Ruby master - Bug #12838

Duplication of UDP packets for DNS responses causing "no address" results for valid hostnames

10/13/2016 08:50 PM - jschneiderhan (Jon-Erik Schneiderhan)

Status: Closed
Priority: Normal
Assignee: akr (Akira Tanaka)
Target version:

<table>
<thead>
<tr>
<th>ruby -v:</th>
<th>Backport:</th>
</tr>
</thead>
<tbody>
<tr>
<td>ruby 2.3.1p112 (2016-04-26 revision 54768) [x86_64-darwin14]</td>
<td>2.1: UNKNOWN, 2.2: UNKNOWN, 2.3: UNKNOWN, 2.7: DONE</td>
</tr>
</tbody>
</table>

### Description

A network that I'm running a Ruby app on has an issue where it is duplicating UDP packets (a separate issue that I need to fix). This is resulting in intermittent "not found" results for valid hostnames.

In my case, my resolver is setup to use multiple search domains, say bad1.com, bad2.com, and good.com. A lookup for hostname 'example' will perform lookups on example.bad1.com, example.bad2.com, example.good.com, and then finally plain 'example'. Say example.good.com is a valid hostname with a corresponding record. What I am seeing is that the duplication of the response s for the first two DNS queries are being read as the response for example.good.com, and I am getting a "no address for example.good.com" error message. Note that this is only happening every once in awhile, when the responses are duplicated.

I have been able to reproduce with the attached server.rb and client.rb files. I also noticed that if I changed the following line to:

```
if (s = sender_for(from, msg)) && s == sender
```

then my problems went away. I have to admit though, I don't really understand the entirely of that file. Not from lack of effort.

You should be able to reproduce the error by running server.rb and client.rb. You may need to use sudo for server.rb in order to bind to port 53 (or you can modify the files to use a higher port).

### Associated revisions

Revision 9682db06 - 11/07/2020 09:12 PM - jeremyevans (Jeremy Evans)

Remove sender/message_id pair after response received in resolv

Once a response for a given DNS request has been received (which requires a matching message id), the [sender, message_id] pair should be removed from the list of valid senders. This makes it so duplicate responses from the same sender are ignored.

Fixes [Bug #12838]

Revision 70c3a195 - 03/20/2021 04:18 AM - nagachika (Tomoyuki Chikanaga)

merge revision(s) 9682db065158da5fa4ec8a3bc267da45b429b92c: [Backport #12838] [Backport #17658]

```
Remove sender/message_id pair after response received in resolv
```

Once a response for a given DNS request has been received (which requires a matching message id), the [sender, message_id] pair should be removed from the list of valid senders. This makes it so duplicate responses from the same sender are ignored.

Fixes [Bug #12838]

Revision bde8e4e6 - 04/04/2021 11:47 PM - usa (Usaku NAKAMURA)

merge revision(s) 9682db065158da5fa4ec8a3bc267da45b429b92c: [Backport #17658]

```
Remove sender/message_id pair after response received in resolv
```

---

09/24/2021
Once a response for a given DNS request has been received (which requires a matching message id), the [sender, message_id] pair should be removed from the list of valid senders. This makes it so duplicate responses from the same sender are ignored.

Fixes [Bug #12838]
---
lib/resolv.rb | 2 +-  
test/resolv/test_dns.rb | 113 +++++++++++++++++++++++++++++++++++++++++++++++++  
2 files changed, 114 insertions(+), 1 deletion(-)

Revision 67929 - 04/04/2021 11:47 PM - usa (Usaku NAKAMURA)
merge revision(s) 9682db065158da5fa4ec8a3bc267da45b429b92c: [Backport #17658]

Remove sender/message_id pair after response received in resolv

Once a response for a given DNS request has been received (which requires a matching message id), the [sender, message_id] pair should be removed from the list of valid senders. This makes it so duplicate responses from the same sender are ignored.

