Akash: Token Model & Mining Economics

Greg Osuri (Akash Network)

September 1, 2019

This paper covers the economics of the Akash Network and introduces the Akash Token (AKT). In this paper, we present various incentives for ensuring the economic security of the Akash ecosystem as well as drive adoption. We propose an inflationary mechanism to achieve economic goals along with estimates for mining rewards and inflation rates. Additionally, we also present mechanisms for allowing a multitude of fee tokens, which will improve the user experience of using the blockchain.

NOTE: This paper is a work in progress and intended for PRIVATE distribution only.

1 Introduction

Akash is a permission-less marketplace trade compute cycles. In this paper, we present the Akash Token (AKT) model that's designed to a) maintain ecosystem sovereignty b) provide economic security c) encourage early adoption. Some definitions:

Akash Token (AKT) AKT is the native token of the Akash Network. The core utility of AKT is to act as a staking mechanism to secure the network. The amount of AKTs staked towards a validator defines the frequency by which the validator may propose a new block and its weight in votes to commit a block. In return for bonding (staking) to a validator, an AKT holder becomes eligible for block rewards (paid in Akash Tokens) as well as a proportion of transaction fees (paid in any of the whitelisted tokens).

Validator Validators secure the Akash network by validating and relaying transactions, proposing, verifying and finalizing blocks. There will be a limited set of validators, initially 100, that are required to maintain a high standard of automated signing infrastructure. Validators will charge Providers a commission fee in Akash Tokens.

Delegator Delegators are holders of the AKT and use some or all of their tokens to secure the Akash chain. While there is no minimum amount of tokens required in order to stake, providers are required to maintain a stake in Akash tokens proportional to the hourly income earned. In return, providers earn a proportion of the transaction fee as well as block rewards.

Provider Providers offer computing cycles (usually unused) on the Akash network and earn a fee for their contributions.

Tenants Tenants lease computing cycles offered by providers for a market-driven price (set during a reverse auction).

1.1 Marketplace Overview

Akash provides a novel spot market to lease containers. A container is a unit of computing ($U \equiv CPU, Memory, Disk$) that can run any type of cloud application. With Akash, early adopters can enjoy over 8x lower cost than the market.

All marketplace transactions are persisted on the Akash blockchain. To lease a container, the tenant (developer) requests a deployment by specifying the type of unit(s), quantity of each unit and attributes to match [*like region (US) or privacy features like (SGX)*]. The fee can be paid in any currency of choice from a set of whitelisted currencies.

Each type of workload in the deployment is defined as a compute unit. For example, if a user's stack contains a web app, a database and a cache server, the user specifies three compute units.

The compute units are grouped into placements with attributes (such as region). The tenant specifies the maximum price they're willing to pay l_u for each compute unit per unit time, defined as $T_u = 1$ second in each placement group p. The tenant also specifies the number of compute units required, U_req for the placement group.

An `Order` $\mathcal{O}$ is created for each placement group when a validator accepts the deployment. Each order contains the maximum price the tenant is willing to pay, $l_{max} = l_u \cdot U_{req}$

The provider(s) p that match all the requirements of the order then place a bid $l_{p,b}$, where $l_{p,b} \leq l_{max}$ by competing on price using a `SubmitFulfillment` transaction. The provider that bid for lowest amount $\min(l_{i,b})$ in the `Order` wins upon which a `Lease` is created between the tenant and the provider for the order.

1.2 Proof of Stake using Tendermint Consensus

Akash employs a blockchain secured by a *Proof-of-Stake* consensus model as a Sybil resistance mechanism for determining participation in its consensus protocol and implements Tendermint(Buchman, Kwon, and Milsosevic, n.d.) algorithm for Byzantine fault-tolerant consensus. Tendermint was designed to address the speed, scalability, and environmental concerns with Proof of Work with below set of properties:

a) Validators take turns producing blocks in a weighted round-robin fashion, meaning it has the ability to seamlessly change the leader on a per-block basis.

b) Strict accountability for byzantine faults allows for punishing misbehaving validators and provide economic security for the network.

Anyone who owns an Akash token can bond (or delegate) their coins and become a validator, making the validator set open and permission-less. The limited resource of Akash tokens acts as a Sybil prevention mechanism.

