Summary of changes for fees in Yeti Finance architecture:

By making our system multi collateral and portfolio based, we needed a new way to make sure that the amount of a particular type of collateral that our system is backed by is not more than a safe amount, while also capturing more revenue for the protocol and the YETI token.

The fee system that we came up with was to use a combination of fee options 0, 1d, 2, and 3 from below. To summarize, we add a variable fee onto the flat fee assessed on the debt amount from the original Liquity system. In this variable fee, we implement a price curve for each collateral type that depends on how much of the entire system is collateralized by that specific asset (backing percent). The higher the percentage, the higher the fee, based on risk parameters that we implement unique to the asset. This allows us to capture more profit for the protocol, since people will be more likely to pay higher fees when there is high demand for borrowing against that riskier asset. In this summary we use wMEMO as an example of risky collateral that we may not want the entire protocol being backed by, and wAVAX as a safe collateral that could safely back the entire protocol. Additionally, we implemented hard caps as a safety mechanism to make sure that users can’t forcefully add collateral that we don’t want backing the stablecoin at that amount.

Below is a much more detailed account of how we came up with the fee system and some of the vulnerabilities that different options gave us. Most important to read Situations D, E, and F

Definitions:

VC - Stands for “Virtual Coin” - is a way for the system to keep track of the value of different collateral values, since we take riskier assets to have less value in the system than safer assets. Essentially standardizes the value of all the collateral in one user’s trove into one collateral value number. Depends on a safety ratio which is defined as a risk parameter when adding the token to the whitelist. $VC = Safety ratio * Token amount * Token price in USD. Example: I have 0.75 wMEMO at $8000 dollars with a safety ratio of ⅔. $VC = 0.75 * ⅔ * 8000 = $4000. So, I can take a loan against this $4000 dollars as if it were $4000 of a safe asset with a safety ratio = 1.

Collateralization Ratio - Ratio of the amount of collateral that a borrower has compared to the amount of debt that they have. $2200 in VC of collateral and $2000 YUSD debt = 2200/2000 = 110% Collateralization ratio. Any collateralization ratio under 110% is vulnerable to liquidation. Higher collateralization ratio means a safer position.

Backing percent - How much of the protocol is backed by that particular asset. If the system has VC value of $1,000,000 and it has $10,000 VC value of wMEMO, then it has 1% backing percent.

Variable fee - The new fee we are adding, calculated on each collateral type that scales on backing percent.

Design Objectives:

a. 0% interest
b. For risky collateral, ensure that there is a cap on the backing percent. If we allow a risky collateral to back our system heavily, then the entire life of the YUSD stablecoin and protocol is dependent on that collateral maintaining its safety / liquidity.
c. Take a larger fee based on collateral risk / yield upside, where they would be ok with a higher fee. Capture larger, untapped, but riskier lending markets like wMEMO. For instance, the MIM pool fills instantly for wMEMO whenever it is replenished.
d. Capture a fee when a user swaps assets in their trove without changing their debt amount. If this wasn't the case, a situation like situation A could be possible.
e. Not punish people for taking a safer position in our system. A user with collateral ratio of 200% should pay less fee than a user with collateral ratio of 110% with the same amount of collateral but more debt.
f. Users should not be able to exploit the fee system to pay a lesser total fee on the collateral they are inputting than is currently assessed by the system
g. Users should not be paying more fees on an adjustment than if they were to open a new trove from another address on the same transaction

Fee System Options

Ideally we capture all objectives in our implementation. These are options for how to implement the fees:

