XEP-0349: Rayo Clustering

Ben Langfeld
mailto:ben@langfeld.me
xmpp:ben@langfeld.me
http://langfeld.me

Martín Pérez
mailto:mperez@tropo.com
http://tropo.com

2017-09-11
Version 0.2

Status	Type	Short Name
Deferred	Standards Track	NOT_YET_ASSIGNED

This specification describes an extension to the Rayo protocol to support clustering of Rayo servers and their presentation as a unified service.

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 Terminology **1**
3.1 Glossary 1
3.2 Conventions 2

4 Session Flow **2**
4.1 Message Routing 2
4.2 Load balancing 7
4.3 Failover 8
4.4 Security 8

5 Determining Support **8**

6 Security Considerations **8**
6.1 Denial of Service 8
6.2 Communication Through Gateways 8
6.3 Information Exposure 9

7 IANA Considerations **9**

8 XMPP Registrar Considerations **9**
8.1 Protocol Namespaces 9
8.2 Namespace Versioning 9

9 Acknowledgements **10**

1 Introduction

Rayo (XEP-0327) [1] documents the behaviour of an individual Rayo server and its interaction with a client. Two of the goals of Rayo, however, are to support clustering of servers and multi-tenancy. This specification outlines how that is to be achieved.

2 Requirements

This specification is required to provide a framework for implementation of the following goals:

- Routing of Rayo messages between clients and server nodes in a cluster.
- Load balancing of calls between multiple clients and server nodes in a cluster.
- Failover of calls between clients and server nodes, and management of the cluster.
- Enforcement of security and multi-tenancy concerns.

3 Terminology

3.1 Glossary

Cluster A collection of one or more Nodes which are coordinated by one or more Gateways.

Gateway The Rayo Gateway is the main element that differentiates a Rayo Cluster from a stand-alone Rayo deployment. The Rayo Gateway is an intermediate component that is in charge of coordinating the communication from clients to Rayo Nodes and from Rayo Nodes to clients. When a cluster contains multiple gateways, the protocol for coordination of multiple gateways is considered implementation specific.

Node The Rayo Node is an individual instance of a Rayo server from Rayo (XEP-0327) XEP-0327: Rayo <https://xmpp.org/extensions/xep-0327.html>., which presents call control capability via Rayo on top of a signaling protocol such as SIP or Jingle. It presents as a member of the cluster under the management of the Gateway(s).

Third-party XMPP server An arbitrary third-party XMPP server used for registration of Rayo clients. MAY form part of the cluster.

[1] XEP-0327: Rayo <https://xmpp.org/extensions/xep-0327.html>.

3.2 Conventions

In examples, the following JIDs are used:

- **juliet@capulet.lit/balcony, romeo@montague.lit/orchard** - Potential controlling parties
- **shakespeare.lit** - The root domain of the Rayo service, presented as the external interface of the Gateway(s).
- **gateway.shakespeare.lit** - The internal domain of the Gateway(s).
- **node1.shakespeare.lit** - The domain of a Node in the Cluster.

4 Session Flow

All communication between the Rayo Gateways and Clients, and the Gateways and Nodes MUST be conformant with Rayo (XEP-0327) [2] or any extensions defined in this specification. A conformant Rayo Gateway MUST NOT derive any critical functionality through proprietary knowledge of the Nodes it is managing.

4.1 Message Routing

The Rayo Gateway communicates both with Rayo Nodes and third-party XMPP servers through S2S XMPP connections. To differentiate the messages that come from Rayo Nodes and from client applications, the Rayo Gateway MUST present two separate domains, an internal (for communication with cluster nodes) and an external (for communication with Rayo clients) interface. In cases of multiple gateways, the internal and external domains MUST be the same, and DNS SHOULD be used for load-balancing.
Inbound call flow is:

1. The Rayo Node generates events for the call as usual, beginning with an offer event.
2. The Rayo Node sends the event to the Rayo Gateway using the internal interface.
3. The Rayo Gateway receives events on its internal interface, decides on the correct client to route the event to (the calls DCP or its set of PCPs).
4. The Rayo Gateway forwards the event to relevant clients via the external interface, replacing the presence ”to” and ”from” attributes. The ”to” attribute is the client’s jid and the ”from” is based on the Rayo Gateway’s external interface.