Fixes [Bug #12838]
---
lib/resolv.rb | 2 +-  
test/resolv/test_dns.rb | 113 +++++++++++++++++++++++++++++++++++++++++++++++++  
2 files changed, 114 insertions(+), 1 deletion(-)

Revision 9edc1625 - 05/11/2021 03:37 AM - rhenium (Kazuki Yamaguchi)
[ruby/resolv] Fix confusion of received response message

This is a follow up for commit 33fb96619711 ("Remove sender/message_id pair after response received in resolv", 2020-09-11).

As the @senders instance variable is also used for tracking transaction ID allocation, simply removing an entry without releasing the ID would eventually deplete the ID space and cause Resolv::DNS.allocate_request_id to hang.

It seems the intention of the code was to check that the received DNS message is actually the response for the question made within the method earlier. Let's have it actually do so.

[Bug #12838] https://bugs.ruby-lang.org/issues/12838
[Bug #17748] https://bugs.ruby-lang.org/issues/17748
https://github.com/ruby/resolv/commit/53ca9c9209

Revision 9c0df2e8 - 05/22/2021 06:20 AM - nagachika (Tomoyuki Chikanaga)
merge revision(s) 9edc16253a4f685332239f6249745ad9b518cbe: [Backport #17781]
[ruby/resolv] Fix confusion of received response message

This is a follow up for commit 33fb96619711 ("Remove sender/message_id pair after response received in resolv", 2020-09-11).

As the @senders instance variable is also used for tracking transaction ID allocation, simply removing an entry without releasing the ID would eventually deplete the ID space and cause Resolv::DNS.allocate_request_id to hang.

It seems the intention of the code was to check that the received DNS message is actually the response for the question made within the method earlier. Let's have it actually do so.

[Bug #12838] https://bugs.ruby-lang.org/issues/12838
[Bug #17748] https://bugs.ruby-lang.org/issues/17748
https://github.com/ruby/resolv/commit/53ca9c9209
---
lib/resolv.rb | 6 +++---
Revision 87d02eac - 06/01/2021 04:40 PM - NAKAMURA Usaku
merge revision(s) 9edc16253a4f68532239f6249745ad9b518cbe: [Backport #17781]

[ruby/resolv] Fix confusion of received response message

This is a follow up for commit 33fb966197f1 ("Remove sender/message_id pair after response received in resolv", 2020-09-11).

As the @senders instance variable is also used for tracking transaction ID allocation, simply removing an entry without releasing the ID would eventually deplete the ID space and cause Resolv::DNS.allocate_request_id to hang.

It seems the intention of the code was to check that the received DNS message is actually the response for the question made within the method earlier. Let's have it actually do so.

[Bug #12838] https://bugs.ruby-lang.org/issues/12838
[Bug #17748] https://bugs.ruby-lang.org/issues/17748
https://github.com/ruby/resolv/commit/53ca9c9209
---
lib/resolv.rb |   6 +++---
1 file changed, 3 insertions(+), 3 deletions(-)

Revision 83c255e8 - 07/03/2021 04:56 PM - usa (Usaku NAKAMURA)
merge revision(s) 9edc1625: [Backport #17781]

[ruby/resolv] Fix confusion of received response message

This is a follow up for commit 33fb966197f1 ("Remove sender/message_id pair after response received in resolv", 2020-09-11).

As the @senders instance variable is also used for tracking transaction ID allocation, simply removing an entry without releasing the ID would eventually deplete the ID space and cause Resolv::DNS.allocate_request_id to hang.

It seems the intention of the code was to check that the received DNS message is actually the response for the question made within the method earlier. Let's have it actually do so.

[Bug #12838] https://bugs.ruby-lang.org/issues/12838
[Bug #17748] https://bugs.ruby-lang.org/issues/17748
https://github.com/ruby/resolv/commit/53ca9c9209
---
lib/resolv.rb |   6 +++---
1 file changed, 3 insertions(+), 3 deletions(-)

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/branches/ruby_2_6@67946 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 67946 - 07/03/2021 04:56 PM - usa (Usaku NAKAMURA)
merge revision(s) 9edc1625: [Backport #17781]

[ruby/resolv] Fix confusion of received response message

This is a follow up for commit 33fb966197f1 ("Remove sender/message_id pair after response received in resolv", 2020-09-11).

