

The Constellation Token Model

Constellation Labs

June 9 2018

Abstract

We propose the Constellation Token Model, a distributed consensus protocol governed by model that achieves equilibria irrespective of speculative market fluctuations.

Introduction

A Generative Economy is a living economy that is designed to generate the conditions for life to thrive, an economy with a built-in tendency to be socially fair and ecologically sustainable ¹. This definition could be paraphrased as essentially a game where the rules of play enforce certain invariants, the result of which is emergent stabilization. Essentially, the act of playing the game itself is what enforces the invariant. Distributed consensus protocols, and the currencies they support, can be modeled as generative economies. Network stability is governed by eventually consistent equilibrium and equilibrium is defined in terms of invariant measures (we want to enforce that something doesn't change, or can be relied upon) that are maintained. Why do we need this? It allows for the the enforcement of real world performance specifics that are decoupled from external market dynamics (gas/exchange prices) which allows for technology focused adoption while also providing an external speculation market similar to traditional commodities markets.

Constellation Economics

Consensus protocols exist to provide a utility and that utility is the basis of a generative economic game. In Constellation, that utility is throughput. Like all

¹Kelly, Marjorie, "Toward A Generative Economy"
<https://www.opendemocracy.net/ourkingdom/marjorie-kelly/toward-generative-economy>

consensus protocols, access to rewards is governed by delegate selection. In turn delegate selection is governed by a seeder/leacher ratio of transactions validated vs transactions submitted. We will use the term 'utility ratio' to describe the throughput a potential delegate has provided vs the throughput used. As nodes play they get ranked based on their utility ratio. Based on ranks they get access to rewards. Fees are charged relative to the scarcity of the utility that the protocol provides and are only charged if an account breaches its allotted utility ratio. The \$DAG token acts as an inflation mechanism like cash, which can be used to pay transaction fees when an account has not or can not maintain their required utility ratio. \$DAG is injected into the pool of validator rewards, and there is incentive to validate transactions that have an accompanying fee attached. Holders of the token assume the role of central bank, deploying \$DAG capital with the understanding that the free market dynamics of the protocol will ensure their account gets access to the network utility when capital is deployed.

Reputation is a unit that represents a 'stake' of total network utility (throughput). It is a perishable good in the constellation economy as accounts that leave the network will watch their reputation recede until 0 and they are replaced by a new player. Accounts' reputation is varying and falls according to various tiers which follow our definition of rank along with a branching factor.

Equilibrium and Market Dynamics

Constellation's value can be thought of as stake within a total computational space. Constellation's data model was designed to follow combinatorial models in distributed computing, where network state can be described in terms of simplicial geometry and transitions are defined in terms of a discrete gradient. The value of our network derives from the total computational space available for verifying transactions i.e. its maximum throughput. We can loosely model the total computational space as the volume of 'space' across nodes with respect to rank r

$$\int_M \epsilon_1 \wedge \dots \wedge \epsilon_r$$

where M is a manifold, ϵ is the sheaf of the vector space of rank r . Within our model this would be explicitly defined as the numerical integration over the outer product of our cohomological definition² which is a vector space given by the tensor product of each space across ranks

$$\sum_{1 \dots r} \Gamma^{\epsilon_1} \otimes \dots \otimes \Gamma^{\epsilon_r}$$

²<https://arxiv.org/abs/1805.07047>

Figure 1: We need two figures, One of the tiered cluster hierarchy and another top down look at the graph showing 'constellations' notes separated by tiers of rep score. Looks like gravity

