RTR-TFM: A Routing Threshold-based Randomized Transaction Fee Mechanism

Anonymous Author(s)
Submission Id: 1330

ABSTRACT

Transaction Fee Mechanism (TFM) design explores the strategic interaction between users and block producers in permissionless networks. Classic TFMs give block producers control over the transactions they include in blocks, creating problems preventing collusion and achieving a socially-optimal fee equilibrium. Given previous research showing existing TFMs only eliminate these problems under strictly-limited conditions, we introduce RTR-TFM, an indirect mechanism which achieves collusion-resilience and incentive compatibility through a novel routing-based technique that derives the right to chain-extension and network payouts separately from the efficiency competing nodes demonstrate at the collection and sharing of transaction fees. This paper introduces the mechanism and demonstrates its robustness against the informational problems that prevent incentive compatibility in other TFMs.

KEYWORDS

Transaction Fee Mechanism, Leonid Hurwicz, Incentive Compatibility, Free-Riding, Collective Action Problems, Blockchain, Censorship Resistance, Collusion-Resilience, Distributed Systems

ACM Reference Format:
Anonymous Author(s). 2025. RTR-TFM: A Routing Threshold-based Randomized Transaction Fee Mechanism. In *ACM Conference, Washington, DC, USA, July 2017*, IFAAMAS, 14 pages.

1 INTRODUCTION

Transaction Fee Mechanisms (TFMs) refer to a class of distributed system in which a consensus mechanism governs the allocation of the same resource used to incentivize its provision. Unlike traditional mechanisms, where the number of honest and dishonest processes is static, in TFMs voting power is dynamic — it flows with the payouts issued by the mechanism. This introduces the ability for attacks intended to extract profits from the mechanism to compromise its stability and subvert its ability to sustain itself in an optimal equilibria.

As more and more of these attacks have been discovered in the wild, academics have named them after the "mechanism-specific" techniques they exploit, resulting in a wide array of terminology such as sybil attacks, block-orphaning attacks, selfish mining attacks, fee manipulation attacks, eclipse attacks, side-contract payments, and others. While most researchers treat these vulnerabilities as isolated technical challenges, a few scholars have applied concepts from economics and particularly mechanism design to ask whether general solutions are possible. Unfortunately, this has led to a series of impossibility results that suggest designing socially optimal TFMs may be infeasible.

This paper challenges these results by identifying the exact equilibrium in which all such attacks are irrational. It argues that three distinct types of *goal conflict* -— *self-interest*, *free-riding*, and *strategic manipulation* —- are what prevent this equilibrium from being implemented by most *TFMs*. A review the earlier work in the field then shows why the problem seems insolvable: a methodological reliance on direct mechanisms and specifically auction models limits the ability of the field to address all three types of goal conflict or even handle the informational complexity necessary to compute the required equilibria in which none apply.

In the language of mechanism design, this paper demonstrates that the social choice rule needed to achieve fee-optimality and collusion-resilience is *pareto optimality*, but the direct mechanisms used to model TFMs are incapable of implementing this rule, as doing so requires multi-dimensional preference revelation across a high-dimensional preference space – a level of informational complexity that composable algorithms cannot handle. While Maskin's Revelation Principle teaches that a direct mechanism must exist for any indirect mechanism, in this case achieving optimality requires decomposable algorithms that use the "no-trade option" to reduce the complexity of computation and limit the scope of the state transitions considered by the mechanism to those consistent with an efficiency shift towards *pareto optimality*.

Since familiarity with economics is needed to understand what goal conflict is and why the variants in *TFMs* cannot be eliminated by the direct mechanisms preferred by the field, the next section of this paper identifies the novel informational characteristics of TFMs, and shows how they create problems with self-interest, free-riding and strategic manipulation. We then discuss why *pareto optimality* is the social choice rule needed to eliminate all three, which leads to a review of the impossibility results mentioned above and a demonstration that their conclusions reflect the informational limitations of their models.

In the second half of this paper, we introduce a novel class of indirect mechanism that is theoretically compatible with *pareto optimality*. We provide the formula for this mechanism and then a game-theoretic treatment which proves its inconsistency with the impossibility results discussed earlier. We then close with a return to economic theory and explanation of how the mechanism overcomes the foundational theoretical problems identified by Samuelson and Hurwicz in the last century as the obstacles to

ACM Conference, , July 2017, Washington, DC, USA. © 2025 Association for Computing Machinery. . . . $ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
. . . $15.00

the design of decomposable mechanisms that implement pareto optimality as a social choice rule.

