Improve IPv6 support

Jaskaran Veer Singh
(jvsg)

IRC nick: jvsg
E-mail : jvsg1303 *at* gmail *dot* com
PGP Key ID : 9E1A6AD8
Fingerprint : 2814 3FB7 A32D 429B 092E 27F0 8AA3 C532 9E1A 6AD8
Timezone: UTC+5:30

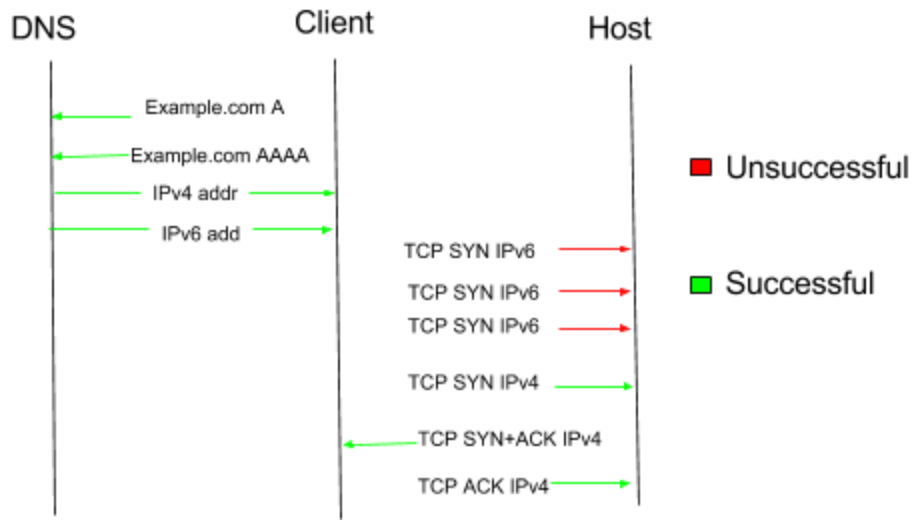
Abstract

Tor works over IPv6, but require some manual configuration. Clients and relays can have the ability to automatically detect IPv6 availability, and configure themselves. Additionally even if we automate this, we would experience a significant connection delay. To overcome this we would require implementing Happy-Balls algorithm that reduces this delay in dual stack clients.

Research[1]

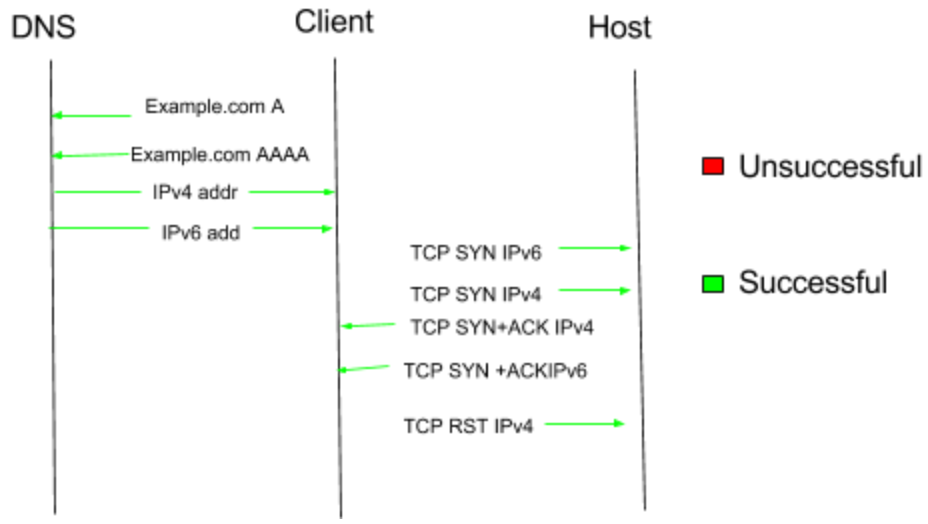
The main part of the solution is to implement a “Happy Eyeballs” Algorithm.

The following diagram shows the vanilla connection handshake situation with just a fallback onto IPv4 connection incase the v6 doesn’t work. The Client gets both the v4 and v6 address from the DNS and tries to make a v6 connection, waits for it, if connection can’t be made, then falls back on IPv4 and proceeds. The problem with this is that it’s slow.



Existing connection handshake

[1] https://tools.ietf.org/html/rfc6555



Happy Eyeballs algorithm

The client sends two TCP SYNs at the same time over IPv6 (6) and IPv4 (7). In the diagram, the IPv6 path is broken but has little impact to the user because there is no long delay before using IPv4. The IPv6 path is retried until the application gives up (10).

