Midas
United non-colluding transaction fees for Bitcoin

By Joannes Vermorel, April 6th, 2018

Status: EARLY DRAFT, INCOMPLETE

In this article, Bitcoin always refers to Bitcoin Cash

Abstract: Transaction fees are an integral component of the Bitcoin social contract. They reward miners into playing the *long game* of Bitcoin, when the monetary mass of Bitcoin will not be growing anymore, not with economic relevance anyway. Through an analysis of Bitcoin looking inward, but even more importantly, looking outward, we demonstrate that soviet economics are required in the short term to set the transaction fees, but that the transition toward market-driven fees should and will happen in the future. The author proposes *Midas,* a pre-consensus signal intended for transaction fees, which preserves the competition within the Bitcoin mining market. Midas unifies miners through their mutual interest of preserving the security model of Bitcoin which includes microlatent transactions. Midas does not require any change to the Nakamoto consensus.

Overview

The Bitcoin social contract proposed by Satoshi Nakamoto offers a practical solution to enable peer-to-peer digital cash. However, as a consensus - the Nakamoto consensus - needs to be established by the participants, each transaction broadcast on the network incurs a cost to the ecosystem as a whole. Bitcoin does not rely on third-party agents willing to subsidize its network. Instead Bitcoin introduces the notion of *transaction fees* paid by the user to the miner precisely as a mechanism for the miner to recoup his costs, and ideally to make a profit as well.

The original Bitcoin article does not clarify how transaction fees should emerge from a practical perspective. While markets are pretty good at letting prices emerge in general, it wasn't overly clear how a *fee-market* within Bitcoin should emerge. The construction of a *proper* social contract around the emergence of this price is of primary importance. If the emergent market price cannot exhibit certain properties, it would put Bitcoin at risk of losing one of its most attractive aspects which is the security model that encompasses microlatent transactions (cf. Ansible).

Fortunately, it appears that this very problem also represents the solution to incentivize miners into a *unity without collusion* scheme where shared interests align themselves into a practical emergence of transaction fees for Bitcoin.

Midas, a peer-to-peer pre-consensus signal intended for the price structure of Bitcoin transactions, is presented in the following as a working solution to precisely achieve this.

Cloud computing platforms as a relevant analogy

Elaborating a rationale about the right pricing for transaction fees is a difficult task. As *bitcoins* represent an entire asset class of their own, the temptation is great to look *inward* within the blockchain to figure out what could possibly be the right transaction fees. An *inward* approach would, for example, try to estimate the “proper” fees based on past observed fees. However, at a conceptual level, this approach is flawed because it entirely delegates the *price making* mechanic to an undefined actor within the market. Bitcoin users may tag-along with whatever transaction fees are deemed required by the miners, assuming this price is low enough, but the point remains: within the miner’s camp, somewhat has to emerge as a *price maker*.

While analogies are a very weak form of proof, the author believes that the present market of *cloud computing platforms* (referred to as *clouds* for the sake of concision) is a relevant analogy for the transaction market, where miners are selling their capacity to include transactions into a block.

Let’s review what *clouds* and *miners* have in common:

- Infrastructure companies, driven by hefty upfront investments.
- On-demand services delivered to decentralized users.
- Distributed by necessity through microlatency requirements (cf. *Ansible*).
- Mission-critical to economy as a whole (cf. *Tokeda*).
- Long game at play, risk-averse for anything touching the security of their users.
- Incentivized to connect heavily to their peers - who happen to be competitors.

Then, the analogy isn’t perfect either because *clouds* do compete on differentiated services, while *miners* are near-perfect substitutes to each other as, by definition, they all abide to the same Nakamoto consensus. However, the similarities are so striking that the author would not be surprised if, a couple of decades from now, *clouds* and *miners* would be largely indifferentiable for all intents and purposes.

Let’s point out some characteristics of the pricing of successful *clouds* :

- Entry-level prices are exposed to the public.
- Fully metered pay-as-you-go pricing.
- Cost-plus strategy is obviously at play in most situations.

While *clouds* represents an imperfect analogy for transaction fees, unless we can conjure a rationale to justify why *miners* should adopt a radically different strategy than *clouds* nowadays, the proposition that *miners* should position themselves like *clouds* do is reasonable. It will be the starting point of multiple discussions in the following.

Short-term accidental non-linearities

The service that is requested by users from miners is the on-demand inclusion of their transactions in blocks. However, unlike *clouds* which have been very carefully engineered to be tremendously scalable, the historical implementation of Bitcoin, known as *Satoshi's client*, is not an implementation intended for unlimited scalability. That being said, the Bitcoin community is working hard to continuously improve those scalability targets.

