

Topic: Dynamic Whale Staking Read 119 times

 forum.biblepay.org/index.php



Dynamic Whale Staking

« **on:** 2019-11-08, 09:36:06 »

Quote

This is a concept to be considered: Dynamic Whale Staking.

The problem statement: We need more users and more investors to make biblepay more popular.

Solution: Make a feature in BBP that allows 'latent BBP' to be 'staked' for a whale-unit reward. Part of the goal is to make this as easy as possible; no complicated smart contracts, no daily wallet unlocks, no leaderboard, just simply buy the BBP on the exchange, load the wallet and run the command that burns the coins.

PROPOSED CHANGES TO EMISSION SCHEDULE WIKI:

https://wiki.biblepay.org/Emission_Schedule_2020

**** NOTE: THERE ARE NO NET CHANGES TO THE EMISSION TOTAL AMOUNT OR TOTAL MONEY SUPPLY,**

however, the changes included here are: A) We allow room for DWS rewards - starting in 2020 - by increasing our deflation percent from 19.5 to 20.04%, and allocating an additional maximum of 10% (of our gross monthly block reward) towards DWS rewards (This does **not** decrease the gross block reward, the sanctuary reward, or the POBH reward or the GSC contract amount). If the DWS rewards are not used, because the DWS stake is never burned, then the BBP is never actually spent from our emissions. (In this case the actual total money supply in BBP would be less than 5.1 B at the end of the schedule depending on how much is never spent).

Why would someone want this: A person who is seeking cryptocurrency exposure will be looking at coins to invest in with certain characteristics. I feel our coin has a bright future because: Its a feel good investment (we do tithe 10% to orphan charity, so the utility of our organization works for you), we are deflationary in emissions per year, we are a HODLing community with governance meaning we have a high % locked up in sancs, and finally - the investor will be comparing ways to get more coin-unit rewards through proof-of-work or mining etc.

Our original money supply covenant: We will need to ensure we don't break our covenant. This would be accomplished by allocating a percent of our emitted coins per year toward this project - in a certain band of years, IE 2020 through 2030, and increasing our deflation rate to pay for the feature. So we would have no net changes in our total supply, but we would modify our coin emission chart to include this feature over 10 years. In laymans terms this means we would emit 10% more total coins for 10 years, but we would increase our deflation % per year to cover the change.

Emission Changes: In this proposal we allocate 10% more than our current annual emission rate (which is currently 612MM per year), therefore we would spend up to 5,083,333 per month on 'dynamic whale staking' deflating at our standard deflation rate (of 19.5%). However note, this is the absolute maximum, if the product is 100% allocated. If the feature is not used, these coins would not be emitted for whale staking. We will also decide how much we need to increase our deflation %.

Full Safeguards: The DWS money supply ratio would be hardcoded in the wallet for full safeguards, to ensure the wallet cannot emit more than the designated existing total plus the DWS for that month. So, if we currently emit 50MM in a given month, the wallet would not allow more than 55MM to be emitted in a given month, due to the hard rules. Additionally, the recipients of each DWS would need to be approved by all the sancs (the sancs who create the daily superblock would evaluate every DWS record and add it to the superblock). So there will be no chance of a miner adding a DWS that is unauthorized.

The core wallet will be generating the coins for the investor from the coinbase, so solvency is guaranteed. BBP cannot go bankrupt from DWS.

Provably identifiable DWS burns: Since DWS uses a provable burn record, it will not be possible to fake a burn, and it will be provably demonstratable that the burn address does not have a corresponding private key. Rob will attach the mathematical proof to this proposal that shows how to create a certifiably provable burn address for Biblepay with no corresponding private key.

EDIT: I am attaching the source code that facilitates the generation of a non-spendable-bbp address for both testnet and Prod, based on the first public key derived from an unusable private key of {0x0} (See attached file: DynamicWhaleStaking.cs).

This code creates a BBP keypair using a blank private blank key - ie hexcode {0x0} - that is a provably unspendable private key. This routine generates this public address : B4T5ciTckWauSqVAcVKy88ofjcSasUkSYU (This is the first public address from the 0x0 private address that is a valid spending address). So if you send BBP to this address it will be lost forever.

=====

Mechanical operation:

We will allow a DWS duration of between 7 and 365 days.

The dynamic-whale-reward will be quoted on the screen in 'test' mode - and will match on every biblepay client until the quote is consumed.

It will be updated block to block.