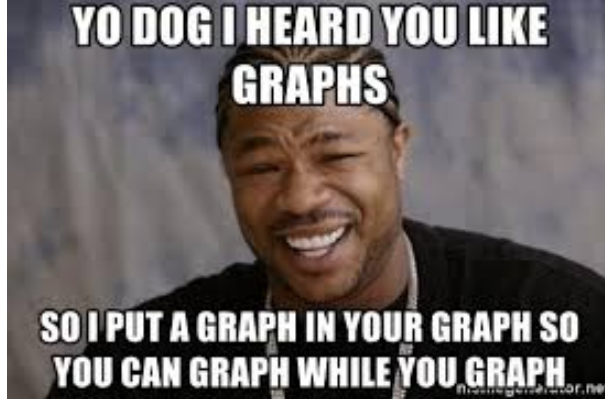
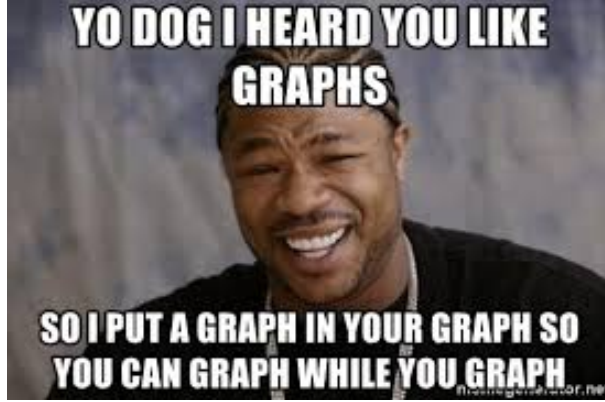


Figure 2: Formulate a ringed space as a hashmap and how each key is like a basis. If there are collisions there is like a singularity that appears. Sheaves are like subsets of the hashmap and represents the newest state of the partition of unity (sheaves are charts)



where each Γ^ϵ is a manifold corresponding to a rank in the differential form above, subordinate to the rank before it. If we define bandwidth (fiber/sec) as the total space³ of the resources given by each node

$$B_{(fiber/sec)} = \sum_r \int \rho_i \theta_i \omega$$

where ρ_i is a partition of unity and θ_i is a discrete diffeomorphism (protocol)

³sec 9.2 ftp.cis.upenn.edu/pub/cis610/public_html/diffgeom4.pdf

on our r-form ω and fibers are Signed Observation Edges⁴ of our data model. (make sure to elaborate that the total space is made up of all actions and the diffeomorphism itself is what determines the solution plane once we solidify into a snapshot. It is essentially a cut on the graph where we keep a subset of bundles not cut as tips new bundles are latched on. The diffeomorphism

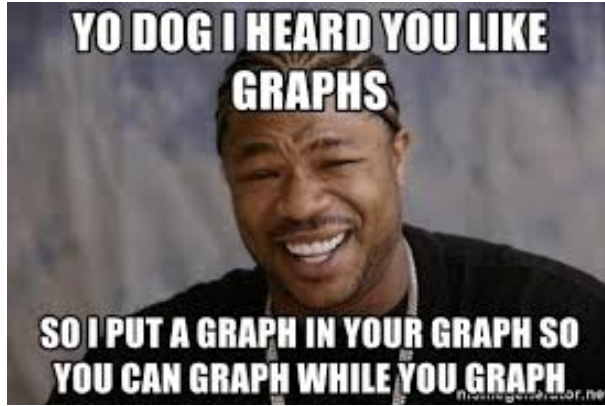
The actual value of our network can be described as the cost to host the resources each node provides.

$$R_{(\frac{s}{sec})} = \sum_r \int \rho_i T_i \omega$$

This is a slight modification to our definition of bandwidth above by replacing θ with T a mapping to instance types and cost. The total space of all value per node is given by the product of R and B , however in order to give a some measure of performance, we can do this by mapping each entry in the bandwidth space to it's normalized Shannon entropy (bits), giving us a measure of relative information gain to cost.

$$S_{(\frac{bits * s}{sec})} = B_{entropy} \oplus R = (\sum_r \int \rho_i H_i \omega) \oplus R$$

Figure 3: Show how the dag gets quilted an how agreed data adds negentropy while disagreed data causes entropy. Use this to quantify a node's shannon entropy (coloring subgraph of each node within the total space of all observations? highlight paths)



where H is our normalized Shannon entropy

$$H_n(p) = - \sum_i \frac{p_i \log_b p_i}{\log_b n}$$

⁴link to arch blog post

and p is a probability measure of how the result of a node's actions deviate from the acceptable state of the protocol and b is our base, bits in the case of Shannon entropy. Deficiency in entropy can be seen as a measure of efficiency. We want to use normalized entropy as using normal Shannon entropy scales with sample size.

