DNS name resolution takes twice as long as it should when primary name server is unavailable

On Linux /etc/resolv.conf can be configured to set the timeout of a nameserver before trying the next one (defaults to 5 seconds). So in the event of a failed name server we expect the time taken to resolve a hostname will be timeout + query time. However testing shows it takes (2 * timeout) + query time. Example resolv.conf to replicate the problem of a failed name server:

```
nameserver 1.1.1.1
nameserver 8.8.8.8
nameserver 8.8.4.4
```

Example ruby code to test:

```ruby
require 'net/protocol'
require 'uri'
require 'benchmark'
uri = URI.parse("http://google.com")
Benchmark.measure { TCPSocket.new(uri.host, uri.port) }.real
```

Example benchmark with a above resolv.conf:

```
irb(main):005:0> Benchmark.measure { TCPSocket.new(uri.host, uri.port) }.real
=> 10.593100978
```

Example benchmark with timeout set to 3 in resolv.conf (options timeout:3):

```
irb(main):005:0> Benchmark.measure { TCPSocket.new(uri.host, uri.port) }.real
=> 6.262138267
```

and without the invalid nameserver:

```
irb(main):005:0> Benchmark.measure { TCPSocket.new(uri.host, uri.port) }.real
=> 0.243385728
```

Example C code that proves the resolver is behaving as expected:

```c
#include <stdio.h>
#include <netdb.h>

int main(int argc, char *argv[])
{
    struct hostent *hstnm;
    if (argc != 2) {
        fprintf(stderr, "usage: %s hostname\n", argv[0]);
        return 1;
    }
    hstnm = gethostbyname(argv[1]);
    if (!hstnm)
        return 1;
    printf ("Name: %s\n", hstnm->h_name);
    return 0;
}
```

Benchmark of C code with failed nameserver:
$ time ./gethostbyname google.com
Name: google.com
real    0m5.189s
user    0m0.000s
sys 0m0.000s

Benchmark of C code without failed nameserver:

$ time ./gethostbyname google.com
Name: google.com
real    0m0.040s
user    0m0.000s
sys 0m0.000s

History

#1 - 08/14/2015 04:54 AM - nemski (Patrick Robinson)

- Subject changed from DNS name resolution takes a twice as long as it should when primary name server is unavailable to DNS name resolution takes twice as long as it should when primary name server is unavailable
- Description updated

#2 - 08/15/2015 10:54 AM - nemski (Patrick Robinson)

Doing an strace I can see the DNS lookup occurs twice (with and without a failing name server):

```
[pid 32129] socket(AF_INET, SOCK_DGRAM|SOCK_NONBLOCK, IPPROTO_IP) = 7
[pid 32129] connect(7, {sa_family=AF_INET, sin_port=htons(53), sin_addr=inet_addr("1.1.1.1")}, 16) = 0
[pid 32129] poll([fd=7, events=POLLOUT]), 1, 0) = 1 ([[fd=7, revents=POLLOUT]])
[pid 32129] sendto(7, "\251\254\1\0\0\1\0\0\0\0\0\0\0\6google\3com\0\0\34\0\1", 28, MSG_NOSIGNAL, NULL, 0) = 28
[pid 32129] poll([fd=7, events=POLLIN]), 1, 3000) = 0 (Timeout)
[pid 32129] socket(AF_INET, SOCK_DGRAM|SOCK_NONBLOCK, IPPROTO_IP) = 8
[pid 32129] connect(8, {sa_family=AF_INET, sin_port=htons(53), sin_addr=inet_addr("8.8.8.8")}, 16) = 0
[pid 32129] poll([fd=8, events=POLLOUT]), 1, 0) = 1 ([[fd=8, revents=POLLOUT]])
[pid 32129] sendto(8, "\271\254\201\200\0\1\0\0\0\0\0\0\0\6google\3com\0\0\34\0\1", 28, MSG_NOSIGNAL, NULL, 0) = 28
[pid 32129] poll([fd=8, events=POLLIN]), 1, 2000) = 1 ([[fd=8, revents=POLLIN]])
[pid 32129] ioctl(8, FIONREAD, [56]) = 0
[pid 32129] recvfrom(8, "\271\254\201\200\0\1\0\0\0\0\0\0\0\6google\3com\0\0\34\0\1...", 1024, 0, {sa_family=AF_INET, sin_port=htons(53), sin_addr=inet_addr("8.8.8.8")}, [16]) = 56
[pid 32129] close(7) = 0
[pid 32129] close(8) = 0
[pid 32129] stat("/etc/resolv.conf", {st_mode=S_IFREG|0644, st_size=226, ...}) = 0
[pid 32129] socket(AF_INET, SOCK_DGRAM|SOCK_NONBLOCK, IPPROTO_IP) = 7
[pid 32129] connect(7, {sa_family=AF_INET, sin_port=htons(53), sin_addr=inet_addr("1.1.1.1")}, 16) = 0
[pid 32129] poll([fd=7, events=POLLOUT]), 1, 0) = 1 ([[fd=7, revents=POLLOUT]])
[pid 32129] sendto(7, "\366\1\0\0\1\0\0\0\0\0\0\0\6google\3com\0\0\34\0\1", 28, MSG_NOSIGNAL, NULL, 0) = 28
[pid 32129] poll([fd=7, events=POLLIN]), 1, 3000) = 0 (Timeout)
[pid 32129] socket(AF_INET, SOCK_DGRAM|SOCK_NONBLOCK, IPPROTO_IP) = 8
[pid 32129] connect(8, {sa_family=AF_INET, sin_port=htons(53), sin_addr=inet_addr("8.8.8.8")}, 16) = 0
[pid 32129] poll([fd=8, events=POLLOUT]), 1, 0) = 1 ([[fd=8, revents=POLLOUT]])
[pid 32129] sendto(8, "\366\1\0\0\1\0\0\0\0\0\0\0\6google\3com\0\0\34\0\1...", 1024, 0, {sa_family=AF_INET, sin_port=htons(53), sin_addr=inet_addr("8.8.8.8")}, [16]) = 220
[pid 32129] close(7) = 0
[pid 32129] close(8) = 0
```

#3 - 08/15/2015 12:04 PM - nemski (Patrick Robinson)

- Ruby -v changed from ruby 2.2.2p95 (2015-04-13 revision 50295) [x86_64-linux] to ruby 2.3.0dev (2015-08-13) [x86_64-linux]

#4 - 02/04/2016 10:54 PM - nemski (Patrick Robinson)

I retested this again today and could only replicate it on Ubuntu Lucid, not Ubuntu Trusty or Debian Jessie.

#5 - 08/13/2019 04:03 AM - jeremyevans0 (Jeremy Evans)

- Status changed from Open to Closed