To protect the system from hogs, we will use two moving averages, annual saturation and monthly saturation.

The monthly saturation will primarily drive the quoted dynamic-whale-reward.

Full example of a DWS whale stake in action:

getdwsquote

Amount staked: 1,000,000

Dynamic-whale-unit reward: 15

This means the user would potentially receive 15000 bbp in reward one year after the coins are burned, if bbp returns them (they are 100% at risk, and we will explain this by posting a risk notice).

(Allowed stakes can be between 100 and 1,000,000 bbp):

dws 1000000 receive_address 365 authorize true

In this case, the whales stake will be sent to the burn address;

365 days later, the whale will receive the original 1mil back, plus 15,000 in additional coins.

The reward algorithm should stabilize at a point where the stake-reward emissions average out to be the allocated projection.

Note: The shorter term rewards are penalized by 50% divided by the span.

So a DWU reward of 100 will only be 50 if the duration is 1 month.

75 for 6 months. 100 for 12 months.

This will encourage DWS whales to lock the coins for longer durations, and free up more of the DWS rewards for more distinct users (because those who choose short spans will receive less rewards). This will be accomplished by storing the duration on the burn itself, so the actual DWU reward will be calculatable off of the base.

Example:

Current DWU-reward level is 50 for 365 days.

DWU-reward is 37.5 for 6 months.

DWU-reward is 25 for 1 month.

User A types : exec dws 10,000 365. They will receive receive 10,000 + 5,000 after 365 days.

User B types : exec dws 10,000 180. They will receive will receive 10,000 + 1,875 in 180 days.

Target Algorithm version 1.1:

"Two saturation levels are monitored. The Annual saturation level is comprised of the total outstanding (owed and unpaid) DWS stakes over the next 365 days.

The Monthly saturation level is comprised of the total outstanding owed over the next 30 days."

Annual and Monthly Saturation Equation:

$$SE_Percent = \frac{Total_Outstanding_Coins_Owed}{Maximum_DWS_Reward_Amount_For_The_Period}$$

1. If Annual Saturation is > 95%, drop DWU reward level to zero (for quotes).
2. If Annual Saturation <= 95%, offer the inverse of the monthly saturation level as a DWU reward.

Inverse of monthly saturation level:

$$IA = 1.0 - Monthly_Saturation_Level$$

$$ROI \text{ quote} = IA * MAX_DWS_DWU$$

Example:

1. Annual saturation level is at 25%. Monthly Saturation is at 50%. Since IA equals 50% (for 365 days), the quote would be $.50 * MAX_DWS_ROI = 100$ DWU.
2. Annual saturation level is at 25%. Monthly Saturation is at 90%. Since IA equals 10% (for 365 days), the quote would be $.10 * MAX_DWS_ROI = 10$ DWU.
3. Annual saturation level is at 96%. Quote = 0 DWU.
4. Annual saturation level is at 94%. Monthly Saturation is at 1%. Quote = $.99 * MAX_DWS = 198$ DWU.
5. Annual saturation level is at 50%. Monthly Saturation is at 99%. Quote = $.01 * MAX_DWS = 2$ DWU.

Risk Rules: (Rules to mitigate risk):

Each burn will be audited in the memory pool before accepted (meaning a burn can be turned down by all the nodes, if for example the burn is created fraudulently, or, if the burn will result in an overage condition for BBP).

Rule 1: Each node will check the components of the burn, to ensure they match the quote available, the duration available, and the availability of the DWS ROI. Check the bounds for each component also. (Otherwise if any of this fails, reject the burn).

Rule 2: No more than 5 mil in gross stakes per day. This will limit our exposure for GSC contracts to never pay more than 5 mil back to whales in a given day.

Rule 3: If the Annual Saturation level > 95%, reject the DWS.

Rule 4: Each node will re-assess the 'total whale metrics' as each transaction occurs.

Burns rejected by the memory pool will automatically refund the funds back to the sender (actually, the transaction will be rejected, the same way a conflict or a double spend is handled).

Howey-test protection, to ensure biblepay stays a utility:

We will add a warning to the DWS burn method:

Your coins are 100% at risk if burned. This is a self directed action by you. Type authorize to acknowledge that burning the coins results in a potential entire loss with no future expectation of profit, and biblepay will be held as a harmless utility for this self directed action. We do not promise you an increase in value for any of your coins! We do not promise to pay you back for any burned coins, and coins are 100% at risk.