Alflect Finance

Reflect Finance explained and beyond

Simon Tian, PhD

08/04/2021

Abstract

`Reflect Finance` introduced a process-based mechanism of distributing a percentage of transaction amount as fees to all token holders in a transactionless manner. The core idea relies on the interplay of a dual-space accounting system. This article is going to lay out the mathematical details of this idea and make an improvement by reducing the system from two spaces to one. This paves the way for the algorithm-based redistribution mechanism in `Alflect Finance`.

Introduction

We first illustrate the idea in `Reflect Finance` with Magic Kingdom R (`MK_R`) and Magic Kingdom T (`MK_T`). The currencies of the two kingdoms have exchange rates strictly tied to the relative size of the two economies. Besides, `MK_R` has an interesting monetary policy that is for every transaction, a certain percentage will be burned and as a result, the size of this economy is reduced by the burned amount. Alice, Bob and Charlie are citizens of both kingdoms and figure out if they make pair-wise transactions in `MK_T`, the third person will not get any benefit. However, if they move tokens from `MK_T` to `MK_R` first, and make transactions there, and exchange tokens back to `MK_T`, everyone can make passive incomes in `MK_T`. This is the basic idea behind `Reflect Finance`.

In order to be mathematically precise, the rest of this article will make concepts more abstract by considering the two kingdoms as `r-space`, and `t-space` (read as reflected space and true space). `t-space` is the true space where people live in and are familiar with. `r-space` is created only for assisting purposes. We denote the total volume or amount of tokens in the two spaces as T_r and T_t, respectively. They correspond to the `_rTotal` and `_tTotal` in the `Reflect Finance` Solidity code.

If a transaction is sent in `t-space` from `A` to `B` and fees are wished to be distributed among all token holders, it would be a very tedious and costly task to update the balance of every single token address. However, as illustrated above, if the transaction is made in `r-space` and reflected back to `t-space`, by updating the exchange rate or relative size of the two spaces $r = \dfrac{T_r}{T_t}$, token holders' balances in `t-space` are instantaneously updated with fees without sending additional transactions. In the following section, more details will be provided.

Methodology

Let b_{A_t}, b_{B_t}, ... denote the balance of accounts `A`, `B`, ... in `t-space`, respectively, and let b_{A_r}, b_{B_r}, ... denote the corresponding balances in `r-space`. At any given moment, we have $b_{A_r} = rb_{A_t}$. The exchange rate $r > 0$ is constantly updated and plays the key role in this mechanism. If we denote the size of a transaction as $S_t > 0$ with a fee $f_t > 0$, then the corresponding quantities in `r-space` would be $S_r = rS_t > 0$ and $f_r = rf_t > 0$.

For a transaction from `A` to `B` in `r-space`, the amount of $S_r - f_r$ is added to the balance of B in `r-space`, i.e., $b^+_{B_r} = b_{B_r} + S_r - f_r$, where $b^+_{B_r}$ denotes the updated balance of `B` after this transaction in `r-space`. Notice, since a fee f_r is charged and burned, the volume of `r-space` is shrank because of it, the exchange rate is changed as well, from $r = \frac{T_r}{T_t}$ to $r^+ = \frac{T_r - f_r}{T_t}$. The new balance of `B` in `t-space` is therefore:

$$b^+_{B_t} = \frac{b^+_{B_r}}{r^+} = \frac{b_{B_r} + S_r - f_r}{r^+} = \frac{rb_{B_t} + rS_t - rf_t}{r}\frac{T_t}{T_t - f_t} = (b_{B_t} + S_t - f_t)\frac{T_t}{T_t - f_t}$$

The last component in the above equation shows the process of reflecting fees over all accounts for account `B` is equivalent with sending a normal transaction of size S_t in `t-space` minus fee f_t and multiplying the new balance by an earning factor $m = \frac{T_t}{T_t - f_t} > 1$. This earning factor is only dependent upon the total volume of `t-space` and this transaction fee in `t-space`. The same pattern can be observed for accounts `A` and `C` as well. Details will be given soon below.

Would the total size of `t-space` be changed? The answer is no for a normal transaction like the one given above. To be more specific, before the transaction, the balance of accounts `A`, `B`, and `C` are b_A, b_B and b_C, respectively. Since only three accounts are in `t-space`, $T_t = b_A + b_B + b_C$. After the transaction, the balances are changed to $(b_A - S_t)m$, $(b_B + S_t - f_t)m$ and $b_C m$, respectively. The sum of the three is

$$(b_A - S_t + b_B + S_t - f_t + b_C)m = (b_A + b_B + b_C - f_t)\frac{T_t}{T_t - f_t} = T_t.$$

The reflection does not change the size of `t-space` at all.

Now to make the notations more generic, we introduce an index $i = 1, 2, ...$ to count transactions, then $b^{(i)}_{B_t}$ and $b^{(i)+}_{B_t}$ denote the balance of `B` in `t-space` before and after the i-th transaction taking place. A natural question would be if the same pattern can be drawn for future transactions. We start from transaction i and check the balances after transaction $i+1$.

