Variance

By Dr. Chloe I Avery

Imagine you set up an Orchid account, connect to providers, and start using service, but after a while, you notice that even though you are in fact sending payments, your balance has not changed. This might feel really strange. It might feel like you are somehow accidentally scamming providers or doing something wrong. But you aren't! Probabilistic Nanopayments, the core technology to Orchid's payment system, operate essentially as lottery tickets. Instead of transferring value with every transaction, each ticket has a probability of payout, called the *win probability*, and a *face value-* how much value the recipient would receive if the ticket is a winner. The *expected value* of a nanopayment is the win probability multiplied by the face value. Money has value because we all collectively decide it does. But in this case, it is rational to agree that nanopayments have value- specifically that they are worth their expected value. Check out my previous piece for an explanation of how nanopayments work and where I argue that nanopayments are worth their expected value. This piece discusses variance- an interesting aspect of using these "lottery ticket" style of payments.

Each time a nanopayment is issued, there is a small independent probability that that ticket is a winner. The first payment you send could be a winning ticket, or, it could take a while before sending a winning ticket and drawing down your balance- essentially giving you free service for a long period of time. Just like if you're repeatedly rolling a die, it is both possible to roll a 1 on the first try, and it is possible to roll a die 100 times in a row and never get a 1. Probability (and my previous piece) says that it all works out after a while. That is, the more payments you send, the closer the average outcome will be to the expected value. But it can still feel weird, especially at first. So, what can we do about this?

First of all, a concern with variance could be that a client's balance could be drawn down faster than expected. However, if a client deposits a multiple of the payout value into their account, the higher the multiple, the less risk of the balance being depleted too quickly. For example, suppose a client is issuing nanopayments with a probability of a winning ticket being 1 in 100, and face value $1. If the client issues 100 tickets, the expected outcome is that one of them is a winner. Of course, since the nanopayments are issued with independent probability (that is, each outcome has no effect on later outcomes), it is possible that two of them are winners. However, it is less likely that a client issues 20 winning tickets out of 1000 than it is for a client to issue 2 winning tickets out of 100. More generally, as the number of tickets issued increases, the probability that the number of winners is close to the expected number of winners also increases, while the probability of extreme outcomes (like having 10 times as many winners as expected) decreases. With a higher balance in an account, a client is essentially allowing for more tickets to be issued, resulting in a higher likelihood of outcomes close to expected.

Another way to deal with variance is to utilize the tradeoff between variance and efficiency. For a fixed expected value, as face value increases, winning tickets are issued less frequently, resulting in fewer transactions on chain and therefore lower transaction fees. However, with winning tickets issued less frequently for higher payouts, there is more variance in average outcome from the expected value.

Variance happens in the nanopayment system because each nanopayment is issued independently. For example, it is possible to issue any number of winning tickets in a row, even though it may be unlikely. One way to eliminate variance is to issue mutually exclusive tickets. That is, if the probability of being a winner is 1 in 100, then exactly one ticket out of 100 issued would be a winner. To do this, instead of a two-party source of entropy (which is what the Orchid protocol currently uses to create nanopayments), an outside source of entropy would be needed, such as using the block hashes on some blockchain. The downside of this is that in order to ensure that the client does not have knowledge ahead of time of whether a nanopayment they issue will be a winner, it is necessary to delay payouts. Otherwise, after a winning ticket is issued, the Client would know that the forthcoming tickets are not winners, which has unfortunate incentive-alignment repercussions. Delaying payouts comes with its own challenges, such as storage difficulties and frustrated recipients, but it is possible.

Overall, yes variance happens. But it is tuneable (see the variance /efficiency tradeoff), and having outcomes far from expected is less likely the more nanopayments are issued. Luckily, if I am paying for something in fractions of pennies, I am likely to make many payments. Personally, I am willing to accept a little bit of weirdness in order to open up the multitude of possibilities that are afforded to me by being able to transfer tiny amounts of value.