by Thomas, with Francesco and Barnabé - February 29th, 2024

Thanks to Luca and Julian for valuable feedback and discussions.

References

tldr

Inclusion List (EIP-7547) End to End Workflow

A detailed description of a version of inclusion lists in which the current slot inclusion list summary is committed within a block.

Concurrent Block Proposers in Ethereum

Proposed design, in which multiple proposers are elected to produce partial blocks that get concatenated to construct the final payload of a block.

Inclusion lists: execution, consensus, & engine spec overview

Latest iteration on single-proposer inclusion list specifications.

Unconditional inclusion lists

Mapping out the design space to evaluate trade-offs between unconditional and conditional inclusion lists.

Fun and games with inclusion lists

Foundational model for economic games (e.g., block stuffing, subsidizing transactions) being played by proposers around ILs

No free lunch – a new inclusion list design

Most recent Vanilla Forward IL proposal

Cumulative, Non-Expiring Inclusion Lists

Cumulative, non-expiring forward IL proposal

Spec'ing out Forward Inclusion-List w/ Dedicated Gas Limits

Specs and implementation of “Forced” ILs (i.e., no block stuffing allowed)

TLDR

Today, a majority of builders and relays censor Ethereum transactions interacting with sanctioned addresses. Advancements in protocol research have led to new designs and specifications for single-proposer inclusion lists, allowing for the improvement of the network's censorship resistance properties by enabling proposers to force the inclusion of transactions in subsequent blocks. However, relying on a single entity to construct inclusion lists introduces potential vulnerabilities to commitment attacks. In this post, we introduce committee-enforced inclusion sets (COMIS): A candidate implementation of a multiplicity gadget on Ethereum, which assigns the construction of an inclusion set to a committee composed of multiple parties, rather than a single proposer. We examine the feasibility of implementing COMIS considering the following properties: Inclusion guarantees, accountability, robustness to commitment attacks, consensus load and overall complexity.

Introduction

Today, 71% of builders and 53% of relays censor Ethereum transactions interacting with sanctioned addresses. In a Proposer-Builder Separation (PBS) world, relying on a few sophisticated parties for block production weakens the network's censorship resistance (CR) properties and calls for action at the protocol level. There has been substantial advancement in the study of inclusion lists (IL), which allow proposers to force include transactions in subsequent blocks. An Ethereum Improvement Proposal (EIP-7547) has been drafted and is being considered for inclusion in the upcoming Electra upgrade (see most recent spec overview here). However, the designs proposed thus far assign the responsibility of constructing the IL to a single entity—the proposer. This property may lead to bribing and extortion attacks, as compromising the proposer to manipulate transaction inclusion becomes economically viable.

In this post, we explore the feasibility of adding a Multiplicity gadget' and assigning the responsibility of constructing an inclusion set (IS

) to a committee composed of multiple parties, rather than to a single proposer.

We hope to provide “something in between the vague explanation of a paper and the exact explanation of a spec”, by mapping out the Multiplicity design space for inclusion sets on Ethereum and highlighting features and trade-offs through a candidate implementation.

If you wish to participate in this line of research, consider applying toROP-9: Multiplicity gadgets for censorship-resistance.

High-level view

Every slot n

, a committee of size m

is randomly chosen amongst validators to construct an global inclusion set IS_n

composed of transactions t

pending in the mempool. If IS_n

is made available on time, the validity of blocks B_{n}

and B_{n+1}

is conditioned on including IS_n

summary and transactions filling its requirements, respectively.

Desired properties

We outline essential factors to be evaluated in multiplicity design proposals.

Inclusion guarantees:

The mechanism should aim to maximise inclusion guarantees.

- Inclusion threshold (IT

):

The Inclusion Threshold (IT

) represents the minimum proportion of committee members that must select a transaction for it to conditionally affect the validity of consecutive blocks B_{n}

and B_{n+1}

. If a transaction must appear in more than one-fifth of all transaction sets constructed by committee members to be guaranteed to influence block validity, IT = 4/5

. Importantly, IT

must be considered in light of the tradeoff between liveness and safety: A high IT

increases chances that a given transaction will be included in the final IS

, regardless of its absence in other committee members’ local sets. However, an increased IT

also elevates the risk that the final IS

could be deemed invalid if committee members are offline or didn’t see the transaction on time. IT

also plays an important role in the short-term censorship resistance guarantees a given mechanism provides: Having very strong inclusion guarantees could allow certain applications relying on those guarantees to be built. Further investigation into how varying IT

levels affect the tradeoffs between inclusion guarantees and liveness should be conducted. Also note that IT

is conditional on the committee aggregator and proposers (for slot n

and n+1

) to stay online.

