

Avalanche Native Token (\$AVAX) Dynamics

2020/06/25

Stephen Buttolph, Amani Moin, Kevin Sekniqi, and Emin Gün Sirer

Abstract. This paper discusses the key implementation details, in particular the token economics (tokenomics), of the native token of the **Avalanche** platform, called **\$AVAX**. The native token secures the network, pays for fees, and provides the basic unit of account between the multiple blockchains deployed on the larger **Avalanche** network. For additional details on **Avalanche**, which serves as a versatile and universal platform, allowing anyone to launch new blockchains with their own rules, virtual machines, and validator sets, we guide the reader to either the accompanying architectural paper [1] or the **Avalanche** docs [2].

Git Commit: 30d3aeb5863f5fcb033f9a34bcad5f21a79a4b7d

Disclosure: The information described in this paper is preliminary and subject to change at any time. Furthermore, this paper may contain forward-looking statements. Forward-looking statements generally relate to future events or our future performance. This includes, but is not limited to, **Avalanche's** projected performance; the expected development of its business and projects; execution of its vision and growth strategy; and completion of projects that are currently underway, in development or otherwise under consideration. Forward-looking statements represent our managements beliefs and assumptions only as of the date of this presentation. These statements are not guarantees of future performance and undue reliance should not be placed on them. Such forward-looking statements necessarily involve known and unknown risks, which may cause actual performance and results in future periods to differ materially from any projections expressed or implied herein. **Avalanche** undertakes no obligation to update forward-looking statements. Although forward-looking statements are our best prediction at the time they are made, there can be no assurance that they will prove to be accurate, as actual results and future events could differ materially. The reader is cautioned not to place undue reliance on forward-looking statements.

1 Introduction

The economic model of any new digital currency/asset is one of the most critical components of the platform that the asset resides on. This is especially true for the native token of a self-sovereign, permission-less platform, like **Avalanche**. In this paper, we discuss the economic

design of the native token, called **\$AVAX**. The discussion is broken down into the governance
 30 properties of the token, its supply, minting (rewards) function of stakers, and other pertinent
 economics details, such as the transactional economy.

1.1 Key \$AVAX Properties

These are the key takeaway properties of the design of the **\$AVAX** economics model:

- The resources spent by a validator for staking are proportional to that validator’s total
 35 stake.
- The rewards accumulated by a validator for validating are proportional to that valida-
 tor’s total stake.
- Since **Avalanche** is leaderless, there is no “rich-get-richer” compounding effects.
- Validators that lock their stake for longer are rewarded more.
- 40 – Validators are incentivized to stay online and operate correctly as their rewards are
 based on proof-of-uptime and proof-of-correctness.
- **\$AVAX** is a capped-supply token, with a maximum cap of 720 million tokens.
- While capped, **\$AVAX** is still governable. The rate at which the maximum cap is reached
 is subject to governance.
- 45 – Fees are not paid to any specific validator. Instead, they are burned, thus increasing
 scarcity of the **\$AVAX**.

2 Governance

We initiate our survey of the **Avalanche** economic design by first discussing governance, as
 it plays a critical role within future components. To enable the system to adapt to changing
 50 economic conditions, the **Avalanche** platform enables key system parameters to be modified
 dynamically based on user input. A workable process for finding globally acceptable values
 for system parameters is critical for decentralized systems without custodians. **Avalanche**
 can use its consensus mechanism to build a system that allows anyone to propose special
 transactions that are, in essence, system-wide polls. Any participating node may issue such
 55 proposals.

Reward rate is an important parameter that affects any currency, whether digital or fiat.
 Unfortunately, cryptocurrencies that fix this parameter might face various issues, including
 deflation or hyper-inflation. To that end, the reward rate will be subject to governance,

within pre-established boundaries. This will allow token holders to choose the rate at which
 60 \$AVAX reaches its capped supply.

Transaction fees, denoted by the set \mathcal{F} , will also be eventually be governed. \mathcal{F} is effectively a tuple which describes the fees associated with the various instructions and transactions supported in future releases. Finally, staking times and amounts will also be governable. The list of these parameters is defined in Figure 1.

