On stake pool reward

Capital Stake Pool [CSP]*

November 14, 2020

*E-mail: csp@capitalstakepool.info.

Contents

1 Parameters definitions

Here we give the definitions of most important parameters for reward calculations:

- R: total reward per epoch to distribute among stake pools.
- k: desired number of active stake pools. From simulations is expected that at equilibrium only k stake pools will survive, i.e. they will not go out of business.
- s: fraction of the stake delegated by the pool owner(s) to the owned stake pool, aka the "pledge".
- a_0: a parameter to adjust the trade-off between decentralization and Sybil attacks resistance of the protocol. It acts as a weight for the pledge in the reward calculations. It can take every values in the range $[0, \infty)$.
- σ: fraction of the total stake of the stake pool, given by $s +$"fraction of delegated ADA".
- τ: the fraction of the produced ₳ in a epoch that goes to the treasury.
- T: total supply of ₳ generated until the last epoch.

2 Where R comes from?

Shelley era of Cardano will introduce "Ouroboros Praos" [3] consensus protocol. This is mainly a security improvement on "Ouroboros Classic" [2] where blocks assignation was public. It's not difficult to convince yourself that a public lottery opens the possibility to maliciously target a pool. However improved security has a cost: we have potential collisions between different pool's slots, aka "slot battles", i.e. two or more different stake pools can be selected for the same slot. This implies also that the number of blocks produced per epoch is not exactly predictable in advance also under ideal conditions, i.e. no forks and no missed blocks. However we can take as a reference value for the expected number of blocks produced per epoch

$$B = n_s \cdot f, \tag{1}$$

where n_s is the number of slots per epoch and $f \in [0, 1]$[1] is the number of active slots. For example, in the ITN (incentivized testnet) an epoch last one day and a slot is created every 2 seconds. So we have $n_s = 43200$. The f chosen for ITN is 0.1, so $B = 4320$.

Consider a virtual pot for all the rewards per epoch that should be distributed among stake pools (to simplify the analysis we will name "stake pool" also a private "stakeholder" or a "private stake pool"). There's a subtle but fundamental difference between a private stake pool and a stakeholder with undelegated stake: **the undelegated stake rewards goes to the treasury (they are said "unclaimed rewards")**. We can subdivide this pot into three additive components:

- Monetary expansion: the number of ₳ is capped to $4,5 \cdot 10^{10}$, but only $3,1 \cdot 10^{10}$ have been created in Byron era. You have to add another 112.483.745 ₳ to the total supply, that comes from ITN. So the residue of $\sim 1,4 \cdot 10^{10}$ ₳ will be distributed to treasury and to stake pools, as a part of the rewards, starting from Shelley era, until the cap value will be achieved (asymptotically).

- Transactions fees. To note that in the Byron era all transaction fees has been destroyed, so they can be effectively seen as a part of future rewards.

[1] To note that f enters also in the probability to get assigned a given block by the stake pool/stakeholder with relative stake σ: $p = 1 - (1 - f)^\sigma$. See [3] equation 1.

- Decayed deposits. This part is composed by: the cost for registering a staking key, the cost for registering a new stake pool and the cost to creating a new UTxO entry.

Monetary expansion is given by ([5], see section 5.4.3)

$$\min\{\eta, 1\} \cdot \rho \cdot (T_\infty - T), \tag{2}$$

where

$$\eta \doteq \frac{\overline{N}}{B}, \tag{3}$$

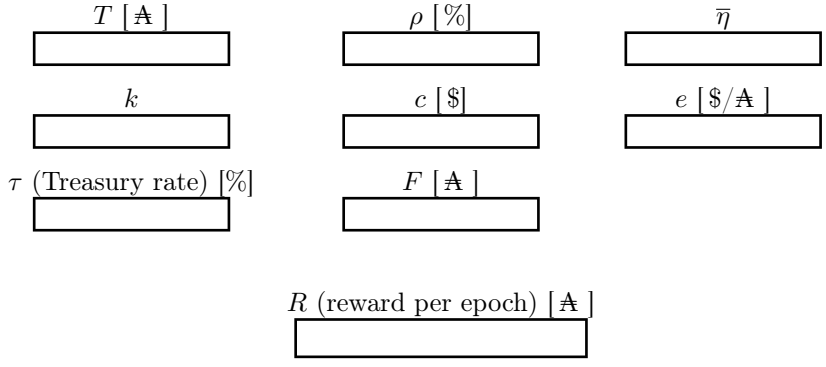
$\overline{N}$ is the number of blocks successfully produced in the epoch, $T_\infty = 4,5 \cdot 10^{10}$ is the hard cap for the total supply and T is the number of ₳ generated by the monetary expansion since the dawn of the Shelley era.

Keep in mind that the product $\min\{\eta, 1\} \cdot \rho$ is bound in the range $[0, 1]$, so that T_∞ is truly and hard cap for the total amount of ₳ ever generated. The official expansion rate is $\rho = 0,0022$ [1] has been announced the 25 June 2020 on the IOHK blog. It's worth to note also that the parameter η ensure that a "rational player" will cooperate with others stake pools to maximize the virtual pot, indeed we are assuming that the average agent/player has no harmful non-economic goals.