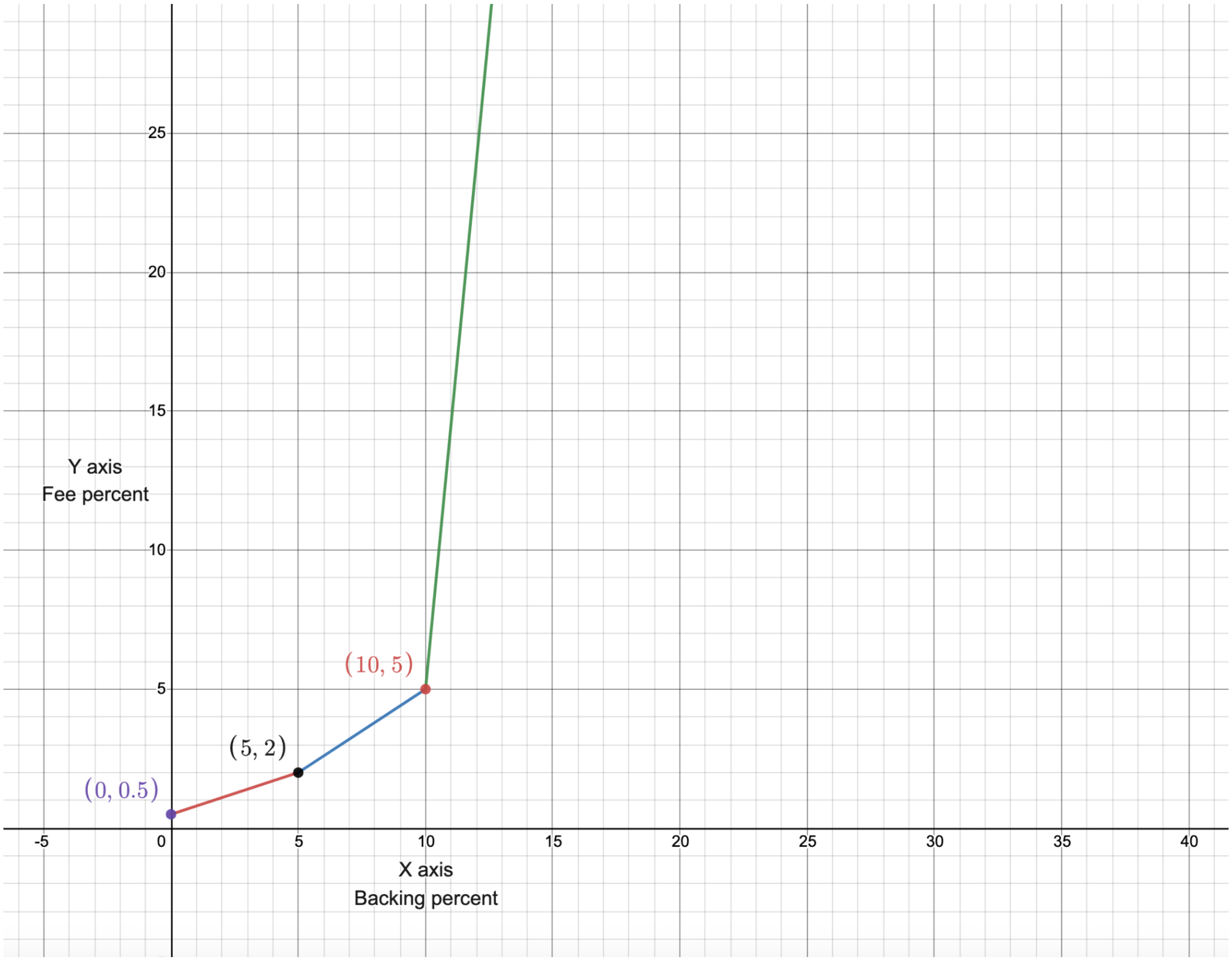
Options for fee system properties:

0. Fee charged purely on new debt amount. Liquity charges a base 0.5% fee. This happens no matter what.
1. Each collateral has a unique fee price curve which scales up based on the backing percent. It reaches a point for risky assets where the fee would scale to 100% of the collateral value. For safer assets this fee is very slow to scale, or zero for something like wAVAX. This is implemented as a price curve with priceCurve(x) = fee %, and we define x as the backing percent. More specifically we use the pre-transaction and post-transaction values of x for a particular asset to calculate the real fee. This is fee % = average(priceCurve(x before any transaction), priceCurve(x after the transaction)). Options are multiplying the fee percent:
    a. By new debt amount

    b. By total debt amount after the tx (for adjust trove, if you are paying back debt then it is total debt after paying off some debt)
    c. By collateral VC Value divided by ICR
    d. By collateral VC Value

2. Have a decay system based on the last fee percent and time assessed for each asset. Then there is a minimum fee based on the previous amount, after some time decay adjustment.
3. Each collateral has a dollar cap based on how much of that collateral is in the system.
4. Charge fee on withdrawal of collateral
    a. On total debt amount after the tx
    b. On collateral VC value

An example of a price curve, axes are denominated in x% * 100



Situations

We have modeled out some potential economic attacks involving the fee options and came up with options 0, 1d, 2, and 3 comprising our fee system. In each situation, some type of implementation of Fee option 1 is always used, since by itself option 1 satisfies objectives a, b(partially), and c.

