2. Glossary

- **Server** — an XMPP server which confirms delivery of message.
- **Sender** — an XMPP client which sends messages and supports this protocol.
- **Recipient** — an entity to whom the message was addressed.
- **Message Archive** — an entity on the server which stores messages.
- **Stanza ID** — a unique identifier created by the **Server** in accordance with Unique and Stable Stanza IDs (XEP-0359)**.**
- **Timestamp** — an identifier created by the **Server**, containing timestamp when message was received by the server.
- **Origin ID** — a unique identifier created by the **Sender** in accordance with Unique and Stable Stanza IDs (XEP-0359).

1.3. Requirements

1. Each message on the **Server** MUST be assigned a unique identifier within the server (hereinafter **Stanza ID**) and a timestamp (hereinafter **Timestamp**) when the message was received.
2. In the **Message Archive**, the message MUST be stored with **Stanza ID** and **Timestamp**.
3. The message MUST be sent with **Stanza ID** and **Timestamp** to the **Recipient**.
4. For messages that require guaranteed delivery of one and only one copy, the **Sender** supporting this protocol MUST add an **Origin ID**.
5. **Server** MUST acknowledge message reception by providing the **Sender** with an information about the **Stanza ID** and the **Message Timestamp** assigned to the message, associating it with the **Origin ID**.
6. After disconnecting and reconnecting, **Sender** MUST retry sending messages for which receipt containing **Stanza ID** have not been received yet.
7. Only the **Sender** that had originally sent the message MAY retry sending it.
8. When retrying to send the message, the **Sender** MUST explicitly indicate that the message is being resent.
9. Upon repeated message delivery, **Server** MUST provide the **Sender** the information about the previous successful message delivery (if there was such), including the **Stanza ID** and the **Message Timestamp** originally assigned to the message, and suppress further processing of the message.

This protocol MUST NOT be used for receiving read and delivery markers/receipts from remote clients.

2. Dependencies

2.1 Unique and Stable Stanza IDs

This protocol depends on XEP-0359: Unique and Stable Stanza IDs and extends its capabilities.

2.2 Message archive

A server implementing this protocol MUST store all incoming messages in **Message Archive** in accordance with XEP-0313: Message Archive Management.

2.3. Message Carbons

Messages forwarded in accordance with XEP-0280: Message Carbons MUST contain a **<stanza-id>** assigned by the server. This tightens the requirement of clause 3.5 of XEP-0313: Message Archive Management.

2.4. Session resumption

The client MUST try to use the same resource resuming a broken connection.

The client MUST comply with the requirements of Section 4.9.3.3 of RFC-6120.

The server MUST implement behavior No. 3 of Section 7.7.2.2 of RFC-6120, i.e. terminate an existing connection in case establishing a new one with the same resource.

The server MUST disconnect the previously established connection restoring the session through XEP-0198 Stream Management, or MUST not support the restoration of sessions through XEP-0198.

3. Business rules

3.1. Message IDs

Sender generates an **Origin ID** to track message delivery to **Server**. It is a client-side identifier that is kept by a client to track delivery receipts and error messages. It MUST NOT be used for any other purposes.

Sender MUST guarantee that the **Origin ID** is unique for all messages of all jid clients. The absence of such guarantee may result in incorrect work of the protocol. It is RECOMMENDED to use UUID for **Origin ID**.

It is RECOMMENDED for the **Sender** to use the **Origin ID** value for ‘id‘ attribute of <message>.

For each stored message, **Server** generates a **Stanza ID**, which MUST be unique within the server. Each message within the server is identified by unique **Stanza ID**. **Sender** SHOULD use **<stanza-id>** for message tracking.

3.2. Delivery receipts

Delivery receipt MUST have type ‘headline’ and MUST NOT be stored in message archive.

Delivery receipt MUST be sent to the **Sender**.

Server SHOULD NOT confirm delivery of messages that are not intended to be stored in **Message Archive**. This includes:

- ‘error’ type messages
- ‘headline’ type messages
- Messages with an empty body.

When searching for possible message duplicates, **Server** MUST search only within the messages of the Sender’s jid.

Example 1.

