DELPHI LABS

Price Oracles for Derivative Assets

Price Oracles for Derivative Assets

Jonathan Erlich
November 2022

1 Introduction

In this piece we'll analyze pre-existing assumptions on price oracles for derivative assets within the context of DeFi lending protocols. Specifically, we'll discuss why using the underlying asset's price as the oracle for a derivative asset is generally a bad practice that should be avoided. Given the prevalence of this approach, we think it's a timely and important matter. We'll focus on two types of derivative assets: bridged assets and liquid staking derivatives. A similar analysis, however, could be generalized for any type of derivative asset.

The piece is divided into three sections: bridged assets, liquid staking derivatives and concluding thoughts.

2 Bridged Assets

When a derivative asset is priced using the underlying asset as a proxy, a fundamental assumption is being made (or is at least implicit in the decision): that the two assets will follow the same price trajectory over time. And generally, this is the case. If everything is working properly the price of the underlying should track closely that of the derivative. However, this assumption breaks when things don't work as intended, which in the case of bridges, has been often. Let's explore an example to understand why.

Assume we have *bridgedETH,* a bridged version of ETH on a different L1 chain. Under normal circumstances, each bridgedETH is backed (on a 1:1 basis) by ETH

locked on a smart contract on Ethereum. However, these normal circumstances don't always hold. Particularly, as has been the case on several occasions, the bridge can get exploited and the underlying assets stolen. For the sake of the example, let's assume that's exactly what happens to our bridgedETH: an attacker exploited the bridge and was able to steal all the ETH backing the bridgedETH.

So, how would this affect a lending protocol using the price of ETH as a proxy for that of bridgedETH? Fundamentally, this means that the protocol is valuing each bridgedETH at 1 ETH, while its real price is 0 ETH (assume that this is its market price as well). This opens up the protocol to the following vulnerabilities:

1. It allows attackers to use bridgedETH as collateral to steal funds from the protocol. Any attacker would be able to buy bridgedETH from the market at ~0, deposit it in the protocol as collateral (where it would be valued at 1 ETH) and borrow (steal) other assets in an effectively uncollateralized fashion. This type of attack has already been used on different live protocols (for instance, to Hundred Finance on Moonriver and to Aave on Harmony).
2. It impedes the correct functioning of liquidations. When the market price of bridgedETH falls from 1 ETH to 0 ETH, a number of positions using bridgedETH as collateral should become liquidatable. However, given that the protocol is still valuing each bridgedETH as 1 ETH, these positions won't become liquidatable. Whether these positions would actually be liquidated if a robust oracle was used is an interesting question and would ultimately depend on how fast the market price of bridgedETH fell. This analysis, however, is outside the scope of this piece. The important point is that under the vulnerable oracle these positions don't even become liquidatable.

As a potential solution to this issue, Aave is considering incorporating Chainlink Proof-of-Reserve feeds. At a high level, these feeds would allow Aave to verify that the assets backing the bridged derivative are actually there. While it might be a viable solution, details of its implementation are still scarce to fully evaluate it. Furthermore, there might be some edge cases that this solution doesn't handle well. For instance, the underlying assets might become inaccessible or "bricked", meaning that a bug in the contract locks the assets in the contract forever. In this case, the assets are actually there, so there wouldn't be an issue with the Proof-of-Reserves. However, if the underlying assets cannot be retrieved, the derivative would be worthless. It's likely that the market price would reflect this, which would open the protocol up to the vulnerabilities explored above. While this is a low-probability event, it should be considered while evaluating the solution.

3 Liquid Staking Derivatives (LSDs)

LSDs are subject to the same vulnerabilities as bridged assets when using the underlying asset's price as a proxy for the derivative asset. Given that we already covered those concerns in the previous section, let's look at a more complex case specific to LSDs within DeFi.

Before getting into the specifics, it's worth briefly exploring how LSDs work. As its name indicates, a LSD is a liquid representation of staked assets (usually in a PoS network). Under the hood, the liquid staking protocol works as follows:

1. It receives the assets to be staked from the user.
2. It delegates them to some validators.
3. It returns a transferable representation of the staked assets to the user: the LSD.

When a user wants to withdraw the underlying asset (going from a LSD to the unstaked asset), the protocol does the following:

1. It receives the LSD.
2. It calculates how much underlying assets it should return to the user using the redemption rate. The redemption rate is just the ratio of the LSD supply to the underlying assets (i.e. it indicates the amount of underlying assets each LSD represents).
3. It unstakes the necessary underlying assets.
4. After the unbonding period, the user is able to claim the assets.

