

## Ruby master - Bug #9525

### Stuck with Socket.pack\_sockaddr\_in

02/17/2014 11:30 AM - sonots (Naotoshi Seo)

<b>Status:</b> Closed	
<b>Priority:</b> Normal	
<b>Assignee:</b>	
<b>Target version:</b>	
<b>ruby -v:</b> 1.9.3p194	<b>Backport:</b> 2.1: DONE
<b>Description</b> We met this trouble with Fluentd <a href="https://github.com/fluent/fluentd">https://github.com/fluent/fluentd</a> .  Fluentd is sometimes stuck at Socket.pack_sockaddr_in line on shutdown. Here is the gist <a href="https://gist.github.com/sonots/9047653">https://gist.github.com/sonots/9047653</a> to explain details.	
<b>Related issues:</b>	
Related to Backport193 - Backport #9538: backport r45047	<b>Closed</b> 02/20/2014
Related to Backport21 - Backport #9536: backport r45047	<b>Closed</b>

#### Associated revisions

##### Revision 2e6b97a4 - 02/19/2014 10:37 AM - akr (Akira Tanaka)

- ext/socket: Bypass getaddrinfo() if node and serv are numeric. Reported by Naotoshi Seo. [ruby-core:60801] [Bug #9525]
- ext/socket/extconf.rb: Detect struct sockaddr\_in6.sin6\_len.
- ext/socket/socketport.h (SET\_SIN6\_LEN): New macro. (INIT\_SOCKADDR\_IN6): Ditto.
- ext/socket/rubysocket.h (struct rb\_addrinfo): Add allocated\_by\_malloc field.
- ext/socket/raddrinfo.c (numeric\_getaddrinfo): New function. (rb\_getaddrinfo): Call numeric\_getaddrinfo at first. (rb\_freeaddrinfo): Free struct addrinfo properly when it is allocated by numeric\_getaddrinfo.

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@45047 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

##### Revision 45047 - 02/19/2014 10:37 AM - akr (Akira Tanaka)

- ext/socket: Bypass getaddrinfo() if node and serv are numeric. Reported by Naotoshi Seo. [ruby-core:60801] [Bug #9525]
- ext/socket/extconf.rb: Detect struct sockaddr\_in6.sin6\_len.
- ext/socket/socketport.h (SET\_SIN6\_LEN): New macro. (INIT\_SOCKADDR\_IN6): Ditto.
- ext/socket/rubysocket.h (struct rb\_addrinfo): Add allocated\_by\_malloc field.
- ext/socket/raddrinfo.c (numeric\_getaddrinfo): New function. (rb\_getaddrinfo): Call numeric\_getaddrinfo at first. (rb\_freeaddrinfo): Free struct addrinfo properly when it is allocated by numeric\_getaddrinfo.

**Revision 45047 - 02/19/2014 10:37 AM - akr (Akira Tanaka)**

- ext/socket: Bypass getaddrinfo() if node and serv are numeric. Reporeted by Naotoshi Seo. [ruby-core:60801] [Bug #9525]
- ext/socket/extconf.rb: Detect struct sockaddr\_in6.sin6\_len.
- ext/socket/sockport.h (SET\_SIN6\_LEN): New macro. (INIT\_SOCKADDR\_IN6): Ditto.
- ext/socket/rubysocket.h (struct rb\_addrinfo): Add allocated\_by\_malloc field.
- ext/socket/raddrinfo.c (numeric\_getaddrinfo): New function. (rb\_getaddrinfo): Call numeric\_getaddrinfo at first. (rb\_freeaddrinfo): Free struct addrinfo properly when it is allocated by numeric\_getaddrinfo.

**Revision 45047 - 02/19/2014 10:37 AM - akr (Akira Tanaka)**

- ext/socket: Bypass getaddrinfo() if node and serv are numeric. Reporeted by Naotoshi Seo. [ruby-core:60801] [Bug #9525]
- ext/socket/extconf.rb: Detect struct sockaddr\_in6.sin6\_len.
- ext/socket/sockport.h (SET\_SIN6\_LEN): New macro. (INIT\_SOCKADDR\_IN6): Ditto.
- ext/socket/rubysocket.h (struct rb\_addrinfo): Add allocated\_by\_malloc field.
- ext/socket/raddrinfo.c (numeric\_getaddrinfo): New function. (rb\_getaddrinfo): Call numeric\_getaddrinfo at first. (rb\_freeaddrinfo): Free struct addrinfo properly when it is allocated by numeric\_getaddrinfo.