Total rewards per epoch:[2] The formula for the total rewards per epoch is given by

$$R = (1 - \tau) \cdot [F + \min\{\eta, 1\} \cdot \rho \cdot (T_\infty - T)] - \frac{k \cdot c}{73 \cdot e}, \tag{4}$$

however it will be full explained in the next section.



[2] *To use this calculator you have to open the pdf with Acrobat Reader or with a pdf reader that supports SpiderMonkey 1.8 JavaScript engine.*

Suggested values:

$T = 31.112.483.745$ ₳ $\quad \rho = 0,22\,\%$ $\quad \overline{\eta} = 1$

$k = 150$ $\quad c = 2.000\,\$$ $\quad e = 0,2\,\$/₳$ or $0,5\,\$/₳$ (if you have faith)

$\tau = 5\%$ $\quad F =$ "unknown", try with 0 ₳

Insert values without symbols or punctuation. For example, for T you have to insert in the corresponding field: 31112483745 or any integer number. Decimals are in the form $1.234\ldots$ to easy of insertion in portable devices.

Legend:

- T total supply of ₳ .
- ρ monetary expansion rate.
- $\overline{\eta}$ mean network performance.
- $k \geq 1$ desired number of active stake pools.
- c average cost to maintain a pool for one year.
- e expected price of one ₳ in \$.
- $\tau \in [0, 1]$ fraction of the reward pot that goes to treasury every epoch.
- F average transaction fee in ₳ paid in one epoch.

3 Explanation and calculation of monetary inflation rate ρ & R calculator

To calculate the monetary expansion rate per epoch in [5] the authors introduced a few more parameters:

- e: exchange rate of ADA in $\$/₳$,
- c: average cost to run a pool for one year in \$,
- F: average fees in ₳ paid in one epoch,
- r: expected ratio of rewards per year per staked ada,

- η: a parameter connected to the network performance, see the previous section (eq. 2).

Let's start from the definition of an epoch. In the terminology of Cardano, an epoch is a window of time in which all assigned blocks, assigned with the lottery mechanism of Ouroboros, should be created. Every epoch is subdivided in "slots". A fraction of the slots correspond to blockchain's blocks produced by the consensus algorithm via the stakeholders/stake pools. These are called "active slots". The length of an epoch is decided in the genesis block of the blockchain (however it's not excluded that in future things will change) and in the Byron era corresponds to 5 days. So, one year corresponds to:

$$\text{Number of epochs in one year} = \frac{365}{5} = 73.$$

Starting with T in a given epoch, after one epoch we have $R + T$. If we call T_n the number of ₳ ever generated until epoch n starting from T, we have [3]

$$T_n = \left(1 + \frac{R}{T}\right)^n \cdot T \tag{5}$$

and after one year

$$T_{73} = \left(1 + \frac{R}{T}\right)^{73} \cdot T.$$

From definition of the parameter r we have also

$$T_{73} = (1 + r) \cdot T, \tag{6}$$

so, by confronting (4) and (5), we arrive at

$$R = (\sqrt[73]{1 + r} - 1) \cdot T. \tag{7}$$

R is also connected to the virtual pot by the equation

$$R = (1 - \tau) \cdot [F + \min\{\eta, 1\} \cdot \rho \cdot (T_\infty - T)] - \frac{k \cdot c}{73 \cdot e}. \tag{8}$$

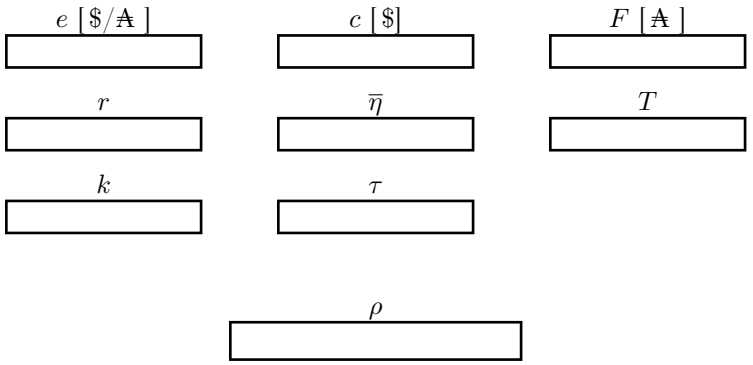
This formula means that the virtual pot $[F + \min\{\eta, 1\} \cdot \rho \cdot (T_\infty - T)]$, composed by the monetary expansion and by the fees, is reduced by the amount that goes to the treasury, so we have $(1 - \tau)$ times the original pot, and by the

[3] Actually we have $T_n = \prod_{k=0}^{n-1} \left(1 + \frac{R_k}{T_k}\right) \cdot T_0$, but using as approximation $\frac{R_k}{T_k} \approx \frac{R_0}{T_0}$ we recover (4).

operational costs in ₳ per epoch, $k \cdot c/73 \cdot e$. Confronting (6) and (7), we finally arrive to a formula for ρ

$$\rho = \frac{(\sqrt[73]{1+r} - 1) \cdot T + \dfrac{k \cdot c}{73 \cdot e} - (1-\tau) \cdot F}{(1-\tau) \cdot \min\{\eta, 1\} \cdot (T_\infty - T)}. \tag{9}$$