2 A SOLUTION

This section introduces RTR-TFM: a Routing Threshold-based Randomized TFM.

RTR-TFMis a Dutch clock auction where producers compete for the right to produce blocks through the collection of transactions and the submission of their fees into a burning mechanism. A costly lottery which follows the production of each block has the potential to resurrect and redistribute these burned fees, with this same lottery providing wrap-around sybil-resistance for the chain. The economic innovation of the approach is that it makes the production of blocks costly for participants who spend their own fees.

RTR-TFMincentives prices to move towards *pareto optimality* by punishing the two forms of activity that push any *TFM* away from optimal fee-levels: the deliberate censorship of transactions which pay theoretically competitive fees, and the inclusion of *fake transactions* whose inclusion is motivated by a desire to manipulate fee-levels.

In the section that follows, we provide game-theoretic characterization of this mechanism. This is accomplished by modelling what Hurwicz referred to as the *formula* or mathematical properties of the approach. We follow this characterization with several game-theoretic proofs that this mechanism evades the impossibility results cited previously, as an *indirect mechanism* that implements *pareto optimality* as its social choice rule.

2.1 Game Theoretic Characterization

In RTR-TFM, when users send transactions to nodes in the network, they include cryptographic routing signatures indicating the first *hop* node. Each node adds its signature as it *propagates* the transaction deeper into the network, creating within each transaction an unforgeable record of the path the transaction has taken from the user to any block producer competing to offer inclusion.

The "routing work" used to purchase blocks is derived from this chain of signatures. Specifically, the amount of routing work that is available to a producer from any transaction is given by $c \cdot \frac{1}{2^{h-1}}$, where c is a network-determined constant and h is the node's hop for that transaction. E.g., a node hearing about a transaction at its third hop receives $\frac{c}{4}$ routing work for that transaction. Each node gathers transactions until they have enough total routing work to meet a network-determined *difficulty threshold*, τ. At this time, the node may become a block producer and broadcast a block with its set of transactions whose total routing work crosses τ.

The existence of multiple nodes processing transactions allows us to model RTR-TFM as a game with a set of $m \in \mathbb{N}$ producers, $\mathcal{P} := [m]$. We consider each producer $i \in \mathcal{P}$ to be *myopic* and *strategic*. To simplify analysis, we assume that each transaction is of the same size, with each block's capacity denoted by $k \in \mathbb{N}$. Furthermore, we let $n \in \mathbb{N}$ denote the total number of users, denoted by $\mathcal{U} := [n]$. We assume that each user $j \in \mathcal{U}$ is also myopic and strategic [? ? ? ? ?]. A user $j \in \mathcal{U}$ is interested in getting a slot in the block for its transaction. Let $\theta_j \in \mathbb{R}_{\geq 0}$ denote user j's private valuation for its transaction's confirmation and $b_j \in \mathbb{R}_{\geq 0}$ as its transaction's public bid.

As is common in distributed consensus mechanisms, each block producer $i \in \mathcal{P}$ has its private copy of the set of outstanding transactions, known as *mempool*. The presence of routing signatures within transactions means that in RTR-TFM producers store both the transaction bids and the specific hop at which they received the transaction. That is, producer i's mempool is the tuple $\mathcal{M}_i = (F_i, H_i)$. $\mathcal{M}_i$ comprises the set of user bids $F_i = (b_1, \ldots, b_n)$ and their corresponding hops $H_i = (h_{i,1}, \ldots, h_{i,n})$.

This lets us define the routing work for any transaction $(b, h) \in \mathcal{M}_i$. Consider a function $\omega : \mathbb{R}_{\geq 0} \times \mathbb{N} \to \mathbb{R}_{\geq 0}$ that represents the amount of routing work gained by a block producer at the h^{th} hop. In RTR-TFM, the routing function ω is:

$$\omega(h) := c \cdot 2^{1-h} \tag{1}$$

That is, RTR-TFM offers 1^{st}-hop nodes $c \in \mathbb{R}_{\geq 0}$ units of routing work, 2^{nd}-hop nodes $\frac{c}{2}$ units of routing work, 3^{rd}-hop nodes $\frac{c}{4}$ units of routing work, and so on.