**Revision 45047 - 02/19/2014 10:37 AM - akr (Akira Tanaka)**

- ext/socket: Bypass getaddrinfo() if node and serv are numeric. Reporeted by Naotoshi Seo. [ruby-core:60801] [Bug #9525]
- ext/socket/extconf.rb: Detect struct sockaddr\_in6.sin6\_len.
- ext/socket/sockport.h (SET\_SIN6\_LEN): New macro. (INIT\_SOCKADDR\_IN6): Ditto.
- ext/socket/rubysocket.h (struct rb\_addrinfo): Add allocated\_by\_malloc field.
- ext/socket/raddrinfo.c (numeric\_getaddrinfo): New function. (rb\_getaddrinfo): Call numeric\_getaddrinfo at first. (rb\_freeaddrinfo): Free struct addrinfo properly when it is allocated by numeric\_getaddrinfo.

**Revision 45047 - 02/19/2014 10:37 AM - akr (Akira Tanaka)**

- ext/socket: Bypass getaddrinfo() if node and serv are numeric. Reporeted by Naotoshi Seo. [ruby-core:60801] [Bug #9525]

- ext/socket/extconf.rb: Detect struct sockaddr\_in6.sin6\_len.
- ext/socket/socketport.h (SET\_SIN6\_LEN): New macro. (INIT\_SOCKADDR\_IN6): Ditto.
- ext/socket/rubysocket.h (struct rb\_addrinfo): Add allocated\_by\_malloc field.
- ext/socket/raddrinfo.c (numeric\_getaddrinfo): New function. (rb\_getaddrinfo): Call numeric\_getaddrinfo at first. (rb\_freeaddrinfo): Free struct addrinfo properly when it is allocated by numeric\_getaddrinfo.

#### Revision 45047 - 02/19/2014 10:37 AM - akr (Akira Tanaka)

- ext/socket: Bypass getaddrinfo() if node and serv are numeric. Reported by Naotoshi Seo. [ruby-core:60801] [Bug #9525]
- ext/socket/extconf.rb: Detect struct sockaddr\_in6.sin6\_len.
- ext/socket/socketport.h (SET\_SIN6\_LEN): New macro. (INIT\_SOCKADDR\_IN6): Ditto.
- ext/socket/rubysocket.h (struct rb\_addrinfo): Add allocated\_by\_malloc field.
- ext/socket/raddrinfo.c (numeric\_getaddrinfo): New function. (rb\_getaddrinfo): Call numeric\_getaddrinfo at first. (rb\_freeaddrinfo): Free struct addrinfo properly when it is allocated by numeric\_getaddrinfo.

#### Revision 667af1ee - 09/05/2014 03:14 PM - nagachika (Tomoyuki Chikanaga)

merge revision(s) r45046,r45047,r45063,r45087,r45146,r45150,r45151,r45152: [Backport #9525]

- \* ext/socket: Wrap struct addrinfo by struct rb\_addrinfo.
- \* ext/socket: Bypass getaddrinfo() if node and serv are numeric. Reported by Naotoshi Seo. [ruby-core:60801] [Bug #9525]
- \* ext/socket/extconf.rb: Detect struct sockaddr\_in6.sin6\_len.
- \* ext/socket/socketport.h (SET\_SIN6\_LEN): New macro. (INIT\_SOCKADDR\_IN6): Ditto.
- \* ext/socket/rubysocket.h (struct rb\_addrinfo): Add allocated\_by\_malloc field.
- \* ext/socket/raddrinfo.c (numeric\_getaddrinfo): New function. (rb\_getaddrinfo): Call numeric\_getaddrinfo at first. (rb\_freeaddrinfo): Free struct addrinfo properly when it is allocated by numeric\_getaddrinfo.
- \* ext/socket/raddrinfo.c (numeric\_getaddrinfo): Use xcalloc. Suggested by Eric Wong. <https://bugs.ruby-lang.org/issues/9525#note-14>
- \* ext/socket/raddrinfo.c (rb\_getaddrinfo): second argument of MEMZERO is type. Coverity Scan found this bug.
- \* include/ruby/win32.h, win32/win32.c (rb\_w32\_inet\_pton): add a wrapper function for inet\_pton minimum supported client is Vista, as well as inet\_ntop.
- \* ext/socket/option.c (inet\_pton): use rb\_w32\_inet\_pton, instead of inet\_ntop directly, which is unavailable on older version Windows.
- \* ext/socket/raddrinfo.c (inet\_pton): use rb\_w32\_inet\_pton, instead of

inet\_pton directly, which is unavailable on older version Windows.

