MultiLend: A Permissionless and Price-feed-free Lending Protocol

MultiLend Labs.
multilends@gmail.com
www.multilend.finance

Abstract. MultiLend is a permissionless and price-feed-free lending protocol that supports any tokens. Its main innovations include: No need for permission to add a lending pool with any token pairs. No risk of price feed manipulation. No inefficient and insecure protocol-level governance. Based on its innovative lending model, a vast shorting market with leverage can also be developed.

1. Overview

The MultiLend protocol facilitates peer-to-pool secured loans without governance and without external price feeds. Current lending and borrowing protocols which utilize smart contracts require active governance (e.g. to set rates and to update contracts) and/or rely on external price feeds (such as oracles like Chainlink[1]). Because the pricing of collateral and parameterization of loans are left to subjective decision making through governance rather than market forces, these protocols carry both solvency and liquidity risk. Governance and maintenance overhead create barriers to entry in the market for lending and borrowing of on-chain assets. MultiLend solves these problems with its unique design, which is defined by the following features:

Permissionless pool creation: Much like the popular DeFi primitive, the "automated market maker," *AMM*, MultiLend pools exist in unique pairs: supply token, provided by lenders and collateral token, provided by borrowers. Pools allow lenders to assess borrower demand for their supply token and for borrowers to assess lender demand for loans backed by their collateral. Pools are created permissionlessly, meaning anyone can create a pool to borrow arbitrary fungible tokens using arbitrary fungible or non-fungible tokens as collateral. Therefore, no governance process is needed to whitelist approved tokens.

Price specified lending: MultiLend replaces external price feeds (oracles) by allowing lenders to input the price at which they're willing to lend. This price is the amount of supply token (i.e. the token they are lending) they are willing to lend per unit of collateral pledged by the borrower. For example, if a lender deposits at price 100, they are willing to lend 100 units of supply token per one unit of collateral. MultiLend pools separate prices into predefined buckets to reduce the complexity of the protocol, prices are therefore hereon referred to as "buckets." Borrowers are then able to borrow from the aggregated liquidity of these various buckets. Depositing in a price bucket allows lenders to have discrete solvency risk based upon their personal risk tolerance. In the worst case scenario, a lender should receive collateral at the price which they lent to the pool in lieu of their supply token deposit.

Market-derived interest rates: MultiLend uses deterministic rules to set interest rates based upon user actions in the pool. These rules are founded on the assumption that aggregate average loan collateralization (*pool collateralization)* is an accurate indicator of the volatility profile of an asset, because borrowers have a natural aversion to liquidation (which they will be forced into over time if they fail to adequately collateralize their loan). MultiLend uses *pool collateralization* to inform the desired utilization ratio (i.e. the equilibrium between utilized and unutilized deposits) for the pool: the *target utilization*. Subsequently, we can look at the *meaningful actual utilization,* which is a short term EMA of the ratio of debt to total lender deposits available to the average loan in the pool (see 8.0 INTEREST RATES). Interest rates adjust based on these values to move the pool towards equilibrium. When meaningful actual utilization is low relative to target utilization, there is a surplus of lenders, and rates can be lowered. When meaningful actual utilization is high relative to target utilization, there is a shortage of lenders and rates can be increased.

Liquidation Bonds: While a loan's eligibility to be liquidated is determined formulaically by the contract, no actual liquidation will occur unless an external user triggers one. In order to trigger a liquidation this user must pledge a liquidation bond. A liquidation bond is effectively a bet, with a payoff curve similar to a short put option spread (effectively a bet that the bidders in the dutch auction will place bids below the neutral price of the loan, see 7.3.1 Determining the Neutral Price), on the outcome of a pay-as-bid dutch auction for the collateral. This mechanism encourages fair

outcomes by imposing a penalty on spurious liquidations. While there is no endogenous incentive to spuriously liquidate a borrower in MultiLend, this introduces an overt disincentive to liquidate loans that are well collateralized with respect to the market price of the collateral (as well as an incentive to liquidate loans that are comparatively undercollateralized).

From a user perspective, the MultiLend Protocol has three core functions which will be described in further detail below. They are:

(1) Lend
(2) Borrow
(3) Trade

Despite the dependency-free nature of its contracts and its additional trading functionality, MultiLend should behave similarly to a traditional on-chain lending and borrowing market.

2. Creating a pool

An MultiLend pool is created using the factory contract. A pool aggregates all lending and borrowing activity of a specific supply token as secured by a specific collateral token type. Three pool types may be created in the MultiLend protocol:

- A fungible pool, consisting of a fungible supply token and a fungible collateral token.

- A non-fungible token, *NFT*, collection pool, consisting of a fungible supply token and an NFT collection (meaning all NFTs from that collection are accepted) as the collateral token.

- An NFT subset pool, consisting of a fungible supply token and a selected subset of NFTs, from the same collection (one or many), as the collateral token.

The token with which the pool is collateralized (i.e. the token that borrowers deposit), which is expressed as the numerator, is the collateral token. The denominator is the supply token lent out in the pool. The factory contract only permits the creation of one pool per unique denominator/numerator token pairing (with the exception of the NFT subset pool). For example, there can only be one ETH/DAI pool where DAI is lent out against ETH but a DAI/ETH pool, where ETH is lent out against DAI, may also exist as its own pool.

Alice would like to lend out her DAI to earn interest. She wants the loans to be collateralized by ETH, but there is no existing ETH/DAI MultiLend pool. She creates one using the MultiLend factory contract.

3. Pool structure

3.1 Price Buckets

MultiLend pools contain an index of prices, discretized into *price buckets*. This is done to simplify the computational requirements of the protocol. Price buckets can simply be understood as the price at which lenders are willing to lend against the collateral. The particular bucket range and spacing was chosen to be inclusive of a wide range of asset prices, in order to maximize the flexibility of the protocol. In a pool there are 7,388 price buckets, spaced 0.5% apart, with 1 being the price of bucket number 3232. This means there are 3,232 price buckets below 1, and 4,155 price buckets above 1. The following are some useful price bucket numbers and their corresponding prices which demonstrate the distribution of price buckets in the pool:

Fenwick index	Bucket index	Price
1	4155	999969141.896623
1077	3079	4669863.09088793
2154	2002	21699.7952738665
3232	924	100.332368143273
4156	0	1
4166	-10	0.95134794069607
4310	-154	0.463902261297398
5260	-1104	0.00406132571375266
5388	-1232	0.00214492403617474
6466	-2310	0.000009917388865692
6647	-2491	0.000004021056399147
7108	-2952	0.000000403446260869
7388	-3232	0.000000099836282891

The formula to generate the price for a given price bucket number is:

$$Price = 1.005^n$$

where n is the bucket index of the price bucket being calculated.

A price bucket consists of the following:

- Supply token (deposit)
- Claimable collateral
- Total liquidity provider balance (Total *LPB*)

When a user provides liquidity to a pool they specify a price bucket ("bucket" for short). Within the contract this is known as *deposit* and is recorded as an amount of supply token owed by the contract to the depositor. Deposits are liabilities of the pool. Users may also place collateral into a bucket, usually via a trade or as the result of a liquidation auction. Each bucket maintains a *claimable collateral* account to reflect the presence of this collateral in the bucket. Collateral is exchangeable with supply tokens at the price of the bucket. Finally, a record is kept of which portion of the content's of the bucket is owed to each liquidity provider. For example, if there are two depositors in a bucket and each has contributed 100 supply token, each would be entitled to 50% of the contents of the bucket, whether it be supply token, collateral, or a combination of the two. If a bucket has both deposit and claimable collateral, users redeeming *LPB* have the choice of which form to withdraw, with the price of the bucket itself being the rate of exchange between deposit and claimable collateral. The exchange rates between LPB in price bucket with price p, and supply token (deposit) and collateral token (claimable collateral) are:

$$LPB_p/QT = (\text{deposit}_p + p \cdot \text{claimableCollateral}_p)/\text{TotalLPB}_p$$

$$LPB_p/CT = (\text{deposit}_p + p \cdot \text{claimableCollateral}_p)/(p \cdot \text{TotalLPB}_p)$$

If a bucket is empty these equations yield 0/0, and the exchange rate could be initialized to anything. In this case, we define the exchange rate between LBP and supply token LPB_p/QT to be 1.0.