- Validation Method

: The mechanism may rely on consensus rounds or block validity conditions, or leverage both, to validate local sets and ISs constructed by committee members.

- Information propagation:

The decision on whether information regarding IS

construction should be gossiped through specific subnets or made available to all validators.

- [Unconditional vs Conditional

](https://ethresear.ch/t/unconditional-inclusion-lists/18500): IS_n

EITHER

acts as a “payload extension” appended to the slot n+1

(unconditional) OR

conditions the enforcement of IS_n

transactions on the existence of extra unused gas in the slot n+1

payload (conditional).

Accountability

The mechanism should identify and differentiate between IS

committee members who censor and those who do not.

- Incentive schemes

can be constructed to reward parties that provide useful inputs to the mechanism or penalise those who do not. Higher levels of accountability allow for more control over incentives such as: * Conditional tipping: IS

transaction tips (e.g., repurposed priority fees) are only distributed across committee members that included these transactions in their local sets. Another idea is to scale conditional tips relative to base fees (h/t Barnabé), so that when demand for block space is high, censoring transactions becomes more expensive

- Weighted rewards: Knowing which proposer included which transactions enables rewards to be weighted according to how many other committee members have included each transaction. This leads to greater incentives to include transactions no one else wants to include (e.g., “censorable transactions”) and helps prevent discouragement attacks. Another idea is to weight rewards based on past performance: If your local set has a weak overlap with those of others, your rewards will be reduced in future slots and vice versa.

- Conditional tipping: IS

transaction tips (e.g., repurposed priority fees) are only distributed across committee members that included these transactions in their local sets. Another idea is to scale conditional tips relative to base fees (h/t Barnabé), so that when demand for block space is high, censoring transactions becomes more expensive

- Weighted rewards: Knowing which proposer included which transactions enables rewards to be weighted according to how many other committee members have included each transaction. This leads to greater incentives to include transactions no one else wants to include (e.g., “censorable transactions”) and helps prevent discouragement attacks. Another idea is to weight rewards based on past performance: If your local set has a weak overlap with those of others, your rewards will be reduced in future slots and vice versa.

- In the end, we might want to use both mechanisms so that (1) users sending transactions pay for its inclusion by committee members, and (2) issuance is used to reward aggregators and proposers and increase the overall costs of censorship (see Rainbow staking post).

Robustness

The mechanism should be robust to commitment attacks and games played to encourage or deter actors from engaging in censorship.

- Any party involved in constructing or validating the IS

is subject to bribery to ensure the IS

is kept empty, invalid, or to include certain transactions.

- Any party (e.g., a censoring proposer) tasked with specific duties, whose successful completion and associated rewards depend on the actions of others, is subject to extortion.

Bandwidth consumption

:

The mechanism should not overload Ethereum's bandwidth consumption.

This can be controlled by:

- IS

size

: The number of transactions per IS

and/or a gas or byte-size limit.

- Committee size (m

)

: The number of parties involved in constructing and/or validating $IS$s.

- Transactions

gas or byte-size limit. Ideally, the mechanism should improve CR properties equally for all types of transactions: * calldata

-light transactions, such as simple swaps or transfers consuming regular gas.

- calldata

-heavy transactions, like transactions which post data from rollups.

- calldata

-light transactions, such as simple swaps or transfers consuming regular gas.

- calldata

-heavy transactions, like transactions which post data from rollups.

Complexity

The mechanism should be compatible with Ethereum's current architecture and planned upgrades, and minimise changes to the code base.

Design

Description:

In the following section, we introduce a candidate implementation of the multiplicity gadget for ISs

on Ethereum.

[

Screenshot 2024-02-08 at 16.18.24

1610×684 100 KB

](https://ethresear.ch/uploads/default/original/2X/5/5fd9c7fce350e54ba1d759ba14048645e76b042b.png)

1. Every slot, a 256

member committee is chosen at random to participate in the construction of an IS

, with one member being appointed as the committee aggregator (1/256

).

1. The remaining committee members are responsible for constructing local sets of transactions based on their local view of the mempool, and gossip them via a dedicated inclusion_set

subnet exclusive to committee members.

1. If IT = 4/5

, the committee aggregator's duty consists in constructing a single, deduplicated IS

that contains at least 4/5

of local sets shared in the inclusion_set

topic. This constraint ensures that the committee aggregator is unable to exclude a transaction present in more than 1/5

of the submitted bundles to build a valid IS

, including:

– A list of raw transactions corresponding to summary entries.

– A summary, consisting in a list of (address

, gasLimit

) pairs, which specify the from

and gasLimit

for each transaction across all local sets. Each pair is referred to as an Entry

, and is associated by a bitlist over the committee to represent which committee members included which transactions. The summary also includes signatures from all committee members.

