Hi,

This is not really a bug but more of a missing feature. Let me layout steps to reproduce what I am trying to achieve:

- Add local entries for example.com

```bash
$ echo '::1 example.com www.example.com
127.0.0.1 example.com www.example.com' | sudo tee --append /etc/hosts
```

- Add rule to DROP ipv6 packets: `sudo ip6tables -P INPUT DROP`
- Start a server: `ruby -rwebrick -e 'WEBrick::HTTPServer.new(Port: 8000).start'`
- Making request using Net::HTTP with timeout (Net::OpenTimeout) after after 60 seconds: `ruby -rnet/http -e "Net::HTTP.get(URI('http://example.com:8000'))"`

As far as I understand, `init_inetsock_internal` needs to fallback to ipv4 for Net::HTTP.get to work. IPv6 route being broke is expected and hence, RFC8305 recommends fallback to IPv4. Support for IPv6 fallback is spotty across languages and http clients, for example, golang supports it but python doesn't, making request with curl works but wget hangs.

Any suggestion on fixing this would very helpful. I meant "Net::HTTP will timeout (Net::OpenTimeout) after after 60 seconds". So far I have tried using Socket.tcp(addr.ip_address, serv).close to preemptively check if resolved addresses are reachable. It works but it is very inefficient. Besides the brute steps explained above, in the wild, this only affects host which have an IPv6 address but their IPv6 route to our CDN is broken. Personal home networks which generally have only IPv4 connectivity seamlessly use the IPv4 address to start the connection (without the need for a fallback).