In the i-th transaction, for `A`, $b^{(i)}_{A_t}$ becomes $b^{(i)+}_{A_t} = \left(b^{(i)}_{A_t} - S^{(i)}_t\right) m^{(i)}$, where $m^{(i)} = \frac{T^{(i)}_t}{T^{(i)}_t - f^{(i)}_t}$; for `B`, $b^{(i)}_{B_t}$ becomes $b^{(i)+}_{B_t} = \left(b^{(i)}_{B_t} + S^{(i)}_t - f^{(i)}_t\right) m^{(i)}$, and for `C`, $b^{(i)}_{C_t}$ becomes $b^{(i)+}_{C_t} = b^{(i)}_{C_t} m^{(i)}$, after the transaction. Lastly, $r^{(i)+} = \frac{T^{(i)+}_r}{T^{(i)+}_t}$.

Then suppose the $(i+1)$-th transaction is again made from `A` to `B` of size $S^{(i+1)}_r$ with fee $f^{(i+1)}_r$ in `r-space`, then in `t-space`, since $r^{(i+1)+} = \frac{T^{(i+1)+}_r}{T^{(i+1)+}_t} = \frac{T^{(i+1)}_r - f^{(i+1)}_r}{T^{(i+1)}_t}$,

$$b^{(i+1)+}_{B_t} = \frac{b^{(i+1)+}_{B_r}}{r^{(i+1)+}} = \frac{b^{(i+1)}_{B_r} + S^{(i+1)}_r - f^{(i+1)}_r}{T^{(i+1)}_r - f^{(i+1)}_r} T^{(i+1)}_t = \left(b^{(i+1)}_{B_t} + S^{(i+1)}_t - f^{(i+1)}_t\right) m^{(i+1)},$$

where $m^{(i+1)} = \frac{T^{(i+1)}_t}{T^{(i+1)}_t - f^{(i+1)}_t}$. Notice for normal transactions, T_t does not change, therefore $T^{(i)}_t = T^{(i+1)}_t$, and $m^{(i)} = \frac{T^{(i)}_t}{T^{(i)}_t - f^{(i)}_t}$ and $m^{(i+1)} = \frac{T^{(i)}_t}{T^{(i)}_t - f^{(i+1)}_t}$. An intuitive conclusion can be drawn right away. If the total volume of `t-space` is unchanged, earning factors $m^{(i)}$ is simply the ratio between the total volume and the volume minus the transaction fee in that transaction.

Similarly, we could have the new balance for `A` and `C` given below:

$$b^{(i+1)+}_{A_t} = \left(b^{(i+1)}_{A_t} - S^{(i+1)}_t\right) m^{(i+1)},$$

$$b_{C_t}^{(i+1)+} = b_{C_t}^{(i+1)} m^{(i+1)}.$$

The balance of `C` provides the critical insight to `Reflect Finance`. The fees are redistributed to each account as if the balance is simply multiplied with the earning factor m for each transaction. This is the foundation of a new fee redistribution algorithm introduced in the second part of this article.

Properties of Earning Factors

- **Independence** A transaction made from `A` to `B` has the same effect as the one made from `C` to `D`, as long as the size and fee rate of the transaction are the same. The earning factors are independent upon trading pairs.
- **Interchangeable** The net effect of two earning factors remains the same if the two transactions are switched in order, i.e., $m * m' = m' * m$, i.e., for an account that is not involved in a transaction, the participating accounts and transaction orders are irrelevant in the net earning factor.
- **Proportionality** Fees are distributed proportionally to the balance of all accounts after the transaction has been made, i.e., $f_A/f_B = b_A^+/b_B^+$, where f_A and f_B denote the fees earned in a transaction and b_A^+ and b_B^+ denote the balances of `A` and `B` after the transaction but before having the earning factors applied on the balances.
- **Unmergeability** In the simplest form, the net earning factor of two transactions is not the same with a single transaction of the size as sum of the two. To see this, suppose two consecutive transactions have earning factors $m^{(1)} = \frac{T}{T-a}$ and $m^{(2)} = \frac{T}{T-b}$, where $a, b > 0$ are both fees. Since fees are linearly additive, the earning factor of the combined transaction with fee $a+b$ would be $m = \frac{T}{T-a-b}$. The resulting earning factors in these two scenarios are $\frac{T}{T-a}\frac{T}{T-b}$ and $\frac{T}{T-a-b}$, respectively. The difference between the two is then $-\frac{T(T^2+ab)}{(T-a)(T-b)(T-a-b)}$, which is always smaller than 0. Therefore, the net earning factor of two smaller transactions is smaller than that of a bigger transaction.

Practical Issues

Excluded Accounts

If some accounts are excluded from accepting fees, it is equivalent with excluding the balance of these accounts from the total economy altogether. To see this, suppose Dave is excluded from fees, it is as if Dave does not have the citizenship of `MK_R` and cannot exchange currencies between two kingdoms. His activities do not have any meaningful impact on the economy of `MK_R`, and therefore, he can be excluded from both economies. The mathematical details of excluding accounts are given below.

