

Ruby - Bug #14997

Socket connect timeout exceeds the timeout value for

08/16/2018 08:56 AM - maciej.mensfeld (Maciej Mensfeld)

Status:	Closed	
Priority:	Normal	
Assignee:		
Target version:		
ruby -v:	2.5.1	Backport: 2.3: UNKNOWN, 2.4: UNKNOWN, 2.5: UNKNOWN

Description

Given a case, where a domain is being resolved to multiple IPs (4 in the following example):

```
dig debug-xyz.elb.us-east-1.amazonaws.com a

; <<>> DiG 9.10.3-P4-Ubuntu <<>> debug-xyz.elb.us-east-1.amazonaws.com a
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 54375
;; flags: qr rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;debug-xyz.elb.us-east-1.amazonaws.com. IN A

;; ANSWER SECTION:
debug-xyz.elb.us-east-1.amazonaws.com. 60 IN A 172.31.86.79
debug-xyz.elb.us-east-1.amazonaws.com. 60 IN A 172.31.109.24
debug-xyz.elb.us-east-1.amazonaws.com. 60 IN A 172.31.119.55
debug-xyz.elb.us-east-1.amazonaws.com. 60 IN A 172.31.71.167

;; Query time: 4 msec
;; SERVER: 172.31.0.2#53(172.31.0.2)
;; WHEN: Tue Aug 14 13:46:18 UTC 2018
;; MSG SIZE rcvd: 132
```

and when connect_timeout is set to a certain value (N), the overall timeout upon non-responsive endpoints that don't immediately throw an exception can reach N * 4.

This can disrupt some time-sensitive systems.

We've experienced it with the following setup:

- TCP server (event machine) behind an AWS NLB
- TCP server process goes down behind NLB but NLB is still responsive
- Socket connect_timeout is set to 100ms
- AWS NLB keeps the connection in the waiting state hoping that the service behind it will get back to normal (but it doesn't)
- Ruby timeouts after 100ms
- Ruby tries to connect to the next IP from the pool (AWS NLB again)
- Due to 4 hosts resolving, the overall timeout is 400ms.

Not sure whether this should be qualified as a bug or a feature, but I believe it should be definitely documented or there should be an option to "hard" block this limit.

Here's the code actually responsible for this behavior: <https://github.com/ruby/ruby/blob/trunk/ext/socket/lib/socket.rb#L631-L664>

Related issues:	
Related to Ruby - Feature #15553: Addrinfo.getaddrinfo supports timeout	Closed

History

#1 - 08/16/2018 08:57 AM - maciej.mensfeld (Maciej Mensfeld)

- Description updated

#2 - 08/24/2018 02:47 PM - maciej.mensfeld (Maciej Mensfeld)

If anyone is actually willing to confirm, that it is indeed an unwanted / unexpected behavior, I offer to fix it.

It could be fixed by tracking how much of the time "pool" has been used and lowering the timeout value appropriate for the next attempts. That would guarantee, that we would never exceed the timeout.

I think this is the most elegant solution.

#3 - 07/02/2019 10:58 AM - tenderlovmaking (Aaron Patterson)

This really sounds like a bug to me. Please make a patch and I will apply it.

#4 - 07/30/2019 03:35 AM - Glass_saga (Masaki Matsushita)

- *Related to Feature #15553: Addrinfo.getaddrinfo supports timeout added*

#5 - 11/26/2019 02:03 PM - kirs (Kir Shatrov)

tenderlovmaking (Aaron Patterson) wrote:

This really sounds like a bug to me. Please make a patch and I will apply it.

Do you mind taking a look at <https://github.com/ruby/ruby/pull/1806>? Based on my testing it's solving the problem.

Together with <https://bugs.ruby-lang.org/issues/15553> (already merged), many of us at Shopify would really love to see that fixed in 2.7 as it would improve resiliency and avoid Ruby processes to hang for 10s (default resolv timeout) when DNS is experiencing issues.

#6 - 08/31/2020 09:38 PM - jeremyevans0 (Jeremy Evans)

- *Status changed from Open to Closed*

I believe this timeout issue is now solved by the Socket.tcp :resolv_timeout option, introduced in [6382f5cc91ac9e36776bc854632d9a1237250da7](#).