Proposal: Decrease Censorship Delay from 24 hours to 4 hours

Category: Constitutional

- Core

Submitted by: Shotaro

Abstract

This Proposal proposes to decrease the censorship resistant delay time for transaction force inclusion from 24 hours to 4 hours.

Moreover, as a consequence, this change restricts the power of the sequencer from backdating transactions down to 4 hours instead of 24 hours.

Motivation

Arbitrum chains today run in a [trusted sequencer mode](#) in which a server operated by Offchain Labs orders L2 transactions.

Censorship resistant L2 transactions however can be submitted through the delayedInbox, and force included after a delay.

Dapps, protocols, and L3s building on Arbitrum chains are constrained by worst case analysis — that is, what happens if the centralized sequencer misbehaves? The sequencer can't produce fraudulent transactions, but it can censor transactions thereby forcing a 24 hour delay in transaction inclusion on L2. In particular, this censorship possibility imposes a 24 hour minimum latency constraint on all optimistic mechanisms designs.

Rationale

As detailed in Arbitrum docs, force inclusion is delayed on purpose so that a trusted sequencer can include transactions in an orderly manner. L1 finality requires at least 12.8 minutes. A trusted sequencer can voluntarily include transactions in the L1 delayed inbox after waiting for L1 finality. This allows the sequencer to give users a soft-guarantee of transaction ordering without waiting 12.8 min for finality.

Ethereum mainnet finality [1] takes ~64-95 slots, so 20 min on the upper range. The sequencer needs additional time to order transactions including any delayed transactions it chooses to include. 1 Hour total should suffice. However, in the interests of an abundance of caution while counter balancing the benefits of low latency 4 hours is chosen as a compromise. A 4 hour delay is a 6x decrease in latency for optimistic mechanisms implemented in any Dapp, Protocol, or L3 deployed on Arbitrum chains. A 4 hour delay maintains a healthy 4-8 times safety margin on the quality of life benefit of soft-finality guarantees provided by the trusted sequencer.

Specifications

Call setMaxTimeVariations in the [SequencerInbox](#) to update its old parameters,

MaxTimeVariation({ delayBlocks = 5760; futureBlocks = 12; delaySeconds = 86400 futureSeconds = 3600; })

to its new parameters

MaxTimeVariation({ delayBlocks = 1200; futureBlocks = 12; delaySeconds = 14400; futureSeconds = 3600; })

Note: futureBlocks and futureSeconds unchanged.

Side note: the current delayBlocks is based on outdated ethpow consensus [2] rules about time synchronization using 15 second max timestamp change versus the ethpos 12 second slot time.

Changes shall be applied to Arbitrum One, Arbitrum Nova, and Arbitrum Goerli SequencerInbox contract deployments [3]. Note that the current MaxTimeVariation parameter for all chains is identical.

- Arbitrum One 0x1c479675ad559DC151F6Ec7ed3FbF8ceE79582B6

- Arbitrum Nova 0x211E1c4c7f1bF5351Ac850Ed10FD68CFfCF6c21b

- Arbitrum Goerli 0x0484A87B144745A2E5b7c359552119B6EA2917A9

Steps to Implement

1. Execution of the parameter change

Timeline

Minimum of 34 days for the constitutional governance process.

A goal of implementing the parameter change within 3 months.

Overall Cost

Governance contract gas cost interactions. I am willing to subsidize the proposal creation and relevant contract interactions if delegates with a proposing quorum are willing to create the proposal on my behalf.

Links

I am a new user so I am limited to 2 links. Posting more resources as strings below

[1] - notes.ethereum(dot)org/@vbuterin

/single_slot_finality

[2] - github(dot)com/ethereum/go-ethereum/blob/00a73fbcce3250b87fc4160f3deddc44390848f4/consensus/ethash/consensus.go#L46

[3] - developer.arbitrum(dot)io/useful-addresses