Voting power is determined by a validator's bonded stake (not reputation or real-world identity).No single actor can create multiple nodes in order to increase their voting power as the voting power is proportional to their bonded stake. Validators are required to post a "security deposit" which can be seized and burned by the protocol in a process known as "slashing".

These security deposits are locked in a bonded account and only released after an "unbonding period" in an event the staker wishes to unbond. Slashing allows for punishing bad actors that are caught causing any attributable byzantine faults to harm to the well-functioning the system.

The slashing condition and the respective attributable byzantine faults and punishments are beyond the scope of this paper.

1.2.1 Limits on Number of Validators

Akash's blockchain is based on Tendermint consensus which gets slower with more validators due to the increased communication complexity. Fortunately, we can support enough validators to make for a robust globally distributed blockchain with very fast transaction confirmation times, and, as bandwidth, storage, and parallel compute capacity increases, we will be able to support more validators in the future.

On Genesis day, the number of validators V_i

is set to $V_i(0) = V_{i,0} = 64$ and the number of validators at time t *year* will be:

$$V_n(t) = \lfloor \log_2(t) \cdot V_{i,0} \rfloor \tag{1}$$

So, in 10 *years*, there will be $V_n(10) = 213$ total number of validators.

2 The Akash Token Model

2.1 Staking Token (AKT)

The primary utility for AKT is staking (providing security to the network) (similar to Atom token at Cosmos(Aggarwal, n.d.)). Although AKT can be used in the marketplace (and elsewhere), it is not intended to be used to pay a fee or used as a currency and thus give you the ability to earn transaction fees and the block rewards. The income stakers generate is proportional to the tokens staked and length of staking commitment.

2.2 Transaction Fees using Multitude of Tokens

In order to avoid issues of network abuse (spam), all `transactions` and `leases` on Akash are subject to a fee. Every transaction has a specific amount of gas associated with it: `GasLimit` for processing the transaction as long as it does not exceed `BlockGasLimit.`

The `GasLimit` is the amount of gas which is implicitly purchased from the sender's account balance. All leases (purchases) require the tenant (buyer) to pay `TakeFee` and the seller (provider) to pay a `MakeFee.`

Akash accepts a multitude of tokens for fees. Each validator and provider on Akash can choose to accept any currency or a combination of currencies as fees.

Resulting transaction fees, minus a network tax that goes into a reserve pool are split among validators and delegators based on their stake (amount and length).

2.3 Maker and Taker Fee Schedule

Leasing compute is either zero-fee or a small fee depending on the user's activity in the last 30 days. Lease fees have a distinction of a `Maker` fee or a `Taker` fee. A ***maker fee*** is paid when you add computing capacity to Akash network by fulfilling an order in the order book when the lease is created. A ***taker fee*** is paid when you remove computing capacity from Akash by placing an order in the book when the lease is created.

For a lease l, the *aggregate trade activity factor* of the a stakeholder of the lease for a given time t is defined by:

$$\kappa_l = \frac{\sum_{t=1}^{T_a} l(1-t)}{\sum_{t=1}^{T_a} L(1-t)}, \tag{2}$$

where $L(t)$ is the aggregate trade activity of the network and $T_a = 30$ *days*.

Given, $\mathcal{P}_l = 0.5$ is the *fee distribution factor*, for that, the maker fee $R_{mk}(l)$ will be:

$$R_{mk}(l) = \frac{10^{1-4\mathcal{P}_l}}{\log(10^{4\mathcal{P}_l})} \cdot \log\left(\frac{\mathcal{P}_l}{\min(\kappa_l, \mathcal{P}_l)}\right), \tag{3}$$

and the taker fee $R_{tk}(l)$ will be:

$$R_{tk}(l) = \frac{10^{1-\frac{7\mathcal{P}_l}{2}}}{\log\left(10^{\frac{7\mathcal{P}_l}{2}}\right)} \cdot \log\left(\frac{\mathcal{P}_l}{\min(\kappa_l, \mathcal{P}_l)}\right). \tag{4}$$

3 Token Economics and Incentives

Providers earn income by selling computing cycles to tenants that lease computing services for a "fee". However, in the early days of the network, there is a high chance the providers will not be able to earn a meaningful income due to a lack of sufficient demand from the tenants (consumers of computing) which in turn hurts demand because of lack of supply.

