Trade Deferral

An Overlooked Feature for Healthy Liquidity Provision

Jamie E. Young*

JEY Labs

March 2021

Abstract

In this note, we explain why trade deferral should be a fundamental feature for a healthy liquidity pool. Deferring trade settlement for a short period of time is a simple yet effective way to reduce impermanent loss for LPs as it mitigates the potential harms inflicted on long-term investors by arbitrageurs. We also explore the mechanism and challenges of implementing trade deferral on a blockchain in practice. Last but not least, we discuss a few liquidity risk checks which should be in place for *all* Liquidity Pools regardless of trade deferral implementation.

1 Introduction

The entire objective of a Liquidity Pool is to have as much liquidity as possible, often characterized by large deposit amounts and low slippage costs. The core mechanism to achieve that is to retain the LPs. Trading fees (and farming yield) are rewards to LPs as market makers to take on such risk. The biggest pain point for them – arguably the only pain point as staking is an artificial enforcement by DEX – is impermanent loss. In order to attract LPs, a Liquidity Pool's singular challenge is to reduce impermanent loss as much as possible.

That said, the simplest yet most effective way to reduce impermanent loss, is to defer trade settlements. This extremely straightforward mechanism is often overlooked. The following note discusses how trade deferral helps reduce impermanent loss and its practical challenges.

2 A Thought Experiment

To see how a trade deferral effectively reduces impermanent loss, we conduct the following thought experiment. One can generally categorize market participants into two groups: arbitrageurs and long-term investors. *Arbitrageurs* are those with "insider information" or some form of information edge. They try to make profits right before market moves, and as a result, are sensitive to execution speed. *Long-Term Investors* would like to have exposure to certain assets as part of their overall portfolios. They are not sensitive to execution speed within a short time frame and generally do not have short-term alpha. Clearly, as a (passive) LP, trading against arbitrageurs has negative expected return with high confidence, while trading against long-term investors or noise traders has zero short-term expected return. Therefore, it would be theoretically beneficial to LPs if only long-term investors or noise traders are allowed to trade in a Liquidity Pool. This is practically infeasible as a priori there is no definitive way to distinguish between arbitrageurs and long-term investors. However, if we defer trade settlements, this would essentially nullify most short-term arbitrage strategies (unless they have alpha even beyond the trade deferral period which is unlikely).

*ENS: professorjey.eth. Contact: @ProfessorJEY.

A real-life parallel to trade deferral in a Liquidity Pool is discussed in *Flash Boys* by Michael Lewis. Brad Katsuyama built IEX, an exchange with the explicit mission to level the playing field for all investors. One of the features of IEX is to have a built-in trade delay so that arbitrageurs (front-runners) would not be able to engage in predatory trading behaviors.

3 Trade Deferral

Trade deferral means at time t, a DEX only processes trade requests that were submitted at $t - \Delta\ t$ or before, but match against the most recent market at t, and once a request was submitted at $t - \Delta\ t$, it would not be cancellable and will be guaranteed to be processed at time t. This would not affect long-term investors or noise traders as much since they are generally not sensitive to execution speed (within a reasonable time frame). This way, deferring trade settlements serves essentially as a deterrence to arbitrageurs and retain the noise traders and long-term investors.

This is indeed the service that a Liquidity Pool should provide; arbitrageurs (especially front-runners) are parasites and should be excluded from the system to the largest extent possible. LPs who participate in a Liquidity Pool are there to make passive income with minimal risk. Why would they want to provide liquidity to those who would cost them (impermanent loss)?

Note that for this to work, a DEX requires a timely price oracle and a mechanism to adjust pool price according to the oracle. By doing this, it essentially separates the "liquidity provision (augmentation)" aspect from the "price discovery" aspect. The primary objective of a Liquidity Pool should be to provide liquidity. Price discovery shall be taken care of somewhere else. Any AMM with a price oracle whose primary objective is to provide liquidity rather than price discovery (typically that is the case) should consider implementing trade settlement deferral feature.