You can do your simulation of ρ [4]:



Suggested values:

$e = 0,5\,\$/₳ \quad c = 2.000\,\$ \quad F = 0\ ₳$

$r = 0,05 \quad \overline{\eta} = 0,9 \quad T = 31.112.483.745\ ₳$

$k = 150 \quad \tau = 0,05$

Insert values without symbols or punctuation. For example, for k you have to insert in the corresponding field: 1000 or any integer number. Decimals are in the form $1.234\ldots$ to easy of insertion in portable devices. **Legend:**

- e expected price of one ₳ in \$.
- c average cost to maintain a pool for one year.
- F average transaction fee in ₳ paid in one epoch.
- r expected ratio of rewards per year per staked ada.

[4] *To use this calculator you have to open the pdf with Acrobat Reader or with a pdf reader that supports SpiderMonkey 1.8 JavaScript engine.*

- $\overline{\eta}$ expected value of the parameter that measures network performance (see previous section).
- T total supply of ₳ produced until last epoch.
- $k \geq 1$ desired number of active stake pools.
- $\tau \in [0, 1]$ fraction of the reward pot that goes to treasury every epoch.

For a discussion on the temporal evolution of the monetary expansion, see the appendix: How many epochs until $T \approx T_\infty$?.

4 Optimal reward per pool

From [4] we know that optimal reward per pool is given by the formula

$$f(s, \sigma) \doteq \frac{R}{1 + a_0} \cdot \left(\sigma' + s' \cdot a_0 \cdot \frac{\sigma' - s' \cdot \frac{z_0 - \sigma'}{z_0}}{z_0} \right), \tag{10}$$

where has been introduced the new parameters:

- $z_0 \doteq 1/k$,
- $\sigma' \doteq \min\{z_0, \sigma\}$,
- $s' \doteq \min\{z_0, s\}$.

Let's see some limiting cases to clarify the meaning of the formula.

4.1 $a_0 = 0$:

This is the case of ITN, where the pledge mechanism was not introduced (see figure 1):

$$f(s, \sigma) = R \cdot \sigma' = \begin{cases} R \cdot z_0, & \text{if } \sigma \geq z_0 \\ R \cdot \sigma, & \text{if } \sigma < z_0 \end{cases} \tag{11}$$

Reward simply saturate to the maximum R/k and there's no incentive to delegate pool owner stake to a single stake pool. For an exemplar case of this behavior, see: 1PCT.

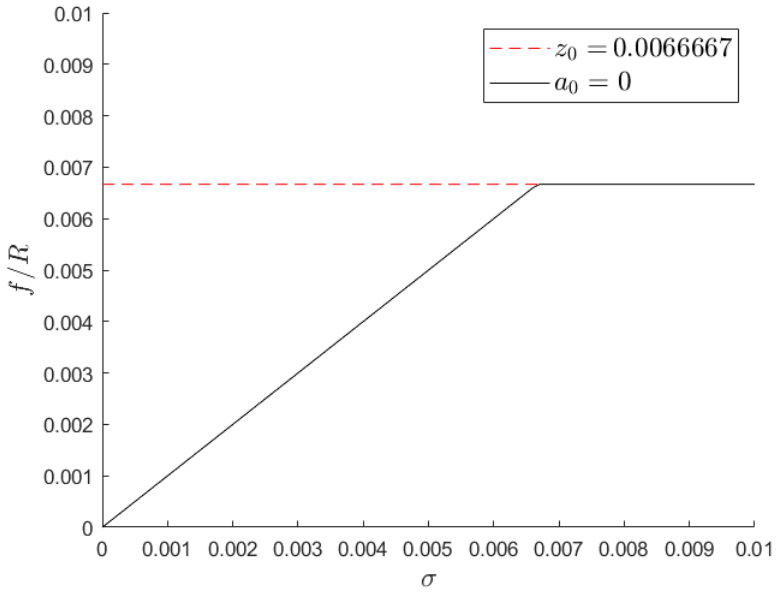


Figure 1: $k = 150$ and $a_0 = 0$.

4.2 $a_0 \neq 0$ and $s \geq z_0$:

This is the case of a private stake pool (see figure 2):

$$f(s,\sigma) = \frac{R}{1+a_0} \cdot \left(z_0 + z_0 \cdot a_0 \cdot \frac{z_0 - z_0 \cdot \frac{z_0 - z_0}{z_0}}{z_0} \right) = R \cdot z_0. \tag{12}$$

Also in this case maximum reward is capped. This maximum maximum (sorry for word play) can be achieved only at expense of delegators participation (other then owner(s)).