To solve this problem, we can incentive the providers using inflation by means of block rewards until a time a healthy threshold can be achieved.

In this section, we present the economics behind mining. An ideal inflation model should the below set of properties:

- Early providers can provide services exponentially lower costs than market, to accelerate adoption.
- The income a provider can earn is proportional to the number of tokens staked.
- The block compensation for a Staker is proportional to their staked amount, the time to unlock and overall locked tokens.
- Stakers are encouraged to stake for longer periods.
- Short term stakers (bear markets) are also encouraged but gain a smaller reward.
- To maximize compensation, stakers are incentivized to 're-stake' their income.

3.1 Motivation

Akash Network intends to gain early adoption by offering exponential cost savings as value proposition in addition to implied productivity benefits of a serverless infrastructure, which is extremely compelling especially to data intensive (machine learning etc) applications.

3.2 Stake and Bind: Mining Protocol

A provider (*staker*) commits to provide services for at least time T and intends to earn service income r every compensation period $T_{comp} = 1\ day$. For that, they stake tokens s and specify an unlock time t_1, where minimal lock-time $t_1 - t$ should not be less than $T_{min} = 30\ days$. Additionally, they delegate (voting power) to validator v by bonding their stake via `BindValidator` transaction.

At any point, a staker can: a) Split their stake (or any piece of their stake) into two pieces. b) Increase their stake l by adding more AKT. c) Increase the lock time T, where $T > T_{min}$.

Stakers choose to split their stake because the compensation is dependent on lock time L which will be addressed in later sections.

3.3 General Inflation Properties

3.3.1 Initial Inflation

If we assume Akash will have the same number of tokens locked as NuCypher (Egorov, Wilkinson, and, n.d.) and DASH (Duffield and Diaz, n.d.): $\lambda = 60\%$ then we'll have $1 - 40\%$ AKT in circulation. The adjusted inflation rate for inflation, I will be:

$$I^* = \frac{I}{1-\lambda}, \tag{5}$$

Considering, ZCash ("ZCash Emmission Rate," n.d.) had $I^* = 350$ (turn around point during the overall bull market), which makes $I = 140\% APR$, it is reasonably safe to set the initial inflation to be $I_0 = 100\% APR$ (meaning 1/365 per day).

3.3.2 Inflation Decay

If we assume that all miners have the maximum compensation rate and we choose the inflation decay factor (time to half the inflation rate) to be 2 in $T_{1/2} = 2\ years$. Inflation depending on the time passed from the Genesis t, looks like:

$$I(t) = I_0 \cdot 2^{-\frac{t}{T_{1/2}}} = I_0 \exp\left[-\ln 2 \frac{t}{T_{1/2}}\right]. \tag{6}$$

In this case, the dependence of the token supply on the time t is:

$$M(t) = M_0 + \int_0^t I(t)\, dt = M_0 + \frac{I_0 T_{1/2}}{\ln 2}\left[1 - 2^{-\frac{t}{T_{1/2}}}\right] \tag{7}$$

If we let I_0 be relative inflation rate, then $I_0 = i_0 M_0$. For 100% APR, $i_0 = 1$ and $I_0 = M_0$, which gives us the maximum number of tokens which will ever be created (as illustrated in fig. 1):

$$M_{\max} = M(\infty) = M_0 \left(1 + \frac{i_0 T_{1/2}}{\ln 2}\right) \approx 3.89\, M_0, \tag{8}$$

where M_0 is initial number of tokens.