The committee aggregator then broadcasts the IS

and its associated list of transactions on the global topic. If IS

is available:

1. B_{n}

validity is conditioned on including the committee aggregator's summary in the BeaconBlock

built by proposer n

.

1. B_{n+1}

validity depends on including transactions filling the requirements of the summary included in slot n

BeaconBlock

.

Evaluation

Inclusion Guarantees

: Given the rule that a transaction must appear in more than one-fifth of the local sets (when IT = 4/5

) for the aggregator to include it in the final IS

, this design inherently assumes a degree of validator honesty and diversity. If a large majority of validators engage in censorship, this behavior could dominate the committee's decisions. Setting IT

carries significant implications, and its impact must be thoroughly investigated.

High thresholds improve inclusion guarantees: At the extreme, IT = 1

means that every single transaction in the local sets from all 256 committee members needs to be included by the committee aggregator for its IS

to be considered valid. However, this comes at the cost of liveness, and one committee member failing to send its local set on time would invalidate the IS

. Inclusion guarantees also depend on the committee aggregator and the proposers for slots n

and n+1

being online.

Accountability

: The process of creating a single deduplicated set of transactions accompanied by bitlists over the committee facilitates the calibration of incentives based on the specific transactions each member includes, and allows incentives schemes like conditional tipping or weighted rewards. In COMIS, we suggest to incentivise:

- Committee members

with weighted rewards based on their contributions. We propose to allocate some amount of the aggregate issuance (e.g., similar to rewards attributed to sync committee duties) of the protocol to committee members for including transactions in their local sets before sending them to the committee aggregator. Note that any local sets not included by the committee aggregator, whether intentionally or not, will be excluded from the reward calculation. In section 1 of the appendix, we propose a simple reward distribution model to compute each member's allocation given the quantity (i.e., count) and uniqueness of transactions included in their local sets.

- The committee aggregator

for packaging a valid IS

. The compensation for this role should be significantly high, suggesting an allocation of a portion of the total issuance to discourage censoring behaviour, whether being elicited by bribery or regulatory concerns. Another possibility is to elect several committee aggregators per committee to increase IS

construction guarantees without necessarily having to incentivise this behaviour with rewards. The limits of this approach lie in the amount of data that would have to be gossiped in the global topic.

- Attesters already receive rewards for making attestations according to their view of the chain, and proposers of slot n

and n+1

receive rewards associated with block proposal.

Robustness:

To evaluate the robustness of our mechanism, we can evaluate the costs of commitment attacks on parties involved in constructing and validating the IS

. A detailed analysis is beyond the scope of this post, but a good way to think about it is to use models from the Fun and Games post and use simulations to:

- Identify parties most susceptible to extortion or bribery given their capacity to censor. In COMIS, a committee aggregator has significantly more power than any single member of the 256-person committee, making it potentially more prone to targeted attacks, or to be an attacker.
- Estimate the cost of commitment attacks on each of these parties and tune the incentives accordingly. Using historical data, we can also assess the frequency with which transactions generate enough MEV to justify the execution of such attacks.

Consensus load

: The responsibility for constructing the IS

is assigned solely to the committee members and the committee aggregator. This ensures that only a select number of validators are tasked with the extra duties of monitoring the mempool and constructing local sets, which are then aggregated into a unified IS

. Nonetheless, the IS

is then propagated in a global topic across all attesters, which requires certain bandwidth provisions and imposes restrictions on the size of the IS

. We suggest setting the IS

cap at 2M gas, accommodating up to 100 transactions (for standard 21000 gas transfers). This would necessitate an additional 60 bytes per committee member (48 bytes for each signature plus roughly 12 bytes for bitlist entries), leading to an increase of about 10 KBs in the BeaconBlock size with 256 committee members involved.

FAQ

- When is the IS constructed by committee members?

- In COMIS, the committee members should start sending their local sets at the beginning of slot n (t = 0s), and the final IS must be constructed and broadcast before a set deadline in slot n (e.g., t = 4s), since the inclusion of its summary by proposer n conditions the validity of B_n.

- In COMIS, the committee members should start sending their local sets at the beginning of slot n (t = 0s), and the final IS must be constructed and broadcast before a set deadline in slot n (e.g., t = 4s), since the inclusion of its summary by proposer n conditions the validity of B_n.

- Where should the IS be appended relative to the BeaconBlock payload?

- While the summary of the IS

should be included in B_n

, transactions satisfying the summary could be appended anywhere in B_{n+1}

, at the top or at the bottom of block B_{n+1}

(see [bottom-of-next-block property] section in the unconditional inclusion lists FAQ for more details). The three options come, as always, with a tradeoff between imposing constraints on the block structure and limiting risks relative to commitment or the rise of secondary markets for ISs. We think giving builders more freedom by letting them insert transactions anywhere in the block might be a good middle ground: More freedom given to builders, and the uncertainty about where transactions will end up being inserted in B_{n+1}

seem to be good features, but more research is needed on that front.