Now that the pool is initialized, Alice deposits 50,000 DAI into the bucket with a price of 1,000. She is issued 50,000 LPB in that bucket, which now has 50,000 in deposit.

3.2 Loans

When a user pledges collateral and borrows supply token, their debt is recorded as a *loan*. Each loan records the amount of debt the borrower has outstanding and the quantity of collateral posted to secure this debt. The borrower may add or withdraw collateral at any time, unless it would leave their loan insufficiently collateralized (see 5.0 BORROW). For loans secured by NFTs, there is an additional mapping in the contract specifying precisely which NFT belongs to each borrower.

By taking a loan's debt and dividing it by the amount of collateral pledged, the loan's *threshold price*, *TP*, can be calculated. The *TP* of a loan is the price at which the value of the collateral equals the value of the debt. This variable is used extensively in the protocol.

Loan creation triggers an origination fee, which is represented by immediately increasing the borrower's debt (see 5.4 Origination Fee). Lenders are faced with deposit penalties under certain conditions. Additionally, the pool collects "net interest margin," *NIM* from the interest collected on loans (see 4.0 LENDING, 8.0 INTEREST RATES)

Bob has ETH, and would like to borrow some DAI. He places 20 ETH into the MultiLend ETH/DAI contract and withdraws a loan of 18000 DAI. His debt is 18,000, his collateral pledged is 20, and his TP is 18000/20 = 900 (we are ignoring the origination fee for the moment)

3.3 Reserves

Origination fees, deposit penalties and net interest margin on loans are accumulated in a pool's "Reserves." Reserves are ultimately used to buy and burn MULTILEND tokens. Reserve balance is calculated using the Reserve Equation:

$$Reserves = debt + balance_{pool} - \sum_{p \in Buckets} deposit_p - \sum_{l \in Liquidations} bond_l - balance_{CRA}$$

In this formula $debt$ is the amount of debt outstanding, $balance_{pool}$ is the balance of supply token in the pool contract, $deposit^p$ is the amount of deposit in bucket p (which is summed over all buckets), $bond^l$ is the liquidation bond posted for liquidation auction l (which is summed over all active auctions), and $balance_{CRA}$ is the balance of the current reserve auction (see 9.1 Buy and Burn Mechanism).

Immediately after Bob has withdrawn his loan, the ETH/DAI pool has 18,000 in debt, 50,000 in total deposit, a DAI balance of 32,000. There are no liquidation auctions or collateral reserve auctions, so the reserves of the pool are 18,000 + 32,000 - 50,000 = 0.
This discussion ignores origination fees, which will be included in the discussion below.

4. Lending

4.1 Deposit

Lenders deposit supply token into specific price buckets and are then credited with *LPB,* in those buckets. Optionally, LPB can be received in the form of an NFT which represents a transferable version of their *LPB* in the pool. The deposited supply token increases the pool's supply token balance, while also increasing the total sum of deposit across all price buckets.

Deposits in the highest priced buckets offer the highest valuations on collateral, and hence offer the first source of liquidity to borrowers. They are also the first buckets that could be used to purchase collateral if a loan were to be liquidated (see 7.0 LIQUIDATIONS). We can think of a bucket's deposit as being *utilized* if the sum of all deposits in buckets priced higher than it is less than the total debt of all borrowers in the pool. The lowest price among utilized buckets or "lowest utilized price" is called the *LUP*. If we were to pair off lenders with borrowers, matching the highest priced lenders' deposits with the borrowers' debts in equal quantities, the *LUP* would be the price of the marginal (lowest priced and therefore least aggressive) lender thus matched (usually, there would be a surplus of lenders that were not matched, corresponding to less than 100% utilization of the pool).

The *LUP* plays a critical role in MultiLend: a borrower who is undercollateralized with respect to the *LUP* (i.e. with respect to the marginal utilized lender) is eligible for liquidation. Conversely, a lender cannot withdraw deposit if doing so would move the *LUP* down so far as to make some active loans eligible for liquidation. In order to withdraw supply token in this situation, the lender must first kick the loans in question.

In other words: in order to remove a deposit, the user must ensure that this action does not cause the *LUP* to move below the "highest threshold price", *HTP*, which is the threshold price of the least collateralized loan. If a lender wishes to remove deposit from a bucket that will cause the *LUP* to move below the *HTP*, they will need to liquidate the loan(s) that sit between the two points.

A lender can liquidate any loan in that range, even though its current *TP* exceeds the *LUP*, by calling the lenderKick method.

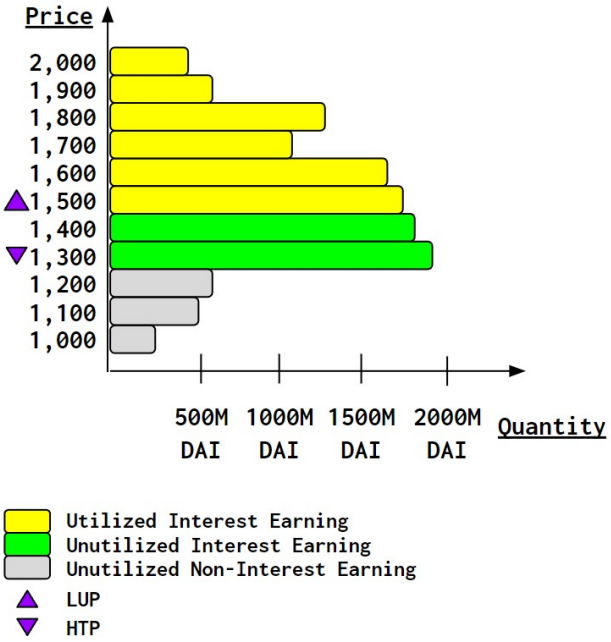


Illustration of MultiLend Pool price buckets and select parameters.

Because all deposits priced above the *HTP* are offering liquidity to every loan in the pool, they all earn interest at the same rate. Conversely, all deposits above the *LUP* are effectively being lent. Therefore, deposits above the minimum of the *LUP* and *HTP* earn interest, while deposits below both do not.

Furthermore, if there are active liquidations in the pool, deposits that are within *liquidation_debt* of the top of the book cannot withdraw until the liquidations are complete (see 7.5 Liquidation Debt). Deposits in these buckets are frozen until the liquidation auctions are settled.

Returning to our ETH/DAI pool above, the only deposit in the pool is Alice's 50,000 at 1000. The total debt in the pool is 18,000, so the LUP is 1000. Carol arrives at the pool and deposits 20,000 DAI at a price of 1100. Now the LUP is 1100, as the deposit at that level exceeds the debt. If Carol had only deposited 10,000, the LUP would have remained at 1000.

4.2 Unutilized Deposit Fee

Lenders are subject to a small fee for making a deposit below the LUP.[2] This fee is applied in order to mitigate MEV attacks on the pool. The unutilized deposit fee accrues in the pool's reserves.

4.3 LPB NFT (Optional)

Lenders are provided the option to mint an NFT representing their *LPB* in a price bucket. The liquidity provider NFT is a wrapping tool that allows external transferability of *LPB*. This NFT can be used in other DeFi protocols or in MultiLend itself if desired.

4.4 Implementation Note

There are a number of technical challenges in implementing this lender/bucket accounting on-chain. In particular, we need a set of data structures and algorithms that enable the following four operations:

1. Evaluating the total deposit over a range of buckets
   (e.g. used for evaluating how much deposit is in a specific bucket or above the HTP, for lender interest calculation)
2. Searching for the highest price that has at least a given amount of deposit above it (e.g. for the LUP calculation)
3. Incrementing/decrementing deposit quantity in a bucket
   (for lending or withdrawing deposit)
4. Multiply all deposit in a range by a give scalar
   (for accruing interest to lenders)

Because of the constraints of the EVM and other blockchain based smart contract implementations, these four operations must be made very efficient, especially in the worst case. MultiLend relies on a modified Fenwick tree structure to hold deposits (actually, two interacting Fenwick trees). The first holds the deposits themselves scaled in a certain way, which enables updates and range sums much like a classic Fenwick tree. The second stores scaling factors that encode interest accumulation over ranges of indices. Together, these structures enable (1)-(4) all in $log(n)$ worst case complexity, with n being the number of buckets (in our case $n = 7388$, so $log(n) < 13$).

[2] The current fee is intended to be implemented as 24-hours of interest.