However, in the short term, miners cannot behave like *clouds* would because, unlike *clouds*, miners are facing the risk of hitting bottlenecks which could crash their existing software implementations altogether. Unlike *clouds*, miners do not yet have the option to simply buy more hardware to cope with the extra demand as it requires a Bitcoin software that can progressively absorb additional computing resources, which has not been delivered yet[1].

This very problem is compounded by the nature of the Nakamoto consensus where miners have to follow the longest chain. Unless a block cap is in place, a *single* Byzantine miner could, over time, manage to permanently bloat the blockchain issuing terabytes of UTXO dust for no other reason but to permanently harm Bitcoin. This Byzantine miner might even manage to profit from the damage incurred by Bitcoin through stakes in a competitor of Bitcoin.

Also, at the time of this writing, the technical fine-print of Bitcoin still needs a few adjustments to make the transaction processing costs strictly linear:

- Canonical transaction ordering should replace the current partial sorting order that Bitcoin implements; otherwise very large blocks incur a larger-than-linear validation cost. This item is already in the roadmap of leading Bitcoin implementations.
- UTXO commitments need to happen in one form or another, so that the blockchain could be almost entirely discarded by miners, who would only need to store the UTXO set. This item is also already in the roadmap of leading Bitcoin implementations.
- Eschatological perspective dictates that some parts of the UTXO sets are growing forever. While this problem is unlikely to be a problem at all for *centuries*, a form of UTXO expiration should probably be considered (cf. Sakura).

Thus, at this point of time, miners are facing *accidental* non-linearities that prevent them from behaving like rational *clouds* who would dynamically adjust their hardware investments to make sure they cope with the demand.

---

[1] The author himself precisely intends to *partially* solve this very problem through his own Terab initiative. However, as the software is not ready at the time of this writing, it cannot be relied upon by the Bitcoin miner.

This explains why, at this point of time, setting the transaction fees remains at the hand of the developers of the leading Bitcoin implementations: the *miners* don't yet have the options they need to actually take this pricing matter into their own hands.

Money emissions are subsidizing transactions

At the time of this writing, the transaction fees of Bitcoin amount for less than 0.1% of the revenues of the miners. This is a commendable outcome of well-chosen settings by the software developers in charge of leading Bitcoin implementations. Indeed, as Bitcoin is still undergoing a relatively rapid inflation, the block reward incentivises miners to keep transaction fees as low as possible, even delivering *zero* fees transactions if they can vet their users somehow.

Indeed, the utmost interest of miners, who play the *long game*, is to foster worldwide adoption of Bitcoin through the two levers that are at their disposal: low transaction fees and fast transaction propagations. The author would even argue that if *miners* had a fair and secure way to invest, through mining, in the acquisition of new users, they would probably do it. The only path for miners to secure their own long-term investments is to grow the Bitcoin userbase.

However, this very incentive remains capable of harming Bitcoin as long as the accidental non-linearities discussed above remain pending. This fundamental problem *forces* software developers into adopting *soviet* economics where software developers have to take it upon themselves to both fix minimal transaction fees and to establish a transaction quota - i.e. cap the block size - to whatever they think is *securely* compatible with the limitations of their own current Bitcoin implementation.

History indicates that soviet economics are an invariable cause of dismal economic results. However, in the specific case of Bitcoin, as long as transaction fees remain extremely low, the problem can empirically be deemed *under control*. While this situation is not satisfying in the long run, it has the merit of giving the Bitcoin ecosystem at large enough time to fix the accidental non-linearities.

Let's point out that the danger of soviet economics is real for Bitcoin. Some competing forks of Bitcoin[2] have managed, through nonsensical blockchain settings defined by their own leading software developers, to cause great economic damage by letting transaction fees rise out of control, which is completely at odds with both users' interests and miners' interests.

The transition to market-driven transaction fees are the only long term viable solution for Bitcoin. However, this transition *cannot and should not happen* until miners are given the option to

[2] The author is referring to the Bitcoin Core fork of Bitcoin, as of late 2017 and early 2018.

dynamically adjust their hardware investments to match the scalability targets as defined by the market demand for Bitcoin transactions.

Public prices are the way to go

When the Bitcoin implementations will grant the miners control over their transaction fees, it will become in their best interest to claim this power taking this important matter into their own hands, as they are the ones who will be both rewarded by setting correct prices - and punished if those prices aren't good enough.

The simplest approach for miners to influence transaction fees is to publish their prices. Considering the very nature of Bitcoin, the most logical place to get those prices published is the blockchain itself, typically in the last 1000 blocks or so (cf. *Ansible* for a similar approach).