Situation A: System currently utilizes fee option 1a.

A user can perform an exploit where they open a trove with non-risky collateral that has no fee or low fee, and then keep the same amount of total collateral VC but switch into some risky collateral. They have essentially dodged the fee.

Concrete example: Alice opens a trove with $4000 VC of wMEMO, and pays the associated fee assessed for that value. Bob wants to open a trove with $4000 VC wMEMO, but does not want to pay the fee. He opens a trove with $4000 VC of wAVAX which has a 0% variable fee, then in another transaction he adjusts his trove to take out all wAVAX and add $4000 of wMEMO simultaneously. His ICR stays the same but he has now paid no fee on the wMEMO in. This is failing objectives d and f.

Situation B: System currently utilizes fee option 1b.

This seemingly fixes the problem from situation A and solves objectives d and f, but now fails objective g, since if a user wants to just add collateral to their trove, their fee is assessed on their entire debt of their trove.

Concrete example: Alice has a trove with $10000 wAVAX collateral, and $5000 debt, so 200% ICR. She would like to take out $500 debt on $2000 of wMEMO. The associated fee would be multiplied by $5500 debt. If Alice were to just transfer her wMEMO to a new wallet address and open a trove here, then she would pay just $500 on her wMEMO. This is why it fails objective g here. The problem is that when adjusting her trove, we don't really know the makeup of the collateralization.

In another situation, she could instead add the wMEMO and then in another transaction, purely add debt. Then, she could dodge the wMEMO fee and it would fail objectives d and f again. As a solution for this second situation with collateral and debt added in separate transactions, we could charge a fee on all collaterals in her trove. But, it would fail objective g again, so, this solution doesn't work.

Situation C: System currently utilizes fee option 1c.

Alice has $4000 VC of wMEMO. Bob has $4000 VC of wMEMO. Alice takes out a 200% collateralized position by taking out $2000 of debt. Bob takes out a 133% collateralized position by taking out $3000 of debt. With option 1c, we would have objective e satisfied, where users with higher collateral ratios pay a lower fee. This is because in this case, Alice would pay (fee assessed for $4000 input) / 200%, and Bob would pay (fee assessed for $4000 input) / 133%. However, this can be exploited because Bob could just add more collateral which has 0% fee (like wAVAX), so that when they open their trove they would have $6000 VC Collateral and $3000 debt so 200% ICR, then just remove the wAVAX collateral in another transaction so they have essentially paid the same fee as Alice, but have the same ICR after everything. Users could actually input a very large amount of some other collateral with the same idea, so that their ICR at open is something like 1000% and pay a very small fee.

A potential solution is to say the factor is instead min(200%, ICR after tx) to make sure that people don't exploit it too much. We are then charging a higher fee for people with under 200% ICR, but those users could perform the same loophole from the beginning if they wanted to open at say 150%. So, essentially we are charging everyone at 200% ICR, at which point we are just doing fee option 1d.

Situation D: System currently utilizes fee option 1d.

By itself, it passes objectives a, b(partially), c, d, and g. It passes g now since it always charges a fee on the collateral amount, so compared to options 1a and 1b, it assesses purely on collateral value and not debt value. So the same collateral VC sent in always has the same fee. However, this means we cannot satisfy objective e. What about objectives b and f? Concrete example of why it doesn't pass b and f currently:

The wMEMO has a backing percent of 5%, and we want this to be hard capped at 10%. The entire system has $100,000 total VC value. So, currently, wMEMO has $5,000 VC, and it should be capped at currently around $10,000 VC in the system. Using the variable fee system, at 5% the variable fee is somewhat high already, definitely higher than at say 0.1%.

