Bancor based utility-enhanced token protocol

As a series of mainnets for high TPS blockchain infrastructure are being implemented in 2018, DApp development is becoming more and more popular recently. However, there is no corresponding industry standard for these DApp utility tokens so far. In this whitepaper, we purpose a kind of new utility token that helps DApp developers design and issue their own utility tokens.

On the Ethereum network, quite much efforts are taken to provide strong liquidity for these traditional ERC20 tokens. However, some DApps do not necessarily require strong liquidity. When DApps are on the different economic scale or in different development stages, different token demands need to be met. Free capital flow, fixed exchange rate, independent token policy -- these three requirements need to be balanced constantly for a DApp utility token. And the performance of the entire crypto currency market starting from 2017 has also proven that the designed strong liquidity does not help DApp developers or users much. It's only good for the market speculators and gamblers for the moment.

In view of the various problems caused by the strong token liquidity, we believe that utility tokens should focus on the token usage scenario and dividends for long-term investors. Therefore, based on the "Bancor Protocol", in this whitepaper, we introduce the token consumption protocol and dividend protocol along with the option protocol and a new trading fee algorithm to make this utility token in line with the current DApp development trend.

1. Token consumption protocol

The core of the Bancor-based token protocol is to consume the tokens held by users and return them to the current token pool. Meanwhile, since the benchmark currency at the other end of the Bancor protocol remains the same, it will increase the value of this utility token for those token holders. The corresponding recursion formulas are as follows:

$$\begin{aligned}
baseCurrencyAmount' &= \frac{(baseCurrencyAmount - baseCurrencyReserve) \times utilityTokenTotalSupply}{utilityTokenTotalSupply - utiltyTokenRemainings - utilityTokenConsumption} \\
baseCurrencyReserve' &= \frac{(baseCurrencyAmount - baseCurrencyReserve) \times (utilityTokenRemainings + UtilityTokenConsumption)}{UtilityTokenTotalSupply - utilityTokenRemainings - utilityTokenConsumption} \\
utilituTokenRemainings' &= utilityTokenRemainings + utilityTokenConsumption
\end{aligned}$$

2. Token dividend protocol

The core of the Bancor-based token dividend protocol is to increase the amount of the base currency, while keeping the remaining amount and total supply of the utility token the same in the meantime, which increase the value of this utility token for those token holders. The corresponding recursion formulas are as follows:

$$\begin{aligned}
baseCurrencyAmount' &= \frac{(baseCurrencyAmount - baseCurrencyReserve + baseCurrencyDividends) \times utilityTokenTotalSupply}{utilityTokenTotalSupply - utilityTokenRemainings} \\
baseCurrencyReserve' &= \frac{(baseCurrencyAmount - baseCurrencyReserve + baseCUrrencyDividends) \times utilityTokenRemainings}{utilityTokenTotalSupply - utilityTokenRemainings}
\end{aligned}$$

3. Token option protocol

All most all so-called team lock-up plan for the ERC-20 tokens on the Ethereum network are performed manually by the project team. These ERC-20 tokens are kept in a public account so all investors could monitor together.

In consideration of the need for the team lock-up plan purpose, we purpose a lock-up release protocol In this whitepaper. The locked tokens will be released gradually during the lock-up period, and the corresponding formulas are as follows:

$$utilityTokenReleasedOptionAmount' = \frac{(currentTime - utilityTokenIssuedTime) \times utilityTokenOptionTotalSupply}{utilityTokenOptionLockupPeriod}$$

The option release process is equivalent to the new token distribution process, and the newly issued utility tokens are authorized to the DApp project team in the mean time. It is necessary to keep the Bancor trading protocol smooth running all the time especially when the new tokens are released. In the EOS RAM extension model, there is a problem that large amount of EOS will be locked in the smart contract eventually. Even RAM holder sell all their RAM, they can not get their locked EOS back from the contract. In this new model, this problem could be avoided so as to protect the token holders' interests. The corresponding formulas are as follows:

$$\begin{aligned} baseCurrencyAmount' &= \frac{(baseCurrencyAmount - baseCurrencyReserve) \times (utilityTokenReleasedOptionAmount + utilityTokenTotalSupply)}{utilityTokenTotalSupply - utilityTokenRemainings + utilityTokenReleasedOptionAmount} \\ baseCurrencyReserve' &= \frac{(baseCurrencyAmount - baseCurrencyReserve) \times utilityTokenRemainings}{utilityTokenTotalSupply - utilityTokenRemainings + utilityTokenReleasedOptionAmount} \\ utilityTokenTotalSupply' &= utilityTokenTotalSupply + utilityTokenReleasedOptionAmount \end{aligned}$$

4. Token trading fee

In order to encourage users to purchase and consume the utility tokens issued by the DApp project team, token purchase fee and consumption fee is waived for users. However, there will be a selling fee charged when users sell tokens. The main reason is that when a DApp is in its early stage, it's economy is rather small. Market speculation behaviours could lead to sharp fluncuations in token price for Bancor protocol based tokens, which has a negative impact on the DApp development process. That is why a relative high selling fee is introduced at the beginning, and it will decrease gradually when the option is released. The recursion formulas for the selling fee are as follows:

$$utilityTokenSellingFee' = \frac{2 \times (initFeeRatio - baseFeeRatio) \times utilityTokenOptionLockupPeriod}{(curentTime - utilityTokenIssuedTime) + utilityTokenOptionLockupPeriod} + (2 \times baseFeeRatio - initFeeRatio)$$