5. Borrow

A borrower may take out a loan by pledging collateral to an MultiLend pool and withdrawing supply token. If a borrower uses an NFT as collateral then the NFT must be pledged in its entirety – the MultiLend protocol does not externally fractionalize NFTs (although fractional portions of NFT collateral can end up in a bucket's claimable collateral, as discussed below. In this case, MultiLend always guarantees an effective way to reconstitute an NFT given the value placed on it by the buckets).

A loan is considered fully collateralized when its debt is less than the value of its collateral, valued at the LUP. Equivalently, each loan is considered collateralized if $TP \leq LUP$. Note that the TP of a loan is entirely under the control of the borrower. If a loan becomes undercollateralized, then it's eligible for liquidation (see 7.0 LIQUIDATIONS). Borrowers should note that even if they are fully collateralized with respect to the current LUP, a lender can initiate liquidation (kick) on their loan if removing that lender's deposit from the book would move the LUP below their TP. The lender does this by calling the lenderKick method.

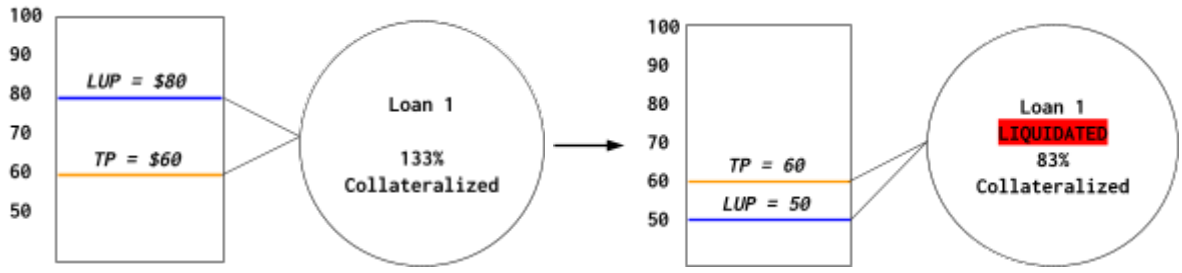


Diagram illustrating the progression of a loan from overcollateralized to undercollateralized due to the LUP crossing below its TP, and thus pushed to liquidation.

5.1 Borrower's Interest Accumulation

Borrowers accrue interest on their debt, which compounds continuously. E.g. if a borrower has $100 of debt and accrues $1 of interest, their new debt balance will be $101. All borrowers pay the same rate of interest. As the debt increases, more collateral may be required to keep the loan fully collateralized.

Because all debt accrues interest at the same rate, we can track debt using a global *inflator*. We define the *borrower inflator* BI at a given time t to be the debt that a borrower would have incurred if they had borrowed a ^{t}single supply token at the initiation of the contract and had never

debt at a later time can be computed by multiplying by the$D$ratio$_{t_1}$ of the$t_1$inflators at the two repaid or taken out any additional debt. If a borrower has debt at time , then their accrued times: t^2

Because all debt accrues interest at the same rate, we can track debt using a global inflator. We define the borrower inflator BI_t at a given time t to be the debt that a borrower would have incurred if they had borrowed a single supply token at the initiation of the contract and had never repaid or taken out any additional debt. If a borrower has debt D_{t_1} at time t_1, then their accrued debt at a later time t_2 can be computed by multiplying by the ratio of the inflators at the two times:

$$D_{t_2} = D_{t_1} \cdot BI_{t_2} / BI_{t_1}$$

In fact, we define the *t-zero equivalent debt* of a given debt at time t as:

$$D_0 = D_t / BI_t$$

This debt at any later time t_1 would then be:

$$D_{t_1} = BI_{t_1} \cdot D_0$$

By storing the t-zero equivalent debt of the loan and not the raw debt, we can always know the actual debt at any time without needing to do per-loan debt accumulations.

Bob's loan was initiated at the start of the pool, when the inflator value was 1.0. So, his t-zero equivalent debt is 18,000, which is recorded in his loan record. Some time passes, and 10% interest has accumulated. The borrower inflator is now 1.10, and Bob's debt is now $18000 \cdot 1.1 = 19800$.

5.2 Borrower's Interest Rates

Borrower interest rates change dynamically based on the pool's "meaningful actual utilization," *MAU*, and "target utilization," *TU*, in response to market forces (see 8. Interest Rates). Meaningful actual utilization is debt relative to all deposit above *DWATP* (the average threshold price of all loans in the pool, with each loan weighted by its debt).

5.3 Borrower's Repayment

Borrowers may pay back their loan in supply token and retrieve their collateral at any time.

5.4 Origination Fee

When borrowing supply token, a borrower is subject to an origination fee which is immediately added to their debt. The origination fee is calculated as the greater of the current annualized borrower interest rate divided by 52 (one week of interest) or 5 bps multiplied by the loan's new debt. This fee is designed to disincentivize interest rate manipulation and accrues to the pool's reserves.

$$\text{originationFee}_{\{\text{loan}\}} = \max\left(\frac{rate_{\{borrow\}}}{52}, 0.0005\right) \cdot \Delta\text{debt\_\{loan\}}$$

In Bob's original loan discussion, we ignored the origination fee. Suppose that the interest rate is 10%. Then Bob's origination fee would be $18,000 \cdot 10\%/52 = 34.61$ *DAI*. When he withdraws his 18,000 DAI, his debt is recorded including this origination fee, for a total of 18,034.61. His threshold price TP is 18,034.61/20=901.73.

Note that this impacts the reserve equation compared to the previous discussion: the total debt is increased, but the deposit and DAI balance of the pool are unchanged. The reserves are

18,034.61 + 32,000 - 50,000 = 34.61. In other words, the origination fee is credited to the contract reserves.

5.5 Minimum Borrow Size

The minimum borrow size is 10% of the average loan size:

$$minimumBorrow = rate_{borrow} / _{loanCount} \cdot 0.1$$

This minimum is designed to deter dust loans, which have the potential to diminish user experience. This check is enforced only when the number of loans exceeds 10 (including the new loan).

6. Trade

While the normal use of MultiLend will involve pledging collateral to the pool with the intention of retrieving it at a later time, or depositing supply token into a bucket and receiving *LPB*, any user may also send collateral to a bucket's claimable collateral account and receive *LPB* in return. This *LPB* may be immediately redeemable for supply token if there is deposit in that bucket (and other requirements, such as the *LUP* check, pass). This effectively amounts to a bid-only order book and the transaction is a trade of collateral for supply token. This functionality encourages arbitrageurs to take action, keeping deposits in an MultiLend pool below external market prices.

6.1 Trading collateral for supply token

Any actor may exercise a trade by simultaneously depositing collateral into a price bucket, minting *LPB*, and exchanging that *LPB* for supply token. Note that the trader in this case can specify a particular price bucket to trade against, in contrast with a normal MultiLend borrower who has to collateralize their loan with respect to the *LUP*. The withdrawn supply token is taken from the deposit of the bucket. This action, as always, cannot push the *LUP* below the *HTP* without the user triggering liquidations. The collateral exchanged in this transaction is put in the claimable collateral account of the bucket, which may then be claimed by exchanging *LPB*.

David owns 1 ETH, and would like to sell it for 1100 DAI. He puts the 1 ETH into the 1100 bucket as claimable collateral (alongside Carol's 20000 deposit), minting 1100 in LPB in return.
He can then redeem that 1100 LPB for supply token, withdrawing 1100 DAI. Note: after David's withdrawal, the LUP remains at 1100. If the book were different such that his withdrawal would move the LUP below Bob's threshold price of 901.73, he would not be able to withdraw all of the DAI.

6.2 Trading Supply Token for Collateral

Symmetrically, actors may purchase collateral in a price bucket by depositing supply token, and exchanging their LPB for collateral. This action increases liquidity, possibly moving the LUP to a more favorable price for borrowers. If the pool uses NFT collateral, the NFTs are claimed in last-in-first-out order.

In our ETH/DAI pool, 1 ETH was deposited in a $1100 price bucket. An actor who desires this trade may deposit 1100 DAI and atomically exchange their LPB to withdraw that bucket's 1 ETH. After this trade, the LPB holder who originally deposited their ETH will begin earning interest on the 1100 DAI in the bucket (assuming that their DAI is in a price bucket above the HTP).