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/branches/ruby\_2\_1@47415 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

### Revision 47415 - 09/05/2014 03:14 PM - nagachika (Tomoyuki Chikanaga)

merge revision(s) r45046,r45047,r45063,r45087,r45146,r45150,r45151,r45152: [Backport #9525]

- \* ext/socket: Wrap struct addrinfo by struct rb\_addrinfo.
- \* ext/socket: Bypass getaddrinfo() if node and serv are numeric.  
Reported by Naotoshi Seo. [ruby-core:60801] [Bug #9525]
- \* ext/socket/extconf.rb: Detect struct sockaddr\_in6.sin6\_len.
- \* ext/socket/sockport.h (SET\_SIN6\_LEN): New macro.  
(INIT\_SOCKADDR\_IN6): Ditto.
- \* ext/socket/rubysocket.h (struct rb\_addrinfo): Add  
allocated\_by\_malloc field.
- \* ext/socket/raddrinfo.c (numeric\_getaddrinfo): New function.  
(rb\_getaddrinfo): Call numeric\_getaddrinfo at first.  
(rb\_freeaddrinfo): Free struct addrinfo properly when it is  
allocated by numeric\_getaddrinfo.
- \* ext/socket/raddrinfo.c (numeric\_getaddrinfo): Use xcalloc.  
Suggested by Eric Wong.  
<https://bugs.ruby-lang.org/issues/9525#note-14>
- \* ext/socket/raddrinfo.c (rb\_getaddrinfo): second argument of  
MEMZERO is type. Coverity Scan found this bug.
- \* include/ruby/win32.h, win32/win32.c (rb\_w32\_inet\_pton): add a  
wrapper function for inet\_pton minimum supported client is  
Vista, as well as inet\_ntop.
- \* ext/socket/option.c (inet\_pton): use rb\_w32\_inet\_pton, instead of  
inet\_ntop directly, which is unavailable on older version Windows.
- \* ext/socket/raddrinfo.c (inet\_pton): use rb\_w32\_inet\_pton, instead of  
inet\_pton directly, which is unavailable on older version Windows.

## History

---

### #1 - 02/17/2014 11:45 AM - akr (Akira Tanaka)

- Status changed from Open to Feedback

Socket.pack\_sockaddr\_in can block with a name resolution.  
If you think the name resolution doesn't block, please explain why.

### #2 - 02/17/2014 11:49 AM - sonots (Naotoshi Seo)

Are there any ways to timeout Socket.pack\_sockaddr\_in?  
If so, this problem should be resolved.

EDIT: And, I was pointing an IP address at the concerned line. Will it still block for the name resolution?

### #3 - 02/17/2014 11:52 AM - akr (Akira Tanaka)

I think it is very difficult to add timeout to getaddrinfo() function in C.

### #4 - 02/17/2014 11:55 AM - akr (Akira Tanaka)

I guess getaddrinfo() doesn't block if an IP address is given.

At least, following command on Debian GNU/Linux (jessie) doesn't show anything.

```
% strace -e socket ruby -rsocket -e 'Socket.pack_sockaddr_in(80, "192.168.1.1")'
```

It means the command create no socket and no communication to another site.

### #5 - 02/17/2014 12:08 PM - sonots (Naotoshi Seo)

Following command on CentOS release 6.2 x86\_64 returned a line

```
$ strace -e socket /usr/lib64/fluent/ruby/bin/ruby -rsocket -e 'Socket.pack_sockaddr_in(80, "192.168.1.1")'  
socket(PF_NETLINK, SOCK_RAW, 0) = 5
```

```
/usr/lib64/fluent/ruby/bin/ruby -v #=> ruby 1.9.3p194 (2012-04-20 revision 35410) [x86_64-linux]
```

#### #6 - 02/17/2014 12:12 PM - akr (Akira Tanaka)

I think that PF\_NETLINK is not a socket to communicate to another host.

#### #7 - 02/18/2014 03:48 AM - naruse (Yui NARUSE)

- Status changed from Feedback to Third Party's Issue

gdb says \_\_check\_pf is guilty.

