



# XMPP

## XEP-0282: DMUC2: Distributed MUC

Carlo von Loesch  
[xmpp:lynx@psyced.org](mailto:xmpp:lynx@psyced.org)

Philipp Hancke  
[xmpp:fippo@ve.symlynx.com](mailto:xmpp:fippo@ve.symlynx.com)

2021-03-04  
Version 0.1.1

Status	Type	Short Name
Deferred	Standards Track	NOT-YET-ASSIGNED

Multi-User Chats, distributed over several nodes in the XMPP network, using a primary/replica architecture

# 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

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Requirements</b>	<b>2</b>
<b>3</b>	<b>Terminology</b>	<b>2</b>
3.1	Dramatis Personae . . . . .	2
<b>4</b>	<b>PRIMARY/REPLICA Protocol</b>	<b>3</b>
4.1	Replica Requirements . . . . .	3
4.2	Creating a Replica . . . . .	3
4.3	Initial Room Roster . . . . .	3
4.4	Initial Room History . . . . .	4
4.5	ENTER . . . . .	4
4.6	LEAVE . . . . .	5
4.7	Presence Update . . . . .	5
4.8	Replica Forwards Stanza to Primary . . . . .	6
4.9	Message Broadcast . . . . .	6
4.10	Primary-Replica Keepalive . . . . .	7
4.11	Netsplit . . . . .	7
4.11.1	Netsplit Detection . . . . .	7
4.11.2	Netjoin . . . . .	8
<b>5</b>	<b>Use-Cases</b>	<b>8</b>
5.1	Entering a Room . . . . .	9
5.2	Sending a Message to All Occupants . . . . .	13
<b>6</b>	<b>Security Considerations</b>	<b>15</b>
<b>7</b>	<b>IANA Considerations</b>	<b>15</b>
<b>8</b>	<b>XMPP Registrar Considerations</b>	<b>15</b>
<b>9</b>	<b>XML Schema</b>	<b>15</b>

## 1 Introduction

*This document is one of several proposals for distributing XMPP chat rooms across multiple chat services. It is expected that the various approaches will be refined and harmonized before a final protocol is developed.*

The architecture is that of a single root node, called PRIMARY and several repeater nodes, each called REPLICA. Every stanza is submitted via the replicas to the primary, control is centralized there. The primary then sends a copy of the stanza to each of the replicas where it is processed and then distributed to each of the replica's leaf nodes, local users at this point. During redistribution, the replica shall not change the stanza's 'from' attribute, which is only possible if the replica is in the same security domain as the user. The result of that is a decreased number of messages on the server-to-server links between the primary's server and each of the replica servers.

```

BigCheese ----> vesa.yeggmaex.irc ----> wallops MUC ----> killrc.
      oulubox.irc ----> Wumpus
                                     killrc.
                                     oulubox.
                                     irc
                                     ---->
                                     Phaedrus
      wallops MUC ----> eris.dclxvii
      .irc ----> WiZ
                                     eris.dclxvii
                                     .irc
                                     ---->
                                     Troy
      wallops MUC ----> vesa.
      yeggmaex.irc ---->
      BigCheese
                                     vesa.
                                     yeggmaex
                                     .irc
                                     ---->
                                     Efchen

```

Note that this only applies to stanzas that are directed to all occupants, such as the change of availability status and messages whose 'type' attribute is set to 'groupchat'.

While 1-1 stanzas such as in-room private messages or vcard-temp requests may also travel along that path they are currently unaffected by this.

## 2 Requirements

This specification addresses the following requirements:

- The existence of the replica shall be transparent for the user.
- Enable detection of connection loss.
- Enable occupants to remain in instance of the conference if connectivity is lost to other instances.
- Enable occupants to leave a chatroom while connectivity is lost.
- Enable syncing of history and room rosters on reconnect.
- Improve distribution of 'broadcast' stanzas'.