4.3 $a_0 \neq 0$ and $s = 0$:

In this case stake pool owner can at most achieve the inferior maximum if she/he will survive to competition (see figure 3):

$$f(s,\sigma) = \frac{R}{1+a_0} \cdot \sigma' = \begin{cases} \frac{R}{1+a_0} \cdot z_0, & \text{if } \sigma \geq z_0 \\ \frac{R}{1+a_0} \cdot \sigma, & \text{if } \sigma < z_0 \end{cases} \tag{13}$$

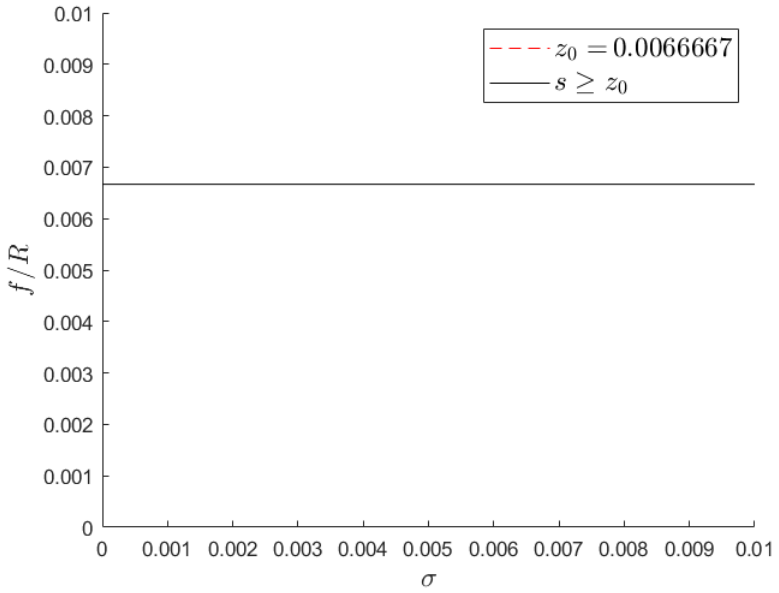


Figure 2: $k = 150$ and $a_0 = 0.3$.

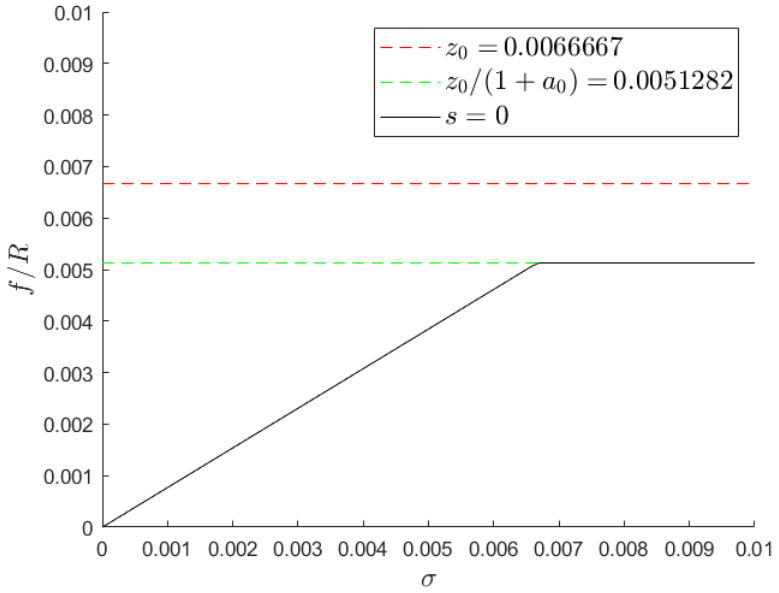


Figure 3: $k = 150$ and $a_0 = 0.3$.

4.4 $a_0 \neq 0$ and $0 < s < z_0$:

This will be surely the most frequent situation that will occur. When equilibrium will be achieved, i.e. $\sigma = z_0$, cap value will be (see figure 4):

$$f(s, z_0) = \frac{R}{1 + a_0} \cdot \left(z_0 + s \cdot a_0 \cdot \frac{z_0 - s \cdot \dfrac{z_0 - z_0}{z_0}}{z_0} \right) = \frac{R}{1 + a_0} \cdot (z_0 + s \cdot a_0). \quad (14)$$

Simply put, maximum reward achievable is directly proportional to pledge. Note that the sum of rewards for every stake pool is obviously less or equal to R:

$$\sum_{i=1}^{k} f(s_i, z_0) = \sum_{i=1}^{k} \frac{R}{1 + a_0} \cdot (z_0 + s_i \cdot a_0) = \ldots \quad (15)$$

$$= \frac{R}{1 + a_0} \cdot k \cdot z_0 + \frac{R}{1 + a_0} \cdot a_0 \cdot \sum_{i=1}^{k} s_i \leq R.$$

Using that

$$k \cdot z_0 = 1 \quad \text{and} \quad \sum_{i=1}^{k} s_i \leq 1.$$

The difference $R - \sum_{i=1}^{k} f(s_i, z_0)$ goes to the treasury.