7. Liquidation

7.1 Liquidation Criterion

Recall that loans which are not fully collateralized with respect to the *LUP* are eligible for liquidation. Mathematically, this means loans that satisfy the following inequality can be liquidated:

$$collateral_{borrow} \cdot LUP < debt_{loan}$$

Equivalently, we can isolate all of the loan-specific quantities on one side of the inequality, and encode whether the loan is eligible for liquidation in terms of its threshold price:

$$LUP < \frac{debt_{loan}}{collateral_{loan}} = TP_{loan}$$

A depositor can kick loans that would be undercollateralized if their deposit were removed from the book. This is to enable withdrawals and prevent loans with TPs just below the LUP from preventing withdrawal. If there is a loan (or even multiple loans) just below the LUP, such that a particular lender would not be able to withdraw their deposit without moving the LUP below the HTP, then that lender can kick those loans and withdraw their deposit. They can even effectively use their deposit to post their bond by obtaining a flash loan to post the bond and then repaying the loan with the withdrawn deposit.

Time passes, and interest accrues on Bob's debt (we will ignore the effect of interest on deposit for now). His prior debt of 18,034.61 has accrued approximately 16.4% interest and now is 21,000. His collateral pledged remains 20, so his threshold price is now 21,000/20=1050.
There is Carol's 20,000 deposit at a price of 1100, which is less than the total debt now, so the LUP has fallen to 1000 where Alice's deposit is. The LUP is less than Bob's threshold price, so his loan is now eligible for liquidation.

7.2 Liquidation Call

Liquidation is triggered by calling the Kick or lenderKick methods. Actors calling either method are referred to as "kickers" since they initiate ("kick") auctions. Kickers are required to post a Liquidation Bond when calling the Liquidate method. A Liquidation Bond is a quantity of supply token that, depending upon the results of the auction, could either earn additional supply token as a reward, or be forfeited in whole or in part as a penalty. Liquidation Bonds have similar payoff functions as put spreads in options trading. Once the Liquidate method has been successfully called the loan is in liquidation. The moment that a loan goes into liquidation its debt is increased by 90 days of interest, based on the current interest rate. This is done to disincentivize borrowers from repaying their loans only after it has been liquidated. The first hour of liquidation is a grace period in which the borrower may recapitalize or pay back their loan. After this hour has passed, anyone may purchase portions of the collateral via a pay-as-bid Dutch auction starting at the greater of $32 \cdot MOMP$ or $32 \cdot NP$ and exponentially decaying towards 0 with a one hour half life. The MOMP, or "most optimistic matching price," is the price at which a loan of average size would match with the most favorable lenders on the book. Technically, it is the highest price for which the amount of deposit above it exceeds the average loan debt of the pool.

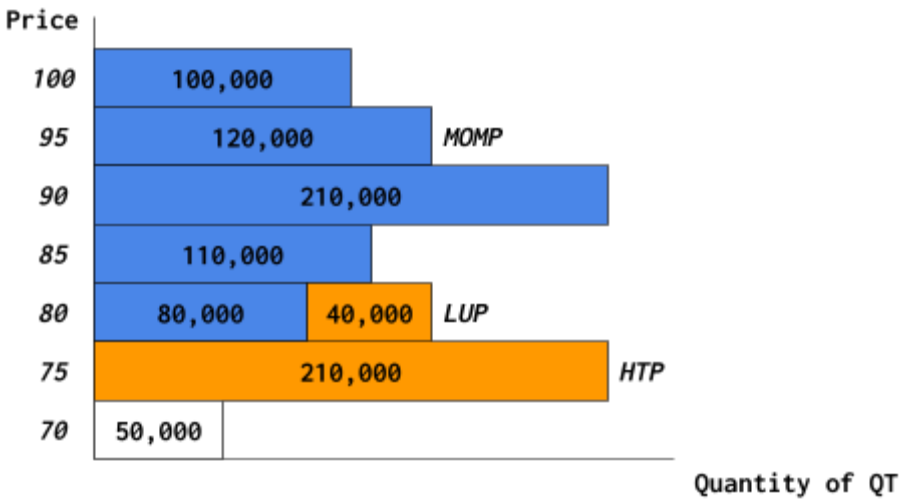
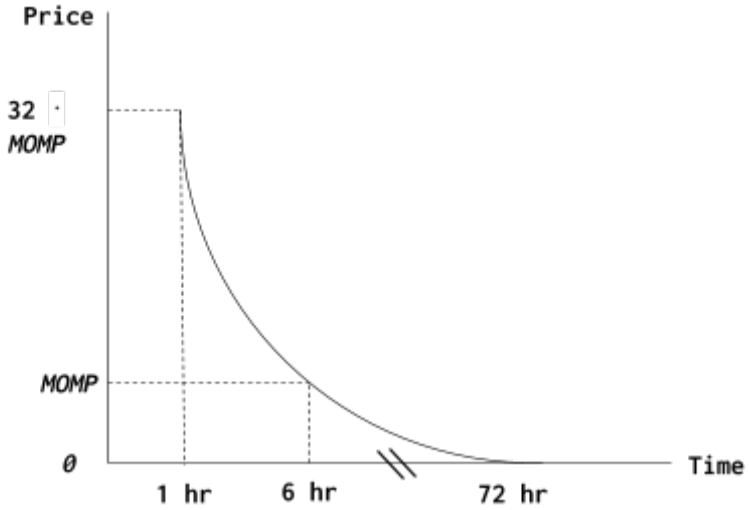


Illustration of a hypothetical pool where the average loan size is 110k QT, and therefore the MOMP is 95.

If the loan becomes fully collateralized again because of paying down debt (either due to purchases in the auction, or the borrower repaying debt), pledges of additional capital, or the *LUP* moving up above the loan's *TP*, the loan is removed from liquidation. If a loan in auction is fully collateralized after a take, arb-take or deposit-take, the loan is also removed from liquidation. It is unlikely that a liquidation would ever settle far below the market value of the collateral because once the auction is showing prices that have deposits, these deposits can be used to bid (see 7.4 Liquidation Auctions).



A graphic showing the timeline of a liquidation auction from its being "kicked" until it expires

7.3 Liquidation Bonds

In order to disincentivize spurious liquidations in an MultiLend pool, users that wish to initiate liquidations ("kickers") must post a bond priced as a percentage of the debt of the loan that they are liquidating. If the liquidation auction yields a value that is over the "Neutral Price," *NP*, the kicker forfeits a portion or all of their bond. The *NP* of a loan is the interest-adjusted *MOMP* (i.e. the *MOMP* adjusted for interest accumulated over time, see 7.3.1 Determining the Neutral Price) at the last time debt was drawn or collateral was removed. In an auction, the *NP* is the price at which a kicker will have their bond returned to them without penalty or reward. If the liquidation auction yields a value below the *NP*, the kicker will have their bond returned plus a reward.

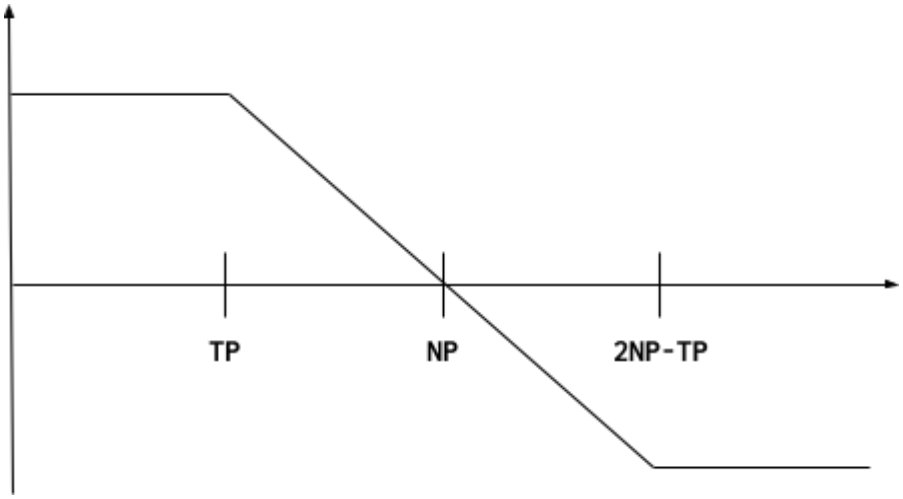


Illustration of the payoff curve for a liquidation bond
The x axis is the auction clearing price and the y axis is the profit/loss to the kicker

The goal of this mechanism is to provide as high of an economic guarantee as possible to borrowers that they will only have their positions liquidated if the ability of lenders to recover their funds is in jeopardy.