The algorithm for calculating the routing work available to block producers allows us to provide the **optimization function**, denoted by $\texttt{OPT}_{\texttt{RTR}}$, which involves each production $i \in \mathcal{P}$ selecting transactions from M_i for inclusion in their proposed blocks:

$$\left.\begin{array}{ll} \underset{S \subseteq \mathcal{M}_i}{\arg\max} & \min_{(b_t,h_t)\in S} b_t \\ \text{s.t.} & \sum_{(b_t,h_t)\in S} \omega(h_t) \geq \tau \\ & |S| \leq k \end{array}\right\} \qquad (\texttt{OPT}_{\texttt{RTR}})$$

The first constraint ensures that $S \subseteq \mathcal{M}_i$ clears the *network-determined threshold for routing work*, τ[1]. For the second constraint, recall that each transaction is of the same size. This implies that the total transactions in a block cannot exceed its capacity, $|S| \leq k$. Throughout this paper, we refer to $S \subseteq \mathcal{M}_i$ as the subset that satisfies these two constraints and $S^\star$ as the solution to $\texttt{OPT}_{\texttt{RTR}}$.

As follows, a producer $i \in \mathcal{P}$ computes $S \subseteq \mathcal{M}_i$, such that the transactions in S clear τ, or,

$$\sum_{(b_t,h_t)\in S} \frac{c}{2^{j-1}} \geq \tau.$$

In order to keep block production stable over time, the consensus mechanism adjusts τ over time to target a constant pace of block production. If fee-throughput increases, τ is increased to force blocktime back into the desired pace by making block production more expensive. If fee-throughput decreases, τ is reduced slightly to make block production cheaper.

We now progress how to payouts are issued. As distinct from other networks, the first thing that happens after a block is produced is that all of its fees are removed from circulation, and then a lottery takes place to distributing up to half of those fees back into circulation as a payout to a random network node. The burning of fees can be done in a pure implementation by having the consensus mechanism simply destroy half of the tokens collected in network fees. A more practical implementation can use a costly method of random-number generation such as hashing to power a post-block

[1]The threshold τ is a network-determined dynamic parameter and increases upon block production, and slowly decreases until the next block is produced as similar to other Dutch clock auctions, similar in principle to the role of the base fee in EIP-1559 [?]. As we consider myopic block producers and users, we omit additional details on the role of τ.

payout lottery and give miners half the block reward. Penalizing fee-throughput spikes is also helpful. Given that this paper focuses on the forumula for routing work, we set the fraction of network fees that are burned in RTR-TFMas 1/2, i.e.

$$\delta(S) := \frac{1}{2} \sum_{(b_t,h_t)\in S} p_t \tag{2}$$

As an aside, while it is not necessary for RTR-TFMto have a second-price payment rule, we adopt it here for the convenience of demonstrating UIC, since one of the purposes of this section is to show that previous impossibility results that make similar assumptions no longer apply to routing work mechanisms. Under this second-price payment rule, the payment collected from *each* user whose transactions are confirmed in S is the lowest winning bid (say) p. The total payment collected is $\frac{1}{2} \cdot |S| \cdot p$ (recall that the other half is burned).

Whether a second-price payment rule is used or not, the lottery that determines the winner of the payout begins after the production of the block. This lottery first selects a random transaction from within the block, and then a random node from within the routing paths of the selected transaction.

The revenue, $\frac{1}{2} \cdot |S| \cdot p$, collected when a block is produced is given to the winner sampled from the following distribution. All sampling is done on-chain, i.e., in a trusted manner [**?** **?**].

(1) Sample a transaction $t^\star \in S$ uniformly, i.e., $t^\star \sim \text{Uniform}(S)$.
(2) From the routing path of $t^\star$, sample a node through a probability distribution that weighs each node by their share of the routing work available at their hop over the total sum of routing work available to all nodes in the routing path of the transaction as included in the block.
   - Let the producers part of $t^\star$'s routing path be (w.l.o.g.) $\mathbf{P}_{t^\star} = \{1, \ldots, l\}$.
   - Any producer's $i \in \mathcal{P}_{t^\star}$ routing work for the transaction $t^\star$ is $\omega(t^\star; h_i)$. Likewise, the total routing work for $t^\star$ is $\sum_{i\in\mathcal{P}_{t^\star}} \omega(t_\star; h_i)$.
   - We sample a producer $i^\star \in \mathcal{P}_{t^\star}$ from the following weighted probability distribution:

$$\Pr(i^\star) \sim \frac{\omega(t^\star; h_{i^\star})}{\sum_{i\in\mathcal{P}_{t^\star}} \omega(t_\star; h_i)}$$

   - The producer $i^\star$ receives the payment $\frac{1}{2} \cdot |S| \cdot p$.

