A Short Description of Nyle

Arnaud Pannatier

School of Basic Sciences

Decentralized and Distributed Systems lab

Master Thesis

September 2019

Responsible
Prof. Bryan Ford
EPFL / DEDIS

Supervisors
Cristina Basescu,
Kelong Cong
EPFL / DEDIS

1 Summary

1.1 What is Nyle?

Nyle is a new system for a cryptocurrency, that answer some classical problems of a blockchain system by using the idea of locality. Two main problems are addressed: WW3 scenarios and approval time for a transaction.

WW3 Scenarios In case of a WW3, we can expect to have at least a long-lasting partition that will split the system in two. This is a problem for classical cryptocurrencies, because for a block to be approved, the users are supposed to wait to have a global consensus. This consensus will not be reached with a long-lasting partition and therefore it will create problems for classical cryptocurrencies. Nyle solve this issue by design using locality.

Approval Time for a Transaction Another issue with waiting global consensus is that it usually takes a long time. If we want to use a cryptocurrency in a daily life, we want to solve that problem to be able to validate (at least partially) a transaction relatively fast. The solution provided by Nyle use locality again: with Nyle a transaction is validated at different levels, and it is up to the user to wait a local, or global validation for a transaction.

1.2 Locality : From CRUX to Nyle

In order to have the locality properties, Nyle uses a similar design than CRUX but applies it in the specific case of a cryptocurrency. It assumes the same Network model as in CRUX (set of nodes that are connected through an Internet-like network,...). It uses the landmark technique from approximate-distance oracles and creates ARAs, with the same strategies. So it will provide the same properties for the network (and bunches, clusters,...) and the ARAs.

So the ARA is the representation of the region. In each of these regions there will be a copy of the same system, in the case of Nyle the system is a blockchain. So each region will have its own blockchain and validate all the transactions between the nodes that are included in it. Some nodes can be included in different regions, and they will send their transactions to all the regions they are part of. Which ensure that each blockchain will be updated each time there is a transaction that concerns one of its nodes.

1.3 Stable environment vs Byzantine evloving system

The big difference between CRUX and NYLE is that the purpose of CRUX is to work in envirnments where machines are relatively "stable" which means that they are not supposed to churn or to crash often, and more, where the machines are not supposed to move. This is not the case for Nyle : if we

have a cryoptocurrency, we can expect to have malicious, deficient and/or moving users. This will add some difficulties for the protocol, that we will have to address by the mean of a "control plane".

1.4 Blockchain System

Each region will have its own blockchain, in Nyle the choice for the blockchain will be chosen between Omniledger or ByzCoin. But it can be generalized to any kind of blockchain.

1.5 Control Plane

Assume you initially have a map of all pairwise latencies and the initial ARAs, i.e., you know the membership of every ARA. As time passes, some nodes leave (or crash) and some nodes join. How would you keep the ARAs up to date?

1.5.1 Definition of the problem

1.6 What is already implemented for Nyle

1.6.1 Transaction validation

We already have a protocol that validates a transaction.

1.6.2 Block storage on node

As each node will participate in different regions (from very local to world-wide), it will need to store the blockchains for all of these region. We have a method that reduce the redundancy, by only storing the hash of a block instead of the full block at each level.

1.6.3 Proof-of-Location

We already have a protocol for controlling the distance from a new node to the rest of the nodes. And that assures no one cheats by giving false distances.

1.7 List of the challenges to build Nyle

- Based on the proof-of-location, build a CRUX-like network
- In each of the region of the regions build a Blockchain (see 1.4)
- Use the transaction validation to give info on the validated region (see 1.1 (Approval time for a transaction))
- Dealing with moving actors. (See 1.5)

- Dealing with double-spending issues (if a node spend the same coin in different regions) (see 1.1 WW3 Scenarios)
- (Investigate if this design is open to other errors)

1.8 Master Thesis: Partition and Region Takeover Recovery

The purpose of my project is to deal with the forth challenge and maybe fifth (controle plane, and double spending). At some point I assume that I would have to deal with challenges 1 to 3. And I expect that at first I will have to understand how the Cothority framework works in detail. Then I will have to design a theoretical protocol to fix the double-spending issue, then to implement it using the Cothority framework, then to test it on multiple benchmark and treating the cases with adversarial nodes. An in-depth analysis of the results will be done as well.

1.9 Implementation

The (only ?) implementation of Nyle, for now, uses the Cothority framework.

The Cothority framework is used to simulate a decentralized system. With it one can create a lot of independent nodes that communicate together in a internet-like network. Each node will have (at least ?) three "layers" : protocol, service and API corresponding to the OSI system (?). The service layer is used to describe some functionalities in a general manner. The protocol layer describe the steps to implement the services. And finally the API is the place to call the different services. The Cothority framework can generate a benchmark consisting of a big set of nodes, and run the different protocol, services, API from each node. To run a program based on the Cothority framework, one should write the protocol, service and API layer code and then writes some test file that will be executed or using CLI.