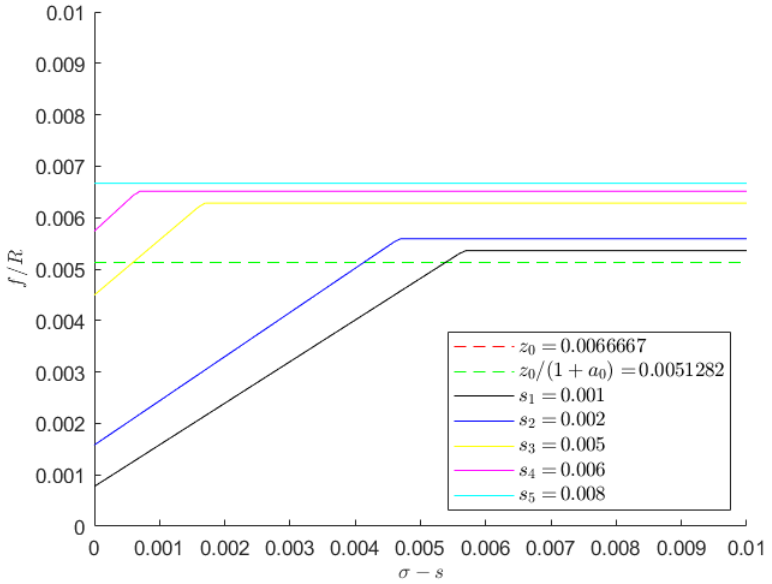


Figure 4: $k = 150$ and $a_0 = 0.3$.

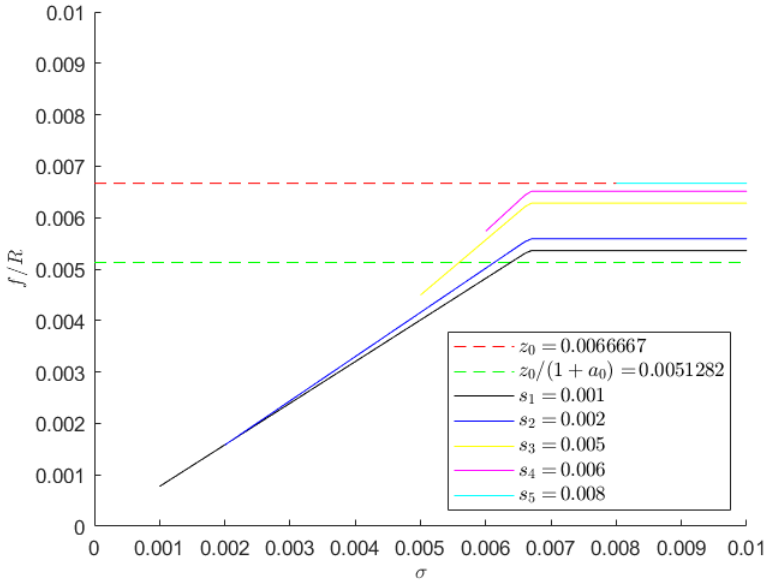


Figure 5: $k = 150$ and $a_0 = 0.3$. Shifted origins to emphasize effect of delegations.

5 Optimal reward calculator

Optimal reward formula is rewritten for convenience

$$f(s,\sigma) \doteq \frac{R}{1+a_0} \cdot \left(\sigma' + s' \cdot a_0 \cdot \frac{\sigma' - s' \cdot \dfrac{z_0 - \sigma'}{z_0}}{z_0} \right).$$

Please note that at odds with the formula, in the calculator fields s and σ are in ₳ . To recover the original definition of s and σ you have to divide them by T. Write your values in the corresponding forms to see results [5]:

a_0 (pledge factor) | k (desired pools) | s (pledge) [₳]

Pool saturation [%] | R (epoch rewards) [₳] | T (total supply) [₳]

Fixed fee/costs | Variable fee [%] | Your stake [₳]

f (optimal reward) [₳] | Delegator share [₳] | Pool owner(s) share [₳]

Optimal desirability

ROS [%]

Suggested values:

$a_0 = 0,3$ $k = 150$ $s = 100.000$ ₳

$R = 31.000.000$ ₳ $T = 31.112.483.745$ ₳ Fixed fee/costs = 340 ₳

Variable fee/stake pool margin = 3%

Your stake = this depends on how many ₳ you want to delegate

Insert values without symbols or punctuation. For example, for R you have to insert in the corresponding field: 27497282 or any integer number. Decimals are in the form 1.234... to easy of insertion in portable devices.

[5] *To use this calculator you have to open the pdf with Acrobat Reader or with a pdf reader that supports SpiderMonkey 1.8 JavaScript engine.*

Legend:

- $a_0 \in [0, \infty)$ = pledge influence factor.
- $(k \geq 1)$ = desired number of active stake pools.
- s = stake pool owner(s) pledge.
- R = total rewards per epoch.
- T = total supply of ₳ .