7.3.1 Determining the Neutral Price

When a loan is initiated (the first debt or additional debt is drawn, or collateral is removed from the loan), the neutral price is set to the current MOMP times the ratio of the loan's threshold price to the LUP, plus one year's interest. As time passes, the neutral price increases at the same rate as interest. This can be expressed as the following formula for the neutral price as a function of time t, where s is the time the loan is initiated.

$$NP_t = (1 + rate_s) \cdot MOMP_s \cdot \frac{TP_s}{LUP\_s} \cdot \frac{BI_t}{BI_s}$$

For each take above the neutral price (see 7.4 Liquidation Auctions), the kicker will lose some of their bond. For each take called below an auction's *NP*, the kicker will gain a reward on their bond. The *NP* can be thought of as the borrower's liquidation price, as it will be uneconomical to liquidate a borrower if the liquidation auction does not clear below this price.

7.3.2 Sizing the Bond

The size of the bond that must be posted by liquidation kickers is equal to the percentage difference between the *MOMP* and the *TP*, multiplied by the amount of debt in the loan being liquidated.

$$\text{BondFactor} = \min\{30\%, \max\{1\%, \frac{\text{MOMP}}{\text{Threshold Price}}\}\}$$

$$Bond\ Size\ (value\ in\ QT) = BondFactor \cdot Debt\ \text{loan}$$

The bond size should increase as the Threshold Price becomes more distant from the *MOMP* because this value reflects the relative collateralization of the position. If the *TP* and *MOMP* are very far apart, indicating that there is more than sufficient liquidity to absorb the average loan of the pool, the bond size will be larger. To illustrate the intended kicker calculus, if there is an asset which tends to fluctuate in price 5% per six hour period, a rational kicker should wait until the market price of the asset is more than 5% below the *NP* of the loan. This is because it will take six hours from when the auction is kicked until the price of the auction reaches the *MOMP*, where the kicker can be more certain that they may offload the debt to the pool (see 7.4 Liquidation Auctions).

Dave sees that Bob's loan is eligible for liquidation. The MOMP at the time Bob's loan origination was 1000. The NP of the loan is this price adjusted for interest: NP = 1164.43. The market price of ETH is currently 1125. As this is below the neutral price of the loan, Dave expects that he would earn a reward for kicking the loan, and calls kick. The bond factor and bond size must be determined. The current MOMP is 1000 and the loan's TP is 1050. Therefore, MOMP < TP, so the bond factor is 1%. The bond size is 1% of the loan debt, which is 210 DAI.

Dave calls kick on Bob's loan and transmits the 210 DAI. Bob's loan is now in liquidation, and the 210 DAI bond is locked in escrow until the auction is resolved.

7.3.3 Determining the Reward or Penalty

The reward or penalty given to the kicker is given by a sum of rewards/penalties over each individual *take* action. Each of these per-take contributions is proportional to the amount of supply token (which may be in the form of deposit) used in the take itself, with the factor of proportionality called Bond Payment Factor, or BPF. If $TP < NP$ then the BPF is given by:

$$\text{BPF} = \text{BondFactor} \cdot \min\{1, \max\{-1, \frac{\text{NP} - \text{price}_{take}}{\text{NP} - \text{TP}}\}\}$$

If $TP \geq NP$, then the second factor in this equation is replaced with the step function that returns +1 if the take price is less than or equal to the NP, and -1 otherwise:

$$BPF = BondFactor \text{ if } \text{price}_{take} \leq NP$$

$$BPF = -BondFactor \text{ if } Price_{take} > NP$$

The amount of collateral, C, purchased in the auction is constrained by three quantities: the amount of supply token used in the take, the amount of collateral remaining in the auction, and the amount of debt remaining in the loan. Because the bond might earn a reward, the amount of debt paid off may be less than the supply token actually accepted in the take. The formula to determine the amount of collateral purchased in the auction is:

$$C = min\left\{BPFcollateral_{loan}, Q/p, debt_{loan}/\left(p \cdot (1 - BPF^{+})\right)\right\} \quad (1)$$

where $BPF^{+} = max\{0, BPF\}$ and, Q, is the amount of supply token used in the take call. The final contribution of the take to the bond reward or penalty is:

$$Reward/Penalty = BPF \cdot C \cdot p$$

These individual per-take contributions are accumulated until the loan exits liquidation to determine the ultimate reward or penalty for the auction kicker. The most that can be lost by the auction kicker is the total amount of the bond.

Bob's grace period expires after an hour, and users can now use take to purchase Bob's collateral.

The MOMP at the time Bob's loan was kicked was 1000, so the auction price starts at 32,000. After 4.9 hours, the price has fallen to $32,000 \cdot 2^{-4.9} = 1071.77$.

Eve is a trader observing the auction, and notices that this auction price is below the prevailing market price of 1125. She attempts to buy as much collateral as she can, calling take with 25000 DAI.

To determine the outcome of this call, we first need to compute the BPF.

Recall that NP=1164.43, so

$BPF = = 0.01 \cdot (1164.43 - 1071.77)/(1164.43 - 1050) = 0.0081$ This means that for every 1 DAI worth of debt paid down by the auction proceeds, 0.0081 DAI are awarded to the kicker, Dave.

The actual amount purchased is $min\{20, 25000/1071.77, 21000/(1071.77 \cdot 0.9919)\}$ which is 19.7538 units of collateral. Therefore, of the 20 units of collateral (ETH) in the loan, 19.7538 ETH are sold to Eve, who pays 21171.5 DAI for them. Dave is awarded 171.5 DAI, Bob's 21000 debt is totally repaid, and the remaining 0.2462 units of ETH are returned to Bob.

Another Example: Suppose a loan with 1000 debt and 12.5 collateral posted (hence TP = 80) with neutral price NP=100 is liquidated. Further, suppose that the current MOMP is 95. Then the Bond Factor would be: min{30%, max{1%, (95-80)/95}} = 15.79% and the size of the bond would be: 15.79% * 1000 = 157.9 Suppose a taker attempts to purchase all 12.5 units of collateral at p=90, sending in 1125 supply tokens. BPF =15.79% min{1, max{-1, (100-90)/(100-80)}=7.895%

And C=min{12.5, 1125/90, 1000/(90(1-7.895%))}=12.0635

Note that the third element of the minimum is the constraining factor. There is not enough debt for the taker to buy all of the collateral, so they will purchase just enough to pay off the debt entirely while compensating the bondholder. The final bond reward is:

min{157.9,7.895%12.063590}=85.7174

resulting in an ROI to the kicker of:

ROI = 85.7174/157.9=54.2%

This risk/reward slope of the bond payoff curve intends to compensate kickers for triggering liquidations on loans that are less collateralized than others, and to fairly penalize them for

triggering a liquidation incorrectly. The kicker is effectively purchasing a short put option spread from the protocol where the strike price is the *NP*.

7.3.4 Bond Refund & Forfeiture Details

In the event that a liquidation is triggered and the borrower recapitalizes their position or the value of the *LUP* increases during the one hour grace period to sufficiently recapitalize the loan, the bond is returned to the kicker. When a bond is penalized because takes occur above the Neutral Price, those penalties accumulate in the contract's reserve (in the Reserve Equation, the bond posted is reduced with no other offsets. Note that when a bond is first posted there is no impact on reserves, as the contract's balance increases by the bond quantity, but this is offset by an identical increase in the bond posted).

7.4 Liquidation Auctions

While a loan is being liquidated, it continues to sit in pools, contribute to the pool level debt accumulator and accumulate interest, but is not included in the *HTP* calculation. The borrower can repay the loan in whole or part, or post additional collateral, which might make the loan solvent. It does enable a new method to be called on the loan, the *take* method. The *take* method implements a pay-as-bid Dutch auction which enables anyone to purchase portions of the collateral at a price that starts at 32 times the greater of the current MOMP or neutral price and exponentially decays with a half life of one hour. Bidders may either use supply token or deposit in the pool to participate. If a deposit is used to participate in the auction, that deposit is canceled and removed from the bucket. If there is collateral remaining in an auction after 72 hours have passed, any actor may settle the auction and claim the remaining collateral (see 7.6 Settling).

Within the liquidation auction, there are three "take" functions available to the purchaser of collateral: take, deposit take and arb take.