## 3 Terminology

### 3.1 Dramatis Personae

Most of the examples in this document use the scenario of the 1990 war that split the Internet Relay Chat into several incompatible networks. The battlefield is represented here by the "wallops@channel.avalon.irc" chatroom. The characters are as follows:

Room Nickname	Full JID	Role
Wumpus	argl@killrc.oulubox.irc/laptop	Moderator
Valdis	valdis@killrc.oulubox.irc/desktop	Participant
Phaedrus	phaedrus@killrc.oulubox.irc/phone	Participant
WiZ	squit@eris.dclxvii.irc/jupiter	Moderator
Troy	troy@eris.dclxvii.irc/screwdriver	Moderator
BigCheese	rubenro@vesa.yeggmaex.irc/shaver	Participant
Efchen	msa@vesa.yeggmaex.irc/bicycle	Participant

In our example we are presuming that three replicas from three hosts have registered with the primary for traffic redistribution. They are doing so using replica@host jids, but they could be using any temporary jid the implementor finds useful to choose.

## 4 PRIMARY/REPLICA Protocol

This section describes the protocol between the primary and the replicas, the interaction between each replica and its users are described in the next section.

### 4.1 Replica Requirements

The replica

- must be able to intercept any stanzas sent to the primary's jid,
- must be able to verify if a local user has sent directed presence to the primary's jid,
- must store room roster,
- must store local room roster
- must store room history (if desired by primary)

### 4.2 Creating a Replica

to follow

Might include things like negotiation of keepalive frequency and amount of room might be useful to set some things like keepalive frequency, keeping/storing room history etc.

### 4.3 Initial Room Roster

After the creation of a new replica, the primary must send the room roster (which at that time only consists of remote participants) to the replica:

Listing 1: Primary Sends Room Roster to Replica

```
<presence
  from='wallops@channel.avalon.irc/Wumpus'
  to='replica@vesa.yeggmaex.irc'>
  <x xmlns='http://jabber.org/protocol/muc#user'>
    <item affiliation='owner' role='moderator' />
  </x>
</presence>
<presence
  from='wallops@channel.avalon.irc/Valdis'
  to='replica@vesa.yeggmaex.irc'>
  <x xmlns='http://jabber.org/protocol/muc#user'>
    <item affiliation='none' role='participant' />
  </x>
```

```

</presence>
<presence
  from='wallops@channel.avalon.irc/Phaedrus'
  to='replica@vesa.yeggmaex.irc'>
  <x xmlns='http://jabber.org/protocol/muc#user'>
    <item affiliation='none' role='participant' />
  </x>
</presence>
<presence
  from='wallops@channel.avalon.irc/WiZ'
  to='replica@vesa.yeggmaex.irc'>
  <x xmlns='http://jabber.org/protocol/muc#user'>
    <item affiliation='admin' role='moderator' />
  </x>
</presence>
<presence
  from='wallops@channel.avalon.irc/Troy'
  to='replica@vesa.yeggmaex.irc'>
  <x xmlns='http://jabber.org/protocol/muc#user'>
    <item affiliation='participant' role='moderator' />
  </x>
</presence>

```

#### 4.4 Initial Room History

After the creation of a new replica, the primary MAY also send the discussion history to the replica. The replica shall store this and deliver it to entering participants.  
 from=primaryjid to=replicajid?

#### 4.5 ENTER

If a new occupant requests to enter the room, the primary first sends a presence update to all participants to inform them of the presence of the new user. Then, the primary sends the affected replica a stanza requesting it to add the user to the distribution list

Listing 2: Primary informs Replica of New Occupant

```

<presence
  from='wallops@channel.avalon.irc/BigCheese'
  to='replica@vesa.yeggmaex.irc'>
  <x xmlns='urn:xmpp:tmp:dmuc:0'>
    <enter jid='rubenro@vesa.yeggmaex.irc/shaver' />
    <history maxstanzas='20' />
  </x>
  ...
</presence>

```

The replica MUST verify that the user has sent directed presence to the primary's JID before. This helps to ensure that the user intended to enter the room. If this is true, the replica shall add the user to the distribution list and send the room roster, occupants own presence in room and discussion history to the full jid of the added user.

#### 4.6 LEAVE

If an occupant sends an unavailable presence to the room, the primary sends the affected replica a stanza requesting it to remove the user from the distribution list.

Listing 3: Primary informs Replica of Occupant's Departure

```
<presence
  from='wallops@channel.avalon.irc/BigCheese'
  to='replica@vesa.yeggmaex.irc'
  type='unavailable'>
  <x xmlns='urn:xmpp:tmp:dmuc:0'>
    <leave jid='rubenro@vesa.yeggmaex.irc/shaver' />
  </x>
  ...
</presence>
```

The replica removes user and forwards the stanza to user.  
The primary then sends a presence update to all replicas to announce the occupants departure.