6 Pledge advantage definition & calculator

We introduce a new metric to make a comparison between different pledge levels, we name it "pledge advantage" P_a [6]:

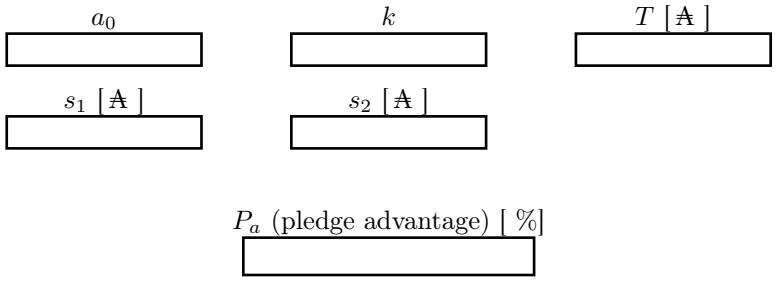
$$P_a(s_1, s_2) \doteq \frac{a_0 \cdot (s_2' - s_1')}{z_0 + s_1' \cdot a_0} \leq \frac{s_2'}{s_1'} - 1. \tag{16}$$

This measures the advantage of a pledge s_2 over a pledge s_1 assuming that either the hypothetical pools are saturated. For example, assuming

$a_0 = 0,3 \quad k = 150 \quad T = 31.112.483.745$ ₳

$s_1 = 0$ ₳ $\quad s_2 = 1.000.000$ ₳

we have $P_a \approx 0,145\%$. This means that if the first pool's reward per epoch is 50.000 ₳ , then the second pool's reward will be $50.072,5$ ₳ , i.e. ca $1,0014$ times the first pool's reward. [7]



[6] This is a relative difference: $[f(s_2, z_0) - f(s_1, z_0)]/f(s_1, z_0)$.

[7] *To use this calculator you have to open the pdf with Acrobat Reader or with a pdf reader that supports SpiderMonkey 1.8 JavaScript engine.*

Suggested values:

$a_0 = 0,3 \quad k = 150 \quad T = 31.112.483.745$ ₳

Insert values without symbols or punctuation. For example, for k you have to insert in the corresponding field: 150 or any integer number. Decimals are in the form $1.234\ldots$ to easy of insertion in portable devices.

7 Desirability definition & calculator

Rewards sharing mechanism includes a formula to rank stake pools that will be included in the wallet specifications. The aim of this ranking is to incentivize the "non-myopic" strategy between stake pools/delegators. This is part of the complex system that constitutes Cardano and, theorically, this will guarantee an healthy growth. Let's introduce the formula for the desirability of a stake pool

$$d(c, m, s, \overline{p}) = \begin{cases} 0 & \text{if} \quad \hat{f} \leq c \\ (\hat{f} - c) \cdot (1 - m) & \text{otherwise} \end{cases} \tag{17}$$

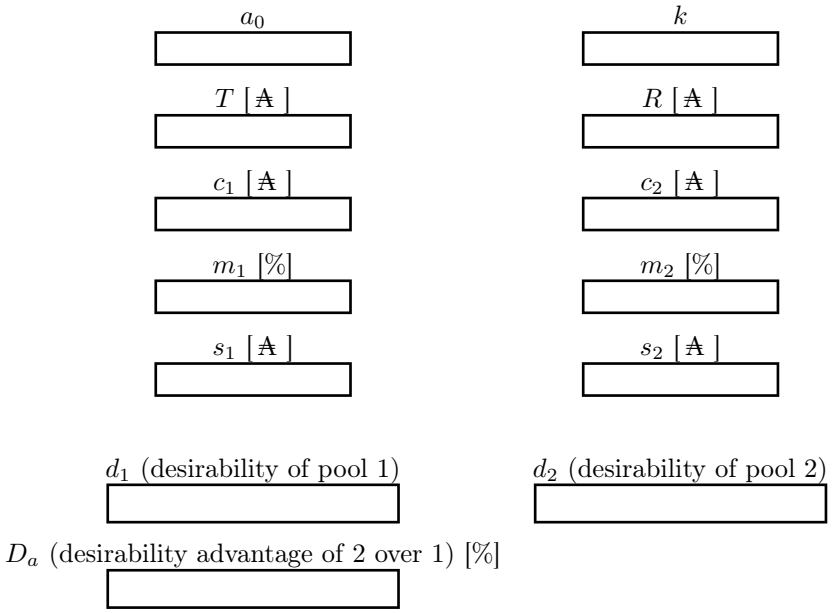
Where the desirability d is a function of: the actual reward per epoch $\hat{f} = \overline{p} \cdot f$, i.e. the optimal reward scaled by the actual performance factor $\overline{p}$ (see next section), fixed costs c and variable fee m. For the sake of simplicity, we will assume an optimal performance $\overline{p} = 1$, an unrealistic hypothesis, and a satured pool, so that we can write

$$d(c, m, s, 1) = \begin{cases} 0 & \text{if} \quad f \leq c \\ \left[\dfrac{R}{1 + a_0} \cdot (z_0 + s' \cdot a_0) - c\right] \cdot (1 - m) & \text{otherwise} \end{cases} \tag{18}$$

In the wallet every stake pool will be ranked by desirability, let's call this rank r. In the long run, it's expected that only those stake pools with $r \leq k$ will survive. We introduce a new metric for desirability of a public stake pool, similar to pledge advantage

$$D_a(c_1, m_1, s_1; c_2, m_2, s_2) = \begin{cases} \text{undefined} & \text{if} \quad f_2 \leq c_2 \quad \text{and} \quad f_1 \leq c_1 \\ -1 & \text{if} \quad f_2 \leq c_2 \quad \text{and} \quad f_1 > c_1 \\ \infty & \text{if} \quad f_2 > c_2 \quad \text{and} \quad f_1 \leq c_1 \\ \dfrac{1 - m_2}{1 - m_1} \cdot \dfrac{f_2 - c_2}{f_1 - c_1} - 1 & \text{otherwise} \end{cases} \tag{19}$$

note also that we are assuming $m_1 < 1$ and $m_2 < 1$, i.e. we are comparing public stake pools. The meaning of D_a is: **desirability advantage of** 2 **over** 1. You can do your computations [8]