Figure 1: RTR-TFM: Revenue Lottery given S (refer $\mathtt{OPT}_{\mathtt{RTR}}$)

For an intuitive example, if a transaction is sampled that has two nodes in its routing path, the total routing work for all nodes in the routing path is $c + \frac{c}{2} = \frac{3c}{2}$. The sampling probability of the first-hop node is $\frac{c}{3c/2} = \frac{2}{3}$ while the sampling probability of the second-hop node is $\frac{c/2}{3c/2} = \frac{1}{3}$.

This allows us to define the probabiltiy of an arbitrary producer i winning the lottery, which depends on the efficiency with which it sends fees into the burning mechanism, denoted by α_i, as :

$$\alpha_i = \sum_{j=1}^{m} \Pr(\mathbb{I}_i = 1 | S_j) \cdot \Pr(S_j) \tag{3}$$

Here, the indicator variable $\mathbb{I}_i = 1$ denotes the event that producer i is selected as the winner (recipient) of the block's payment; $\mathbb{I}_i = 0$ otherwise. $\Pr(S_j)$ denotes the confirmation of the set S_j owned by the j^{th} producer.

The dynamics of the routing work mechanism provide security against classes attacks of attacks which are impossible to eliminate in other mechanisms. Producers minimize their losses in the payout lottery if they spend their own fees, but doing so also burns half of their own money. Adding transactions which have been routed by other nodes adds fees that can subsidize the unlock cost, but also introduce competing claims-on-payout from other routers that grow faster than the work provided. As our next sections will show, in a competitive dynamic this lose-lose situation dissuades rational producers from using their own money to extend the chain once the network is in equilibrium, ceteris paribus.

2.2 Incentive Compatibility

The standard way TFM papers test for UIC and MIC is to establish incentive compatibility for users following Myerson's Lemma, and then examine whether producers have an incentive to faithfully implement the mechanism assuming that the probability of block production – and thus the utility offered to users for transaction inclusion – is held constant. In this section we take the same approach to prove the impossibility results of earlier papers do not apply to RTR-TFM.

User Incentive Compatibility. As mentioned above, Myerson's Lemma [**?**] provides a condition under which any mechanism (like an auction) ensures users bid the maximum amount they are willing to pay irrespective of what every other user does. According to the lemma, the allocation rule must be monotone in the user bids, given other bids are constant. Further, it must follow the proposed payment characterization. E.g., it is well known that the generalized second-price auction (or VCG) is a special case of Myseron's Lemma and thus incentive compatible for users. The TFM literature considers the single-demand, homogeneous setting, i.e., each user has a requirement of at most one item, and all the available items are copies of a single item. The VCG auction allocates to the highest k users and charges them the $(k+1)^{th}$ bid.

In RTR-TFM, the block producers must consider both the bids and the routing work corresponding to each transaction. Due to the additional requirement of the routing work threshold, producers may not follow the standard VCG allocation. That is, the highest k bids may not clear the routing work threshold if they have propagated deeply into the network and their transactions provide less "routing work" for the production of a block. Therefore, in order to demonstrate that Myerson's Lemma holds we must first show that the proposed allocation rule is monotonic.

Lemma 2.1. *For any user $i \in \mathcal{U}$, RTR-TFM allocation rule $\boldsymbol{x}$ is monotone with respect to their bid (transaction fees), given the remaining bids $\mathcal{U} \setminus \{i\}$ do not change.*

Proof. A strategic producer selects transactions that clear the routing work threshold and satisfy the block capacity constraint, captured by $\mathsf{OPT}_{\mathsf{RTR}}$'s feasibility constraints. Note that, the routing work of any transaction is independent of the user's bids. Let $\mathcal{S}$ be the set of the subset of feasible transactions. The producer selects the subset that maximizes the minimum bid (objective of $\mathsf{OPT}_{\mathsf{RTR}}$). If a user's transaction belongs to any feasible subset, increasing the bid will have the following effect.

If the said bid is the minimum in S, increasing it will increase the chance of confirmation. It will not affect the chance of confirmation if it is not the minimum in S. Changing the bid does not have any effect if the transaction does not belong to any feasible subset (due to the constraints in $\mathsf{OPT}_{\mathsf{RTR}}$). Hence, the allocation is non-decreasing with increasing bid. □

We note that RTR-TFM has a monotonic allocation rule, it does not entirely satisfy Myerson Lemma's [**?**] payment characterization in the absence of a price-setting transaction. As we have yet to establish that it is costly for producers to include their own price-setting transactions in the block. Therefore, similar to **?**], we suggest using the minimum bid in $S^\star$ as the price-setting bid. Theorem 5.2 shows that this payment rule ensures almost URC. That is, when there are sufficient transactions and the difference between transaction pairs is small, the incentive from deviating is negligible.