<https://gist.github.com/sonots/9047653#file-gistfile3-txt>

```
(gdb) bt  
#0 0x00000032f2ce659d in recvmsg () from /lib64/libc.so.6  
#1 0x00000032f2d0c2c5 in make_request () from /lib64/libc.so.6  
#2 0x00000032f2d0c6fa in __check_pf () from /lib64/libc.so.6  
#3 0x00000032f2ccfb47 in getaddrinfo () from /lib64/libc.so.6  
#4 0x00007fb3139d3818 in nogvl_getaddrinfo (arg=<value optimized out>) at raddrinfo.c:161  
#5 0x00000000004f600c in rb_thread_blocking_region (func=0x7fb3139d3800 <nogvl_getaddrinfo>, data1=0x7fb0d5dd0140, ubf=<value optimized out>, data2=<value optimized out>) at thread.c:1129  
#6 0x00007fb3139d2eaf in rb_getaddrinfo (node=<value optimized out>, service=<value optimized out>, hints=<value optimized out>, res=<value optimized out>) at raddrinfo.c:181  
#7 0x00007fb3139d2fcb in rsock_getaddrinfo (host=<value optimized out>, port=22017, hints=0x7fb0d5dd0600, socktype_hack=1) at raddrinfo.c:359  
#8 0x00007fb3139d37ed in rsock_addrinfo (host=0, port=140397479001552, socktype=<value optimized out>, flags=659) at raddrinfo.c:379  
#9 0x00007fb3139c939a in sock_s_pack_sockaddr_in (self=<value optimized out>, port=140397479001552, host=35158400) at socket.c:1307
```

By glibc's commits, once \_\_check\_pf is always called [6f3914d5a3269c00e70506bd95f816fef6b635ce](https://github.com/glibc/glibc/commit/6f3914d5a3269c00e70506bd95f816fef6b635ce) (it is fixed at [fa3fc0fe5f452d0aa7e435d8f32e992958683819](https://github.com/glibc/glibc/commit/fa3fc0fe5f452d0aa7e435d8f32e992958683819))

The difference between akr's and sonots' seems because of this.

Therefore this is glibc's old bug and RHEL/CentOS's backport issue.

#### #8 - 02/18/2014 05:24 AM - sonots (Naotoshi Seo)

Thank you! Let me write a note here.

On my current environment, the glibc version was glibc-2.12-1.47.el6.x86\_64.

I checked the latest centos rpm <http://vault.centos.org/6.5/os/Source/SPackages/glibc-2.12-1.132.el6.src.rpm>, but it looked RHED/CentOS is not backporting it yet (cf. <https://gist.github.com/sonots/9065060>)

#### #9 - 02/18/2014 03:38 PM - akr (Akira Tanaka)

I think we can accept a workaround if there is a good patch.

#### #10 - 02/18/2014 06:57 PM - kosaki (Motohiro KOSAKI)

I checked a glibc code. I don't think fa3fc0fe5f fixed this issue. It is a mere optimization patch. I'm not sure why kernel's netlink doesn't reply anything. But it may be worth to try upgrade your kernel.

Thanks.

#### #11 - 02/18/2014 07:35 PM - kosaki (Motohiro KOSAKI)

Akr-san,

I'm not an expert this area. But I guess we don't need to call getaddrinfo() in this case because the name was already resolved.

How about bypass to call getaddrinfo when 'host' is given by IP address?

#### #12 - 02/18/2014 11:48 PM - akr (Akira Tanaka)

Motohiro KOSAKI wrote:

I'm not an expert this area. But I guess we don't need to call getaddrinfo() in this case

because the name was already resolved.  
How about bypass to call getaddrinfo when 'host' is given by IP address?

Yes. It is the workaround I said.

**#13 - 02/20/2014 06:35 AM - akr (Akira Tanaka)**

I tried to workaround this issue at r45047.  
However I don't have an environment to reproduce the problem.

Would anyone test the problem at latest trunk?

**#14 - 02/20/2014 07:53 AM - normalperson (Eric Wong)**

[akr@fsij.org](mailto:akr@fsij.org) wrote:

I tried to workaround this issue at r45047.  
However I don't have an environment to reproduce the problem.

