There's a popular trend among some new blockchains to have literally zero transaction fees, instead relying on a mechanism where accounts are rate-limited based on their balance - that is, there is a global constant (or possibly dynamically adjusting value) k

such that if you have N coins, you can make at most N * k

transactions per day (or alternatively, make transactions that spend at most N * k

gas per day).

This has the marketing benefit that it allows you to claim you have "free transactions", and it arguably has the real benefit that a DoS attacker would have to temporarily buy up a lot of coins, making the attack very expensive if it actually succeeds in causing the cryptocurrency's price to drop. However, behind the appeal of "free transactions" there are a lot of hidden costs to this scheme that in my view make it definitely not worth it.

Capital lockup cost is still cost

To fairly compare the benefits of balance-based rate-limiting versus fees, and avoid accidentally introducing the completely separate discussion of consensus algorithms, let us hold the capacity of the blockchain constant - say, to 8 million gas per block. From empirical evidence and theory, it is clear that if the demand

for including transactions in the chain is less than 8 million per block, then fees can be tiny, especially if we use full proof of stake which ensures that there is no marginal cost to the miner for making a block larger as long as they publish early enough (because full PoS is not a Poisson process). Hence, the case that matters is the case where people theoretically want to use more

than 8 million gas per block.

Suppose that there is 20 million gas per block of demand (at near-zero prices), but only 8 million gas per block of supply. The rate-limiting proposal would prevent this situation by requiring accounts to hold a balance of ETH proportional to what they want to send. This would mean that 12 million gas of demand would be excluded, because they do not hold enough ETH, and many users would be holding more ETH than they would otherwise want to in order to be able to send the number of transactions that they want. This influences their financial exposure in ways that they would otherwise not want, or alternatively forces them to find ways to hedge their ETH exposure - all costs that are nevertheless suffered by participants in the system even if these costs are not officially given the label "transaction fees".

Uncertainty

The next question is, how will rate-limiting handle uncertainty. That is, suppose that there is some equilibrium value for k

such that, right now, the blockchain's capacity is 8 million gas, and there is 8 million gas worth of transactions coming in. What if either (i) global demand increases, or (ii) an individual user's need to send transactions increases?

If global demand increases, there are several solutions. The first is to set k

so low that the total budget of all ETH in the system is only 8 million gas, so usage can never exceed supply as a matter of mathematical certainty. However, given that most whales are totally uninterested in sending transactions, this would mean that the blockchain is greatly underutilized in the average case, possibly by a factor of over 100x.

The second is to adjust k

. The simplest way to do this is, if usage starts exceeding 8 million, make a transaction that would previously consume, for example, 1% of one's daily budget instead consume 1.2% of one's daily budget, and keep ratcheting the percentages up until usage comes back to normal. This essentially transforms the problem of global uncertainty into the problem of individual uncertainty - what happens if a single user suddenly needs to send more transactions than they thought they would?

Suppose you as a user expect that you will need to send ~3 transactions per day. You buy enough ETH to allow you to send 3 transactions per day. Then suddenly you discover some new app, and really love the app, and this increases your budget requirement to 6 transactions per day. You immediately (or, if the system offers some slack, quickly) need to buy up more coins to sustain your new level of usage. If the system charges transaction fees, on the other hand, then you can simply burn through your existing balance of coins more quickly, and have to make your next purchase earlier

.

Suppose that there is a spike in usage, and your use case is very-high-value. In transaction fee land, you would simply up the gasprice. In N * k transactions per day land, you would have no such easy way to economically express your desire to "be in the express lane". A forgiving version of the N * k transactions per day scheme could allow you to send 2N * k today in exchange for sending fewer tomorrow, but that does not work for the "demand over capacity" problem - if demand is over capacity, then you need to force demand down, now

, and so the only way to “raise one’s gasprice” would be to get an ultra-short-term loan for a large amount of ETH into your account.

So all in all, the N * k transactions per day scheme must choose one of two terrible options: (i) be far under capacity in the normal case so that it almost never reaches full capacity, like a fractional reserve bank with capital requirements far above the average daily amount withdrawn though still less than its full obligations, or (ii) tolerate “crunch times” with no clear bidding process for high-priority transactions.

Markets will exist whether you want them or not

If you give an agent the power to choose from a list of transactions which ones to include in the next block, transaction senders naturally have the incentive to bribe that agent to make sure they get in first. The genius of traditional blockchains is that they simply accept this effect as a given, and even deliberately formalize it into the benign market of transaction fees. But even if you don’t “officially” put a transaction fee field into the transaction data format, underhanded ways to pay transaction fees will inevitably appear.

“Transaction accelerators” run by mining pools already exist in the context of bitcoin, where high and unpredictable transaction fees, plus the lack of effective replace-by-fee infrastructure to bump up the fee of a transaction within the protocol, effectively necessitate them (lack of replace-by-fee has benefits too, like being able to more easily trust zero-confirmation transactions, though this is much less relevant in an Ethereum context because block times are much faster than bitcoin). It should be reasonable to expect transaction accelerators to appear and be in demand during congestion time on a chain running an N * k transaction per day regime as well, and for it to be de-facto mandatory for high-value users to have a contract with at least one such service.

But it gets worse. Any

user that is using less than their full transaction allotment is leaving an unused resource, and in any reasonably sophisticated smart contract system it is possible to design applications that allow one user to send transactions on another user’s behalf, perhaps by adding the signature of the user making the request into the data of the transaction, and then verifying it in contract code. Vlad Zamfir’s DAOist protocol from 2014 is an example of a design for a market for this kind of transaction proxying.

Many users hold much more ETH than they would ever need to send transactions; they would have the incentive to send such proxy transactions, even if initially at a cost of 0.000…0001 ETH / gas. If this market is reasonably efficient, then pretty soon two “market rates” would naturally emerge: the “proxying gasprice” of how much ETH you need to pay to get someone else to proxy 1 gas for you, and the “interest rate” of how much ETH of proxying fees per day you can get with a balance of 1 ETH; the ratio between these two rates would necessarily be k

(ie. the amount of gas an account with 1 ETH can consume per day). Everyone who is underutilising their transaction allotment would be participating in this proxying market, and everyone who is fully utilising theirs is implicitly participating by using their allotment to send transactions instead of earning interest by proxying. Hence, the end result is economically equivalent to a transaction fee market plus paying some interest rate to all ETH holders, which is itself economically equivalent to just having a transaction fee market - except this would happen through a proxying layer that adds inefficiencies instead of a simple transparent marketplace.

Conclusion: if you want to make blockchains affordable, the only way to do it is the obvious way: by doing hard technical work to make them more scalable, thereby lowering transaction fees through supply and demand mechanics, until eventually they are “too cheap to psychologically care about” much like every meter of driving in your car burning gasoline.