A theoretical analysis of the Keep Network random beacon using agent based modeling

Prashanth Irudayaraj
Keep Network
prashanth@keep.network

Antonio Cordozo
Keep Network
antonio@keep.network

Promethea Raschke
Keep Network
promethea@keep.network

Markus Fix
Keep Network
markus@keep.network

Liam Zebedee
Keep Network
liam@keep.network

***Abstract*—ABM is an effective tool for analysis of complex systems with long term emergent behavior. In this study we apply ABM to analyse the long term behavior of Keep's random beacon. We look for the emergence of steady state behavior, at which point we evaluate the sensitivity of specific group and signature characteristics to various parameters. The results of this study illustrates the effectiveness of ABM as an aid to the design of novel distributed systems.**

***Index Terms*—Token Engineering, Agent Based Model, Distributed Systems**

I. Introduction

II. Agent Based Modeling (ABM)

ABM has traditionally been a tool to simulate complex dynamic systems such as the spread of pathogens [1], social psychology [4], and financial markets [2]. It is well suited to systems that resist simple analytical solutions due to the interaction of complex individual agents with varying attributes. As a bottom up approach, ABM has been gaining in popularity over tradtional methods such as Discrete Event Simulation, and can provide more convincing theoretical analysis than approaches such as general equilibrium analysis [5].

The nascent field of token engineering is currently establishing its own set of tools and processes. As such, ABM lends itself well to analyzing these systems to support the design process and evaluate system state after launch.

III. Analysis of Token based systems

Discuss the use of ABM for token engineering. Cite recent papers.

IV. Overview of the Keep Random Beacon

The Keep Network random beacon uses a threshold relay simuliar to the one proposed by DFINITY and based on the BLS signature scheme [] (NEED CITATION)

A. Role of the model in the design process

The initial architecture of the Keep beacon was designed by (need input from Antonio) what purpose did the simulation serve?

Identify applicable funding agency here. If none, delete this.

V. Key research questions

A. When could steady state behavior emerge?

Emergence of steady state behavior occurs as a confluence of several factors and is difficult to predict at the design stage. Specifically, process that take several blocks to complete such as DKG that often occur asynchronously can create large swings in values such as number of active groups or percentage of compromised groups.

B. What are the effects of node failures on system behavior?

Nodes may go offline or fail entireley. Since their participation in groups is critical to the success of the threshold relay, it is important to understand the impact of various levels of failure on group characteristics. In particular, failures can result in disproportionate ownership of groups which may cause a byzantine fault to occur.

C. What are the effects of stake distributions on system behavior?

Stake distributions can also skew ownership of groups and can lead to a byzantine fault. A more centralized distribution could once again lead to a greater ownership of groups by a few entities thus creating conditions for the group to be compromised.

VI. Model Creation

A. Key Terms

- Node: A computational entity with memory and computational power sufficient to to run a keep client
- Node Owner: may own 1 or more nodes and can allocate different levels of stake to each node owned
- Stake: A bond that make a node eligible to participate in a group
- Group: A collection of nodes who have successfully completed DKG together
- DKG (Distributed Key Generation): The processes of generating keys for each node enabling them to sign using Keep's version of Threshold ECDSA [3].
- Signature: The process of securely generating a random number using Threshold ECDSA [3]
- Ownership Percent: The level of ownership a specific entity has in a group or signature
- Lynchpin: A node who's ownership exceeds the maximum malicious threshold

B. Model Structure

We construct the model using the MESA ABM framework (cite). MESA consists of an agent class with attributes and methods. We generate 3 types of agents Nodes, Groups, and Signatures. TABLE shows the specific differences in each of these.

ADD table for Agent types

The simulation model consists of the model class which instantiates the various agents and steps through the simulation. At each step the state of the model and each agent is updated. The scheduler manages the sequence of state changes. For this model we use a simulataneous activation scheduler, which first stages updates for each agent and then advances them simultaneously. (NEED TO CHECK SCHEDULER SEQUENCE AGAIN)

Usually steps measure a change in time. Therefore for our simulation we assume 1 step = 1 block.

C. Runs

We perform two sets of experiments to answer our research questions.

- Single run: Our single run of 1000 steps provide a first look at the performance of the sim, quickly. We also use this first experiment to evaluate when steady state behavior could occur. We also perform some initial evaluations of the impact of different stake distributions.
- Multiple runs: The second experiment consists of multiple runs with varying parameters. For each change in parameter we perform 6 runs. By varying parameters in these runs we attempt to identify sensitivity. We measure this sensitivity after the start of steady state behavior which we identify in the single run.

D. Assumptions

Stochastic Assumptions

To simplify the model we make stochastic assumptions for exogenous processes.

