XEP-0436: MUC presence versioning

JC Brand
mailto: jc@opkode.com
xmpp: jc@opkode.com

Matthew Wild
mailto: mwild1@gmail.com
xmpp: me@matthewwild.co.uk

2020-05-05
Version 0.1.0

This specification defines a versioning mechanism which reduces the amount of presence traffic in a XEP-0045 MUC.
Legal

Copyright

This XMPP Extension Protocol is copyright © 1999 – 2020 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).
1 Introduction

Many modern-day non-XMPP groupchat implementations have discarded the metaphor of physical presence inside a room that a user must enter and exit, as implemented by Multi-User Chat (XEP-0045)\(^1\). The newer Mediated Information eXchange (MIX) (XEP-0369)\(^2\) has therefore made presence subscriptions optional.

Often it no longer makes sense for a chat service to require that a user is "present" in order for them to be addressed by other occupants or to receive messages, especially if the chat implementation will inform you out-of-band, for example via push notifications or email. The notion of "room presence" is therefore less relevant than before, and in some cases can be done away with entirely.

Broadcasting all XEP-0045 MUC participants’ presences to one another scales quadratically (O(n^2)) and can greatly increase the amount of network traffic, for potentially negligible gain.

Even though the metaphorical concept of presence inside a room might no longer be relevant for a groupchat implementation, \(<\text{presence}/>\) stanzas might still contain useful metadata, such as the user’s affiliation or their Hats (XEP-0317)\(^3\).

This XEP defines a versioning mechanism (similar to roster versioning in RFC 6121\(^4\)) whereby the amount of presence traffic in a MUC may be greatly reduced. It also describes additional measures which may be taken to further reduce the amount of presence traffic.

2 How it works

A client that supports presence versioning needs to keep track and store the presence statuses of all MUC occupants, across multiple MUC sessions. Similarly, a MUC service which supports presence versioning will also need to maintain a changelog of version numbers and presence states.

Before the client enters a MUC, it SHOULD use service discovery to check whether presence versioning is supported (see determining support below.). If presence versioning is supported, the client MAY include a ‘ver’ attribute set to the last known presence version in the \(<\text{path}http://jabber.org/protocol/muc}\x> tag of the \(<\text{presence}/>\) stanza, which it sends to join the MUC.

If presence versioning is not supported by the server, the client MUST NOT include a ‘ver’ attribute.

Listing 1: User specifies the last known presence version when seeking to enter a MUC

---

The MUC will return only those presences that have changed since the version indicated by the client, and in the self-presence of the joining user it will add a 'ver' attribute with the latest version number on the <path xmlns='http://jabber.org/protocol/muc'/> tag. The client must save the version number and use it next time it joins the MUC.

When presence versioning is enabled, every subsequent <presence/> stanza sent by the server MUST include a new version number, which replaces the existing one saved by the client.

### 3 Determining support

If a MUC implements presence versioning, it MUST specify the 'urn:xmpp:muc-presence-versioning:0' feature in its service discovery information features, as specified in Service Discovery (XEP-0030)\(^5\).

---

Listing 4: The MUC advertises support for presence versioning

```
<iq type='result'
    to='romeo@montague.example/home'
    from='room@muc.shakespeare.example'
    id='info1'>
  <query xmlns='http://jabber.org/protocol/disco#info'>
    ...
    <feature var='urn:xmpp:presence-versioning:0'/>
    ...
  </query>
</iq>
```

4 Business Rules

If a MUC receives a presence version number that’s so old, so that it no longer has the corresponding state available, it needs to send all presence statuses back to the client. If the client has not yet saved a presence version number and the corresponding presence states, then it MUST bootstrap presence versioning by sending a 'ver' attribute set to the empty string (i.e., ver="").

Even if the client did not include a 'ver' attribute in its “join” <presence/> stanza, the server SHOULD still set a 'ver' attribute on the relevant <presence/> stanzas.

5 Additional measures

There are a number of Multi-User Chat (XEP-0045) features that a client and server may decide to configure and/or implement in order to further reduce the number of presence stanzas being sent around.

5.1 Only broadcast presence for affiliated users

A MUC MAY be configured to only broadcast presence from occupants above a certain affiliation, (see the presence broadcast section of XEP-0045), for example in a MUC where non-affiliated users are allowed to view the discussion but aren’t allowed to send messages.

5.1.1 Let users register themselves with a MUC

This step can be taken in addition to letting users register themselves as members in the MUC. XEP-45 describes in section 7.10 "Registering with a Room how a user may register themselves with a room, thereby receiving the "member" affiliation and having their preferred nickname

---

5.2 Restrict presence states to available and unavailable

In some cases, it makes sense to reduce the number of presence statuses to only a subset, in order to reduce to total amount of states the server needs to keep track off. In the simplest case, this would mean keeping track only of two statuses, ‘available’ and ‘unavailable’.

5.2.1 Send presence stanzas for unavailable members

XEP-0045 documents the status code 102, which is used to indicate that a room shows unavailable members. By also sending to newly joined users the presence stanzas of unavailable members, it’s possible to provide any necessary presence metadata of all relevant users in a groupchat and not just the currently ”present” users.

6 IANA Considerations

None.

7 XMPP Registrar Considerations

7.1 Protocol Namespaces

The XMPP Registrar includes ‘urn:xmpp:muc-presence-versioning:0’ in its registry of protocol namespaces (see <https://xmpp.org/registrar/namespaces.html>).

- urn:xmpp:muc-presence-versioning:0

7.2 Protocol Versioning

If the protocol defined in this specification undergoes a revision that is not fully backwards-compatible with an older version, the XMPP Registrar shall increment the protocol version number found at the end of the XML namespaces defined herein, as described in Section 4 of XEP-0053.

---

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/>.