XEP-0310: Presence State Annotations

Kevin Smith
mailto:kevin.smith@isode.com
xmpp:kevin.smith@isode.com

2012-01-10
Version 0.1

Status	Type	Short Name
Deferred	Standards Track	PSA

This document provides a protocol using which a server is able to provide information to clients indicating that previously received presence data may be stale.

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 1

2 Requirements 1

3 Discovery 1

4 Use Cases 1
- 4.1 Remote Server Unavailable . . . 2
- 4.2 Local User Paused . . . 3
- 4.3 MUC loses contact with occupant . . . 4

5 Business Rules 5

6 Security Considerations 6

7 IANA Considerations 6

8 XMPP Registrar Considerations 6

9 XML Schema 6

1 Introduction

Presence data received by a client XMPP IM [1] are typically cached and displayed until they are replaced by an update (whereupon the new data will be cached and displayed...), for example graphically annotating availability of contacts in a roster. Where the entity from which the client has received the presence data is remote (residing upon a different server), unavailability of the server to server link will render the client unable to receive further presence updates from the entity but unaware that this is the case. Where the remote entity is a contact, it may cause the contact appear to be online (or offline, or away etc.) to the user of the client when they are not. Where the remote entity is a Multi-User Chat (XEP-0045) [2] room the case is more severe, as the MUC room may have ejected the client due to the server-to-server (S2S) link being unavailable to send stanzas but the client may still consider itself present in the room.
This extension provides a simple mechanism by which the local server can resend previous presence data to the client, annotated such that the client knows it would be unable to receive future updates.

2 Requirements

Allow servers to annotate presence stanzas sent to clients indicating lack of availability of a client or server link necessary for receipt of updated presence.

3 Discovery

Clients supporting this should have a feature of ”urn:xmpp:psa” in caps. Servers autoenable based on this.

4 Use Cases

There are three main cases that links may be unavailable, leading to Romeo (a user of the montague.lit server) receiving presence state annotations:

- That a connection from montague.lit to a remote server, such as capulet.lit, is unavailable, leading to remote presence from users on capulet.lit (e.g. juliet@capulet.lit) being potentially stale.

[1] RFC 6121: Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence <http://tools.ietf.org/html/rfc6121>.
[2] XEP-0045: Multi-User Chat <https://xmpp.org/extensions/xep-0045.html>.

- That a connection from a client to montague.lit has been paused (e.g. because the client uses Stream Management and has a ready-to-resume session, or has a paused BOSH session), such that Romeo's representation of a paused session such as benvolio@montague.lit's is not accurate.
- That the connection between a MUC room in which Romeo is an occupant and another of that room's occupants is affected by either of the cases above, making Romeo's view of the room occupancy stale.

Here we address each in turn. In each of these examples Romeo's client is assumed to have already indicated support for PSAs, as described in 'discovery' above.

4.1 Remote Server Unavailable

In this example, Romeo (romeo@montague.lit) has received presence from Juliet (juliet@capulet.lit/orchard), either because she's in his roster or because she sent directed presence.

Listing 1: Romeo receives Juliet's presence

```
<presence
    from='juliet@capulet.lit/orchard'
    to='romeo@montague.lit/lane'>
  <status>Available</status>
</presence>
```

The server to server connection between capulet.lit and montague.lit then becomes unavailable for some reason (montague.lit can't know if this is a network error, a server failure, or some other condition). Juliet may now be updating her presence (e.g. she might now have gone offline), and Romeo would not be able to receive the presence update stanzas. As such, montague.lit MUST push the old presence to Romeo, annotated that this may now be stale due to an s2s outage.
The state-annotation element alerts Romeo's client to the unavailability of the server link through the server-unavailable child. The state-annotation MUST have a 'from' attribute set to the entity annotating the stanza (in this case montague.lit), and an OPTIONAL description child MAY provide human-readable text describing the change.

Listing 2: Romeo receives Juliet's presence, annotated that the S2S is unavailable

```
<presence
    from='juliet@capulet.lit/orchard'
    to='romeo@montague.lit/lane'>
  <status>Available</status>
  <state-annotation xmlns="urn:xmpp:psa" from="montague.lit">
    <server-unavailable/>
```

```
    <description>The capulet.lit server is no longer available</
       description>
  </state-annotation>
</presence>
```

When the link between montague.lit and capulet.lit again becomes available, montague.lit MUST send an empty annotation of the previous presence to Romeo. It MAY subsequently attempt to refresh this presence by reprobing the remote entity.

Listing 3: Romeo receives Juliet's annotated presence, annotated that S2S is restored

```
<presence
    from='juliet@capulet.lit/orchard'
    to='romeo@montague.lit/lane'>
  <status>Available</status>
  <state-annotation xmlns="urn:xmpp:psa" from="montague.lit"/>
</presence>
```

When an S2S connection is deemed not to be available, or to return to availability afterwards (TODO: guidelines on what this means), the server MUST send a corresponding annotated presence for every remote full JID on the unavailable server to every local client requesting PSAs where the local client has received presence from the remote JID and the most recent presence is an available type (i.e. not error or unavailable). Subscription requests are excluded from this processing.