#### 4.7 Presence Update

Presence updates are distributed by the primary to all replicas.

Listing 4: Primary sends presence update

```
<presence
  from='wallops@channel.avalon.irc/BigCheese'
  to='replica@killrc.oulubox.irc'>
  <x xmlns='http://jabber.org/protocol/muc#user'>
    <item affiliation='none'
          jid='rubenro@vesa.yeggmaex.irc/shaver'
          role='participant' />
  </x>
  ...
</presence>
<presence
  from='wallops@channel.avalon.irc/BigCheese'
  to='replica@eris.dclxvii.irc'>
  <x xmlns='http://jabber.org/protocol/muc#user'>
```



```

    <item affiliation='none'
          jid='rubenro@vesa.yeggmaex.irc/shaver'
          role='participant' />
  </x>
  ...
</presence>
<presence
  from='wallops@channel.avalon.irc/BigCheese'
  to='replica@vesa.yeggmaex.irc'>
  ...
</presence>

```

Note that the primary MUST NOT send the full jid of the user to any replicas without members that are moderators.

When rebroadcasting this stanza to its local occupants, the replica MUST remove the participants full JID subject to the rules of XEP-0045. In addition, the replica stores the resulting changes to the room roster, so that it can send the correct state to entering users.

#### 4.8 Replica Forwards Stanza to Primary

The replica may relay messages from local users to the primary. When doing so, it MUST NOT modify the stanzas 'from' attribute but MUST retain the original address of the sender.

Listing 5: Replica Relays a Message to the Primary

```

<message
  from='rubenro@vesa.yeggmaex.irc/shaver'
  to='wallops@channel.avalon.irc'
  type='groupchat'>
  <body>Phaedrus , we're_all_having_a_truly_rotten_time.</body>
</message>

```

#### 4.9 Message Broadcast

When relaying a message, the primary SHOULD add a urn:xmpp:delay element inside dmuc element so that the replica can provide proper timestamps to new users. The primary then sends a copy of the stanza to each replica.

Listing 6: Message from Primary to all Replicas

```

<message
  from='wallops@channel.avalon.irc/BigCheese'
  to='replica@killrc.oulubox.irc'
  type='groupchat'>
  <body>Phaedrus , we're_all_having_a_truly_rotten_time.</body>
  <<x xmlns='urn:xmpp:tmp:dmuc:0'>

```

```

<delay xmlns='urn:xmpp:delay'
  from='wallops@channel.avalon.irc/BigCheese'
  stamp='1990-10-13T23:58:37Z' />
</x>
</message>
<message
  from='wallops@channel.avalon.irc/BigCheese'
  to='replica@eris.dclxvii.irc'
  type='groupchat'>
  <body>Phaedrus, we're all having a truly rotten time.</body>
  <x xmlns='urn:xmpp:tmp:dmuc:0'>
    <delay xmlns='urn:xmpp:delay'
      from='wallops@channel.avalon.irc/BigCheese'
      stamp='1990-10-13T23:58:37Z' />
    </x>
  </message>
<message
  from='wallops@channel.avalon.irc/BigCheese'
  to='replica@vesa.yeggmaex.irc'
  type='groupchat'>
  <body>Phaedrus, we're all having a truly rotten time.</body>
  <x xmlns='urn:xmpp:tmp:dmuc:0'>
    <delay xmlns='urn:xmpp:delay'
      from='wallops@channel.avalon.irc/BigCheese'
      stamp='1990-10-13T23:58:37Z' />
    </x>
  </message>

```

After saving the stanza for the purpose of keeping a room history, the replica SHOULD remove the urn:xmpp:tmp:dmuc:0 element and send the stanza to each local user.

## 4.10 Primary-Replica Keepalive

To ensure a working connection, the primary SHOULD send an XMPP Ping stanza to each replica if there has been no traffic for a certain amount of time.

Likewise, each replica should ping the primary if there has been no traffic for more than the usual amount of time.

## 4.11 Netsplit

### 4.11.1 Netsplit Detection

A 'lost connection' can be detected by either replica or primary when a stanza sent to the primary or a replica respectively bounced.