Finally we can define a rewards function VR for an account a representing one or more nodes in the network as the tensor $S(a) \rightarrow \$DAG_{(\frac{bits}{sec})}$ which is just the sum over entries of S (the nodes) which correspond to an account a

$$S(a) = \sum_{i,r} h_i | h_{i,r} \in a, \forall h \in S$$

For our study, we can make a very simple approximation. We define the total computational space as the sum of all available resources $f(n)$ each node n provides

$$\Delta(n) = \sum_{i=0}^r e^{\beta f(n)} dn$$

where dn is a differential over the sorted (po-)set of all n by instance size, such that $f(n-1) \leq f(n)$, β is a "branching factor" or approximation of our partitions of union p_i and $f(n)$ is a distribution of computational resources. The relationship between $\$DAG$ and our network's value is that $\Delta(n) = VR_{\$DAG}$ where VR represents the rate of validator rewards distribution, a decaying function that will eventually disburse 40% of all total $\$DAG$, call the $\$DAG$ reserve owned by investors call it supply I . The VR is effectively 'locked' into the total space of $\Delta(n)$ and there is a conversion rate C that converts $\$DAG$ into some subset of compute space (stake) of Δ such that total $\$DAG$ is conserved. Thus, since $VR - C * I = 0$

$$\Delta(n) = C * I$$

which we can use to relate the influx of value with respect to nodes joining/leaving

$$\frac{\partial \Delta}{\partial n} = C(t) * \frac{\partial I}{\partial n}$$

where C is our equilibria condition, a 'maximum conversion rate' for spending $\$DAG$ for a subset of Δ . When capitol is injected from I the validator rewards increase by a factor of $C * di$ where di is the subset of total supply injected to

Δ . We can account for the increase in validator rewards with respect to our conversion rate, $C(t)$ which is equal to our 'minting' curve

$$C(t) = e^{-mt}$$

where m is the minting rate and we use time (t) as a continuous approximation of fiber (SOE) creation which in reality is discretized.

Analysis and Results

Figure 4: bitcoin validator rewards chart

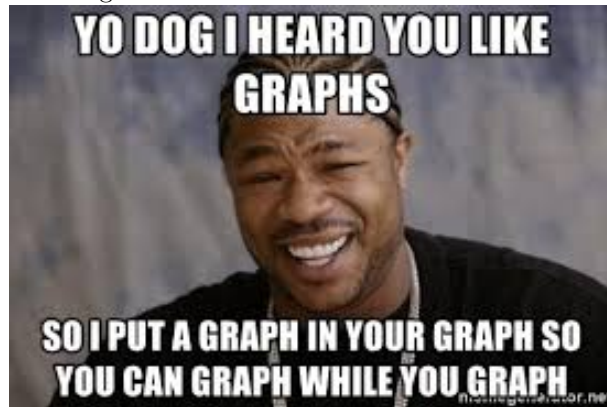


Figure 5: top left next to above bitcoin rewards according to hash power.

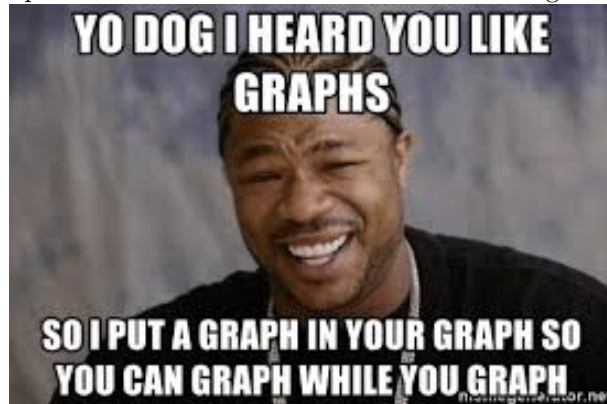


Figure 6: We need a basic plot of validator rewards with respect to both time and fiber creation, fiber creation will be discretized and time should roughly follow fiber creation. Top left corner of square.