4.2 Local User Paused

In this example, Romeo (romeo@montague.lit) has received presence from Benvolio (benvolio@montague.lit/lane), who is using BOSH.

Listing 4: Romeo receives Benvolio's presence

```
<presence
    from='benvolio@montague.lit/lane'
    to='romeo@montague.lit/lane'>
  <status>He ran this way, and leap'd this orchard wall</status>
</presence>
```

Benvolio's client then uses the BOSH 'pause' feature for suspending the session. The montague.lit server MUST then send an annotated presence to Romeo's client saying that Benvolio's session has been paused. This MUST include the connection-paused element, MUST include a 'from' attribute of the entity doing the annotating (montague.lit) and MAY include a human-readable description element

Listing 5: Romeo receives Benvolio's presence annotated for the paused session

```
<presence
    from='benvolio@montague.lit/lane'
    to='romeo@montague.lit/lane'>
  <state-annotation xmlns="urn:xmpp:psa" from="montague.lit">
    <connection-paused/>
    <description>benvolio's␣BOSH␣session␣has␣been␣paused</description>
␣␣</state-annotation>
␣␣<status>He␣ran␣this␣way,␣and␣leap'd this orchard wall</status>
</presence>
```

After pausing the session, Benvolio's client can either resume it or allow it to time out. If the session is resumed, the server MUST send an empty state-annotation to Romeo's client. If the session times out, it will simply send an unavailable presence as normal.

Listing 6: Romeo receives Benvolio's presence annotated for the resumed session

```
<presence
    from='benvolio@montague.lit/lane'
    to='romeo@montague.lit/lane'>
  <state-annotation xmlns="urn:xmpp:psa" from="montague.lit"/>
  <status>He ran this way, and leap'd␣this␣orchard␣wall</status>
</presence>
```

Listing 7: Romeo receives Benvolio's synthesized unavailable presence for the terminated session

```
<presence
    from='benvolio@montague.lit/lane'
    to='romeo@montague.lit/lane' type='unavailable'/>
```

When a server detects that a client's session has been temporarily suspended (either through a BOSH pause, or through a disconnection while using 198 prior to the resume period timing out) it MUST send annotated presence, as above, to every local and remote JID that has received the client session's presence (and where the presence is not subscription-related and has not been cancelled by a corresponding unavailable presence) and requests PSAs. If a paused session is resumed, the server MUST then send a corresponding empty state-annotation to each of these JIDs, as above. If the session is terminated the server MUST send unavailable presence using the usual rules from RFC6121.

4.3 MUC loses contact with occupant

In this example, Romeo (romeo@montague.lit) is an occupant in the orchard@chats.shakespeare.lit MUC room, alongside Juliet (juliet@capulet.lit). (For these examples, the usual MUC payloads are elided; they should be present as required by xep45).

Listing 8: Romeo receives a presence update from Juliet in the MUC room

```
<presence
    from='orchard@chats.shakespeare.lit/Juliet'
    to='romeo@montague.lit/lane'>
  <status>Ay me!</status>
</presence>
```

Some time later, the MUC service chats.shakespeare.lit loses server to server connectivity to capulet.lit (the server upon which Juliet's account resides). If the server does not immediately remove Juliet as an occupant of the room, it MUST send Romeo an annotated presence, following the rules for server-unavailable defined above in s2s.

Listing 9: Romeo receives an annotated presence from the MUC room that Juliet's server is unavailable

```
<presence
    from='orchard@chats.shakespeare.lit/Juliet'
    to='romeo@montague.lit/lane'>
  <state-annotation xmlns="urn:xmpp:psa" from="chats.shakespeare.lit">
    <server-unavailable/>
    <description>The user's␣server␣is␣no␣longer␣reachable</description
       >
␣␣</state-annotation>
␣␣<status>Ay␣me!</status>
</presence>
```

If the S2S link becomes available and the MUC service has not yet ejected Juliet from the room, it MUST send an empty state-annotation to indicate this. If, instead, it has ejected Juliet in the meantime there is no additional stanza to send (as the ejection will already have been communicated to Romeo's client following the rules in xep45).

Listing 10: Romeo receives an annotated presence from the MUC room that Juliet's server is again available

```
<presence
    from='orchard@chats.shakespeare.lit/Juliet'
    to='romeo@montague.lit/lane'>
  <state-annotation xmlns="urn:xmpp:psa" from="chats.shakespeare.lit"/
     >
  <status>Ay me!</status>
</presence>
```

5 Business Rules

- When getting 'available again' for a remote MUC, MAY message yourself in the MUC and rejoin if not present.

- When MUC service re-establishes S2S, may send a probe to see if remote client is still online.

6 Security Considerations

Although this involves sending stanzas on behalf of another entity, these are always sent by entities that are already within trust domains able to spoof such stanzas (e.g. the local server sending to a local user, or a remote server sending on behalf of its local users).

7 IANA Considerations

None.

8 XMPP Registrar Considerations

Needs a namespace.

9 XML Schema

When advanced.