Sender send message with **Origin ID**:

```
<message xmlns='jabber:client' type='chat' id='fa20384a-75ea-4d4e-bb39-49e0fd55473b'
from='juliet@capuliet.it/phone' to='romeo@montague.it'>
   <body>Hi!</body>
   <origin-id xmlns='urn:xmpp:sid:0' id='fa20384a-75ea-4d4e-bb39-49e0fd55473b' />
</message>
```

Server replies by sending delivery receipt to the **Sender**:

```
<message from='juliet@capuliet.it' to='juliet@capuliet.it/phone' type='headline'>
   <received xmlns='https://xabber.com/protocol/delivery'>
      <time by='juliet@capuliet.it' stamp='2020-03-16T11:46:33.970745Z'/>
      <origin-id id='fa20384a-75ea-4d4e-bb39-49e0fd55473b' xmlns='urn:xmpp:sid:0'/>
      <stanza-id by='juliet@capuliet.it' id='1584359193970745' xmlns='urn:xmpp:sid:0'/>
   </received>
</message>
```

3.3. Message archiving

Server MUST ensure that the value of attribute ‘stamp‘ of **<time>** tag added to the message under the current protocol is equal to the value of attribute ‘stamp‘ of **<delay>** tag added by **Message Archive**. This is necessary for the compatibility with the existing protocols.

Server MUST send generated <stanza-id> back to **Sender**. This is necessary to track message delivery. This requirement conflicts with clause 3.5 of XEP-0313: Message Archive Management, that servers MUST NOT include the **<stanza-id>** element in messages addressed to JIDs that do not have permissions to access the archive. However, in XEP-0359: Unique and Stable Stanza IDs there are no restrictions on relaying **<stanza-id>** to clients. Clause 5 of XEP-0359 states that **<stanza-id>** value SHOULD NOT be secret. Consequently, this protocol does not observe this requirement of clause 3.5 of XEP-0313.

Use of receipts MUST NOT clutter message history. Therefore, delivery receipts MUST NOT be stored in the message history.

3.4. Resending

By message re-sending **Sender** MUST include <retry/> tag to the message and send it with original **Origin ID** to the **Server**.

Server MUST check if the message was stored in the **Message Archive.** If the message is found **Server** MUST stop processing the message and send a receipt to the **Sender.**

Example 2. Sender try to re-send message:

Sender try to re-send a message:

```
<message xmlns='jabber:client' type='chat' id='fa20384a-75ea-4d4e-bb39-49e0fd55473b'
from='juliet@capuliet.it/balcony' to='romeo@montague.it'>
<body>Hi!</body>
<origin-id xmlns='urn:xmpp:sid:0' id='fa20384a-75ea-4d4e-bb39-49e0fd55473b' />
<retry xmlns='https://xabber.com/protocol/delivery'/>
</message>
```

Server searches for this message in the **Message Archive**. If **Server** found message he MUST send delivery receipt:

```
<message from='juliet@capuliet.it' to='juliet@capuliet.it/balcony' type='headline'>
<received xmlns='https://xabber.com/protocol/delivery'>
<time by='juliet@capuliet.it' stamp='2020-03-16T11:46:33.970745Z'/>
<origin-id id='fa20384a-75ea-4d4e-bb39-49e0fd55473b' xmlns='urn:xmpp:sid:0'/>
<stanza-id by='juliet@capuliet.it' id='1584359193970745' xmlns='urn:xmpp:sid:0'/>
</received>
</message>
```

And MUST NOT send the message to the **Recipient**.

In case if the message is not found **Server** MUST work in the usual way.

3.5. Error handling

If **Sender** receives an error in response to sent message, **Sender** MUST consider that message delivery is not possible and MUST NOT attempt to resend message using resending mechanism described in current protocol. Sender MAY send a message again without the <retry> tag in the usual way.

Example 3. Sender got a error by message delivery

Sender sends message:

```
<message xmlns='jabber:client' type='chat' id='fa20384a-75ea-4d4e-bb39-49e0fd55473b'
from='juliet@capuliet.it/phone' to='romeo@montague.it'>
<body>Hi!</body>
<origin-id xmlns='urn:xmpp:sid:0' id='fa20384a-75ea-4d4e-bb39-49e0fd55473b' />
</message>
```

Server sends a error message:

```
<message xmlns='jabber:client' type='chat' id='fa20384a-75ea-4d4e-bb39-49e0fd55473b'
from='juliet@capuliet.it/phone' to='romeo@montague.it'>
<body>Hi!</body>
<origin-id xmlns='urn:xmpp:sid:0' id='fa20384a-75ea-4d4e-bb39-49e0fd55473b' />
<error code='404' type='cancel'>
<remote-server-not-found xmlns='urn:ietf:params:xml:ns:xmpp-stanzas'/>
<text xml:lang='en' xmlns='urn:ietf:params:xml:ns:xmpp-stanzas'>DNS lookup failed: non-existing
domain</text>
</error>
```

</message>