Theorem 2.2. *RTR-TFM is incentive compatible for users*

Proof. We prove UIC through a case-by-case analysis.

Let $S^\star$ be the block producer's optimal subset of transactions based on the bids, computed via $\mathsf{OPT}_{\mathsf{RTR}}$. The utility to the user is the value of inclusion in the blockchain at the level of security generated by the user if they bid their true value.

Let $B = \min_{(f,h)\in S^\star} f$ be the minimum accepted transaction.

- **Case 1.** $\theta_i < B$ for any user i, if $b_i = \theta_i$ the user does not get selected in $S^\star$ and gets zero utility. If the user under-bids, i.e., $b_i < \theta_i$ the utility remains zero. Upon overbidding, i.e., $b_i > \theta_i$, the user might get selected, but the user's utility will be $\theta_i - B < 0$. For Case 1, bidding true value maximizes the utility.
- **Case 2.** $\theta_i > B$, if $b_i = \theta_i$ and $b_i \in S^\star$, i.e., the user is truthful and other constraints (independent of bid) ensures the selection of i and utility of $\theta_i - B$. As long as the bid value $b_i > B$, the user might get a utility $\theta_i - B$. If the bid $b_i < B$, the utility will be zero. Hence, the maximum utility is obtained at truthful bidding. In the other scenario where $b_i = \theta_i$ and $b_i \notin S^\star$, i.e., the user does not get included due to other constraints, the user's utility is zero. Changing the bid does not impact its inclusion; thus, the utility remains zero.
- **Case 3.** $\theta_i = B$, in this case, the user can deviate by bidding the lowest value needed to qualify for $S^\star$. Since this deviation explicitly lowers fee-throughput relative to the optimal level at which user utility is assumed, τ is lowered by consensus and the amount of utility received by the user is also lowered. As per our starting assumptions, this is a suboptimal outcome as the reduction of the fee is not costless in terms of the utility purchased and the user is in a suboptimal equilibrium – if they preferred this equilibrium they should have bid it originally as per the Revelation Principle.

This proves the theorem. □

Producer Incentive Compatibility. The standard way in which MIC is examined is to demonstrate that block producers with a temporary monopoly over block production have incentives to manipulate fee-levels. In this section we show the same assumptions other papers treat as universal limitations lead to different results in RTR-TFM. To do this, given the presence of a routing payout and the potential for strategic attacks on it, we first show that RTR-TFM incentivizes producers to propagate transactions without engaging in malicious routing strategies: either the hoarding of transactions or the addition of fake identities on the routing network. We then show that the inclusion of fake transactions is only rational if it pushes the network towards a *pareto optimal* equilibrium, and thus constitutes a form of strategic behavior that the mechanism leverages to achieve fee-optimality.

Lemma 2.3. *In RTR-TFM, routing is a Dominant Strategy over hoarding transactions for any block producer $i \in \mathcal{P}$.*

Proof. Consider four block producers, say A_1, A_2, B_1, B_2, such that A_1 and A_2 are connected (i.e., messages from A_1 reach A_2 in single hop). Also, consider B_1 and B_2 as connected. We assume A_1 and B_1 receive the same transaction as first hop nodes. Now, we examine 2 cases: (1) when B_1 hoards transactions, and (2) when B_1 routes transactions. We show that, in either case, A_1 receives a higher utility on routing than hoarding.

For the proof, we quantify $u(A_1 \text{ routes}|B_1 \text{ hoards})$ as the utility A_1 receives from routing the transaction in the event B_1 decides to hoard it. Further, $u(A_1 \text{ hoards}|B_1 \text{ hoards})$ denotes the utility for A_1 when both choose to hoard. Likewise, $u(A_1 \text{ hoards}|B_1 \text{ routes})$ and $u(A_1 \text{ routes}|B_1 \text{ routes})$ correspond to utilities for A_1 when B_1 decides to route to B_2.

Case 1: B_1 **hoards the transaction.** If A_1 hoards then the probability of A_1 and A_2 producing the block is $\Pr(A_1) = \Pr(A_2) = \frac{1}{2}$, that is, both are equally likely. Let p be the payment received, implying A_1's utility is $u(A_1 \text{ hoards}) = \frac{1}{2} \cdot p$. When A_1 propagates instead of hoarding and given $\Pr(A_1) = \Pr(A_2) = \Pr(B_1) = \frac{1}{3}$, i.e., all the three nodes involved are equally likely to produce a block, $u(A_1 \text{ routes}) = \Pr(A_1) \cdot p + \Pr(A_2) \cdot \frac{2}{3} \cdot p = \frac{5}{9} \cdot p$. Thus $u(A_1 \text{ routes}|B_1 \text{ hoards}) > u(A_1 \text{ hoards}|B_1 \text{ hoards})$.