In the case of a take, supply token is provided by the taker and sent to the contract to pay both the bond reward (if necessary) and repay some or all of the debt in the auction. The moment that the first take is called in an auction, a 7% liquidation penalty is added to the borrower's debt. In the case of deposit take and arb take, the supply token used to purchase collateral is in the form of deposit in the contract, swapping one liability of the contract (deposit) for another (claimable collateral). The kicker's award is granted in the form of *LPB* in the bucket with claimable collateral. The deposit of the bucket is adjusted to compensate for this additional LPB, and the debt of the loan is increased (or, more precisely, reduced by less than it otherwise would be given the amount

of collateral purchased and the price) to compensate for the increase in deposit. In the event that an actor calls one of the take functions and this take would leave the loan at less than the minimum borrow size, this call will revert.

Within the liquidation auction, there are three "take" functions available to the purchaser of collateral:

7.4.1 Take

Calling take means that the purchaser is sending supply token to the auction in exchange for collateral.

An auction is kicked to recover 100 supply token of debt. Over time the pay-as-bid dutch auction reaches a price of 200 supply token per unit of collateral. Eve comes to the protocol with 100 supply tokens and calls *take* to atomically purchase the collateral, covering the borrower's debt and settling the auction.

7.4.2 Deposit take

Calling deposit take means that the caller is using existing deposit (likely utilized) from the MultiLend pool to purchase the collateral on behalf of lenders. In effect, this participant is not purchasing the collateral for themselves, but causing a trade between a lender and the borrower. For example, if the highest price bucket in the pool is 100 and the auction price is also 100, anyone may call deposit take to put the lenders in the 100 bucket with the collateral. The implication of this feature is that the pool should rarely incur bad debt, and can operate without secondary market liquidity to back up liquidations.

Instead of Eve purchasing collateral using take, she will call deposit take. After 4.9 hours, the price has fallen to $32,000 \cdot 2^{-4.9} = 1071.77$, which is below the price of 1100 where Carol has 20000 deposit. Deposit take will purchase the collateral using the deposit at price 1100 and the neutral price is 1164.43, so the BPF calculation is:

$$BPF = 0.01 \cdot \frac{0.01 \cdot 1164.43}{1164.43 - 1050} = 0.0056$$

The collateral purchased is $min\{20, 20000/(0.9944 \cdot 1100), 21000/(0.9944 \cdot 1100)\}$ which is 18.28 units of ETH. Therefore, 18.28 ETH are moved from the loan into the claimable collateral of bucket 1100, and the deposit is reduced to 0. Dave is awarded LPB in that bucket worth $18.28 \cdot 1100 \cdot 0.0056 = 112.6\ DAI$. The debt repaid is 20000, so the loan remains with 1000 debt and 1.72 units of ETH.

Because the only remaining deposit is Alice's at 1000, the LUP is now 1000. The loan is fully collateralized with respect to this LUP so the loan's auction ends, but the loan remains active.

Note that Eve received nothing for calling deposit take, so the caller would typically have other motivations. She might have called it because she is Carol, who wanted to buy the ETH but not add additional DAI to the contract. She might be Bob, who is looking to get his debt repaid at the best price. She might be Alice, who is looking to avoid bad debt being pushed into the contract. She also might be Dave, who is looking to ensure a return on his liquidation bond.

7.4.3 Arb take

This function enables actors to arbitrage an MultiLend liquidation auction against the book. If the auction price is below the highest book price, a taker could buy collateral in the auction and deposit it at the top of the book. The taker would then be able to withdraw supply token to cover the cost of the original auction purchase, with no net cost, but with some "free" (net of gas cost) excess LPB remaining in the top bucket. This operation will reduce the debt by the amount of deposit utilized, and therefore will not move the *LUP* downward.

In detail, suppose that the auction price is p with bond payment factor BPF, and there is at least Q supply token in price bucket with price $q > p$. A user could buy using Q units of supply token in the auction, put them all the resulting Q/p units of collateral into bucket q, and redeem Q/q worth of LPB in the bucket in the form of deposit (supply token). This user would then have put in zero net supply token and collateral, but would still have $Q \cdot (q - p)$ worth of LPB remaining.

There is a wrinkle, however: because the bondholder also needs to be paid in supply token, the borrower's debt is only reduced by $(1 - BPF) \cdot Q$. Because this is less than the reduction in deposit Q (at least if $BPF > 0$), this could result in the LUP moving down. To avoid this, we require that the bondholder be paid in LPB in bucket q instead of supply token. In order for that to be fair to the other LPB holders in that bucket, we need to move over some additional collateral from the

loan to compensate for that additional LPB minted. The final result is setting $Q = Deposit(q)/(1 - BPF)$ and:

$$C = min\left\{loan_{collateral}, Q \cdot p, debt_{loan}/p \cdot (1 - BPF^{+})\right\} \quad (2)$$

that we can move C units of collateral from the loan into the bucket, reduce the deposit in bucket q and the loan's debt by $(1 - BPF) \cdot Q$, issue $Q \cdot (q - p)$ LPB to the taker and $p \cdot C \cdot BPF$ to the bondholder.

Finally, we will work through the example of Eve calling arb take. As before, the price is 1071.77.

Eve will call arb take with that auction price, against the bucket priced at 1100 with 20000 deposit.

In arb take, there are always two prices in play: p=1071.77 and q=1100.

The BPF calculation is the same as in section 7.3.3, so BPF=0.0081.

The collateral purchased is (using the arb-take collateral equation above):
$min\{20, 20000/(0.9919 \cdot 1071.77), 21000/(0.9919 \cdot 1071.77)\}$

which is 18.81 units of collateral. Therefore, 18.81 ETH are moved from the loan into the claimable collateral of bucket 1100, and the deposit is reduced to 0.

Dave is awarded LPB in that bucket worth $18.81 \cdot 1071.77 \cdot 0.0081 = 163.3\ DAI$. The debt repaid is 20000, so the loan remains with 1000 debt and 1.19 units of ETH. Eve is awarded the arbitrage profit of $18.81 \cdot (1100 - 1071.77) = 531$ DAI worth of LPB in bucket 1100.
As in the take case, the only remaining deposit is Alice's at 1000, the LUP is now 1000. The loan is fully collateralized with respect to this LUP so the loan's auction ends, but the loan remains active.

Note that for NFTs the *take* function only allows an integral amount of NFTs to be purchased. When calling *deposit take* for an NFT, the asset may end up fragmented in the pool, and a non-integer amount of collateral could remain in the loan. For such a loan to be considered re-collateralized and hence removed from liquidation, the debt must be fully collateralized with respect to the LUP relying only on the integer amount of collateral remaining. In other words, the liquidation criterion discussed in Section 7.1 is replaced with:

$$floor(collateral_{loan}) \cdot LUP > debt_{loan}$$

When this occurs, the loan is moved out of auction with only $floor(collateral$ collateral, $loan$) with the remainder fractional bit of collateral moved into the claimable collateral account of the price bucket nearest the liquidation price at that time. The borrower is issued LPB in that price bucket corresponding to the value of that fractional amount of collateral, valued at the price of the bucket. This ensures that all loans not in liquidation have integer amounts of collateral in them, and that the total amount of claimable collateral in the book is an integer once all loan liquidations have settled.

In this circumstance, anyone may combine *LBP* and supply token to retrieve the full NFT. For example, if after an auction an NFT is split 50/50 between the 100 and 90 price buckets, and a user has 50 *LPB* in the 100 price bucket, they can retrieve the full NFT by redeeming their LPB and depositing 45 QT into the 90 price bucket. Because the liquidation mechanism ensures that there will always be integral amounts of collateral claimable across the book, MultiLend provides a clean method for reconstituting and claiming entire NFTs by anyone willing to provide enough supply token, with the factional NFT holding buckets (and their LPs) receiving fair compensation.