[2] XEP-0327: Rayo <https://xmpp.org/extensions/xep-0327.html>.

5. The client(s) receive the event, responding to the Gateway. The Gateway correspondingly rewrites commands received from the client to the relevant Node for each call, and responses to those commands in reverse.

Listing 1: Simple inbound call scenario

```
<presence from='9f00061@node1.shakespeare.lit'
          to='gateway.shakespeare.lit'>
  <c xmlns='http://jabber.org/protocol/caps'
     hash='sha-1'
     node='urn:xmpp:rayo:call:1'
     ver='QgayPKawpkPSDYmwT/WM94uAlu0='/>
  <offer xmlns='urn:xmpp:rayo:1'
         to='tel:+18003211212'
         from='tel:+13058881212'/>
</presence>

<presence from='9f00061@shakespeare.lit'
          to='juliet@capulet.lit/balcony'>
  <c xmlns='http://jabber.org/protocol/caps'
     hash='sha-1'
     node='urn:xmpp:rayo:call:1'
     ver='QgayPKawpkPSDYmwT/WM94uAlu0='/>
  <offer xmlns='urn:xmpp:rayo:1'
         to='tel:+18003211212'
         from='tel:+13058881212'/>
</presence>

<iq from='juliet@capulet.lit/balcony'
    to='9f00061@shakespeare.lit'
    type='set'
    id='hd721'>
  <accept xmlns='urn:xmpp:rayo:1'/>
</iq>

<iq from='gateway.shakespeare.lit'
    to='9f00061@node1.shakespeare.lit'
    type='set'
    id='hd721'>
  <accept xmlns='urn:xmpp:rayo:1'/>
</iq>

<iq from='9f00061@node1.shakespeare.lit'
    to='gateway.shakespeare.lit'
    type='result'
    id='hd721'/>

<iq from='9f00061@shakespeare.lit'
    to='juliet@capulet.lit/balcony'
    type='result'
```

```
    id='hd721'/>

<iq from='juliet@capulet.lit/balcony'
    to='9f00061@shakespeare.lit'
    type='set'
    id='f3wh8'>
  <hangup xmlns='urn:xmpp:rayo:1'/>
</iq>

<iq from='gateway.shakespeare.lit'
    to='9f00061@node1.shakespeare.lit'
    type='set'
    id='f3wh8'>
  <hangup xmlns='urn:xmpp:rayo:1'/>
</iq>

<iq from='9f00061@node1.shakespeare.lit'
    to='gateway.shakespeare.lit'
    type='result'
    id='f3wh8'/>

<iq from='9f00061@shakespeare.lit'
    to='juliet@capulet.lit/balcony'
    type='result'
    id='f3wh8'/>

<presence from='9f00061@node1.shakespeare.lit'
          to='gateway.shakespeare.lit'
          type='unavailable'>
  <end xmlns='urn:xmpp:rayo:1'>
    <hangup-command/>
  </end>
</presence>

<presence from='9f00061@shakespeare.lit'
          to='juliet@capulet.lit/balcony'
          type='unavailable'>
  <end xmlns='urn:xmpp:rayo:1'>
    <hangup-command/>
  </end>
</presence>
```

Outbound call flow is:

1. A client sends a dial command to the Rayo Cluster, arriving on the external interface of a Gateway.
2. The Rayo Gateway finds an available Rayo Node and forwards the command to it, replacing the ”from” attribute with its own internal interface and the ”to” attribute based on

the Node's domain.

3. The Rayo Node receives the dial command, processes the outbound call and returns responses/events to the Gateway for rewriting to the client.