Case 2: B_1 **routes the transaction to** B_2**.** If A_1 hoards then $u(A_1 \text{ hoards}) = \frac{1}{3} \cdot p$ where $\Pr(A_1) = \frac{1}{3}$. If A_1 decides to route to A_2, and given that all the four nodes involved are equally likely to produce the block, we get $u(A_1 \text{ routes}) = \Pr(A_1) \cdot p + \Pr(A_2) \cdot \frac{2}{3} \cdot p = \frac{1}{4} \cdot p + \frac{1}{4} \cdot \frac{2}{3} \cdot p = \frac{5}{4} \cdot \frac{1}{3} \cdot p$. Thus, $u(A_1 \text{ routes}|B_1 \text{ routes}) > u(A_1 \text{ hoards}|B_1 \text{ routes})$. □

While we can observe that forwarding transactions does modify the probability of producers proposing a block, probability analysis shows that forward-propagation is still statistically dominant. As with our section on UIC, what is really happening is that the impossibility results created by the assumption of "temporary monopoly" are overcome by the use of a work function that explicitly links fee-levels to the pace of block production.

Similar logic shows that fake transactions (producer-initiated fees) are also disincentivized under temporary-monopoly assumptions.

Fake Transactions

We consider the case of a block producer who is able to produce a block that solves $\mathsf{OPT}_{\mathsf{RTR}}$using at least a subset of the transactions in their mempool. The question is whether this block producer is advantaged by the manipulation of the set transactions in their block. We prove on a case-by-case basis that they are not by showing that there is only one situation in which fee-manipulation can be profitable and then showing that this situation is incentive compatible with *pareto optimality*:

THEOREM 2.4. *Accellerating the burn fee is the only non-losing strategy for producers*

PROOF. Let $S^\star$ be the block producer's optimal subset of transactions based on the bids, computed via $\mathsf{OPT}_{\mathsf{RTR}}$. The utility to the producer is at most the value of half of the fees in the block minus at minimum the value of the half the fees in the block that originate from the block producer.

Let $B = \min_{(f,h)\in S^\star} f$ be the minimum accepted transaction.

- **Case 1.** If the block producer eliminates *Bitnolongerhasadequateroutingworktoproduceablockandhasexpectedreducedincomefromincreasedcompetition*
- **Case 2.** If the block producer replaces *Bwithanidenticaltransaction, itsprobabilityofblockproductionisheldconstantbutitreplacesaprofitabletransaction* *replacementstrategy*.
- **Case 3.** If the block producer replaces *Bwithaself−generatedtransactionthatpaysahigherfee, alltransactionsintheblockwillpaythehigherfee.Thisistheonlyp* *generatedtransactionthathasthepotentialtobeprofitable*. □

The situation that must be analyzed to determine whether the mechanism is incentive compatible (with pareto optimality) is the third case. But note a fundamental difference between RTR-TFMand other TFMs. In this case, by attempting to increase the fees they are able to collect, the block producer is forcing up the burn-fee and pushing the network into a higher-throughput equilibrium that is more secure and more costly to re-organize. The decision to speed-up the blockchain is also tantamount to an increase in the overall supply of blockspace. Supply is expanding to fill an increase in demand.

Any strategy that accellerates the burn fee involves the block producer *subsidizing* security for the subset of users who have paid a higher fee than *Bandwhohavesignalledapreferenceforfasterinclusionatahigherrate.Thisisbydefinitionafaithfulimplementationofthemechanism, since* *levelsthatrequiresbothtomoveinunion.IfwetakethenarrowdefinitionofMICasfaithfulimplementationofthemechanismthenourproducersarebydefiniti* *levelsthatusershavealreadyindicatedisoptimal*.

Genuine Incentive Compatibility

We can move beyond "faithful implementation" and towards full incentive compatibility. To see this, observe that the decision to self-generate a transaction is rational if the block producer earns enough in profit from the transaction fees paid by other users to outweigh the costs they bear from the inclusion of their own fee-bearing transaction. Even in the hypothetical case where generating such a block is profitable, there is an explicit cost that the block producer pays in the form of accepting a lower marginal profitability.