After performing the above procedure, the client learns whether connections to the host's IPv6 or IPv4 address were successful. The client MUST cache information regarding the outcome of each connection attempt, and it uses that information to avoid thrashing the network with subsequent attempts. In the example above, the cache indicates that the IPv6 connection attempt failed, and therefore the system will prefer IPv4 instead. Cache entries should be flushed when their age exceeds a system-defined maximum on the order of 10 minutes.

The algorithm works in the following way:

1. Call getaddrinfo(), which returns a list of IP addresses sorted by the host's address preference policy.
2. Initiate a connection attempt with the first address in that list (e.g., IPv6).
3. If that connection does not complete within a short period of time (Firefox and Chrome use 300 ms), initiate a connection attempt with the first address belonging to the other address family (e.g., IPv4).
4. The first connection that is established is used. The other connection is discarded.

5. After the connection is made. The information about which address worked is Cached. This is to prevent thrashing the network with subsequent attempts. The Cache would indicate

Advantages

1. Provides fast connection for users, by quickly attempting to connect using IPv6 and (if that connection attempt is not quickly successful) to connect using IPv4.
2. Avoids thrashing the network, by not always making simultaneous connection attempts on both IPv6 and IPv4.

Implementation

The following tickets need to be fixed in order to completely add support for IPv6.

#4806 Detect and warn when running IPv6-using client without IPv6 address privacy

Last 48 bits of IPv6 address has the potential to reveal a device's MAC address. This is very dangerous to the privacy of a user. We cannot mask the MAC addresses ourselves (because we aren't root). So we'll have to check whether the outgoing MAC address matches to any of the MAC address of the clients, if so, then output a warning and a way to mask the real MAC address. For example, the following solution could be used by the user to mask the address:

```
sysctl net.ipv6.conf.all.use_tempaddr=2
sysctl net.ipv6.conf.default.use_tempaddr=2
```

#11360 Listen on IPv6 by default for SocksPort

- We have an Ipv4 listener on 127.0.0.1:9050.
- Now, we need to open another *Port listener on the same port on [::1]: As long as there is no conflicting listener explicitly configured on [::1] or [::]

Things to keep in mind:

- Listener shutdown should close both listeners.
- Check those areas in code that use port_cfg_t. They shouldn't rely on localhost being 127.0.0.1, or there being exactly one listener per port_cfg_t.

#17217 Change clients to automatically use IPv6 if they can bootstrap over it

Currently Tor avoids using IPv6 for OR and Directory connections. We could introduce a way to always check for IPv6 availability first and then fallback to IPv4 if that fails. But this means the user would experience delay in connection. So to prevent this we would use “Happy-Eyeballs” algorithm, that is efficient in testing for IPv6 availability and falling back on IPv4 if that isn’t present.

#17782 Relays may publish descriptors with incorrect IP address

This bug affects those instances which:

- Don’t have an address in their torrc
- Provide a hostname that doesn’t resolve, or doesn’t resolve into public IP address
- Are behind NAT

A fix for this would be to check whether we can connect to our own ORPort and DirPort.

#17845 Add unit tests for IPv6 relay descriptors

Once chutney gets the ability to use IPv6 addresses, we could update them by adding desc that use IPv6 addresses.

#6939 Missing IPv6 ORPort reachability check

Currently, Tor performs and logs a check to determine whether IPv4 ports are reachable but does not do the same for IPv6. To implement this ticket, we need Tor to be able to connect to its own IPv6 ORPort, using at a 3-hop path.

#4847 Bridges binding only to an IPv6 address doesn't work

If a relay has only IPv6 OR port, it doesn’t have a descriptor for itself, doesn't consider itself publishable and won't upload its descriptor.For this the two functions need to be fixed:

- router_rebuild_descriptor() // decides whether to rebuild a descriptor
- decide_if_publishable_server() //decides if desc needs to be published

#5940 Figure out own IPv6 address

We need a way for relays to figure out their own IPv6 addresses. get_interface_address6() could be used for this purpose. It maybe possible that the relay operator might not want to expose IPv6 address to the public. Hence, we need an option “IPv6Relay” that the relay operator could set before we detect the address.

#17011 Teach chutney to verify over IPv6

Things that need to be fixed:

Chutney doesn't give clients an IPv6 address to connect to when verifying even when chutney is using IPv6 exits. So in a way exit nodes and bridges are listening on IPv6 but clients are only provided IPv4 to connect to.

Timeline

I. May 4 - May 30 (Community Bonding Period)

- Attend Network team meetings to get in pace with the development.
- Start with fixing #4806 - Detect and warn when running IPv6-using client without IPv6 address privacy.
- This tickets needs_revision, So I'll be able to work on this besides preparing for my exams.