I don't know how to reproduce the problem, either. netlink sockets should not block like that.

r45047 looks correct. A few minor comments:

- backwards for loop for list is confusing to me, any reason for not reversing list declaration?
- why xmalloc + MEMZERO? xcalloc is shorter and generates smaller code

**#15 - 02/20/2014 08:57 AM - kosaki (Motohiro KOSAKI)**

- Related to Backport #9538: backport r45047 added

**#16 - 02/20/2014 08:58 AM - kosaki (Motohiro KOSAKI)**

- Related to Backport #9536: backport r45047 added

**#17 - 02/20/2014 08:58 AM - kosaki (Motohiro KOSAKI)**

- Status changed from Third Party's Issue to Closed

- Backport changed from 1.9.3: UNKNOWN, 2.0.0: UNKNOWN, 2.1: UNKNOWN to 1.9.3: REQUIRED, 2.0.0: REQUIRED, 2.1: REQUIRED

**#18 - 02/20/2014 01:23 PM - akr (Akira Tanaka)**

Eric Wong wrote:

- backwards for loop for list is confusing to me, any reason for not reversing list declaration?

I prefer ordering consistency between "list" and "res".

- why xmalloc + MEMZERO? xcalloc is shorter and generates smaller code

xcalloc is good idea.

**#19 - 03/04/2014 07:09 AM - sonots (Naotoshi Seo)**

Akira Tanaka wrote:

Would anyone test the problem at latest trunk?

I had never ran fluentd with ruby 2.1.1, but now I am trying it.  
If it works well, I will try ruby-trunk next.

PS. Sorry, I struggled, but could not create a small subset to reproduce this issue so that anyone can check.  
What I can tell here is only that this issue would sometimes occur on an environment which is heavy-loaded, and uses many threads, but I am not sure.

**#20 - 03/06/2014 05:04 PM - sonots (Naotoshi Seo)**

I applied following consecutive patches for ruby 2.1.1, and tried.

<https://github.com/ruby/ruby/commit/948ce9decb97e5ff0833e53a392aa9f1d42c9b0d>

<https://github.com/ruby/ruby/commit/dd1c3a75096b97c1ebcb8597c001761ddf3c1bf>  
<https://github.com/ruby/ruby/commit/2e6b97a45d077979121b29484a8831034d47ef50>

Before (ruby 2.1.1):

The stuck was able to be reproduced with ruby 2.1.1.

I restarted my entire fluentd cluster 12 times, and it occurred at 4th and 12th, that is, 2 times out of 12.

After (ruby 2.1.1 + patch)

I restarted the fluentd cluster 36 times, and got stuck 3 times, but the stuck point was changed.

See <https://gist.github.com/sonots/9392668>

I did not see that Fluentd was stuck at Socket.pack\_sockaddr\_in anymore.

I will continue to work for the new stuck point at Fluentd issue <https://github.com/fluent/fluentd/pull/257>. I think the problem of stuck at Socket.pack\_sockaddr\_in was resolved.

**#21 - 04/30/2014 07:02 AM - usa (Usaku NAKAMURA)**

memo: related commits: 45046,45047,45063,45087,45150,45151,45152

**#22 - 08/25/2014 02:24 PM - nagachika (Tomoyuki Chikanaga)**

Is there any progress?

**#23 - 09/05/2014 02:54 PM - nagachika (Tomoyuki Chikanaga)**

I have confirmed that this issue is already fixed.

<https://twitter.com/sonots/status/507804708153483264>

Add related commits: r45146

**#24 - 09/05/2014 03:15 PM - nagachika (Tomoyuki Chikanaga)**

- Backport changed from 1.9.3: REQUIRED, 2.0.0: REQUIRED, 2.1: REQUIRED to 1.9.3: REQUIRED, 2.0.0: REQUIRED, 2.1: DONE

r45046, r45047, r45063, r45087, r45146, r45150, r45151 and r45152 were backported into ruby\_2\_1 branch at r47415.

**#25 - 07/13/2016 07:11 AM - mariakatosvich (maria katosvich)**

- Backport deleted (1.9.3: REQUIRED, 2.0.0: REQUIRED, 2.1: DONE)

Related issue: <https://bugs.ruby-lang.org/issues/9536>

This problem is a big issue for Fluentd.

We hope to backport this commit to Latest Rubies.

Thanks,

[sky\\_contact\\_number\\_uk@fixithere.net](mailto:sky_contact_number_uk@fixithere.net)

**#26 - 07/13/2016 10:19 AM - nobu (Nobuyoshi Nakada)**

- Backport set to 2.1: DONE