Method	Description	Q	C	Actor	Debt/Deposit/QT	Collateral	LPB (in QT)
Take	Buy Using QT	Quote Token	$\min\left\{\begin{matrix} C_{loan} \\ Q/p \\ D_{loan}/(p\cdot(1-BPF)) \end{matrix}\right\}$	Borrower	$+(1-BPF)\cdot C\cdot p$	$-C$	
				Taker	$-C\cdot p$	$+C$	
				Bond	$+BPF\cdot C\cdot p$		
Deposit Take	Buy Using Deposit	$Deposit(p)$	$\min\left\{\begin{matrix} C_{loan} \\ Q/((1-BPF)\cdot p)) \\ D_{loan}/(p\cdot(1-BPF)) \end{matrix}\right\}$	Borrower	$+(1-BPF)\cdot C\cdot p$	$-C$	
				Bucket p	$-(1-BPF)\cdot C\cdot p$	$+C$	Mint $BPF\cdot C\cdot p$
				Bond			$+BPF\cdot C\cdot p$
Arb Take	Arb Deposit in bucket $q>p$	$\frac{Deposit(q)}{1-BPF}$	$\min\left\{\begin{matrix} C_{loan} \\ Q/p \\ D_{loan}/(p\cdot(1-BPF)) \end{matrix}\right\}$	Borrower	$+(1-BPF)\cdot C\cdot p$	$-C$	
				Taker			$+C\cdot(q-p)$
				Bond			$+BPF\cdot C\cdot p$
				Bucket q	$-(1-BPF)\cdot C\cdot p$	$+C$	Mint $C\cdot q-(1-BPF)\cdot C\cdot p$

Summary of take functions

The left columns are inputs: which function is called, with what quantity of supply token

The right columns show the resulting flows of supply token, collateral and LPB among the various actors and buckets involved

7.5 Liquidation Debt

When a loan is pushed to liquidation the amount of its debt is added to the *liquidation debt* accumulator. This causes the equivalent amount of deposit to be frozen in price buckets from the *HPB* downwards. If a bucket only partially falls into this range, it is still fully frozen. For example, if a loan with 1,000,000 supply token in debt is being liquidated, the liquidation debt accumulator

will be incremented by 1,000,000 and the equivalent amount of deposit on the book will be frozen until the liquidation is complete. Assuming that there was 750,000 supply token in the 100 price bucket, 200,000 supply token in the 99 price bucket, and 100,000 supply token in the 98 price bucket, buckets 100, 99, and 98 would all be frozen until the loan was restored to good standing or the liquidation auction was settled. When a bucket is frozen, depositors cannot withdraw or move supply token from it.

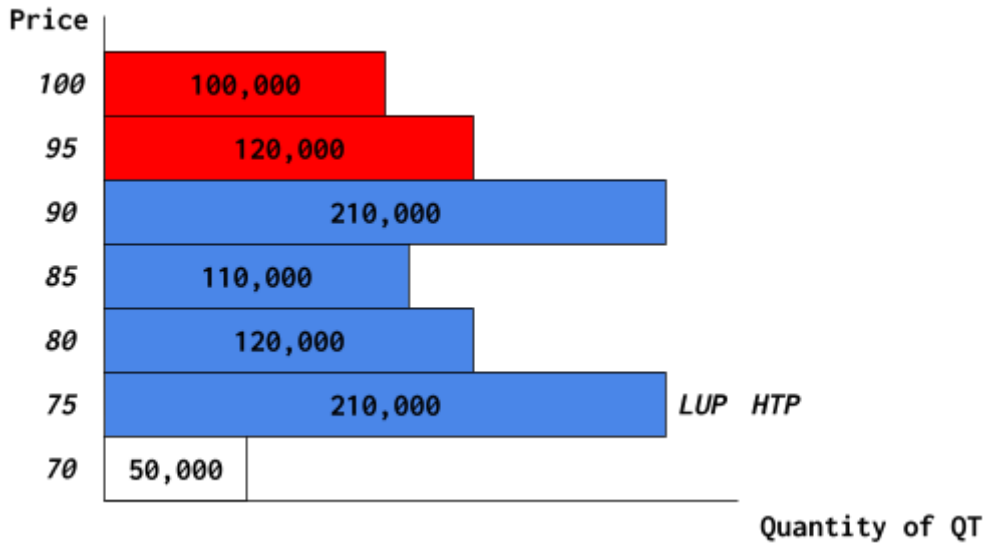


Illustration of a hypothetical pool with 110k of QT in liquidation debt, rendering the top two buckets frozen. Frozen buckets are colored red.

When kick was called, this incremented the contracts Liquidation Debt to 21,000. Because there is 20,000 in deposit in bucket 1100 and 50,000 in bucket 1000, both of these buckets are now within Liquidation Debt of the top of the book, and are frozen.

7.6 Settling

All auctions eventually become irrelevant as they run out of debt to repay, collateral to sell, or enough time has passed that the auction price has become miniscule. MultiLend forces these auctions to be removed from the set of active auctions, using a *settling* mechanism. Formally, an auction is *settleable* if any of the three criteria listed above are met (with the time limit being set to 72 hours). A settleable auction can be *settled* by any participant, which does the following:

1. If debt is zero, it's simply removed from the list of auctions. The borrower, having no debt, is then free to remove their remaining collateral
2. If the debt is non-zero and there is remaining collateral, collateral is placed into the top priced bucket's claimable collateral, reducing the deposit by the corresponding amount up to the total amount of deposit available. The loan's debt is credited the same amount as the reduction in deposit. This process is repeated until either the debt or deposit is zero.
3. If debt is non-zero but collateral is zero, the debt is reduced by min{debt, reserves} (see 3.3 Reserves). This also reduces the reserves by the same quantity, as the debt is an asset of the pool, and there is no offsetting reduction in pool liabilities. After this operation, either the debt is zero, or the reserves are zero. This operation effectively uses the reserves to protect lenders from bad debt.
4. If there is still non-zero debt in the loan, but reserves are zero and the loan has no collateral, the deposits are reduced by the quantity of debt in the loan, from the highest priced bucket downwards iteratively, with the loan's debt reduced by the same amount. This repeats until the loan's debt is zero. In this case, the most aggressive lenders are being forced to take on the remaining bad debt.

Settling the auction also finalizes the bond reward/penalty, and allows the bondholder to claim their reward and bond or what remains of the bond in the case there was a net penalty applied. In order to force all auctions to eventually be settled, the contract stores them in queue. No lender is allowed to redeem LPB if the head of the queue is settleable. Because every loan will be settled within 72 hours, this forces all loans to eventually be settled if lenders are ever going to withdraw their deposits or claim collateral. Furthermore, liquidation bondholders can't claim their rewards unless the auction settles, providing another incentive to settle auctions.

8. Interest rates

Each MultiLend pool has a global borrower interest rate, which is the annualized rate of interest paid by all borrowers on debt. The interest rate is initialized to a value between 1-10% by the pool creator. After pool creation it changes in +- 10% increments with minimum 12 hours between changes according to the pool's rate algorithm. Once initiated, rates are bounded by a 0.1% minimum and a 50,000% maximum.

Meaningful deposit MD is defined as the amount of deposit above the pool's debt-weighted average threshold price $DWATP$, less debt in auction, and is used to compute the Meaningful Actual Utilization MAU, defined as $debt/MD$. DWATP is the average threshold price of all loans in the pool, weighted by the debt of the loan:

$$DWATP = \frac{\sum_i D_i \cdot \frac{D_i}{C_i}}{\sum_i D_i} = \frac{\sum_i b \cdot T_i \cdot \frac{b \cdot T_i}{C_i}}{\sum_i b \cdot T_i} = b \cdot \frac{\sum_i \frac{T_i^2}{C_i}}{\sum_i T_i}$$

Meaningful actual utilization is the ratio of the 12 hour EMA of the debt to the 12 hour EMA of deposit above the DWATP. The debt-weighted average collateralization ratio (DWAC) of loans in a pool is used to determine the Target Utilization TU. If D_i and C_i are the debt and collateral pledged for loan i, b is borrower inflator and T_i is the t0-equivalent debt, then DWAC is:

$$DWAC = \frac{\sum_i D_i \cdot \frac{D_i}{lup \cdot C_i}}{\sum_i D_i} = \frac{\sum_i b \cdot T_i \cdot \frac{b \cdot T_i}{lup \cdot C_i}}{\sum_i b \cdot T_i} = \frac{b}{lup} \cdot \frac{\sum_i \frac{T_i^2}{C_i}}{\sum_i T_i}$$

Target Utilization, TU is the ratio of the 3.5 day exponentially weighted moving average (EMA) of the numerator of the DWAC $b \cdot \sum_i \cdot \frac{T_i^2}{C_i}$ to the EMA of the denominator $lup \cdot \sum_i T_i$.