II. May 30 - June 9 (Possibility of exams during this time)

I'm not sure as the exams date haven't been out till now. But if there are exams during this time, I'll try to begin work during Community bonding period itself.

III. June 10 - June 17 (Fix #11360)

- #11360 - Listen on IPv6 by default for SocksPort
- Add another listener on the same port to [::1] or [::]
- Check for those areas which rely on only having one listener on port 9050

IV. June 18 - June 23 (#17217)

- #17217 Change clients to automatically use IPv6 if they can bootstrap over it.

V. June 24 - June 30 (#17217)

- Continue with previous week's work.
- Would need to implement Happy EyeBalls algorithm.

---------- Phase-1 Evaluation ----------

VI. July 1 - July 7 (#17782)

- #17782 Relays may publish descriptors with incorrect IP address.
- Check whether we can connect to our own ORPort and DirPort.

VII. July 7 - July 14 (#17845)

- #17845 Add unit tests for IPv6 relay descriptors.

VIII. July 15 - July 21 (#6939)

- #6939 Missing IPv6 ORPort reachability check.
- Connect own to own port through guard relay, relay, and exit node.

IX. July 22 - July 28 (#4847)

- #4847 Bridges binding only to an IPv6 address doesn't work
- Fix router_rebuild_descriptor() and decide_if_publishable_server()

---------- Phase-2 Evaluation ----------

X. July 29 - August 04 (#5940)

- #5940 Figure out own IPv6 address
- Add a new configuration option, something on the lines of IPv6Relay.
- Use get_interface_address6() to get the v6 address.

XI. August 5 - August 11 (#17011)

- #17011 Teach chutney to verify over IPv6
- Would need to implement a method for clients to get IPv6 address and connect over it.

XII. August 12 - August 18 (Buffer Period & Documentation)

Would use this time as a buffer period in case anything goes wrong. Would Blog and improve documentation/man-pages corresponding to any changes made.

---------- Pencils Down ----------

XIII. After GSoC ends -

Fix bugs that are reported. Stay involved with the community to improve Tor in other areas.

About Me / Why I'm fit for this project

My friends call me "annoying privacy evangelist" as I keep poking them with the mistakes they regularly do online that compromises their privacy. I have been a privacy/anonymity advocate and a supporter of FOSS for long.

I was selected and completed a project in Libreoffice during GSoC 2016 and my work involved C++ and a HUGE codebase. So I guess now I'm fairly trained in handling large codebases. I have been associated with Libreoffice even after completing my GSoC, and most probably soon I will be a member of the TDF (Libreoffice's parent organisation).

I am associated with the Tor project since some time. I used to lurk around at the dev channel at OFTC and tried to learn by the discussions that used to happen there.

Point us to a code sample:

My work in Libreoffice (around 60 commits, mostly C++) : https://goo.gl/eIvldP

My work in Orcus (around 50 commits, mostly C++) : https://goo.gl/k1jDHR

I had realized there aren't many File Encryption and storage programs available for Linux unlike Mac and Windows. So I've created this : https://github.com/jvsg/mitron . It is a work under progress.

Why do I want to work with The Tor Project in particular?

I come from a country where Right to Privacy isn't considered a fundamental right. So every citizen is responsible for his/her privacy. I believe that freedom of speech can only free fully exercised when a person is anonymous.

Quite a large percentage of people do not care about their anonymity/privacy online and more efforts should be taken to reach out to them. Few of the establishments and organisations have interest in breaking the anonymity system that Tor provides. Hence it becomes even more important to constantly seek ways of improving Tor.

Not only philosophically and politically, but I'm also fascinated by Tor from a technological point of view. I have been using Tor since a long time now and I believe it's my turn now to give back to the community.

Will I be working full-time on the project for the summer?

Yes, but most probably during the first week of June I would have my exams. So apart from just one week, I would be available full time during the rest of the summer.

My Academic Pursuits

I am a undergrad student at a University in New Delhi, India. My major is Electronics and Communication Engineering. Sometime in the future, I would like to pursue research in the area of Anonymity systems.

Is there anything else that you should know that will make you like my project more?

I think many applications such as Google Chrome and Firefox make use of Happy Eyeballs algorithm for connections. So, it time that Tor also does.

Am I applying to other projects for GSoC and, if so, what would be my preference if you're accepted to both? Having a stated preference helps with the deduplication process and will not impact if you accept my application or not.

I'm not applying to the other organisation, but I'm applying to another project(Count-Unique IPs) under Tor. That will be my priority.