If the primary receives a stanza with type=error from the replica JID, it MUST:

- mark the replica as 'split', making sure that any further broadcasts are not sent to the affected replica
- start sending pings to the replica with a higher frequency to get a timely notification if the replica reappears
- resync when the replica reappears as described below

In addition, the primary MAY send a presence update for each user on the affected replica, marking them as away.

If the replica receives a stanza with type=error from the primary's JID, it MUST:

- stop submitting messages to the primary
- react to people sending presence updates or leaving the room and broadcast those changes to local users

In addition, the replica MAY (and be careful when using this):

- enable local participants to continue their in-room conversation with other local participants
- enable local participants to continue 1-1 messaging in the context of a room (such as private chat or vcard retrieval)
- enable local users to enter the room (DANGER!!!)
- enable moderators (as designated by the primary) to kick participants
- mark non-local users as away

#### 4.11.2 Netjoin

to follow

## 5 Use-Cases

This section narratively describes the primary-replica protocol in context with client interactions.

## 5.1 Entering a Room

The user seeks to enter the wallops chatroom with the room nickname BigCheese:

Listing 7: User Seeks to Enter Room

```
<presence
  from='rubenro@vesa.yeggmaex.irc/shaver'
  to='wallops@channel.avalon.irc/BigCheese'>
  <x xmlns='http://jabber.org/protocol/muc' />
</presence>
```

The primary sends the presence update to each replica to inform the current occupants of BigCheese's arrival:

Listing 8: Primary Sends Presence Update to all Replicas

```
<presence
  from='wallops@channel.avalon.irc/BigCheese'
  to='replica@killrc.oulubox.irc'>
  <x xmlns='http://jabber.org/protocol/muc#user'>
    <item affiliation='none'
          jid='rubenro@vesa.yeggmaex.irc/shaver'
          role='participant' />
  </x>
  ...
</presence>
<presence
  from='wallops@channel.avalon.irc/BigCheese'
  to='replica@eris.dclxvii.irc'>
  <x xmlns='http://jabber.org/protocol/muc#user'>
    <item affiliation='none'
          jid='rubenro@vesa.yeggmaex.irc/shaver'
          role='participant' />
  </x>
  ...
</presence>
<presence
  from='wallops@channel.avalon.irc/BigCheese'
  to='replica@vesa.yeggmaex.irc'>
  ...
</presence>
```

Each replica then distributes this presence update:

Listing 9: Replicas Distribute Presence Update to Users - killrc Replica

```
<presence
  from='wallops@channel.avalon.irc/BigCheese'
```

```

    to='argl@killrc.oulubox.irc/laptop'>
  <x xmlns='http://jabber.org/protocol/muc#user'>
    <item affiliation='none'
          jid='rubenro@vesa.yeggmaex.irc/shaver'
          role='participant' />
  </x>
  ...
</presence>
<presence
  from='wallops@channel.avalon.irc/BigCheese'
  to='valdis@killrc.oulubox.irc/desktop'>
  ...
</presence>
<presence
  from='wallops@channel.avalon.irc/BigCheese'
  to='phaedrus@killrc.oulubox.irc/phone'>
  ...
</presence>

```

Listing 10: Replicas Distribute Presence Update to Users - eris Replica

```

<presence
  from='wallops@channel.avalon.irc/WiZ'
  to='squit@eris.dclxvii.irc/jupiter'>
  <x xmlns='http://jabber.org/protocol/muc#user'>
    <item affiliation='none'
          jid='rubenro@vesa.yeggmaex.irc/shaver'
          role='participant' />
  </x>
  ...
</presence>
<presence
  from='wallops@channel.avalon.irc/WiZ'
  to='troy@eris.dclxvii.irc/screwdriver'>
  <x xmlns='http://jabber.org/protocol/muc#user'>
    <item affiliation='none'
          jid='rubenro@vesa.yeggmaex.irc/shaver'
          role='participant' />
  </x>
  ...
</presence>

```

Listing 11: Replicas Distribute Presence Update to Users - vesa Replica

```

<presence
  from='wallops@channel.avalon.irc/BigCheese'
  to='rubenro@vesa.yeggmaex.irc/shaver'>
  ...
</presence>