With this background, let's go back to the pricing problem. The approach some protocols have taken is using both the underlying token's price and the redemption rate to calculate the LSD price as follows:

$$LSD\ Price\ =\ Underlying\ Price\ *\ Redemption\ Rate$$

Where:

$$Redemption\ Rate\ =\ \frac{LSD\ supply}{Total\ underlying\ assets}$$

By incorporating the redemption rate into the calculation, this approach solves for the case explored in the previous section, where the underlying assets could be stolen (or slashed in the case of PoS protocols). For instance, if the underlying

assets were stolen, the redemption rate would reflect this and thus, the price would also be impacted.

Then, what's the issue with this approach? Let's use an example to explore this question. For this exercise, assume we have a LSD of ATOM called *stakedATOM,* which works in a similar way to the way we described LSDs above and has an unbonding period of 21 days. Using the pricing approach described above, the price of stakedATOM would be calculated as follows (assume we want the price in USD):

$$stakedATOM/USD \;=\; ATOM/USD \;*\; stakedATOM\ supply/Underlying\ staked\ ATOM$$

Where:

- $ATOM/USD$ is retrieved from any available source. We can assume it's accurate.
- $stakedATOM\ supply/Underlying\ staked\ ATOM$ is the redemption rate.

There's a fundamental issue with this approach: the real market price of stakedATOM can diverge from the calculated price explored above. That's because while there's a clean arbitrage opportunity when the market price of stakedATOM is above the calculated price, the same isn't true when the price of stakedATOM is below the calculated price. In other words, while the market price of stakedATOM has a hard peg to the upside (equivalent to the calculated price), it doesn't have one to the downside. To understand why, let's explore how the arbitrage opportunity works in both cases. The upside case is as follows:

- Assume the redemption rate is 1 ATOM/stakedATOM.
- The market price is 1.1 ATOM/stakedATOM (stakedATOM is overvalued in the market).
- An arbitrageur could stake 1 ATOM, immediately receive 1 stakedATOM and then sell that stakedATOM in the market for 1.1 ATOM, for a profit of 0.1 ATOM.

The above will happen until the arbitrage opportunity no longer exists. This mechanism guarantees that the market price will tend to have a ceiling equal to the calculated price. To the downside, however, this isn't the case. The reason for this is that the 21 day unbonding period breaks the clean arbitrage opportunity. An example:

- Assume the redemption rate is 1 ATOM/stakedATOM.
- The market price is 0.9 ATOM/stakedATOM.

- If there were no unbonding period, an arbitrageur could buy 1 stakedATOM for 0.9 ATOM in the market, unstake the stakedATOM and receive 1 ATOM back, for a profit of 0.1 ATOM.
- However, the unstaking period lasts for 21 days, so there's no immediate opportunity to take advantage of this arbitrage.

So while there's a hard peg to the upside, there's more of a soft peg to the downside. This soft peg means that over the long run the market price should tend to follow the redemption rate, but in the short term there isn't really a floor to the price of stakedATOM. This isn't just a theoretical or abstract finding. We've already seen this play out in reality with the infamous stETH "depeg" and other LSDs such as stLUNA. Basically, when enough people want to exit the LSD without waiting for the unbonding period the price will tend to "depeg" to the downside.

For lending protocols using the calculated price as an oracle for LSDs, the above dynamic could lead to insolvencies. Let's go over an example to understand how this can happen.

Assume we have the following position within a DeFi lending protocol:

- A user deposited 100 stakedATOM as collateral, with a maximum LTV of 70%.
- The user borrowed $600 worth of another asset using the stakedATOM as collateral.

Now, let's explore how this position evolves over three timesteps (T1, T2 and T3), depicted in the figure below:

	Timestep	T1	T2	T3
	stakedATOM			
1	ATOM price (USD)	$ 10	$ 10	$ 5
2	stakedATOM redemption rate (ATOM)	1,0	1,0	1,0
3	stakedATOM market price (ATOM)	1,0	0,8	0,8
4	stakedATOM calculated oracle price (USD)	$ 10	$ 10	$ 5
5	stakedATOM market price (USD)	$ 10	$ 8	$ 4
	Position			
6	Collateral (stakedATOM)	100	100	100
7	Maximum LTV	70%	70%	70%
8	Value of collateral in USD (oracle price)	$ 1.000	$ 1.000	$ 500
9	Value of collateral in USD (market price)	$ 1.000	$ 800	$ 400
10	Borrowed assets (USD)	$ 600	$ 600	$ 600
11	Health Factor (oracle price)	1,2	1,2	0,6
12	Health Factor (market price)	1,2	0,9	0,5
13	Collat. Factor (oracle price)	**1,7**	**1,7**	**0,8**
14	Collat. Factor (market price)	**1,7**	**1,3**	**0,7**
	Liquidation profit			
15	Liquidation Bonus	10%	10%	10%
16	Debt repaid on liquidation	$ 600	$ 600	$ 600
17	Collateral received (stakedATOM)	66	66	100
18	Market value of collateral received in USD	$ 660	$ 528	$ 400
19	Liq. Profit	**$ 60**	**-$ 72**	**-$ 200**

Figure 1: Calculation Example

- In T1, the market price of stakedATOM (row 4) perfectly reflects the calculated oracle price (row 5), so everything works well. Specifically, the health factor ($\frac{Collateral * Max.\ LTV}{Debt}$) is above 1, as well as the collateralization factor ($\frac{Collateral}{Debt}$). In other words, the account is healthy and overcollateralized.
- In T2, the market price of stakedATOM deviates from the redemption rate (and thus from the oracle price). Specifically, while the redemption rate is 1 ATOM, the market price of stakedATOM is 0.8 ATOM (a 20% discount). This has a couple implications worth highlighting:
    - The health factor calculated using the market price of stakedATOM (row 12) is no longer above 1, meaning that measured by the market price the position is unhealthy and should be liquidatable. However, given that the protocol's oracle uses the oracle price, which hasn't changed, the health factor as calculated using the oracle price (row 11) is still exactly the same as in T1 and above 1. Thus, the position won't be liquidatable.

    - Note that at this point, the position would already be unprofitable to liquidate for a liquidator (see last row). This is due to the fact that the protocol calculates the amount of stakedATOM collateral to pay the liquidator based on the oracle price, which is overvaluing stakedATOM. If the oracle used the market price instead, this position would be liquidatable *and* profitable to liquidate at this point (see [1] below for a more detailed explanation).
    - While this scenario isn't ideal, given that a position that should be liquidated isn't, it's not that bad since the position is still solvent. In other words, it's still overcollateralized (collateralization factor above 1).
- In T3, the deviation between the market price and the redemption rate stays the same as in T2, but the price of ATOM drops from $10 to $5. This would effectively lead to an insolvent position, as it wouldn't be liquidated and now the collateralization ratio has fallen below 1.

Fundamentally, the issue with this oracle methodology is that the system can become insolvent even when everything is working as intended and no manipulation has taken place, as explored above.

4 Closing Thoughts

[1] The liquidation process works as follows:

1. The liquidator repays a certain amount of debt. In the case above $600.
2. The protocol calculates the value of the collateral to be returned to the liquidator as: Debt Repaid * (1 + Liquidation Bonus) = $660.
3. Then, using the oracle price, the protocol computes the number of stakedATOM tokens to return to the liquidator as: $660/$10 = 66 stakedATOM tokens.
4. However, these tokens are overvalued and in the market each can only be sold for $8, so the real value wouldn't be $660 but rather $528 (66 * $8). Thus, this liquidation wouldn't happen as it would be unprofitable for the liquidator.
5. Using the real market price as an oracle, this wouldn't be the case. Specifically, on step 3, the protocol would compute the amount of stakedATOM to be returned as $660/$8 = 82.5 tokens (instead of 66).
6. These 82.5 tokens would be worth $660 if sold on the market, for a profit of $60 for the liquidator.

Oracles lie at the heart of DeFi lending protocols. They are so important that they often determine the future of entire protocols. A single non-robust oracle could put millions of dollars' worth of assets at risk, which is why we spend so much time analyzing oracle implementations.

With this piece, we hope to shed some light on the usage of oracles for derivative assets. Particularly, we've shown why some commonly used practices are not ideal and should be avoided. The purpose of this piece isn't to discourage listing derivative assets though, as we understand they are some of the best assets to be used as collateral. The purpose is to discourage listing them with non-robust oracles, which could lead to painful outcomes.

We understand that derivative assets tend to be less liquid than their underlying assets, and this makes it difficult to build robust oracles specifically for them. We don't think, however, that this should be an excuse to use non-robust oracles. We strongly encourage protocols to wait for liquidity to build up and robust oracles for the specific asset to be developed before using non-robust oracles.