



**GSoC Final Report on :**  
***“eRPC - An Efficient Relay Partition Checker”***

## **Project WiKi :**

<https://gitlab.torproject.org/rishadbaniya/erpc/-/wikis/home>

## **Project Repository :**

<https://gitlab.torproject.org/rishadbaniya/erpc>

## **Proposed Plan :**

The proposed plan was to build a relay partition checking tool that can check for partitions among relays in the Tor Network.

## **Current State :**

As of right now, the relay partition checking tool is divided into two parts, one being primary worker and other being secondary worker, primary worker handles everything from checking for partition among relays(attempts to create two hop circuit among relays and logs the output), handling influx of new relays, has OnionPerf integrated to reduce the no of circuits being created, has functionality to resume the previous state from sqlite database. The secondary worker, it can be deployed anywhere and it will connect to the primary worker for work being assigned(two hop circuits to build).

The primary worker produces output on the database “sqlite or neo4j”, and internal graph data structure called “Petgraph” which can be then used to visualise partition among relays by segregating the errors manually(as of now).

There are certain goals i missed and plan to work in the upcoming days, such as

- A more concrete way for filtering relays to be scanned rather than just creating two hop circuits from higher consensus weight to lower as mentioned in the [Wiki of the project](#)
- Segregating errors, there were different errors being produced and I failed to list out explanations for every single possible error that were occurring there.
- Implement some graph algorithms for visualising partition among relays.
- Possible deployment of the tool

## **Future Plans :**

The adaptation to the environment that I've gotten within the Tor Community in the last few weeks has made me want to contribute more to this community in any form. I've fixed my mind on improving "erpc", adding more features into it, contributing to "arti" related crates and getting myself attached to this community more in the coming days. As myself being an undergraduate computer science student, this is the best way for me to hone my skills too i.e by getting suggestions from senior developers, communicating properly with senior developers on specific tickets/issues, and developing the habit of asking "proper questions".

Some of the tickets/issues that I will be tackling starting from this week(related to the project and personal level) will be..

- (On Personal Level) Be more adapted to the workflow of the tor community i.e raising tickets for different works and working through them, documenting more verbosely, posting reports regarding the project on the mailing list, working on different branches more frequently in contrast to how i currently do i.e work on the main branch directly.

Related to the “eRPC” project..

- Update and fix the schema, for saving the output and updating the resume functionality within the application.
- Explore different tor log files and Onionperf analysis files in depth.
- Having a strong filtering mechanism to scan for partition among specific relays.
- Provide different algorithms for visualising partition through the graph databases.
- Explore all possible errors that can cause partitions and interpret how the errors logged by “arti” match to them.

If you have any questions/suggestions or tickets/issues to raise, please feel free to raise them at

<https://gitlab.torproject.org/rishadbaniya/erpc>. I would be more than happy to recognize the voids that are in the project and things I have to do to fill them and make this project more concrete and useful to the community.

Lastly, I would like to thank the entire Tor community for helping me grow as a person and a programmer, giving me the opportunity to contribute to this amazing community and reflect on my work.