For account `X`, we have $r = \frac{T_r}{T_t}$ before excluding `X`, and $r' = \frac{T_r - b_{X_r}}{T_t - b_{X_t}}$ after it. Notice $T_r - b_{X_r} = rT_t - rb_{X_t}$, therefore, $r' = r$. Similarly for an inclusion, $r = \frac{T_r}{T_t}$, and $r' = \frac{T_r + b_{X_r}}{T_t + b_{X_t}}$. Since $T_r + b_{X_r} = rT_t + rb_{X_t}$, $r' = r$. This shows exclusion and inclusion of an account do not change the exchange rate.

For the earning factor $m = \frac{T_t}{T_t - f_t}$, an exclusion reduces T_t by the balance of the excluded account `X`, that is to say, after the exclusion, the earning factor is $m' = \frac{T_t - b_{X_t}}{T_t - b_{X_t} - f_t} = \frac{T_t'}{T_t' - f_t}$, where T_t' is the reduced volume of `t-space`. This shows the exclusion of an account `X` does affect the earning factor. Given $b_{X_t} \geq 0$,

we can conclude $m \leq m'$ due to the basic inequality property that $\frac{a}{b} <= \frac{a-x}{b-x}$, for $a, b > 0$, $x \geq 0$, $a-x > 0$ and $b - x > 0$. This is aligned with the intuition that if one less account is to earn fees, if all the other conditions are the same, the other accounts should earn more. Similarly, the inclusion of account `X` will create an opposite effect, given $m^* = \frac{T_t + b_{X_t}}{T_t + b_{X_t} - f_t} = \frac{T_t^*}{T_t^* - f_t} \leq m$, where T_t^* is the increased volume of `t-space`. This is also an intuitive result. Since one more account comes to share fees, if all the other conditions are the same, all the other accounts earn less.

Notice that the same conclusions can be applied to minting and burning tokens. Minting and burning are directly modifying the total size of the economy.

Another question that can be asked is if the size of the economy is increased (or decreased) by Δ, how much the new fee should be if the earning factor remains the same as if there is no change. Suppose the earning factors are m and m^* before and after the change, and the corresponding fees are f and f^*, respectively, then we have $m = \frac{T}{T-f}$ and $m^* = \frac{T+\Delta}{T+\Delta-f^*}$. Let $m = m^*$, then we have

$$\frac{T}{T-f} = \frac{T+\Delta}{T+\Delta-f^*},$$

$$f^* = f\frac{T+\Delta}{T}, \text{ or } \frac{f}{f^*} = \frac{T}{T+\Delta}.$$

The last equation shows if the economy is increased by Δ, a higher fee $f^* - f = \frac{\Delta}{T}f$ is needed to keep the earning factor the same. Likewise, if the economy is decreased by Δ, the amount of fee can be reduced to $f - f^* = \frac{\Delta}{T}f$.

For an excluded account, its earning factor relative to the running earning factor should always be 1, i.e., no additional earning is made across transactions.

Numbers on Return Rates

The passive incomes are made by accruing small fees. To illustrate the effect of reflection, numerical results are given based on the parameters in `RFI` smart contract, in which, $T_t = 1e7$, and f_t is 1% of the amount of every transaction S_t. Suppose the size of a typical transaction be $1,000$ tokens, then the earning factor is $m = \frac{1e7}{1e7 - 1000 * 0.01} = 1.000001$, roughly a 0.0001% increase per such transaction and a 1% increase after $10,000$ such transactions. The total number of transactions so far for this token is $72,455$. That is translated into a 7.51% return for 258 days in existence, and 10.36% annualized. If you have $10,000$ tokens in your wallet, after a year, that number would go up to $11,036$.

Other Fees

The core of `Reflect Finance` is about redistributing fees across all token holders. Other fees such as marketing fees, liquidity fees, burning fees, and etc., do not get redistributed across token holders' accounts. Therefore, the framework laid out in this article does not apply to these fees.

Volume of `r-space`

The two particular lines of `RFI` solidity code caught many people's attention, and the second line is what make people start thinking in terms of spaces, since the total of one total is one type of residual of the other.

```
uint256 MAX = ~uint256(0);
```

```
_rTotal = (MAX - (MAX % _tTotal));
```

Questions that can be asked include 1) should `r-space` be bigger or smaller than `t-space`? 2) if bigger, how much bigger? How is it determined? 3) Would another initial quantity such as `uint256 MAX = uint256(~uint128(0));` work just as fine?

These can be easily answered based on the mathematical insights drawn above. In fact, the size of `r-space` does not matter at all. The earning factors only depend on the size of `t-space` and the transaction fee in `t-space`. It is this very insight that motivates the design of a new mechanism with one space.

A Single Space Implementation

The above dual-space interplay does provide a workable process-based mechanism of redistributing fees over accounts, however, lack of intuition in logic and high gas costs, especially when a few accounts are excluded from fees, leave a significant room for improvement. A natural improvement is to implement the mechanism with a single space. `Alflect Finance` is going to introduce such an idea that is intuitive, mathematically equivalent with the current approach and having a much improved readability in code.