Indeed, a *layer 2* approach can be considered for the publication of those prices. Yet, if this *layer 2* happens to be mission-critical for Bitcoin, then it should become part of Bitcoin itself. There is no good reason to introduce an external point of failure within Bitcoin which is precisely intended to be failure resistant. As we will see in the following, as it only takes a couple of numbers to define a reasonable pricing strategy for transaction fees, the overhead upon Bitcoin to support this approach is minimal.

Then, public prices aren't, by themselves, a silver bullet, as the resulting market dynamics of having inconsistent prices put on display are a bit puzzling. Should the users be expected to adjust their fees according to the most expensive public price put on display by any miner in order to be sure that their transaction will be included in the next block? This would be at odds with the security model of Bitcoin which is intended to be resilient to the presence of a single dishonest miner, as a single miner would be given the power to *game* the transaction fees on its own.

Moreover, the very idea that users ought to be voting with their money for the intended latency of inclusion of their transaction into a block appears nonsensical to the author. Indeed, Bitcoin should deliver secure *microlatency* transactions (cf. Ansible). If the transaction is already secure within 10 ms, why should a *payee* be concerned by the actual settlement delay? If microlatency transactions are secure, then the *payee* is not concerned, which is exactly what participants expect from Bitcoin in the first place.

The concern of having inconsistent prices put on display can be resolved satisfyingly, as will be demonstrated in the following. Meanwhile, we still need to address why miners would abide *at all* with making their price public. The answer to this question is simple: economic actors who can afford public prices are more competitive than the ones who cannot. A casual observation of the economy at large clearly shows that the markets that are the most fiercely competitive are

all markets where prices are public[3]. Thus, miners don't *have* to put their prices on display, yet, those who don't will simply be outcompeted and pushed out of the market given enough time.

Simple parametric transaction pricing model

As Bitcoin is intended to be close to an asymptotic approximation of a perfect free market, the only pricing strategy that makes sense is a *cost-plus* pricing strategy where miners compete on an ever diminishing margin. This insight drastically simplifies the setting of the transaction fees as *cost-plus* is only a matter of establishing the *cost* of a transaction, and a sustainable[4] margin on top.

Measuring the *cost* of a transaction is somewhat complex because Bitcoin allows a wide variety of transactions to exist. Thus, any attempt to modeling the *cost* of a transaction is a necessary tradeoff between *practicality* and *precision*. The model should remain simple enough to be usable in practice. The model should be precise enough so that the miner which relies on it doesn't harm itself by doing an incorrect economical assessment of its own situation.

For any Bitcoin node, the cost incurred with a transaction can be modeled as follows:

- **A fixed cost for the transaction**: there are many operations that a Bitcoin node has to perform merely because the transaction exists, irrespectively of its content. For example, every transaction comes with a 32 byte identifier that does not depend at all on the content of the transaction.
- **A per-byte cost for bandwidth**: every Bitcoin node, upon reception of a transaction, is incentivized to relay this transaction to all its fellow peers, with efforts proportional to the estimated hashrate of every peer. Assuming that the number of relevant peers is slowly varying, this bandwidth cost is essentially linear in the number of bytes of the transaction.
- **A per-byte delta-cost for the UTXO growth**: most of the data found in a transaction can be pruned from the blockchain entirely (more details below). Yet, the Bitcoin node has to persist parts of the transaction into the UTXO set, committing itself to preserve this data essentially until the UTXO entries are claimed, which may well never happen. However, a transaction can actually generate a *net decrease* of the UTXO set, if it consumes more entries from the UTXO than it introduces.
- **A per-OpCode cost for processing**: as the transaction embeds a piece of logic, reified as sequence of OpCodes in a Forth-like language named *Script,* a Bitcoin node incurs a processing cost for the validation the transaction. The design of the Bitcoin OpCodes

[3] We leave to the current reader the task of imagining how a company could succeed at outcompeting Amazon - either the ecommerce or the cloud computing platform - while not putting any price on display.
[4] Catastrophes happen all the time: fires, floods, tornados, terrorist attacks... A miner playing the long game must anticipate that there are irreducible risks that can never be accounted for. Thus, the miner needs to maintain profits, not just to reward its shareholders, but foremost to ensure its own survival against catastrophes.

has been carefully vetted to ensure that no OpCode would introduce a supra-linear processing cost compared to the number of OpCodes. Naturally, in practice, the actual processing cost varies from OpCode to OpCode. At this point, we are adopting a simplified perspective for the sake of clarity and concision.