```

The primary then informs the replica of the entered user about a new occupant:

Listing 12: Primary Informs Replica of New Occupant

```
<presence
  from='wallops@channel.avalon.irc/BigCheese'
  to='replica@vesa.yeggmaex.irc'>
  <x xmlns='urn:xmpp:tmp:dmuc:0'>
    <enter jid='rubenro@vesa.yeggmaex.irc/shaver' />
  </x>
  ...
</presence>
```

The replica sends presence from existing occupants to new occupant:

Listing 13: Replica Sends Presence from Existing Occupants to New Occupant

```
<presence
  from='wallops@channel.avalon.irc/Wumpus'
  to='rubenro@vesa.yeggmaex.irc/shaver'>
  <x xmlns='http://jabber.org/protocol/muc#user'>
    <item affiliation='owner' role='moderator' />
  </x>
</presence>
<presence
  from='wallops@channel.avalon.irc/Valdis'
  to='rubenro@vesa.yeggmaex.irc/shaver'>
  <x xmlns='http://jabber.org/protocol/muc#user'>
    <item affiliation='none' role='participant' />
  </x>
</presence>
<presence
  from='wallops@channel.avalon.irc/Phaedrus'
  to='rubenro@vesa.yeggmaex.irc/shaver'>
  <x xmlns='http://jabber.org/protocol/muc#user'>
    <item affiliation='none' role='participant' />
  </x>
</presence>
<presence
  from='wallops@channel.avalon.irc/WiZ'
  to='rubenro@vesa.yeggmaex.irc/shaver'>
  <x xmlns='http://jabber.org/protocol/muc#user'>
    <item affiliation='admin' role='moderator' />
  </x>
</presence>
<presence
  from='wallops@channel.avalon.irc/Troy'
  to='rubenro@vesa.yeggmaex.irc/shaver'>
  <x xmlns='http://jabber.org/protocol/muc#user'>
```

```

    <item affiliation='participant' role='moderator' />
  </x>
</presence>
<presence
  from='wallops@channel.avalon.irc/Efchen'
  to='rubenro@vesa.yeggmaex.irc/shaver'>
  <x xmlns='http://jabber.org/protocol/muc#user'>
    <item affiliation='none' role='participant' />
  </x>
</presence>

```

and concludes the room roster by sending the new occupant's presence to the new occupant:

Listing 14: Replica Sends New Occupant's Presence to New Occupant

```

<presence
  from='wallops@channel.avalon.irc/BigCheese'
  to='rubenro@vesa.yeggmaex.irc/shaver'
  ...
</presence>

```

After that, the replica will send the discussion history to the new occupant:

Listing 15: Replica Sends Discussion History to New Occupant

```

<message from='wallops@channel.avalon.irc/Trillian'
  to='rubenro@vesa.yeggmaex.irc/shaver'
  type='groupchat'>
  <body>Honest! He did!!</body>
  <delay xmlns='urn:xmpp:delay'
    from='wallops@channel.avalon.irc/Trillian'
    stamp='1990-01-23T19:50:01Z' />
</message>
<message from='wallops@channel.avalon.irc/BlondeBoy'
  to='rubenro@vesa.yeggmaex.irc/shaver'
  type='groupchat'>
  <body>Was that absolutely necessary?????????</body>
  <delay xmlns='urn:xmpp:delay'
    from='wallops@channel.avalon.irc/BlondeBoy'
    stamp='1990-01-23T19:50:30Z' />
</message>
<message from='wallops@channel.avalon.irc/DCLXVI'
  to='rubenro@vesa.yeggmaex.irc/shaver'
  type='groupchat'>
  <body>SUUUUUUUURE he did</body>
  <delay xmlns='urn:xmpp:delay'
    from='wallops@channel.avalon.irc/DCLXVI'
    stamp='1990-01-23T19:50:51Z' />
</message>

```

```

<message from='wallops@channel.avalon.irc/Cerebus'
  to='rubenro@vesa.yeggmaex.irc/shaver'
  type='groupchat'>
  <body>He did. really truly he did.</body>
  <delay xmlns='urn:xmpp:delay'
    from='wallops@channel.avalon.irc/Cerebus'
    stamp='1990-01-23T19:51:20Z' />
</message>