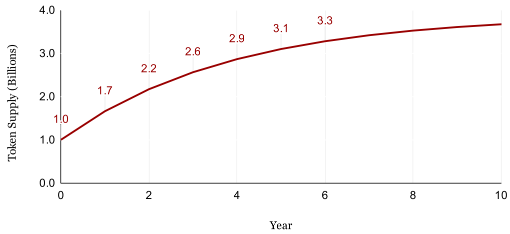

Figure 1: Token Supply Over Years. It takes 89 years to reach maximum supply of 3.89B tokens with inflation reaching near 0% APR in 15 years when initial inflation is 100% APR and halving every 2 years

3.3.3 Staking Time and Token Emission

We will reward the full compensation ($\gamma = 1$) to the stakers who are committed to stake at least $T_1 = 1$ *year* (365 days), however those who stake for $T_{\min} = 1$ *month* will get close to half the compensation ($\gamma \approx 0.54$).

$$\gamma = \left(0.5 + 0.5 \frac{\min(T_i, T_1)}{T_1}\right), \tag{9}$$

$$T_{i,\text{initial}} > T_{\min}, \tag{10}$$

Where, the unlocking time T_i means the time left to unlock the tokens $t_1 - t$. The initial T_i cannot be set smaller than $T_{min} = 1$ *month*, but it eventually becomes smaller than that as the time passes, and as the stake gets close to unlocking the stake.

Shorter stake periods (for lower rewards) result in a lower daily token emission. Considering, miners will most likely stake for short periods during a bear market; lower emissions will provide much better price and stability as a result.

The emission half decay time $T^*_{1/2} = T_{1/2}/\gamma^*$, where γ^* is the mean staking parameter, is also prolonged when $\gamma < 1$. $T_{1/2}$ prolongs to 4 years instead of 2 if all stakers have $\gamma^* = \gamma = 0.5$.

The total supply over time (eq. 7) at $\gamma^* \neq 1$ will then look like:

$$M(t) = M_0 \left[1 + \frac{i_0 \gamma^* T^*_{1/2}}{\ln 2}\left(1 - 2^{-\frac{t}{T^*_{1/2}}}\right)\right]. \tag{11}$$

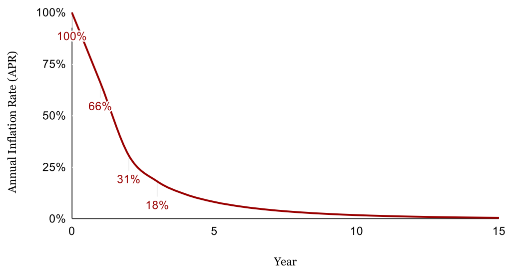

Figure 2: Inflation over the Years when $\gamma = 0.75$. The rate reaches near 0% in 15 years and single digits by 5th year

3.4 Delegate Pool Distribution

The exponential is a solution of a differential equation where inflation is proportional to the amount of not yet mined tokens:

$$I(t) = \frac{\ln 2}{T_{1/2}} \left(M_{\max} - M(t)\right) \tag{12}$$

$$dM = I(t)\, dt. \tag{13}$$

where $M(t)$ is the current token supply with $M(0) = M_0$ and dt can be equal to the mining period (1 day). Each validator can trivially calculate its dM using few operations using the token supply M from the last period. So, the amount of mined tokens for the validator pool p in the time t will be:

$$\delta m_{v,t} = \frac{s_v}{S} \frac{\ln 2}{T_{1/2}} \left(M_{\max} - M_{t-1} \right), \tag{14}$$

$$dM_t = \sum_v dm_{v,t}, \tag{15}$$

where s_v is the number of tokens bound to the validator delegate pool v and S is the total number of tokens locked. Instead of calculating all the sum over v, each validator can add their portion $\delta m_{v,t}$.

The distribution factor for a delegate bound to pool p is:

$$\kappa = \frac{1}{2} \left(\frac{\gamma}{\gamma_v} + \frac{s}{S_v} \right), \tag{16}$$

γ_v is the aggregate stake compensation factor for the pool and S_v is the sum of all tokens bound to the pool.

3.5 Mining strategies and expected compensation

In this section, we look at three possibilities: a staker liquidating all the compensation while extending the lock time (Liquidate mining compensation), a staker adding all the compensation to their current stake, and a miner waiting for their stake to unlock after time T. Each of these possibilities could have different distributions of γ. Let's consider $\gamma = 1$ and $\gamma = 0.5$ as two marginal values. Let's take the amount of tokens locked to be $\lambda = 60\%$, as in DASH.