Suggested values:

$a_0 = 0,3 \quad k = 150 \quad T = 31.112.483.745$ ₳

$R = 27.497.282$ ₳ $\quad c = 340$ ₳

Insert values without symbols or punctuation. For example, for k you have to insert in the corresponding field: 150 or any integer number. Decimals are in the form $1.234\ldots$ to easy of insertion in portable devices.

[8] *To use this calculator you have to open the pdf with Acrobat Reader or with a pdf reader that supports SpiderMonkey 1.8 JavaScript engine.*

8 Pool performance and actual reward

Optimal reward doesn't consider the effect of missed blocks due to an off-line node, to soft forks and to collisions between slots. To have pool's actual reward, we have to consider also a performance metric, defined as

$$p \doteq \frac{n}{\max\{1, N\}}. \tag{20}$$

Where n is the number of blocks produced by the pool in the epoch and N is the number of blocks assigned to the pool at the start of the epoch. We said in the section "where R comes from?" that with Ouroboros Praos consensus protocol is not possible to publicly know the number of blocks assigned to the pool. So, in order to assign a performance metric to a stake pool, we have to renounce to the aforementioned metric, which uses unobservable quantities, and rely on an apparent performance metric, defined as

$$\overline{p} \doteq \frac{\beta}{\sigma} = r_e \cdot p. \tag{21}$$

r_e is a factor of proportionality between actual performance and apparent performance. New parameter β is the fraction of blocks produced by the pool in the epoch

$$\beta \doteq \frac{n}{\max\{1, \overline{N}\}}, \tag{22}$$

$\overline{N}$ being the number of blocks added to the blockchain in the epoch and σ is pool's fraction of stake (see previous sections). Using a little of algebra, we find that

$$r_e = \frac{N}{\sigma \cdot \max\{1, \overline{N}\}}, \tag{23}$$

so we conclude that r_e is a quantity sensible to the various random effects due to leader election (included in N and $\overline{N}$), forks (included in $\overline{N}$) and so on. With this definitions we can write the actual reward of a stake pool as

$$\hat{f}(s, \sigma, \overline{p}) \doteq \overline{p} \cdot f(s, \sigma) \leq f(s, \sigma). \tag{24}$$

From the (20) follows that

$$\sum_{i=1}^{k} \hat{f}(s_i, z_0, \overline{p}_i) = \sum_{i=1}^{k} \frac{R \cdot \overline{p}_i}{1 + a_0} \cdot (z_0 + s_i \cdot a_0) \leq R. \tag{25}$$

From [5] we know that the difference

$$R - \sum_{i=1}^{k} \hat{f}(s_i, z_0, \overline{p}_i) \tag{26}$$

goes to the treasure.

A How many epochs until $T \approx T_\infty$?

To simplify notation we introduce a new variable

$$\Delta T_k = T_\infty - T_k. \tag{27}$$

We know that the monetary expansion goes as

$$T_{k+1} = T_k + \min\{\eta_k, 1\} \cdot \rho \cdot \Delta T_k,$$

where η_k measures the network performance and it's obviously not predictable. The previous formula can be rewritten in terms of ΔT_k only

$$\Delta T_{k+1} = (1 - \min\{\eta_k, 1\} \cdot \rho) \cdot \Delta T_k. \tag{28}$$

However, if we assume as approximation an average suboptimal performance $\eta_k \approx \eta$, $\eta \leq 1$, we can write

$$\Delta T_{k+1} \approx (1 - \eta \cdot \rho) \cdot \Delta T_k = (1 - \eta \cdot \rho)^{k+1} \cdot \Delta T_0 = e^{-\frac{k+1}{\tau}} \Delta T_0, \tag{29}$$

where

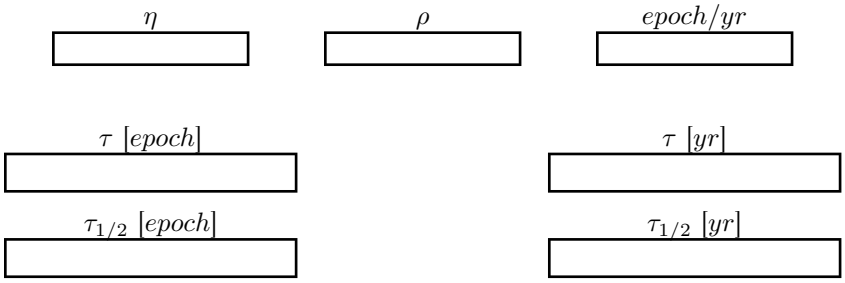
$$\tau \doteq -[\ln(1 - \eta \cdot \rho)]^{-1} \tag{30}$$

is the mean lifetime of the monetary expansion. The intuitive meaning of τ is the number of epochs necessary to reduce ΔT_0 to ca 36% of its value. Actually epochs are integer, so we have to round τ. If you prefer, you can use the "half-life" defined as $\tau_{1/2} = \ln 2 \cdot \tau \approx 0,69 \cdot \tau$, which corresponds to the time necessary to reduce ΔT_0 to ca 50% of its original value (see table 1). For example, given $\eta = 0,9$ and $\rho = 0,0022$ (see fig. 6)

$$\tau \approx 505\,\text{epochs} \quad \text{and} \quad \tau_{1/2} \approx 350\,\text{epochs}.$$