3.1 Trade Deferral in Practice

In practice, it is not trivial to implement trade deferral because there is no auto wakeup mechanism on the blockchain - someone needs to proactively send in the transaction, to be precise, call the appropriate function, and pay for the gas. One possible implementation might be as follows: when a trader sends in a trade against a Liquidity Pool, in addition to paying gas for the current transaction, she also allocates some funds into a smart contract that will be used to pay gas later for the trade settlement. Then a set of "keepers" will be constantly looking for trades that are eligible to become settled (just like blockchain miners that are constantly looking for pending transactions to be mined), where eligibility is defined as the request timestamp plus the trade deferral period (such as 5 minutes). As long as trades become eligible, keepers are motivated to "confirm" the trade by sending the corresponding transaction, paying the transaction gas with the pre-allocated funds, and taking from the remaining funds as their "mining" profits. If funds are used up before a trade is settled, the trade would be reverted, just like when gas is used up before a function call is finished. In practice, gas fees and "confirming fees" shall be deducted simultaneously; for instance, after each line of code execution, one unit of gas fee is deducted, and at the same time one unit of "confirming fees" is also deducted. The execution will terminate when either the pre-allocated funds run out, or the code has finished execution, in which case the remaining funds will be sent back to the original sender.

4 Liquidity Risk Checks

With the implementation of trade deferral, an AMM should consider various liquidity risk checks between two consecutive oracle updates to ensure a functioning Liquidity Pool and protect LPs. In fact, these checks should be in place for *all* Liquidity Pools regardless of trade deferral implementation.

1. A maximum percent r_x of \$X and r_y of \$Y can be taken out of the pool, i.e. the pool balance x_1 of \$X and y_1 of \$Y, at any time, must satisfy: $x_1 \geq (1 - r_x) \times x_0$, $y_1 \geq (1 - r_y) \times y_0$, where x_0 and y_0 are the reference balance of \$X and \$Y at the beginning of the epoch. r_x and r_y are constant (upgradable)

parameters. If a trade would result in a violation of this rule, the trade needs to revert. This refreshes after an oracle update (i.e. new epoch). Note that by setting $r_x = 100\%$ and $r_y = 100\%$, this check should be effectively turned off (never triggered).

2. A maximum absolute amount Q_x of \$X and Q_y of \$Y can be taken out of the pool, i.e. the pool balance x1 of \$X and y1 of \$Y, at any time, must satisfy: $x1 \geq x_0 - Q_x$, $y_1 \geq y_0 - Q_y$, where x_0 and y_0 are the reference balance of \$X and \$Y at the beginning of the epoch. Q_x and Q_y are constant (upgradable) parameters. If a trade would result in a violation of this rule, the trade needs to revert. This refreshes after an oracle update (i.e. new epoch). Note that by setting $Q_x = +\infty$ and $Q_y = +\infty$, this check should be effectively turned off (never triggered).

3. A maximum percent r of price fluctuation can occur, i.e. the pool price p (of \$X in terms of \$Y), at any time, must satisfy: $-r \leq \ln(\frac{p}{p_0}) \leq r$ [or equivalently, $p_0 \times e^{-r} \leq p \leq p_0 \times e^r$], where p_0 is the pool price at the beginning of the epoch (which should be the oracle price). If a trade would result in a violation of this rule, the trade needs to revert. This refreshes after an oracle update (i.e. new epoch). Note that by setting $r = +\infty\%$, this check should be effectively turned off (never triggered).

   (a) A maximum percent r of price fluctuation can occur, i.e. the pool price p (of \$X in terms of \$Y), at any time, must satisfy: $-r \leq \frac{p}{p_0} - 1 \leq r$ [or equivalently, $p_0 \times (1 - r) \leq p \leq p0 \times (1 + r)$], where p_0 is the pool price at the beginning of the epoch (which should be the oracle price). If a trade will result in a violation of this rule, the trade needs to revert. This refreshes after an oracle update (i.e. new epoch). Note that by setting $r = +\infty\%$, this check should be effectively turned off (never triggered).

5 Conclusion

We discuss a simple trade deferral mechanism as a fundamental feature of a healthy Liquidity Pool. This often overlooked yet effective feature would reduce potential profits of arbitrageurs (especially front-runners) and benefit long-term investors and LPs immensely.