3.5.1 Liquidate Mining Compensation

In this scenario, all stakers in the pool are liquidating all the earnings every T_{comp} period. The total amount of tokens staked in the network can be expressed as $S = \lambda M$. Assuming all the delegates has equal amount of stake bound to the pool. The amount of stake stays constant in this case and equal to $m_i = s$ making $m_v = s_v$ and $\gamma = \overline{\gamma_v}$ where, $\overline{\gamma_v}$ is the mean staking parameter of the pool. Then, the pool mining rate (i.e. the cumulative pool reward) is:

$$\frac{dr_p}{dt} = \overline{\gamma_v} \frac{l}{\lambda M(t)} \frac{\ln 2}{T_{1/2}} \left(M_{\max} - M(t) \right). \tag{17}$$

When we substitute $M(t)$ from eq. 11 and integrate over time, we find total pool compensation:

$$r_p(t) = l \frac{\gamma}{\gamma^* \lambda} \ln \frac{M(t)}{M_0}, \tag{18}$$

If $\Delta r_p(t) = r_p(t) - \mathcal{C}$ where $\mathcal{C}$ is validator's commission, that brings individual staker's compensation to be:

$$r(t) = \kappa \cdot \Delta r_p(t). \tag{19}$$

If $\gamma = 1$ (staking for 1 *year*) and $\lambda = 60\%$ (60% of all nodes in the network are staking). With $\mathcal{C} = 0.1 \cdot r(t)$, staker's compensation starts from 0.45% *per day* in AKT, or 99.2% during the first year of staking.

We should note that if other miners stake for less than a year ($\gamma^* < 1$), the inflation rate decays slower, and the compensation over a given period will be higher.

3.5.2 Re-stake mining compensation

Instead of liquidating mining compensation, it could be re-staked into the pool in order to increase the delegate's stake and. In this case, the actual stake s is constantly increasing with time:

$$\frac{ds}{dt} = \gamma \frac{s}{\lambda M(t)} \frac{\ln 2}{T_{1/2}} \left(M_{\max} - M(t) \right). \tag{20}$$

If we substitute $S(t)$ from eq. 11 and solve this differential equation against s, we get:

$$s(t) = s(0) \left[\frac{M(t)}{M_0} \right]^{\frac{\gamma}{\gamma^* \lambda}}. \tag{21}$$

Assuming the validator commission is 1%, if $\gamma = 1$ (staking for 1 *year+*) and $\lambda = 60\%$ (60% of all nodes in the network are staking), delegate's compensation starts from 0.45% *per day* in AKT tokens, or $s(1) - s(0) = 176.5\%$ during the first year of staking.

3.5.3 Take mining compensation and spindown

When the node spins down, the miner doesn't extend the time for end of staking t_1, and the compensation is constantly decreasing as the time left to unlock becomes smaller and smaller, effectively decreasing γ gradually towards 0.5. That's the default behavior: to avoid that, the miner should set t_1 large enough, or increase t_1 periodically.

3.5.4 FAQ

How many tokens will ever be in existence? We'll start with 1 *billion* tokens, and the maximum amount of tokens ever mined will be 3.89 *billion*.

What's the inflation rate? The inflation rate will depend on how many short-term miners and long-term miners are working in the system. Depending on this, the initial inflation will be between 50%~APR (if all miners are very short term) and 100%~APR (if all miners commit for a long term). The inflation will decay exponentially every day, halving sometime between 2~years (if all the miners are long term) and 4~years (if all the miners are short term).

Aggarwal, Sankalp. n.d. "Cosmos Multi-Token Proof of Stake Token Model." https://github.com/cosmos/cosmos/blob/master/Cosmos_Token_Model.pdf.

Buchman, Ethan, Jae Kwon, and Zarko Milsosevic. n.d. "The Latest Gossip on BFT Consensus." https://arxiv.org/abs/1807.04938.

Duffield, Evan, and Daniel Diaz. n.d. "Dash: A Payments-Focussed Cryptocurrency." https://github.com/dashpay/dash/wiki/Whitepaper.

Egorov, Michael, MacLane Wilkinson, and. n.d. "NuCypher: Mining & Staking Economics." https://www.nucypher.com/static/whitepapers/mining-paper.pdf.

"ZCash Emmission Rate." n.d. https://z.cash/technology/.