Listing 2: Simple outbound call scenario

```
<iq from='juliet@capulet.lit/balcony'
    to='shakespeare.lit'
    type='set'
    id='h7ed2'>
  <dial xmlns='urn:xmpp:rayo:1'
        to='tel:+13055195825'
        from='tel:+14152226789'/>
</iq>

<iq from='gateway.shakespeare.lit'
    to='node1.shakespeare.lit'
    type='set'
    id='h7ed2'>
  <dial xmlns='urn:xmpp:rayo:1'
        to='tel:+13055195825'
        from='tel:+14152226789'/>
</iq>

<iq from='node1.shakespeare.lit'
    to='gateway.shakespeare.lit'
    type='result'
    id='h7ed2'>
  <ref xmlns='urn:xmpp:rayo:1' uri='xmpp:9f00061@shakespeare.lit'/>
</iq>

<iq from='shakespeare.lit'
    to='juliet@capulet.lit/balcony'
    type='result'
    id='h7ed2'>
  <ref xmlns='urn:xmpp:rayo:1' uri='xmpp:9f00061@shakespeare.lit'/>
</iq>

<presence from='9f00061@node1.shakespeare.lit'
          to='gateway.shakespeare.lit'>
  <ringing xmlns='urn:xmpp:rayo:1'/>
</presence>

<presence from='9f00061@shakespeare.lit'
          to='juliet@capulet.lit/balcony'>
  <ringing xmlns='urn:xmpp:rayo:1'/>
</presence>

<presence from='9f00061@node1.shakespeare.lit'
```

```
          to='gateway.shakespeare.lit'>
  <answered xmlns='urn:xmpp:rayo:1'/>
</presence>

<presence from='9f00061@shakespeare.lit'
          to='juliet@capulet.lit/balcony'>
  <answered xmlns='urn:xmpp:rayo:1'/>
</presence>

<iq from='juliet@capulet.lit/balcony'
    to='9f00061@shakespeare.lit'
    type='set'
    id='f3wh8'>
  <hangup xmlns='urn:xmpp:rayo:1'/>
</iq>

<iq from='gateway.shakespeare.lit'
    to='9f00061@node1.shakespeare.lit'
    type='set'
    id='f3wh8'>
  <hangup xmlns='urn:xmpp:rayo:1'/>
</iq>

<iq from='9f00061@node1.shakespeare.lit'
    to='gateway.shakespeare.lit'
    type='result'
    id='f3wh8'/>

<iq from='9f00061@shakespeare.lit'
    to='juliet@capulet.lit/balcony'
    type='result'
    id='f3wh8'/>

<presence from='9f00061@node1.shakespeare.lit'
          to='gateway.shakespeare.lit'
          type='unavailable'>
  <end xmlns='urn:xmpp:rayo:1'>
    <hangup-command/>
  </end>
</presence>

<presence from='9f00061@shakespeare.lit'
          to='juliet@capulet.lit/balcony'
          type='unavailable'>
  <end xmlns='urn:xmpp:rayo:1'>
    <hangup-command/>
  </end>
</presence>
```

4.2 Load balancing

The Gateway(s) in a Cluster are responsible for managing the routing of calls between relevant nodes and clients, and SHOULD retain knowledge of the presence of each for this purpose. Nodes and Clients SHOULD NOT be aware of each others identity of presence, and SHOULD only communicate with the Gateway(s).
The Gateway(s) in a Cluster MUST attempt to evenly balance outbound calls among Nodes; at a minimum they MUST implement round-robin dispatch of dial commands. Gateways MAY attempt load-based distribution by monitoring the number of active sessions (inbound and outbound) per Node and distributing accordingly.
The rules by which the PCPs for an inbound call are determined is implementation specific. In cases where a server permits registration of multiple JIDs as PCPs, it MAY opt to load-balance offers between them by an unspecified algorithm, though it may not assume any knowledge of the clients outside of this specification or Rayo (XEP-0327) [3].
In order for a Rayo Node to be considered available for processing dial requests, it MUST first notify the Gateway that it is available for such by sending directed presence to the Gateway internal interface with a <show/> element containing 'chat' as in the example:

