

## Ruby trunk - Bug #13407

### We have recv\_nonblock but not send\_nonblock... can we add it?

04/06/2017 12:26 PM - ioquatix (Samuel Williams)

<b>Status:</b> Assigned	
<b>Priority:</b> Normal	
<b>Assignee:</b> akr (Akira Tanaka)	
<b>Target version:</b>	
<b>ruby -v:</b>	<b>Backport:</b> 2.2: UNKNOWN, 2.3: UNKNOWN, 2.4: UNKNOWN
<b>Description</b> We have recv_nonblock, read_nonblock, write_nonblock but not BasicSocket#send_nonblock. Is this a mistake?	

#### History

##### #1 - 04/06/2017 08:32 PM - normalperson (Eric Wong)

[samuel@oriontransfer.org](mailto:samuel@oriontransfer.org) wrote:

Bug #13407: We have recv\_nonblock but not send\_nonblock... can we add it?  
<https://bugs.ruby-lang.org/issues/13407>

We have recv\_nonblock, read\_nonblock, write\_nonblock but not BasicSocket#send\_nonblock. Is this a mistake?

I've sometimes wondered that too...

However we have sendmsg\_nonblock, is that insufficient?

So I also wonder why recv\_nonblock exists, since recvmsg\_nonblock is a superset of its functionality; and redundant methods waste memory...

It's too late to remove recv\_nonblock, of course.

##### #2 - 04/07/2017 12:03 AM - ioquatix (Samuel Williams)

Thanks for the quick reply and interesting information.

I think I'd like to see send\_nonblock, but I'm not sure if its implementation is different than sendmsg\_nonblock. If it's not different, an alias may be sufficient.

##### #3 - 04/07/2017 12:23 AM - ioquatix (Samuel Williams)

I've been playing around with sendmsg\_nonblock, but I can't see how to make a wrapper around it with the equivalent API to send. It seems like it should be possible, but it always requires ancillary data, can't provide nil?

##### #4 - 04/07/2017 12:36 AM - normalperson (Eric Wong)

[samuel@oriontransfer.org](mailto:samuel@oriontransfer.org) wrote:

I've been playing around with sendmsg\_nonblock, but I can't see how to make a wrapper around it with the equivalent API to send. It seems like it should be possible, but it always requires ancillary data, can't provide nil?

Huh? The following works for me:

```
require 'socket'
a, b = UNIXSocket.pair
b.sendmsg_nonblock("HI", exception: false)
p a.recv(2)
# prints "HI"
```

##### #5 - 04/07/2017 01:37 AM - ioquatix (Samuel Williams)

Okay, so I found the issue.

Firstly, sendmsg doesn't work for UDP sockets, I get EINVAL on Darwin OS.

Secondly, UDPSocket overrides #send but only in a specific case:

```
static VALUE
udp_send(int argc, VALUE *argv, VALUE sock)
{
    VALUE flags, host, port;
    struct udp_send_arg arg;
    VALUE ret;

    if (argc == 2 || argc == 3) {
        return rsock_bsock_send(argc, argv, sock);
    }
    rb_scan_args(argc, argv, "4", &arg.sarg.mesg, &flags, &host, &port);

    StringValue(arg.sarg.mesg);
    GetOpenFile(sock, arg.fptr);
    arg.sarg.fd = arg.fptr->fd;
    arg.sarg.flags = NUM2INT(flags);
    arg.res = rsock_addrinfo(host, port, rsock_fd_family(arg.fptr->fd), SOCK_DGRAM, 0);
    ret = rb_ensure(udp_send_internal, (VALUE)&arg,
                   rsock_freeaddrinfo, (VALUE)arg.res);
    if (!ret) rsock_sys_fail_host_port("sendto(2)", host, port);
    return ret;
}
```

If you call `udp_socket.send(data, flags, host, port)` it uses `sendto` which works, otherwise it uses `rsock_bsock_send` which fails with `EINVAL`.

If you already constructed a `sockaddr`, for example, trying `udp_socket.send(data, flags, sockaddr)` will fail. It's all a bit confusing. Additionally, for `UDPSocket`, there is no `send_nonblock` which is my real issue here, and using `sendmsg_nonblock` fails.

#### #6 - 04/07/2017 01:51 AM - normalperson (Eric Wong)

[samuel@oriontransfer.org](mailto:samuel@oriontransfer.org) wrote:

Firstly, sendmsg doesn't work for UDP sockets, I get EINVAL on Darwin OS.

Can you show us a sample code of what you're trying?

I wonder if it's something that works on Linux but not Darwin or if there's something else... I don't have too much experience with UDP, but maybe it's something I can still help with.

#### #7 - 04/07/2017 02:17 AM - ioquatix (Samuel Williams)

Okay.

Here is the working example.

```
#!/usr/bin/env ruby

require 'socket'

port = 6778

server = UDPSocket.new.tap{|socket| socket.bind("localhost", port)}
client = UDPSocket.new

data = "Matz is nice so we are nice."

t1 = Thread.new do
    packet, (_, remote_port, remote_host) = server.recvfrom(512)
    server.send(packet, 0, remote_host, remote_port)
end

t2 = Thread.new do
    client.send(data, 0, "localhost", port)

    response, _ = client.recvfrom(512)

    puts "Got response: #{response.inspect}"
end
```

```

[t1, t2].each(&:join)

puts "Finished."

Here is one that fails with EINTR:

#!/usr/bin/env ruby

Thread.abort_on_exception = true

require 'socket'

port = 6778

server = UDPSocket.new.tap{|socket| socket.bind("localhost", port)}
client = UDPSocket.new

data = "Matz is nice so we are nice."

t1 = Thread.new do
  puts "Server waiting for packet..."
  packet, (_, remote_port, remote_host) = server.recvfrom(512)
  server.send(packet, 0, remote_host, remote_port)
end

t2 = Thread.new do
  address = Addrinfo.udp("localhost", port)

  puts "Sending data to #{address.inspect}"
  # Should call ssize_t sendto(int sockfd, const void *buf, size_t len, int flags, const struct sockaddr *de
st_addr, socklen_t addrlen), but calls send which fails with EINTR.
  result = client.send(data, 0, address.to_sockaddr)

  response, _ = client.recvfrom(512)

  puts "Got response: #{response.inspect}"
end

[t1, t2].each(&:join)

puts "Finished."

```

#### #8 - 04/07/2017 02:44 AM - nobu (Nobuyoshi Nakada)

Please file a new issue.

#### #9 - 04/07/2017 05:49 AM - ioquatix (Samuel Williams)

[nobu \(Nobuyoshi Nakada\)](#) here is the issue <https://bugs.ruby-lang.org/issues/13409> for the specific problem mentioned above.

But, if possible, I'd like send\_nonblock too :)

#### #10 - 06/16/2017 08:56 AM - akr (Akira Tanaka)

I cannot remember why I didn't add send\_nonblock.

However I feel adding send\_nonblock is considerable because sendmsg is much more complex than send/sendto.

#### #11 - 08/31/2017 05:31 AM - hsbt (Hiroshi SHIBATA)

- Assignee set to akr (Akira Tanaka)

- Status changed from Open to Assigned

#### #12 - 05/11/2018 02:09 AM - ioquatix (Samuel Williams)

I have since worked around this issue, by using sendmsg as an alias.

<https://github.com/socketry/async-io/blob/dbcfb05d209a41d2283e49c75c407a93179a64b6/lib/async/io/socket.rb#L47>

#### #13 - 08/08/2018 03:10 AM - ioquatix (Samuel Williams)

We can close this issue, as there exists a satisfactory alternative.