- While the summary of the IS

should be included in B_n

, transactions satisfying the summary could be appended anywhere in B_{n+1}

, at the top or at the bottom of block B_{n+1}

(see [bottom-of-next-block property] section in the unconditional inclusion lists FAQ for more details). The three options come, as always, with a tradeoff between imposing constraints on the block structure and limiting risks relative to commitment or the rise of secondary markets for ISs. We think giving builders more freedom by letting them insert transactions anywhere in the block might be a good middle ground: More freedom given to builders, and the uncertainty about where transactions will end up being inserted in B_{n+1}

seem to be good features, but more research is needed on that front.

- Is COMIS compatible withsingle-proposer inclusion lists?

- Yes! We think these two mechanisms are complimentary, and single-proposer inclusion lists can act as a fallback mechanism in case the committee aggregator intentionally or unintentionally didn’t propose the IS

on time.

- Yes! We think these two mechanisms are complimentary, and single-proposer inclusion lists can act as a fallback mechanism in case the committee aggregator intentionally or unintentionally didn’t propose the IS

on time.

- How does COMIS differ from theConcurrent Block Proposer (CBP) design?

- The goals are very similar: Trying to limit the “unilateral authority a proposer has over the set of transactions included” in a given block. However, the main difference resides in where the aggregation of multiple inclusion sets happens. In COMIS, the inclusion set in constructed beforehand by the committee members, and conditions the block validity of proposer n

and n+1

if available. In CBP, two or more proposers “produce a block concurrently, collaboratively building the final payload by concatenating the constituent blocks”: this concatenation and deduplication happens later on and affects voting and weights given to partial blocks during consensus (e.g., proposer boost), leading to (1) potentially complex voting calculations required when one or more partial blocks are missing (see proposer boost section), (2) a need to solve for conflicts between transactions included in different payloads, and (3) bandwidth requirements increasing exponentially relative to the number of concurrent proposers involved in construction the final block.

- The goals are very similar: Trying to limit the “unilateral authority a proposer has over the set of transactions included” in a given block. However, the main difference resides in where the aggregation of multiple inclusion sets happens. In COMIS, the inclusion set in constructed beforehand by the committee members, and conditions the block validity of proposer n

and n+1

if available. In CBP, two or more proposers “produce a block concurrently, collaboratively building the final payload by concatenating the constituent blocks”: this concatenation and deduplication happens later on and affects voting and weights given to partial blocks during consensus (e.g., proposer boost), leading to (1) potentially complex voting calculations required when one or more partial blocks are missing (see proposer boost section), (2) a need to solve for conflicts between transactions included in different payloads, and (3) bandwidth requirements increasing exponentially relative to the number of concurrent proposers involved in construction the final block.

Appendix 1: Reward Distribution Model

We propose a simple implementation for weighted incentives by defining a function to calculate rewards for a given member, factoring in the quantity (i.e., count) and uniqueness of transactions included in their local sets.

Variables

- R

: Total reward pool available for distribution among committee members.

- T

: Total number of transactions included in the block or epoch by all members.

- t_i

: Number of transactions included by member i.

- $u(t)$

: A function that returns the uniqueness weight of a transaction t

, which could be defined as the inverse of the number of members including that transaction. For example, if a transaction is included by 5

members, its uniqueness weight could be $1/5$

.

- S_i

: Contribution score of member i

, calculated based on the transactions they included and their uniqueness weights.

- N

: Total number of committee members.

Assumptions

- Each transaction’s uniqueness weight is calculated prior to the reward distribution process.
- The total contribution score (S_{total}

) is the sum of all individual member scores, where each member’s score is the sum of the uniqueness weights of the transactions they included.

Contribution Score Calculation

The contribution score S_i

for member i

is calculated as:

$$S_i = \sum_{j=1}^{t_i} u(t_j)$$

where t_j

is each transaction included by member i

, and $u(t_j)$

is the uniqueness weight of transaction t_j

.

Reward Calculation for a Given Member

The reward R_i

for member i

is then:

$$R_i = \left(\frac{S_i}{S_{total}} \right) \times R$$

where $S_{total} = \sum_{i=1}^{N} S_i$

is the total of all members’ contribution scores.

Given the above, the function to calculate the reward for a given member i

simplifies to:

$$\left(\frac{\sum_{j=1}^{t_i} u(t_j)}{\sum_{k=1}^{N} \sum_{j=1}^{t_k} u(t_j)} \right) \times R$$