



University Blockchain
Research Initiative



UBRI Connect



UNIVERSITY *of* NICOSIA

Virtual Event

October 11–13, 2020

ubri.ripple.com

The Team



Klitos Christodoulou, PhD

IFF Research Manager, DLRC Lab Leader

christodoulou.kl@unic.ac.cy



Elias Iosif, PhD

IFF Project Manager, DLRC Lab Co-Leader

iosif.e@unic.ac.cy



Antonios Inglezakis

Senior Researcher

touloupos.m@unic.ac.cy



Marios Touloupou

Researcher/ PhD Student

touloupos.m@unic.ac.cy



<https://github.com/unic-iff>



UNIC | Distributed Ledgers
Research Centre

Benchmarking the Behaviour of Blockchain Protocols

Explore: Launch Benchmarking Framework with *rippled* daemon

- Clone our repository (*git clone https://github.com/UNIC-IFF/ripple-docker-testnet.git*)
- Move into the cloned repository '*cd ripple-docker-testnet*'
- Check for running containers in the background '*docker ps -a*'
- Check for the docker image '*docker images*'
 - Pull the rippled image from our docker hub '*docker pull uniciff/rippled-runner:v1.5.1*'
- Deploy **testnet** with **n** number of nodes/validators '*./run_testnet 5*'
 - During this process, keys will be generated for our validators
 - Config Files will also be generated for the validators
 - Each node will connect with each other (a full mesh network)
 - Monitoring network is deployed (using Prometheus, Grafana, Push Gateway, Alert Manager, Influxdb)
- Run '*docker ps -a*' in order to check if everything is up and running with no errors in the 'status' column

Benchmarking the Behaviour of Blockchain Protocols

Explore: Generate Accounts and spread XRP (Traffic Generator)

- While in the cloned repository, change dir to `/ripple_traffic_generator`
- Then, change dir to `/config`, a config file is located. That file contains 4 important parameters.
 - **host_domain**: provide the IP of where the rippled daemon is running. If you are on the same machine leave as is – **localhost**
 - **port**: The default port of rippled daemon for **WS** communication is 6006. If not sure, leave it as is.
 - **account_addr**: This is the genesis account address which will hold all generated XRP during genesis of the network. This is the account we will first use to spread XRP in the network. You can change this later, in case you would like to send XRP from another account.
 - **account_secret**: This is the secret key of the account address above. This is needed for signing the transactions etc. If you change the account address, you also need the secret key for that account as well.
- Generate Accounts and spread XRP: `./traffic_gen.sh 5 1000`
 - 5 is the number of how many new accounts to generate.
 - 1000 is the number of XRP to be transferred from the account we have set before in the config file.
 - At the start, the newly created account addresses will be displayed in the console, while later the transaction output will also be logged.
 - Complete logging could be found in directory `/output_data`

Benchmarking the Behaviour of Blockchain Protocols

Explore: Retrieve Account Info

- Use `'node acc_info.js <account_address>'` to find more details e.g., account balance of the account we provided.
- directory `/output_data` – to print all the account addresses we have generated in the previous step by typing `'cat accounts_to_pay.txt'`
- The balance of an already created account should be 1000 XRP (that's how much we have transferred in a previous step)

Benchmarking the Behaviour of Blockchain Protocols

Explore: Monitoring Module

- Type `'docker ps -a'` to make sure that the monitoring system is up and running.
- Going back to our repo under branch '3rd-article' there is a directory called exporters. This directory contains custom exporters for capturing monitoring data from the running validators. For now, we have an exporter which is running inside each validator. Every 5 secs, it is scheduled to push its data towards the **push gateway**.
- A push gateway is used by our architecture as a separate component to order to avoid the need of monitoring each validator independently. That way any number of validators could connect or disconnect to the network without affecting the monitoring system.
- We can visit the push gateway in its Web interface with a web browser at: <http://localhost:9091>
 - There, we will see several “**jobs**” each one of them containing different monitoring data.
 - **Job=validator1-5** are the jobs responsible for keeping the data of each validator running in our network.
 - **Job=tx1-5** are the jobs responsible for keeping the data of the transactions we have produced earlier.
 - We can also visit Prometheus Web interface at: <http://localhost:9090>
 - Prometheus uses a scrap method to get the data from the push gateway. Additionally, it provides features for visualization of the data such as time plots.

Benchmarking the Behaviour of Blockchain Protocols

Explore: Monitoring Module

- A time series database i.e., **Influxdb** is used to make data persistent as time series. Check whether our data is pushed in the database by connecting to the database using a client.
- When connected, a database named '**ripple**' should exist. In this db we keep our data in time series for further analysis.
- Go ahead and query the database requesting data for a certain time period of your choice. *Interesting?*

Benchmarking the Behaviour of Blockchain Protocols

Stop TestNet and Clear your Environment

- The ***rippled*** topology is stopped when executing `./stop_testnet.sh`
 - This will stop all running containers, however we preserve all generated keys and config files for future usage.
- Finally, check if all services are down successfully, execute `'docker ps -a'`.

Thank you for your attention!

Learn more @



<https://www.unic.ac.cy/iff/dlrc>



Medium

<https://medium.com/ripple-series>



<https://github.com/unic-iff>



Klitos Christodoulou (christodoulou.k@unic.ac.cy)

Elias Iosif (iosisf.e@unic.ac.cy)

Antonios Inglezakis (inglezakis.a@unic.ac.cy)

Marios Touloupou (touloupou.m@unic.ac.cy)