- Δ : Staking amount, denominated in \$AVAX. This value defines the minimal stake required to be placed as bond before participating in the system. The default value on genesis will be 2,000 \$AVAX.
 - δ_{min} : The minimal amount of time required for a node to stake into the system. The default value on genesis will be 2 weeks.
 - δ_{max} : The maximal amount of time a node can stake. The default value on genesis will be 52 weeks.
 - γ, λ : The two key parameters in governing the minting rate function.
 - \mathcal{F} : the fee structure, which is a set of governable fees parameters that specify costs to various transactions.

Fig. 1. Key governable parameters used in Avalanche.

65 Ultimately, we note that governance in \$AVAX has hysteresis, meaning that changes to parameters are highly dependent on their recent changes. There are two limits associated with each governable parameter: time and range. Once a parameter is changed using a governance transaction, it becomes very difficult to change it again immediately and by a large amount. These difficulty and value constraints relax as more time passes since the last
 70 change. These restrictions are in keeping with the design philosophy of predictability: the system must never change drastically over a short period of time, allowing users to safely predict system parameters in the short term, while having strong control and flexibility for the long term.

3 Token Economics

75 \$AVAX has a capped-supply of 720,000,000 (720M) tokens. The genesis block will have 360M \$AVAX tokens. The rest of the 360M tokens will be minted according to Equation 1. For a graphical representation, Figure 2 shows the token emissions curve between \$AVAX and BTC. The principle of the emissions function chosen for Avalanche is simple: reach a capped

supply, in a fashion similar to Bitcoin's emissions curve, and yet maintain the ability to
 80 govern the rate at which the system reaches this limit.

3.1 Minting Function

R_j is total number of tokens at year j , with $R_1 = 360M$, and R_l representing the last year
 that the values of $\gamma, \lambda \in \mathbb{R}$ were changed; c_j is the yet un-minted supply of coins to reach
 720M at year j such that $c_j \leq 360M$; u represents a staker, with $u.s_{amount}$ representing
 85 the total amount of stake that u possesses, and $u.s_{time}$ the length of staking for u .

$$R_j = R_l + \sum_{\forall u} \rho(u.s_{amount}, u.s_{time}) \times (c_j/L) \times \left(\sum_{i=0}^j \frac{1}{\left(\gamma + \frac{1}{1+i^\lambda}\right)^i} \right) \quad (1)$$

where,

$$L = \left(\sum_{i=0}^{\infty} \frac{1}{\left(\gamma + \frac{1}{1+i^\lambda}\right)^i} \right) \quad (2)$$

At genesis, $c_1 = 360M$. The values of γ and λ are governable, and if changed, the function
 is recomputed with the new value of c_* . We have that $\sum_* \rho(*) \leq 1$. $\rho(*)$ is a linear function
 that can be computed as follows ($u.s_{time}$ is measured in weeks, and $u.s_{amount}$ is measured
 90 in \$AVAX tokens):

$$\rho(u.s_{amount}, u.s_{time}) = (0.002 \times u.s_{time} + 0.896) \times \frac{u.s_{amount}}{R_j} \quad (3)$$

If the entire supply of tokens at year j is staked for the maximum amount of staking time
 (one year, or 52 weeks), then $\sum_{\forall u} \rho(u.s_{amount}, u.s_{time}) = 1$. If, instead, every token is staked
 continuously for the minimal stake duration of two weeks, then $\sum_{\forall u} \rho(u.s_{amount}, u.s_{time}) =$
 0.9. Therefore, staking for the maximum amount of time incurs an additional 11.11% of
 95 tokens minted, incentivizing stakers to stake for longer periods. Due to the capped-supply,
 the function above guarantees that regardless of the number of governance changes, we will
 never exceed a total of 720M tokens. Therefore,