We also note that if producers spend their own money to push up the burn fee, they increase the cost of repeating this strategy, since the cost of this strategy rises as the burn fee rises relative to the networks' volume of organic fee-throughput. Equilibrium must be reached at the point where the losses from self-generation are not larger than the profits available from the other fees included in the block. It follows that any strategic exploitation of Case 3 pushes the network into an equilibrium where supply expands until the marginal profitability of providing more blockspace matches the marginal utility of users purchasing it.

In game theoretic terms, the decision to self-generate is a strategic decision which reveals private information available only to the block producer: information about their marginal profitability given their private cost structure for attracting whatever mixture of public and private fees exists in their mempool. Producers who are efficient at gathering high-fee transactions may choose to self-generate if they fear competition or desire more rapid inclusion. But all shifts of this sort push the network a higher-throughput equilibrium in which the utility delivered to users increases relative to the fees they are paying, and larger losses must be borne to speed up the chain further.

We thus have a functioning market. Users desire transaction inclusion at the lowest rates possible. Producers desire transaction inclusion at the highest rates possibe. The mechanism induces honest revelation from users of the fees they are willing to pay for the utility that the blockchain offers at its equilibrium level as represented by the burn fee. Producers reveal private information about their own cost structure through their decision to generate a transaction that speeds up the blockchain but burns a portion of their own wallet as the cost. Truthful preference revelation and strategic behavior occurs on both sides of the market, and the network reaches an equilibrium at the point where these two forces come into balance, the point of *pareto optimality* at which neither users nor producers can profitably extract more utility from the network by allocating their resources to it in any different proportion.

3 INCENTIVE COMPATIBILITY

In the body of this paper, we demonstrated previous impossibility results simply universalize the limitations of their preferred *direct mechanisms* and introduced an *indirect mechanism* which is not subject to these limitations.

While the previous section shows the preceding impossibility results do not apply to RTR-TFM, in order to conclusively illustrate that RTR-TFMsuccessfully overcomes the informational hurdles to implementing *pareto optimality* we must return to economics and show how RTR-TFMovercomes the foundational impediments to achieving *pareto optimality* discussed in our preceding review of economic theory: Samuelson's objection based on the existence of public goods, and Hurwicz's objection on the ability for participants to costlessly subvert efficient price-discovery in the informationally decentralized environment.

Samuelson and Free-Riding

The objectio that Samuelson offered to achieving *pareto optimality* was based on the two-good equation for the *utility possibilities frontier* which describes all points at which economic production is pareto optimal. His observation was that achieving this equation and thus pareto optimality is problematic in the presence of public goods with non-excludable forms of utility: rational actors will not keep the equation that balances utility production with its cost-of-production in equilibrium if they can enjoy the benefits of goods without the need to allocate resources in payment of the costs:

$$\sum_{i=1}^{s} \frac{u^i_{pub+priv}}{u^i_b} = \frac{F_{pub+priv}}{F_b}$$

To observe how RTR-TFMsolves this problem, note that transaction inclusion in our framework is neither a private good as conceptualized by Roughgarden nor a public good as conceptualized by Fox. The fee paid for blockspace is privately-collected and can be privately-negotiated, but collective security is maximized only to the extent its existence induces competition between producers for the right to collect the fee. This is why transaction hoarding strategies typically manifest in *TFMs* with transaction fees: restricting the dissemination of transactions can limit the degree of competition for fee collection, and reduce the need for nodes to spend competitively on the security function.

RTR-TFMskirts this problem through two approaches. The first involves the derivation of the work required to produce a block directly from the transaction fee itself. This eliminates the ability for producers to hold expected utility constant while offering a lower fee to users. Producers who offer participants discounted rates through off-chain payments must add their own fees back into blocks in a separate transaction in order to make up for the shortfall in routing work that results from any underpayment.

The second reason routing work eliminates *free-riding* pressures comes from the explicit incentive it provides participants to broadcast transactions. We can see how this avoids the problem that Samuelson raised by modifying his cost function and adding a variable x that reflects the probabiliy that transactions and fees are circulating publicly, such that open competition thus exists for collection of the transaction fee:

$$\sum_{i=1}^{s} \frac{u^i_{(pub*x)+priv}}{u^i_b} = \frac{F_{priv}}{F_b}$$

Theoretically, we know that users prefer widespread distribution of their fee as this maximizes the speed of transaction inclusion. And producers prefer to have private access to fees as this improves their relative profitability. Given the fact that routing work incentivizes producers to cooperatively share transactions, we can see that these mechanisms avoid the problems Samuelson flagged with suboptimality as the equation for the *utility possibilities frontier* simplifies to the following once x becomes 1:

$$\sum_{i=1}^{s} \frac{u^i_{pub+priv}}{u^i_b} = \frac{F_{priv}}{F_b}$$

Pareto optimality is achievable in this situation since *free-riding pressures* are fully eliminated.