- Node Connection Delay: We apply a random uniform distribution to a user specified range (NEED JUSTIFICATION OR A JUSTIFIED SAMPLING)
- Node Connection Failure: This is one of the parameters we intend to adjust to test for sensitivity. Therefore we randomly pick a percentage of nodes to fail using a uniform distribution.
- Node Death: We uniform randomly pick nodes to die at a user specified rate.
- Signature Delay: A delay between when a signature is triggered and when it is executed. We use a poisson distribution. (NEED JUSTIFICATION or REFERENCE)
- Node Owner Assignment: We use a normal distribution to assign owners to nodes. (NEED JUSTIFICATION)

Stake distribution

Since one of our research questions involves understanding the effects of centralization on system behavior, we use three different token distribution models with varying degrees of decentraliation to evaluate this impact.

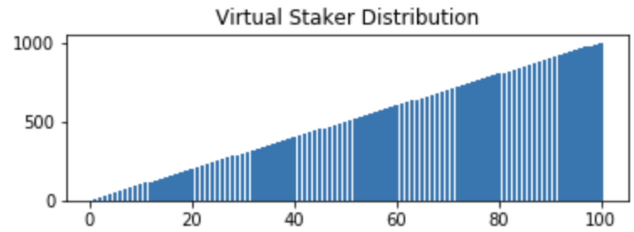

Fig. 1. Linear Stage Distribution

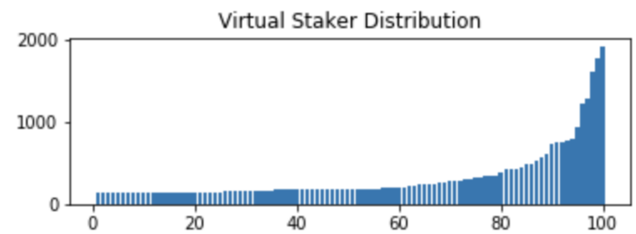

Fig. 2. Top 30 Percent Ethereum Distribution

- Linear Distribution: We assume a simple linear distribution as our most decentralized case. We take 50000 stakes and allocate them linearly to 100 nodes as shown in Figure 1
- Ethereum Distribution: In the spectrum of decentralization, we assume ETH to be moderateley decentralized. We therefore take the distribution of the top 30 percent account holders and apply it with some normalization in Figure 2. (NEED TO JUSTIFY TOP 30 PERCENT)
- Assumed Stake Distribution: Should we disclose?

Assumptions Model Assumptions

VII. Verification and Validation

Benchmark using analytical methods - from Promethea

VIII. Results

A. Emergence of steady state behavior

Using the single run experiment, we discover that steady state behavior begins at around 400 blocks. This appears due to the smoothening out of bootstrapping effects. (NEED TO CHECK SIM AS STEADY STATE SEEMS TO BE IN FLUX)

B. Effects of Node failures on group and signature characteristics

- Effects of stake distributions on group and signature characteristics
- Additional analyses?

IX. Conclusion and Future work

- Dynamic systems -Transfer functions instead of stochastic representations?
- Utility functions
- Genetic Algorithms
- Additional questions

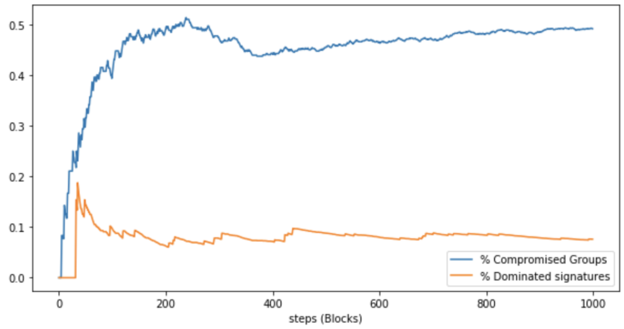

Fig. 3. Emergence of Steady State Behavior

REFERENCES

[1] Amy L. Bauer, Catherine A.A. Beauchemin, and Alan S. Perelson. Agent-based modeling of host-pathogen systems: The successes and challenges. *Information Sciences*, 179(10):1379–1389, 2009.

[2] Ling Feng, Baowen Li, Boris Podobnik, Tobias Preis, and H. Eugene Stanley. Linking agent-based models and stochastic models of financial markets. *Proceedings of the National Academy of Sciences of the United States of America*, 109(22):8388–8393, 2012.

[3] Rosario Gennaro and Steven Goldfeder. Fast multiparty threshold ECDSA with fast trustless setup. *Proceedings of the ACM Conference on Computer and Communications Security*, pages 1179–1194, 2018.

[4] Eliot R. Smith and Frederica R. Conrey. Agent-based modeling: A new approach for theory building in social psychology. *Personality and Social Psychology Review*, 11(1):87–104, 2007.

[5] L. Wang, K. Ahn, C. Kim, and C. Ha. Agent-based models in financial market studies. *Journal of Physics: Conference Series*, 1039(1), 2018.