- **A lower limit on the satoshis within each UTXO entry**: an UTXO entry has to be kept forever in the data storage of the miners. Through this mechanism, a user - intentionally or not[5] - can put an unlimited economic burden on the Bitcoin network. Yet, the miner can counter an adversarial intent by forcing a minimal amount of satoshis to be left pending in the UTXO entry itself. Indeed, this balances the situations by putting an unlimited economic burden on the user too, because until the UTXO entry gets spent, the money is *sleeping*. The user incurs the *opportunity costs* of not putting his money to a productive use which would generate economic interests. For an eschatological perspective on UTXO costs, please refer to *Sakura*.

Based on those 5 values, the author argues that in the future - once the accidental non-linearities are solved - the *cost* associated with a given transaction will become accurately assessable by a Bitcoin node with a simple parameter model, essentially similar to the one presented above.

Thus, by merely publishing those numbers in every block it produces, a miner can signal its own price structure which covers any single potential transaction. In order to achieve the intended *cost-plus* effect, the margin is embedded in those numbers by simply scaling them accordingly.

Midas, a peer-to-peer pre-consensus signal on *fees*

We propose to introduce *Midas,* a peer-to-peer pre-consensus signaling system that is largely similar to the *Ansible* scheme. Let's immediately point out that Midas does not require any change from the Nakamoto consensus. Any transaction deemed valid by the consensus remains valid, irrespectively from the fact that it abides or not to the Midas signaling mechanism. In practice, *Midas* has so many similarities with *Ansible* that it could even be seen, and implemented, as an extension of the *Ansible* itself. For the sake of concision, we are not going to redescribe the entire scheme here, merely outline the distinctive aspects.

Through Midas, members - i.e. miners who recently produced a block which publicly embeds their Midas membership - elect a *Midas master* among themselves. This master is responsible for establishing the pre-consensus price on transaction fees. Any member, master included, can be revoked at any time through a simple majority vote of the other members.

For every one of the 5 cost parameters as introduced in the previous section, the master - like any observer of the blockchain - computes the *median* parameter price based on the

[5] An honest but uncaring user can lose its private keys, effectively freezing the UTXO entry forever.

parameters put on display by any non-revoked member of Midas. The median - rather than the average - is desirable because it delivers a robust[6] statistical estimator.

Whenever a member has any doubt of current relevant pricing parameters, the member queries the master to sign a *Midas pricing message* that contains both the pricing parameters and also a timestamp of the *master of the Ansible*. This response is propagated among Midas-capable nodes of the Bitcoin network. Whenever the *Midas master* decides that the relevant pricing parameters should be changed - merely acknowledging the consensual pricing structure put forward by Midas - the master takes the initiative of broadcasting such a pricing message on his own.

The Midas-capable nodes within the Bitcoin network abide to the pre-consensus pricing signal put forward by the Midas master. As a result, those nodes only relay transactions that exhibit fees deemed compliant with the pre-consensus signal.

From a Bitcoin wallet perspective, either a transaction propagates immediately through Midas-capable nodes, and the transaction can be deemed immediately secure; or the transaction does not propagate at all, and can be considered as rejected by the network because deemed *insecure*.

Indeed, a wallet app that lets a user propagate a transaction while there is an objectively high probability to have the transaction absent form the next block[7] is doing a disservice to its users. Users should never have to even consider the possibility that their transaction might not make it to the next block. If Bitcoin can deliver microlatent secure transactions (cf. Ansible), there is no reason to break this highly desirable property of Bitcoin through incorrect transaction fee assessments.

Unity, not collusion, through a self-fulfilling prophecy

Midas is *unifying* scheme, not a colluding one. Every miner remains capable, proportionally to his own hashrate, to get the transaction fees moving up or down. If a miner has a competitive advantage over its peers because it can operate with lower transaction fees while being increasingly profitable thanks to an overall increase of the market demand[8], then it is rational for the miner to publish prices that align with its own capacity.

Midas does not prevent miners to compete on price. Midas incentivizes miners to stand united in their competition so that one of the cornerstones of the Nakamoto consensus, which is to deliver security even for microlatent transactions, remains accessible to users at large. Honest

---

[6] See https://en.wikipedia.org/wiki/Robust_statistics

[7] There is still the case of a race condition between a propagation of the transaction and the propagation of a new block. The “next” block obviously refers to whatever block is next *after* the transaction is fully propagated.

[8] Increasing the demand by lowering the price of the product being sold is 101 economics.

miners are playing the *long game*. The preservation of the economic value of their rewards, *bitcoins*, depends on their capacity to preserve the properties of the secure model offered by Bitcoin.