The lambda used for the TU EMAs is $\lambda = exp\left(-\frac{\Delta t}{3.5\ days} ln\ 2\right)$, where Δt is the time elapsed since the last EMA update, so that the EMA update equation is:

$$EMA_{t+\Delta t}(x) = \lambda \cdot EMA_t(x) + (1 - \lambda) \cdot x_t$$

here, x_t could be either time series used in the definition of TU

For the actual utilization related EMAs the equation is similar, but the denominator in the exponential is 12 hours.

The interest rate, R, can be changed when MAU and TU are in a certain relationship as described below. Intuitively, when MAU is low relative to TU there is an excess of lenders relative to borrowers, and rates should be lowered. Conversely, if MAU is high then there is excess demand for borrowing, and rates should be increased.

8.1 Rate Algorithm

If *TU* is higher than *MAU* plus the "no change" window, the system will decrease rates. If *MAU* is higher than *TU* plus the "no change" window, the system will increase rates.

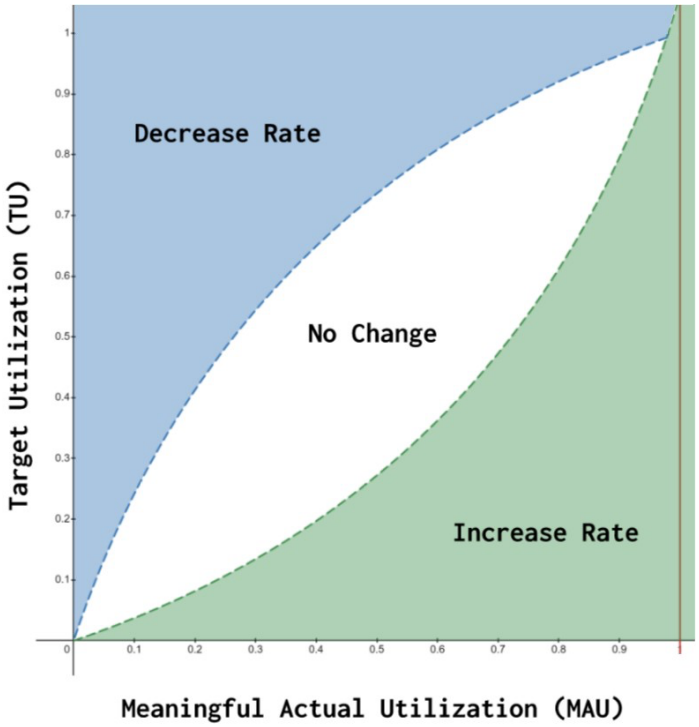


Illustration demonstrating the relationship between rates and MAU/TU, illustrating the "no change" window

The relationship between rates and *MAU/TU* is defined by the following equations:

If

$$4(tu - 1.02 \cdot mau) < (tu + 1.02 \cdot mau - 1)^2 - 1$$

then raise rates.

Else, if

$$4(tu - mau) >- (tu + mau - 1)^2 + 1$$

then decrease rates. The 1.02 coefficient ensures that rates can always be raised if the mau exceeds ~98%.

The check for rate changes occurs any time a user makes, removes, or moves a deposit, draws or repays debt, withdraws collateral, executes a trade, or triggers a liquidation. Rate changes are

implemented in +- 10% increments. If the rate is rising, the previous rate is multiplied by 1.1; if the rate is decreasing, the previous rate is multiplied by 0.9. Rates can only be changed a minimum of 12-hours after the previous rate change.

A portion of the interest accrued by borrowers is not credited to any deposit and hence, is not ultimately credited to lenders. This fraction of interest, called the "net interest margin," NIM, accrues in the reserves and is used to provide a liquidity cushion to lenders, and absorb the first bit of bad debts if some should occur. The NIM is given by the following formula:

$$NIM = (1 - MAU)^{1/3} \cdot 0.15$$

The rest of the interest is credited to lenders by scaling all deposit at or above the lesser of the HTP and the LUP by the appropriate amount. A bucket earns interest if and only if it has positive deposit above the highest threshold price of any loan in the pool or if there is less deposit above it than the current debt. Two other useful characterizations of interest earning deposits: a deposit earns interest if it can finance every loan in the book, or, if it provides liquidity to all other deposits in the sense that they can be withdrawn (including those priced above above the LUP). Observe that every unit of additional supply token deposited above the HTP enables a unit of existing supply token deposited above the LUP to be withdrawn while passing the LUP check.

8.2 Rate Reset

If the pool rate becomes too high for a protracted period, all debt could eventually exit the pool. This would result in low utilization, so eventually the pool would move back to an equilibrium interest rate, but it could take a long period of time. To return the pool to a reasonable range of market interest rates quickly, there is an interest rate reset mechanism: if the debt EMA is less than 5% of the meaningful deposit EMA and the interest rate exceeds 10%, then the rate will immediately be reset to 10%.

9. Use cases

MultiLend's no-governance, no-oracle design and permissionless nature make it an ideal integration for the DeFi ecosystem. It may be used as a composable building block where users can rely on an immutable set of contracts. A non-exhaustive list of potential use cases are as follows:

9.1 Money-market for the long tail of crypto assets

Nearly all DeFi money-market platforms today require significant governance, oracle and operational overhead to onboard new supply and collateral assets. It can take months to be onboarded, and that is assuming the assets in question have sufficient secondary market liquidity. Major platforms like Compound and Aave carry solvency risk, as all assets are cross-collateralized, while smaller platforms like Euler have isolated risk but are constrained by manual governance and Uniswap liquidity. With MultiLend users may initiate a pool with any arbitrary pairing of assets, isolating risk and skipping the cumbersome onboarding process. No secondary market liquidity outside of the MultiLend pool itself is required.

9.2 Interest earning limit orders

Current limit order books and AMMs only pay liquidity providers if they are taking active inventory risk (aka “impermanent loss”). In MultiLend, as long as a deposit is within the utilized range, those looking to purchase collateral at a discount to its current price may post an order (like in a limit order book) and be paid for providing liquidity without suffering impermanent loss. For example, if ETH is trading $2,000 and a user wishes to purchase it at $1,000, they can place a deposit in MultiLend at the $1,000 price bucket and be paid interest for lending out their stablecoins. If the price of ETH decreases to below $1,000, the user can simply do nothing and be put with the collateral.

9.3 Efficient shorting mechanism

While most money-market pools are denominated in stablecoins, the collateral and supply tokens can be flipped to create a shorting market. For example, a user can create a pool where DAI is the collateral and ETH is the supply token, and may therefore borrow ETH to achieve a short position.

9.4 On-chain convertible notes for protocols

DAOs with fundraising needs are often forced to sell protocol tokens, depressing the price. These organizations can create an MultiLend pool for their protocol token, pledge it as collateral and borrow against it, and pay back the loans with interest should they be commercially successful. If they are not commercially successful and they do not have the funds to repay the loans, the depositors will be “converted” (i.e. put with) into the protocol tokens.

9.5 Semi-permissioned pools for real world assets

When interacting with the legacy financial system, asset issuers have certain compliance obligations to their regulators. If a real world asset originator wishes to seek liquidity in an MultiLend pool, they can create a pool where the supply token is a wrapped version of a stablecoin where all holders have been KYC'ed and are therefore whitelisted to hold the wrapped stablecoin (they cannot transfer this stablecoin to non-whitelisted addresses). From here, the issuer is free to operate in the market without sacrificing compliance requirements.

9.6 Source for NFT liquidity

MultiLend pools can accept single NFTs, or NFT collections as collateral. This represents a new marketplace in which to facilitate loans against non-fungible assets. The problem with previous markets is the need for secondary market liquidity, which is nearly impossible to achieve at scale without fungibility. In MultiLend, the in-pool liquidity is sufficient for safe borrowing and lending. Additionally, MultiLend has a unique defragmentation scheme for NFTs that are "split" internally within the pool due to a liquidation auction.

9.7 Volatility market (when paired with a long put option)

By pairing an MultiLend deposit with a similarly denominated long put option (on the collateral asset) with a strike price that is the same as the MultiLend deposit's price bucket, the user is effectively long the volatility of the collateral asset. If volatility of the collateral asset increases, MultiLend interest rates would presumably increase as well, exceeding the premium paid for the long put option. Conversely, a user could wrap their MultiLend LPB and the long put option into a new token, and sell this token to short the volatility of the collateral asset.