Hurwicz and the Incentive to Truthfulness

The objection that Hurwicz offers to achieving incentive compatibility is based on the informational need for participants to engage in a process of price discovery priour to allocating resources or computing their own utility-maximizing strategies. This is the source of Hurwicz' distinction between the "pre-exchange negotiation stage" in which participants share information and the "action stage" in which they effect the resulting trades. Hurwicz argues that this distinction is always needed to achieve *pareto optimality* as all algorithms capable of optimizing prices over time require users to form resource allocation strategies on the basis of a pre-computed understanding of the relative costs of different forms of utility.

As Hurwicz makes clear in his 1972 paper on this topic, it is consequently the lack of an "incentive to truthfulness" creates the potential for participants to engage in *strategic manipulation*. Costless misrepresentation of the informational environment is what induces others to a suboptimal allocation of their own resources. When Roughgarden and his peers argue that costless transaction inclusion is a fundamental limitation of all *TFMs*, they are offering a technologically-instantiated version of this critique, and a tautological assumption that makes it impossible to solve once incorporated into their equations.

The first way in which RTR-TFMovercomes this issue is by moving the information needed to calculate prices out of the hands of adversarial peers and into what Hurwicz called the mechanism "environment". By listing the cost of transaction inclusion listed directly in the block header in the form of the burn fee needed to produce blocks, and with a wrap-around cost for chain-extension rooted in the real-world hash expenses needed to unlock payouts, calculating the market rate for transaction-inclusion becomes a mathematical exercise that can be performed without the need for off-chain price discovery. Participants can theoretically model the market price by examining the blockchain at whatever level historical granularity is needed for the purposes of price estimation.

But don't users get environmental information from their peers? While it might seem that block producers can forge the information as a form of strategic manipulation – this is possible in many mechanisms – RTR-TFMoffers a curious design that makes this quantifiably costly.

Abstractly, we can consider *pareto optimality* as targeting an unknown price level which is most efficient at producing utility. We do not know this specific price level, but we know that we will reach this point if all transactions which are willing to pay the market rate are included, and no transactions which do not pay the market rate are included. The ability to create a mechanism that punishes both the exclusion of work funded by others and the inclusion of self-funded work thus creates a mechanism that imposes a cost on pushing prices away from their most efficient levels.

An asymmetrical cost is thus created that punishes *strategic manipulation* by making the communication of fraudulant information costly – imposed by regulating costs according to the measured efficiency of the topological channels through which the fees paid for broadcast flow – thus overcoming Hurwicz' fundamental objection and permit algorithmic compatibility with pareto optimality. The mechanism removes all attacks motivated by *strategic manipulation* by removing the ability for participants to manipulate price expectations.

On a closing note, we also observe that RTR-TFMovercomes the other barriers to achieving *pareto optimality* which are not discussed in computer science literature but nonetheless exist. The existence of a historical chain of blocks permits the user of price-discovery algorithms that require *inertia* to achieve price optimality. And we note that the presence of algorithmic smoothing in both the cost and payout functions of the mechanisms adds for slight friction in the price-adjustment process that overcomes the objections of critics like Jordan (1986).

4 CONCLUSION & FUTURE WORK

In this paper, we introduced RTR-TFM: a novel TFM that addresses the incentive misalignment in classic transaction fee mechanisms (TFMs) by introducing a novel routing-based block production rule and a revenue scheme. RTR-TFM rewards block producers in proportion to their contribution to the propagation of transactions. Such a reward ensures that block producers actively participate in the blockchain network upkeep instead of free-riding on other participating nodes. We also provide a game-theoretic characterization of the underlying game in RTR-TFM. We prove that RTR-TFM effectively discourages transaction hoarding, ensures Sybil resistance, and achieves incentive compatibility for both users and block producers under reasonable assumptions.

In addition to demonstrating these properties on the technical level, we also show that RTR-TFMaddresses the underlying informational problems that create the forms of abstract *goal conflict* that lead rational actors to launch byzantine attacks on *TFMs*, creating a new kind of informational environment more conducive to the implementation of *pareto optimality* as a social choice rule in distributed mechanisms.

Temporary page!

LaTeX was unable to guess the total number of pages correctly. As there was some unprocessed data that should have been added to the final page this extra page has been added to receive it.

If you rerun the document (without altering it) this surplus page will go away, because LaTeX now knows how many pages to expect for this document.