Design & Implementation Document

Reward system for subnet validators of custom blockchains/vm on Avalanche

version : initial draft (08 Sept’2021)

1. Introduction

This document describes the design and implementation done by our team, as part of the Hackathon event hosted by Gitcoin and sponsored by Ava-Labs titled : “Subnet and Virtual Machine”.

2. Challenge Description

Avalanche subnets allow anyone anywhere to spin up a taylor-made network w/ custom virtual machines and complex validator rulesets. They enable permissionless and permissioned networks to launch with optional privacy and regulatory compliance.

Virtual machines are the application-level logic of your blockchain. ...

3. Our understanding of the Challenge Description

Currently on Avalanche, the validators have the ability to add Nodes to the Primary subnet and are rewarded as per the Avalanche staking reward system. Validators can also add their nodes to other subnets, however we donot know of any mechanism of getting rewards if they validate additional subnets other than the Primary subnet (viz., P, X & C).

4. Our Proposed Design

We have based our work on the template provided by Avalanche - timetampvm and extended it for defining a reward system. These modification proposed can be reused for any other custom VM.

High level tasks

1. Identify the Node-ids that are part of a custom subnet
2. Get the required details of each Node-id like start time, end time, reward address, uptime, etc., through existing workflow
3. Track the events when Nodes are joined and leaving the subnet validation.
4. Calculate rewards accumulated for each Node and distribute as per policy defined.

Details are given below.

Identify the Node-ids : A time.ticker is implemented which will poll at regular intervals the membership of the custom subnet. The default time to poll is 1 hour, and cannot be changed in runtime.

Get required details of each Node-id : We use the inbuilt rpc api command and call the platform chain’s GetCurrentValidators for the custom subnet. The output for custom subnet does not have information like reward address, uptime, etc., so we have to make a call for the primary subnet also to get the missing details.

Track the events : We maintain a data structure with the membership information and also current status of each Node. During polling if a new Node-Id is seen, its added to this data structure. We also maintain the endTime info of each Node and keep track of the ones which should be removed from this data structure at each polling interval.

Calculate rewards : We can have multiple reward incentive schemes based on the subnet ecosystem requirements.

a. Fixed amount

- i) Distribute fixed number of Avalanche Native tokens in AVAX
- ii) Distribute fixed number of Custom Tokens

b. Variable amount

- i) Distribute variable number of AVAX Tokens based on staking amount, etc.,
- ii) Distribute variable number of Custom Tokens based on staking amount, etc.,

Another scheme is

- a. Reward distribution at regular intervals
- b. Reward distribution at the endTime of the subnet staking

In the current implementation for hackathon, we use very basic scheme of using Fixed number of Custom Tokens distributed at regular 'reward epoch' interval. Code for other schemes are also available for reference in the implementation which can be changed accordingly.

Custom Token : We create a custom Token via the Variable Cap Asset on X-Chain for rewarding validators for this specific subnet. We use this method to show the technical feasibility only, the economics of the Token generation etc., is not explored and not in scope here.

Addition of Node-ids to subnet : Subnets are a powerful primitive that allows the creation of permissioned blockchains. To add the validators (Node-id's), the subnet owner / administrator will facilitate to add the Node-Ids as and when required for the staking duration. After addition the reward scheme will take into effect.

Another option is to have a permissionless network using Subnets, where any validator can add their node into this subnet via custom VM's API.

5. Other Design Considerations

The above design is developed for the submission of the hackathon event. However there are other alternatives that can be thought of or developed based on various pros and cons. We are listing some of them.

Custom Token : For the reference implemenation, we use the custom tokens minted on the X-Chain to distribute rewards to validators. However this can also be achieved via ERC-20 Tokens on the C-chain. One advantage of using ERC-20 tokens is that they can be bridged across other vendor blockchains and swapped on exchanges.

6. User Workflow

We give below the steps to enable the workflow for a validator in getting rewards for adding into subnet of custom VM. This for the sake of completeness of this document. Please look for any updated instructions in the github link provided in the reference[c].

7. References

a. Hackathon event link : https://gitcoin.co/issue/ava-labs/open-defi-hackathon/3/100026354
b. Avalanche Developer Documentation : https://docs.avax.network/
c. Github link of our hackathon submission : https://github.com/shri4net/ava-labs-reward-system

8. Further Contacts

1. ksk12345@yahoo.com
2. shri4net@gmail.com