Using that 73 epochs = 1 year

$$\tau \approx 7\,\text{years} \quad \text{and} \quad \tau_{1/2} \approx 4,8\,\text{years}.$$



Suggested values:

$\eta = 0,9 \quad \rho = 0,0022 \quad epoch/yr = 73$

Insert values without symbols or punctuation. Decimals are in the form $1.234\ldots$ to easy of insertion in portable devices.

Number of $\tau_{1/2}$ elapsed	Fraction remaining	Percentage remaining
0	$\frac{1}{1}$	100%
1	$\frac{1}{2}$	50%
2	$\frac{1}{4}$	25%
3	$\frac{1}{8}$	$12,5\%$
4	$\frac{1}{16}$	$6,25\%$
5	$\frac{1}{32}$	$3,125\%$
6	$\frac{1}{64}$	$1,5625\%$
7	$\frac{1}{128}$	$0,78125\%$
$\vdots$	$\vdots$	$\vdots$
n	$\frac{1}{2^n}$	$100\%/2^n$

Table 1: Fraction/percentage of remaining monetary expansion in function of $\tau_{1/2}$.

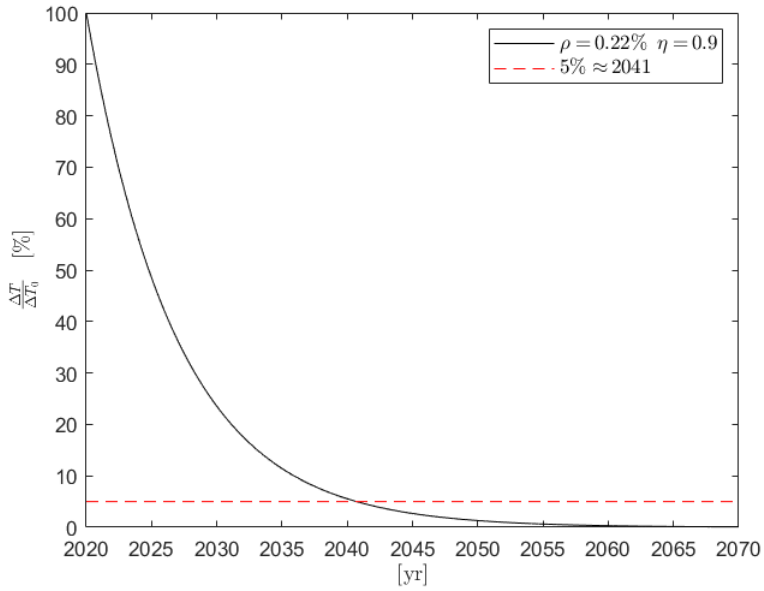


Figure 6: Monetary expansion: $\rho = 0,22\%$ and $\eta = 0,9$.

9 Acknowledgments

My acknowledgments goes to LordWotton [RIOT-VAULT-APEX] for the precious suggestions to improve the article and to feqifei [CALM-COOL] for checks of formulas and typos. [9]

[9] If you found this article useful, feel free to offer me a coffee:
`addr1qyxpmx9uq9mvpdkpnm9vdql05ss2yjuzg374cad4pdz7sueylxgsue6m4lsdpg9qamp5c5m7dkvezstyzkhwntuvhk7qv5yf32`

References

[1] Lars Brünjes: Iterating for growth with IOHK research: Building key values into the Cardano ecosystem. https://iohk.io/en/blog/posts/2020/06/25/iterating-for-growth-with-iohk/.

[2] Aggelos Kiayias, Alexander Russell, Bernardo David & Roman Oliynykov: "Ouroboros: A Provably Secure Proof-of-Stake Blockchain Protocol". DOI: https://doi.org/10.1007/978-3-319-63688-7_12.

[3] Bernardo David, Peter Gaži, Aggelos Kiayias & and Alexander Russelly: "Ouroboros Praos: An adaptively-secure, semi-synchronous proof-of-stake blockchain". DOI: https://doi.org/10.1007/978-3-319-78375-8_3.

[4] Lars Brünjes, Aggelos Kiayias, Elias Koutsoupias & Aikaterini-Panagiota Stouka: "Reward Sharing Schemes for Stake Pools". URL: arXiv:1807.11218.

[5] Philipp Kant, Lars Brünjes & Duncan Coutts: Design Specification for Delegation and Incentives in Cardano. https://hydra.iohk.io/build/2339958/download/1/delegation_design_spec.pdf