$$\lim_{j \rightarrow \infty} R_j = 720M \quad (4)$$

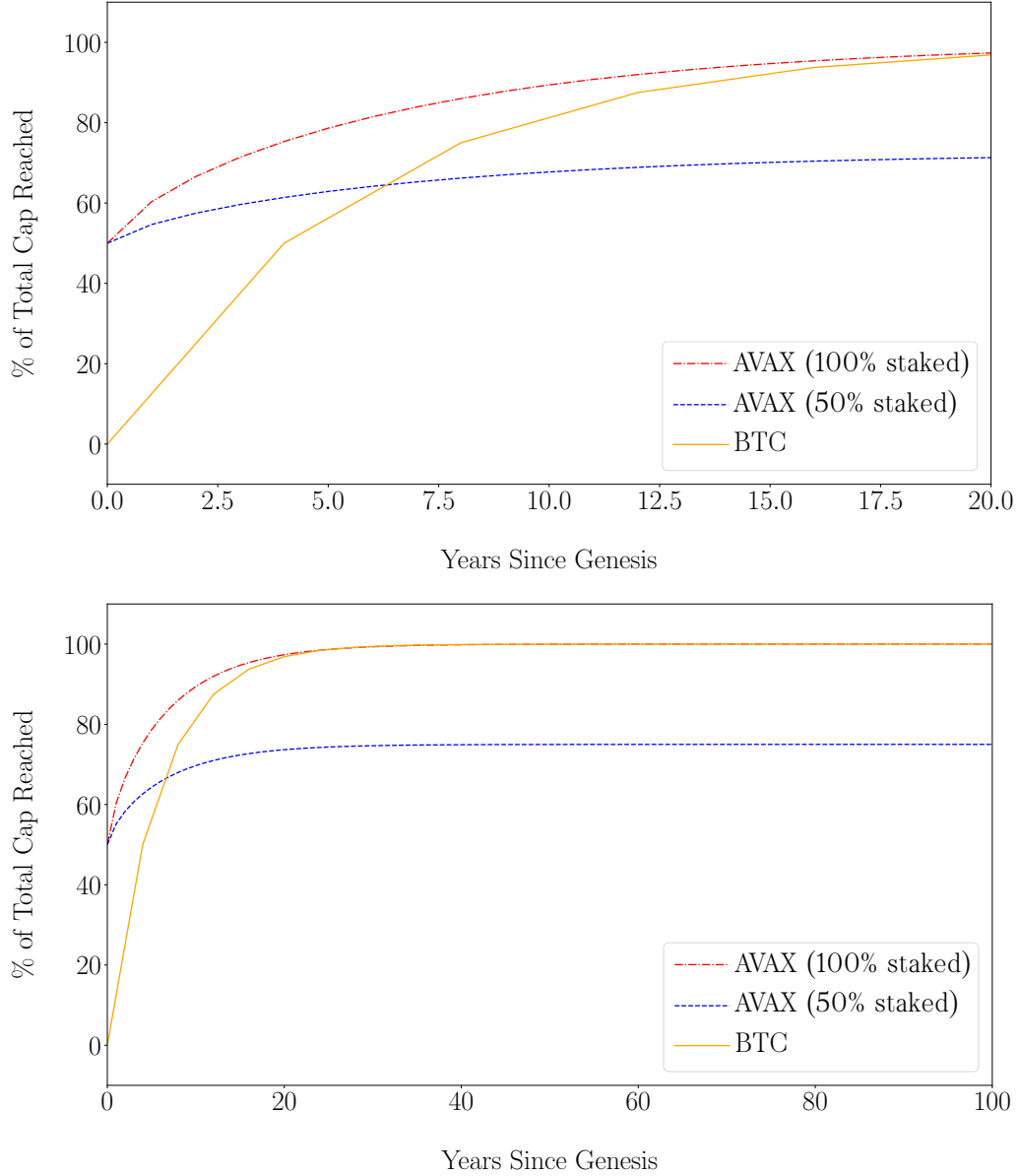


Fig. 2. Token emissions between \$AVAX and BTC, calculated over a 20 year and 100 year horizon, with $\gamma = 1.15$ and $\lambda = 1.1$. The curve for “\$AVAX (100% staked)” represents the case where every token is being staked repeatedly for the maximum staking duration of one year, i.e. $\sum_{\forall u} \rho(u.s_{amount}, u.s_{time}) = 1$. On the other hand, the curve of “\$AVAX (lower)” represents the case where only 50% of the tokens are being staked repeatedly over the minimal staking duration of two weeks, i.e. $\sum_{\forall u} \rho(u.s_{amount}, u.s_{time}) = 0.45$. We note that, for simplicity, these graphs represent the case where γ and λ are fixed at genesis and never governed afterwards. The goal of changing γ and λ is to increase total supply of tokens in case the empirically observed total staked supply is too low.

