Ruby master - Bug #15468
Net::Protocol::BufferedIO#write raises NoMethodError when sending large multi-byte string
12/27/2018 04:45 AM - eitoball (Eito Katagiri)

Status: Closed
Priority: Normal
Assignee:
Target version:
ruby -v: ruby 2.6.0p0 (2018-12-25 revision 66547) [x86_64-linux]
Backport: 2.4: DONTNEED, 2.5: DONTNEED, 2.6: DONE

Description
ruby-2.6.0で大きなJSON文字列をNet::HTTPでPOSTすると以下のようにNoMethodErrorが発生するようになっ

$ ruby -rnet/http -rjson -v -e "Net::HTTP.post(URI('http://httpbin.org/post'), { text: 'あ'*100_000 }.to_json, 'Content-Type' => 'application/json')"

ruby 2.6.0p0 (2018-12-25 revision 66547) [x86_64-linux]
Traceback (most recent call last):
 19: from -e1:in `<main>'
 18: from lib/ruby/2.6.0/net/http.rb:500:in `post'
 17: from lib/ruby/2.6.0/net/http.rb:605:in `start'
 16: from lib/ruby/2.6.0/net/http.rb:920:in `start'
 15: from lib/ruby/2.6.0/net/http.rb:502:in `block in post'
 14: from lib/ruby/2.6.0/net/http.rb:1281:in `post'
 13: from lib/ruby/2.6.0/net/http.rb:1493:in `send_entity'
 12: from lib/ruby/2.6.0/net/http.rb:1479:in `request'
 11: from lib/ruby/2.6.0/net/http.rb:1506:in `transport_request'
 10: from lib/ruby/2.6.0/net/http.rb:1506:in `catch'
 9: from lib/ruby/2.6.0/net/http.rb:1507:in `block in transport_request'
 8: from lib/ruby/2.6.0/net/http/generic_request.rb:123:in `exec'
 7: from lib/ruby/2.6.0/net/http/generic_request.rb:189:in `send_request_with_body'
 6: from lib/ruby/2.6.0/net/protocol.rb:247:in `write'
 5: from lib/ruby/2.6.0/net/protocol.rb:265:in `writing'
 4: from lib/ruby/2.6.0/net/protocol.rb:248:in `block in write'
 3: from lib/ruby/2.6.0/net/protocol.rb:275:in `write0'
 2: from lib/ruby/2.6.0/net/protocol.rb:275:in `each_with_index'
 1: from lib/ruby/2.6.0/net/protocol.rb:275:in `each'
lib/ruby/2.6.0/net/protocol.rb:280:in `block in write0': undefined method `bytesize' for nil:NilClass (NoMethodError)

https://github.com/ruby/ruby/pull/2058

Associated revisions
Revision 66582 - 12/27/2018 04:49 AM - nobu (Nobuyoshi Nakada)
Fix Net::Protocol::BufferedIO#write when sending large multi-byte string

This commit should fix Net::Protocol::BufferedIO#write when sending large multi-byte string like following example.

$ ruby -rnet/http -rjson -v -e "Net::HTTP.post(URI('http://httpbin.org/post'), { text: 'あ'*100_000 }.to_json, 'Content-Type' => 'application/json')"
	ruby 2.6.0p0 (2018-12-25 revision 66547) [x86_64-linux]
Traceback (most recent call last):
 19: from -e1:in `<main>'
 18: from lib/ruby/2.6.0/net/http.rb:500:in `post'
 17: from lib/ruby/2.6.0/net/http.rb:605:in `start'
 16: from lib/ruby/2.6.0/net/http.rb:920:in `start'
 15: from lib/ruby/2.6.0/net/http.rb:502:in `block in post'
 14: from lib/ruby/2.6.0/net/http.rb:1281:in `post'
 13: from lib/ruby/2.6.0/net/http.rb:1493:in `send_entity'
 12: from lib/ruby/2.6.0/net/http.rb:1479:in `request'
 11: from lib/ruby/2.6.0/net/http.rb:1506:in `transport_request'
 10: from lib/ruby/2.6.0/net/http.rb:1506:in `catch'
Fix GH-2058

From: Eito Katagiri eitoball@gmail.com

Revision 9548d04 - 01/12/2019 09:02 PM - naruse (Yui NARUSE)
merge revision(s) 66582: [Backport #15468][Backport #15472]

Fix Net::Protocol::BufferedIO#write when sending large multi-byte string

This commit should fix Net::Protocol::BufferedIO#write when sending large multi-byte string like following example.

```
$ ruby -rnet/http -rjson -v -e "Net::HTTP.post(URI('http://httpbin.org/post'), { text: '?'*100_000 }.to_json, 'Content-Type' => 'application/json')"
```

ruby 2.6.0p0 (2018-12-25 revision 66547) [x86_64-linux]