As the @senders instance variable is also used for tracking transaction ID allocation, simply removing an entry without releasing the ID would eventually deplete the ID space and cause Resolv::DNS.allocate_request_id to hang.

It seems the intention of the code was to check that the received DNS message is actually the response for the question made within the method earlier. Let's have it actually do so.

[Bug #12838] https://bugs.ruby-lang.org/issues/12838
Bug #17748] https://bugs.ruby-lang.org/issues/17748
https://github.com/ruby/resolv/commit/53ca9c9209

---
lib/resolv.rb | 6 +++---
1 file changed, 3 insertions(+), 3 deletions(-)

History

#1 - 10/14/2016 02:58 PM - jschneiderhan (Jon-Erik Schneiderhan)
- File check-sender.diff added

I added a diff file (check-sender.diff) instead of using the issue description to show the change I made to fix the error on my machine. The change and diff were made against the v2_3_1 tag in the git repo mirrored on github. I'm not necessarily suggesting it as the fix for the issue, just mentioning it as something interesting that I noticed.

#2 - 10/14/2016 03:20 PM - jschneiderhan (Jon-Erik Schneiderhan)

I looked through some of the DNS RFCs, at the suggestion of a colleague, to see if there was any mention of a standard way of dealing with duplicate responses. I didn't see anything specifically calling out duplicates, but I did find this section in the "Resolver Implementation" section of RFC 1035:

```
The next step is to match the response to a current resolver request. The recommended strategy is to do a preliminary matching using the ID field in the domain header, and then to verify that the question section corresponds to the information currently desired. This requires that the transmission algorithm devote several bits of the domain ID field to a request identifier of some sort.
```

I think the current problem is that the transaction ID in the response is not being matched up with the transaction ID in the request. "sender_for(from, msg)" is looking up the sender based on the ID in the response, but it is never checked to see if the ID matches the ID of the message sent earlier on in the "request" method.

#3 - 09/11/2020 10:46 PM - jeremyevans0 (Jeremy Evans)
- Assignee set to akr (Akira Tanaka)
- Status changed from Open to Assigned

This is still an issue in the master branch. The underlying problem is resolv does not remove the [sender, message_id] pair once a response has been received. I have submitted a pull request to fix this issue: https://github.com/ruby/ruby/pull/3536

#4 - 11/07/2020 09:12 PM - jeremyevans (Jeremy Evans)
- Status changed from Assigned to Closed

Applied in changeset git|9682db065158da5fa4ec8a3bc267da45b429b92c.

Remove sender/message_id pair after response received in resolv

Once a response for a given DNS request has been received (which requires a matching message id), the [sender, message_id] pair should be removed from the list of valid senders. This makes it so duplicate responses from the same sender are ignored.

Fixes [Bug #12838]

#5 - 03/20/2021 04:18 AM - nagachika (Tomoyuki Chikanaga)
- Backport changed from 2.1: UNKNOWN, 2.2: UNKNOWN, 2.3: UNKNOWN to 2.1: UNKNOWN, 2.2: UNKNOWN, 2.3: UNKNOWN, 2.7: DONE

ruby_2_7 70c3a195f9763dcdff9367d0c9b7e815431a41a merged revision(s) 9682db065158da5fa4ec8a3bc267da45b429b92c.

Files

<table>
<thead>
<tr>
<th>File</th>
<th>Size</th>
<th>Date</th>
<th>Commiter</th>
</tr>
</thead>
<tbody>
<tr>
<td>server.rb</td>
<td>2.17 KB</td>
<td>10/13/2016</td>
<td>jschneiderhan (Jon-Erik Schneiderhan)</td>
</tr>
<tr>
<td>client.rb</td>
<td>239 Bytes</td>
<td>10/13/2016</td>
<td>jschneiderhan (Jon-Erik Schneiderhan)</td>
</tr>
<tr>
<td>check-sender.diff</td>
<td>429 Bytes</td>
<td>10/14/2016</td>
<td>jschneiderhan (Jon-Erik Schneiderhan)</td>
</tr>
</tbody>
</table>