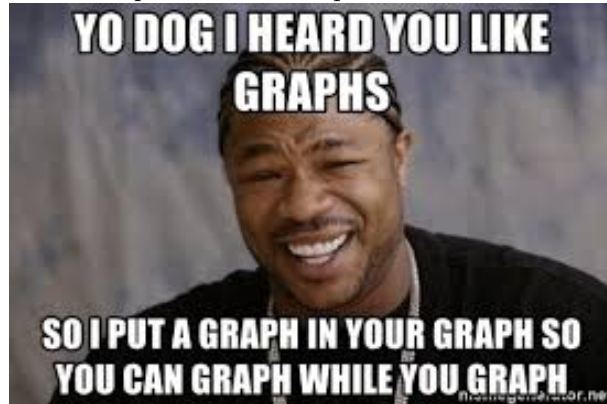


Figure 7: Constellation validator rewards chart. The number of nodes the network starts with num nodes in the end. Plot adoption ad exponential. As more nodes join an existing node gets 'shifted' up and they are allowed to process more txs (get more rewards)

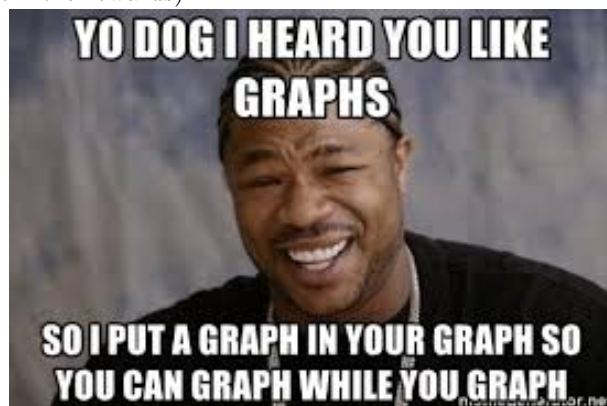


Figure 8: Also show that per node reward is higher after having been on the system for a while. Having been on the system for a while has the ability to earn more as nodes grow 'under' them

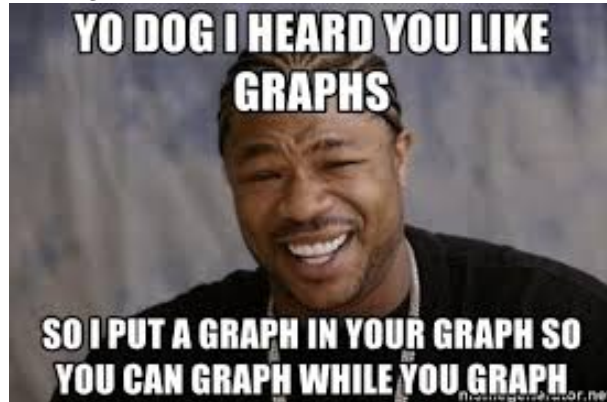


Figure 9: Put units of dollars in terms of cost per Ethereum tx (assumption they could perform this throughput). Show that value increases as more nodes join and peg to ethereum values (error bars around network cost vs tx rates since gas fluctuates).