If Alice were to open a trove right now with $200,000 VC collateral of wMEMO, then the system would be $205,000 / $300,000 = about ⅔ backed by wMEMO. Then, Alice would be paying an insane fee so it would not be economically favorable. However, it is still possible to make the system this much backed by wMEMO, which is why it fails objective b.

Instead, Alice opens a trove with $100,000 debt, and $5,000,000 VC of wAVAX, an asset with 0% variable fee. Now, wMEMO comprises $5,000 /$ 5,100,000 = about .1% of the system. She adds $200,000 VC of wMEMO, and after the transaction wMEMO comprises $205,000 / $5,300,000 = about 4% of the system. The fee will be assessed at this time. Then, Alice, in another transaction, can pull out all wAVAX from her trove so that her ICR is now 200%, and the system is now $205,000 / $300,000 = about ⅔ backed by wMEMO, without paying more than 4% fee, which is what it would have been if she had added a trivial amount currently. This example could be done with more drastic numbers to make the fee even smaller. This is why it fails objective f.

Situation E: System currently utilizes fee options 1d and 2

Currently inheriting from situation D, it passes objectives a, b(partially), c, d, and g. Option 2 allows the system to pass objective f. Essentially, in the situation detailed above in situation D, Alice could open the trove with a different amount of collateral first, and pay a lower fee on this. Fee option 2 fixes this because even if she adds wAVAX first, we save the backing percent at the time of the last fee assessed. As a reminder, the fee is calculated with: fee % = average(priceCurve(x before any transaction), priceCurve(x after the transaction)). So, instead of using current x before Alice's transaction, where she can manipulate it to be .4%, we can instead use the previous fee percentage at 5%, and assess the post transaction x using that as well. The post transaction x would be 5% + (fee with VC of wMEMO added). So instead, Alice would be paying a much higher fee, equal to about 7% instead of 3%, and so she can't realistically dodge the fee. Essentially it puts a minimum on the fee that she pays based on what has happened previously. If the fee at 7% is ridiculous, then Alice can't perform this system exploit. This max fee decays according to some previously set decay time, and as time goes on and the min fee slowly drops, anyone has the opportunity to add this collateral type at a lower fee so it is not an exploit, the system TVL has just gone up. However, she can still forcefully add wMEMO to the system, then take out her wAVAX later if she is fine with that fee. This passes f, but does not pass b.

Situation F: System currently utilizes fee options 1d, 2, and 3

Currently inheriting from situation E, it now passes objective b also, bringing the list to all objectives except e. Fee option 3 allows for objective b to pass, since it puts a hard cap on the amount of collateral that the system can be forced to take. Hard caps are not ideal, but will likely be set to larger than reasonable. We were unable to find a system which was able to allow for objective b with absolute safety without hard caps. A remedy to this is to protect against this "large addition and subsequent withdrawal of 0% variable fee collateral" attack could be fixed by potentially adding a small fee like 0.1% - 0.5% to these safe collaterals, but currently undecided whether this is reasonable for wAVAX etc. We still need to fine tune the number tradeoff between interest and additional flat fee that users are willing to pay to make final adjustments.

Objective e is not possible in this system as discussed in the beginning of situation D. The 0.5% YUSD fee still applies here but objective e is not satisfied currently on the fee assessed from collateral value.

Situation G: System uses anything with fee option 4

Fee option 4 could be used along with fee option 1 to satisfy all the objectives that option 1 gives, but we decided that users should know exactly how much their debt is once they add collateral, otherwise it would be a bad UX. Also, if someone adds wMEMO when it has a low fee, they shouldn't have to pay a high fee on withdrawal just because the protocol filled up on wMEMO.

Summary:

We were able to capture almost all objectives in Situation F by using this combination of systems, and making a compromise on objective e is a reasonable sacrifice. The final system has fee options 1d, 2, and 3, and provides all objectives except e. This is implemented in ThreePieceWiseLinearCurve.sol, called by Whitelist.sol, where each collateral is assigned a particular price curve. The whitelist function is called by BorrowerOperations.sol when there is new collateral passed in, for openTrove and adjustTrove.