4 Minting Mechanism

Minting in \$AVAX is designed to incentivize nodes to behave in a way that positively helps
 100 global outcomes. This is accomplished by special *minting transactions*. A node earns the
 right to mint by first putting up a stake and then participating actively in the consensus
 process. Specifically, node rewards are directly linked to their uptime and response latency.
 Every node maintains local information about the liveness and behavior of each other node
 with which it interacted. Whenever a node v is sampled by u , the latter maintains a local
 105 tuple of (response bit, timestamp). The first entry is a single bit representing whether v
 responded within the timeout, and the second represents the timestamp of the response. In
 other words, minting in **Avalanche** is done via proof-of-uptime and proof-of-responsiveness.
 This mechanism has important consequences. In particular, since there is no “leader” accu-
 mulating rewards, there is no “rich-get-richer” compounding effects.

110 5 Transaction Economy

5.1 Fees Structure

The fee structure in the **Avalanche** platform carries several differentiating features that distinguish it from other existing and upcoming platforms.

Staker Fees. Unlike other protocols that pay all fees to the elected leader, such as in Bitcoin,
 115 in **Avalanche** fees are simply burned. Therefore, payment is global and for the good of the
 entire ecosystem. Fee burning increases scarcity of tokens in the system. The minting process
 offsets the transaction fee burning, therefore there is no danger of the system grinding to a
 long term halt due to gradual destruction of coins.

Transaction Costs. In **Avalanche**, transaction fees differ depending on the type of transac-
 120 tion. Instantiations of new subnetworks carry the heaviest fees. In contrast, other types of
 transactions, such as simple payments of \$AVAX, carry little cost. For other subnets, trans-
 actions pay fees in that subnet’s native token, as well as some amount in the \$AVAX token.
 A transaction native to a subnetwork may specify its own transaction fee structure, and it
 is up to the creator of the subnet to choose a fee structure that incentivizes validation for
 125 open, permissionless subnetworks.

Sliding Cost Function. Transaction fees carry a sliding-cost function. The fee is not set by the issuer of the transaction, but rather by a globally verifiable fee-function. As the congestion in the network increases, fees increase. At the end of some specific period of time, the function is recalculated to accommodate natural increases in transaction volume in the network.

Transaction Tiers. Unlike in a model such as Ethereum's, where every transaction invocation must pay some gas, **Avalanche** adopts a different model that incorporates two types of transaction processing mechanisms. All keys with positive account balances will be able to immediately interact with the platform, where the fees will be based on an allotment mechanism, functionally similar to a tiered payments model adopted in cloud computing platforms. Every transaction will name a sender address (i.e. the invoker), which will be checked for current invocation allotment. If the address still has free invocations left, the transaction does not have to carry any fees attached by the sender. Past a certain amount of calls, the sender will need to attach some fees based on the resources used to compute the transaction. Additionally, users may opt instead pay for their transactions using computation. To that end, future releases will support free frequency-limited transactions, which do not require fees in coins but require some pre-computation. Whenever a new transaction is generated, the user will compute and attach a valid PoW on the transaction, which can be checked by all other parties.

5.2 Spam Management

Although simple payments do carry fees, the value will be virtually zero. This, however, can lead to spam in the network. In future releases, to prevent congestion, each transaction carries with it a local PoW. The PoW is initially of low difficulty and therefore a transaction can be immediately issued with very little overhead. However, if a specific key generates a large amount of transactions in a short amount of time, each subsequent transaction will carry a larger amount of difficulty in its PoW puzzle. This mechanism works in conjunction with the fee burning.

References

1. AVA: Avalanche platform paper - architectural overview (2019), <https://files.avalabs.org/papers/platform.pdf>
2. AVA: Avalanche platform documentation (2025), <https://docs.avax.network>