Listing 3: Node presents itself as available to the Rayo Gateway

```
<presence from='node1.shakespeare.lit'
          to='gateway.shakespeare.lit'>
  <c xmlns='http://jabber.org/protocol/caps'
     hash='sha-1'
     node='urn:xmpp:rayo:node:1'
     ver='QgayPKawpkPSDYmwT/WM94uAlu0='/>
 <show>chat</show>
</presence>
```

Conversely, when a Rayo Node wishes not to process dial requests, it SHOULD send directed presence to the Gateway with a <show/> element containing 'dnd' as in the example:

Listing 4: Node presents itself as unavailable to the Rayo Gateway

```
<presence from='node1.shakespeare.lit'
          to='gateway.shakespeare.lit'>
  <c xmlns='http://jabber.org/protocol/caps'
     hash='sha-1'
     node='urn:xmpp:rayo:node:1'
     ver='QgayPKawpkPSDYmwT/WM94uAlu0='/>
 <show>dnd</show>
</presence>
```

[3] XEP-0327: Rayo <https://xmpp.org/extensions/xep-0327.html>.

4.3 Failover

A Rayo Gateway MAY transparently retry failed operations like dial requests, or MAY automatically remove from rotation the Rayo Nodes that fail to satisfy such requests repeatedly.

4.4 Security

The Rayo Gateway MUST validate permissions on incoming Rayo commands from Clients (check that they are one of the call's DCP/PCP as appropriate to the rules defined in Rayo (XEP-0327) [4]). The Rayo Gateway MUST enforce its own rules on Node membership of the Cluster, ensuring communication via its internal interface with only trusted Nodes. The rules by which inbound calls are permitted are implementation specific. When configured as members of a cluster, Rayo Nodes SHOULD accept communication *only* with the gateway.

5 Determining Support

Rayo gateways MUST advertise support for "urn:xmpp:rayo:1" on their external interface, and "urn:xmpp:rayo:gateway:1" on their internal interface. Rayo nodes MUST advertise support for "urn:xmpp:rayo:node:1", indicating that they may be used as part of a cluster, and additionally "urn:xmpp:rayo:1" if they may also be used separately from the cluster.

6 Security Considerations

6.1 Denial of Service

Rayo sessions can be resource-intensive. Therefore, it is possible to launch a denial-of-service attack against an entity by burdening it with too many Rayo sessions. Care must be taken to accept sessions only from known entities and only if the entity's device is able to process such sessions.

6.2 Communication Through Gateways

Rayo communications can be enabled through gateways to non-XMPP networks, whose security characteristics can be quite different from those of XMPP networks. For example, on some SIP networks authentication is optional and "from" addresses can be easily forged. Care must be taken in communicating through such gateways.

[4] XEP-0327: Rayo <https://xmpp.org/extensions/xep-0327.html>.

6.3 Information Exposure

Mere negotiation of a Rayo session can expose sensitive information about the parties (e.g. IP addresses). Care must be taken in communicating such information, and end-to-end encryption should be used if the parties do not trust the intermediate servers or gateways.

7 IANA Considerations

This document requires no interaction with the Internet Assigned Numbers Authority (IANA) [5].

8 XMPP Registrar Considerations

8.1 Protocol Namespaces

This specification defines the following XML namespaces:

- urn:xmpp:rayo:gateway:1
- urn:xmpp:rayo:node:1

The XMPP Registrar [6] includes the foregoing namespaces in its registry at <https://xmpp.org/registrar/namespaces.html>, as governed by XMPP Registrar Function (XEP-0053) [7].

8.2 Namespace Versioning

If the protocol defined in this specification undergoes a major revision that is not fully backward-compatible with an older version, or that contains significant new features, 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.

[5] 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/>.

[6] 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/>.

[7] XEP-0053: XMPP Registrar Function <https://xmpp.org/extensions/xep-0053.html>.

9 Acknowledgements

The authors would like to acknowledge the input of teams at Tropo, Mojo Lingo and Grasshopper in the development of the specification.
Specific individuals who have contributed to the specification or to software significant to its completion include:

- Ben Langfeld
- Chris Rienzo
- Martín Pérez