Traceback (most recent call last):
  19: from -e:1:in `<main>'
  18: from lib/ruby/2.6.0/net/http.rb:500:in `post'
  17: from lib/ruby/2.6.0/net/http.rb:605:in `start'
  16: from lib/ruby/2.6.0/net/http.rb:920:in `start'
  15: from lib/ruby/2.6.0/net/http.rb:502:in `block in post'
  14: from lib/ruby/2.6.0/net/http.rb:1281:in `post'
  13: from lib/ruby/2.6.0/net/http.rb:1493:in `send_entity'
  12: from lib/ruby/2.6.0/net/http.rb:1479:in `request'
  11: from lib/ruby/2.6.0/net/http.rb:1506:in `transport_request'
  10: from lib/ruby/2.6.0/net/http.rb:1506:in `catch'
  9: from lib/ruby/2.6.0/net/http.rb:1507:in `block in transport_request'
  8: from lib/ruby/2.6.0/net/http/generic_request.rb:189:in `send_request_with_body'
  7: from lib/ruby/2.6.0/net/http/generic_request.rb:247:in `write'
  6: from lib/ruby/2.6.0/net/protocol.rb:275:in `write0'
  5: from lib/ruby/2.6.0/protocol.rb:265:in `writing'
  4: from lib/ruby/2.6.0/protocol.rb:248:in `block in write'
  3: from lib/ruby/2.6.0/protocol.rb:275:in `write0'
  2: from lib/ruby/2.6.0/protocol.rb:275:in `each_with_index'
  1: from lib/ruby/2.6.0/protocol.rb:280:in `block in write0': undefined method `bytesize' for nil:NilClass (NoMethodError)
```

[Fix GH-2058]

From: Eito Katagiri eitoball@gmail.com

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/branches/ruby_2_6@66799 b2ddd03e8-39d4-4d8f-98ff-823fe69b080e

Revision 66799 - 01/12/2019 09:02 PM - naruse (Yui NARUSE)
merge revision(s) 66582: [Backport #15468][Backport #15472]

Fix Net::Protocol::BufferedIO#write when sending large multi-byte string

This commit should fix Net::Protocol::BufferedIO#write when sending large multi-byte string like following example.

```
$ ruby -rnet/http -rjson -v -e "Net::HTTP.post(URI('http://httpbin.org/post'), { text: '?'*100_000 }.to_json, 'Content-Type' => 'application/json')"
```

ruby 2.6.0p0 (2018-12-25 revision 66547) [x86_64-linux]

Traceback (most recent call last):
  19: from -e:1:in `<main>'
  18: from lib/ruby/2.6.0/net/http.rb:500:in `post'
  17: from lib/ruby/2.6.0/net/http.rb:605:in `start'
  16: from lib/ruby/2.6.0/net/http.rb:920:in `start'
  15: from lib/ruby/2.6.0/net/http.rb:502:in `block in post'
  14: from lib/ruby/2.6.0/net/http.rb:1281:in `post'
  13: from lib/ruby/2.6.0/net/http.rb:1493:in `send_entity'
  12: from lib/ruby/2.6.0/net/http.rb:1479:in `request'
  11: from lib/ruby/2.6.0/net/http.rb:1506:in `transport_request'
  10: from lib/ruby/2.6.0/net/http.rb:1506:in `catch'
  9: from lib/ruby/2.6.0/net/http.rb:1507:in `block in transport_request'
  8: from lib/ruby/2.6.0/net/http/generic_request.rb:189:in `send_request_with_body'
  7: from lib/ruby/2.6.0/net/http/generic_request.rb:247:in `write'
  6: from lib/ruby/2.6.0/net/protocol.rb:275:in `write0'
  5: from lib/ruby/2.6.0/protocol.rb:265:in `writing'
  4: from lib/ruby/2.6.0/protocol.rb:248:in `block in write'
  3: from lib/ruby/2.6.0/protocol.rb:275:in `write0'
  2: from lib/ruby/2.6.0/protocol.rb:275:in `each_with_index'
  1: from lib/ruby/2.6.0/protocol.rb:280:in `block in write0': undefined method `bytesize' for nil:NilClass (NoMethodError)
```

[Fix GH-2058]
Can we get this issue closed and the with right backport field values?

#2 - 01/11/2019 02:32 AM - nobu (Nobuyoshi Nakada)
- Backport changed from 2.4: UNKNOWN, 2.5: UNKNOWN, 2.6: UNKNOWN to 2.4: UNKNOWN, 2.5: UNKNOWN, 2.6: REQUIRED
- Status changed from Open to Closed

#3 - 01/11/2019 06:34 AM - hsbt (Hiroshi SHIBATA)
- Backport changed from 2.4: UNKNOWN, 2.5: UNKNOWN, 2.6: REQUIRED to 2.4: DONTNEED, 2.5: DONTNEED, 2.6: REQUIRED

#4 - 01/12/2019 09:03 PM - naruse (Yui NARUSE)
- Backport changed from 2.4: DONTNEED, 2.5: DONTNEED, 2.6: REQUIRED to 2.4: DONTNEED, 2.5: DONTNEED, 2.6: DONE

ruby_2_6 r66799 merged revision(s) 66582.

#5 - 01/23/2019 09:41 PM - ioquatix (Samuel Williams)
Can we please fast track the release of this? It's breaking a ton of stuff.

Stepping back a little bit from this issue, I think we see a picture of Ruby where:

- Handling binary data can cause bugs like this (mismatch between binary and encoded string).
- Reading and writing strings is tricky, e.g. [https://bugs.ruby-lang.org/issues/13626](https://bugs.ruby-lang.org/issues/13626)
- There are some odd edge cases which can cause problems, e.g. [https://bugs.ruby-lang.org/issues/14975](https://bugs.ruby-lang.org/issues/14975)
- Reading and writing can put pressure on GC even though no additional strings should be allocated, see [https://bugs.ruby-lang.org/issues/15560](https://bugs.ruby-lang.org/issues/15560) for ideas to fix this.

I feel like some improvements to handling binary data could be a very useful addition to Ruby.

I welcome further discussion here and on the above issues so that we can avoid issues like this in the future. Ideally, we provide APIs such that this kind of bug is impossible.