```

## 5.2 Sending a Message to All Occupants

Listing 16: Occupant Sends a Message to All Occupants

```

<message
  from='rubenro@vesa.yeggmaex.irc/shaver'
  to='wallops@channel.avalon.irc'
  type='groupchat'>
  <body>Phaedrus, we're all having a truly rotten time.</body>
</message>

```

The primary will broadcast this message to all replicas:

Listing 17: Primary Distributes Message to all Replicas

```

<message
  from='wallops@channel.avalon.irc/BigCheese'
  to='replica@killrc.oulubox.irc'
  type='groupchat'>
  <body>Phaedrus, we're all having a truly rotten time.</body>
  <x xmlns='urn:xmpp:tmp:dmuc:0'>
    <delay xmlns='urn:xmpp:delay'
      from='wallops@channel.avalon.irc/BigCheese'
      stamp='1990-10-13T23:58:37Z' />
  </x>
</message>
<message
  from='wallops@channel.avalon.irc/BigCheese'
  to='replica@eris.dclxvii.irc'
  type='groupchat'>
  <body>Phaedrus, we're all having a truly rotten time.</body>
  <x xmlns='urn:xmpp:tmp:dmuc:0'>
    <delay xmlns='urn:xmpp:delay'
      from='wallops@channel.avalon.irc/BigCheese'
      stamp='1990-10-13T23:58:37Z' />
  </x>
</message>
<message
  from='wallops@channel.avalon.irc/BigCheese'

```



```

    to='replica@vesa.yeggmaex.irc'
    type='groupchat'>
    <body>Phaedrus, we're all having a truly rotten time.</body>
  </x xmlns='urn:xmpp:tmp:dmuc:0'>
  <delay xmlns='urn:xmpp:delay'
  <from='wallops@channel.avalon.irc/BigCheese'
  <stamp='1990-10-13T23:58:37Z' />
</x>
</message>

```

And each replica distributes to local occupants

Listing 18: Replica Distributes Message to Occupants - killrc Replica

```

<message
  from='wallops@channel.avalon.irc/BigCheese'
  to='argl@killrc.oulubox.irc/laptop'
  type='groupchat'>
  <body>Phaedrus, we're all having a truly rotten time.</body>
</message>
<message
  <from='wallops@channel.avalon.irc/BigCheese'
  <to='valdis@killrc.oulubox.irc/desktop'
  <type='groupchat'>
  <<body>Phaedrus, we're all having a truly rotten time.</body>
</message>
<message
  from='wallops@channel.avalon.irc/BigCheese'
  to='phaedrus@killrc.oulubox.irc/phone'
  type='groupchat'>
  <body>Phaedrus, we're all having a truly rotten time.</body>
</message>

```

Listing 19: Replica Distributes Message to Occupants - eris Replica

```

<message
  from='wallops@channel.avalon.irc/BigCheese'
  to='squit@eris.dclxvii.irc/jupiter'
  type='groupchat'>
  <body>Phaedrus, we're all having a truly rotten time.</body>
</message>
<message
  <from='wallops@channel.avalon.irc/BigCheese'
  <to='troy@eris.dclxvii.irc/screwdriver'
  <type='groupchat'>
  <<body>Phaedrus, we're all having a truly rotten time.</body>
</message>

```

Listing 20: Replica Distributes Message to Occupants - vesa Replica

```
<message
  from='wallops@channel.avalon.irc/BigCheese'
  to='rubenro@vesa.yeggmaex.irc/shaver'
  type='groupchat'>
  <body>Phaedrus, we're_all_having_a_truly_rotten_time.</body>
</message>
<message
  from='wallops@channel.avalon.irc/BigCheese'
  to='msa@vesa.yeggmaex.irc/bicycle'
  type='groupchat'>
  <body>Phaedrus, we're all having a truly rotten time.</body>
</message>
```

## 6 Security Considerations

to follow  
Careful with channel overtakes :-)

## 7 IANA Considerations

This document requires no interaction with the [Internet Assigned Numbers Authority \(IANA\)](#)<sup>1</sup>.

## 8 XMPP Registrar Considerations

This document requires no interaction with the [XMPP Registrar](#)<sup>2</sup>.

## 9 XML Schema

to follow

---

<sup>1</sup>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/>.

<sup>2</sup>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/>.