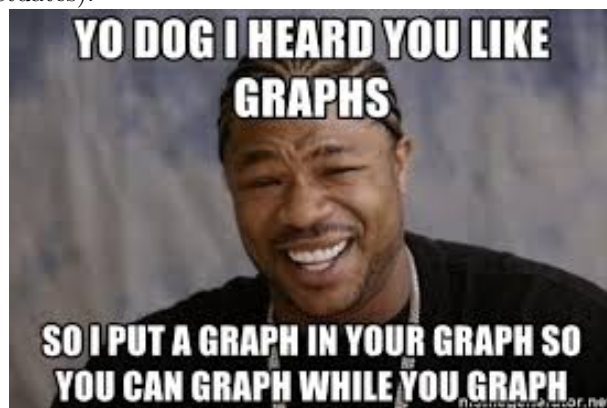


Figure 10: As node count increase, exponentially more value (compared to gas prices. Constellation transaction value/throughput chart. Show gas prices exponentially increasing as nodes grow linearly and ours doing the same but shifting to the right as nodes join (do different node counts on x axis and color shifts right) Next to this we want to also plot how transaction time vs cost increases ratio decreases as throughput increases on both eth and btc (wait times vs gas prices) and compare to us which stays flatlinekeychain

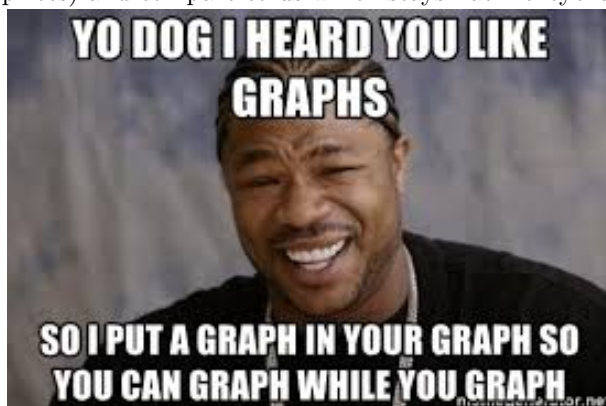


Figure 11: We need a plot that connects entropy and seeder leacher ratios. We can do this by having one main curve with error bars being a gaussian of deviations from acceptable states. We will want to pair this with a discrete plot of seeder leacher ratios to VR then superimpose a curve over the discrete intervals. Error bars around that smooth curve represent a rate of fibers accepted to fibers created

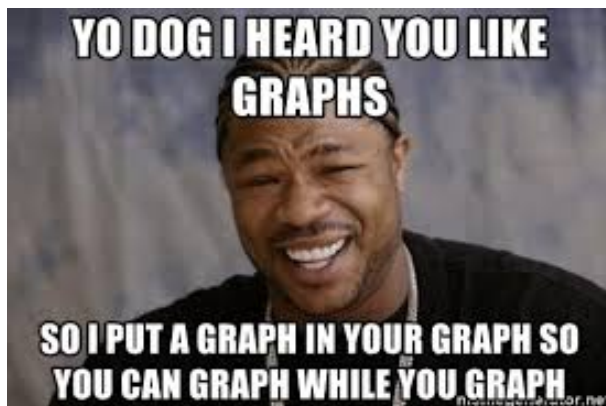


Figure 12: We need a plot that shows the increase in value network according to instance types and nodes. Set this next to a plot of throughput give instance types and nodes

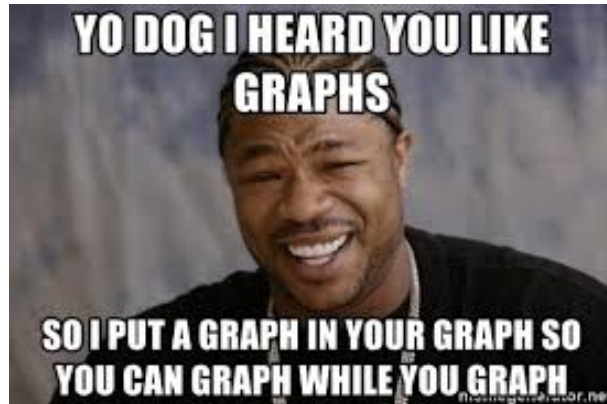


Figure 13: We need a plot that connects seeder leacher ratios to information gain/loss. This will prob be hardest

