

# BUIP146: Add config parameter to delay mining transactions

Submitted by: Jonathan Silverblood

Date: 2019/5/11

## Summary

The purpose of this BUIP is to make Bitcoin Unlimited a better choice for miners when used in a high-frequency network where transactions are created at a high rate, but propagation has imperfect latencies.

## Proposal

This BUIP proposes that:

- Make a short study on a testnet to determine if there are possible benefits to delaying inclusion of a transaction into the block template
- Add a new configurable parameter to determines how long the node should wait before considering a new transaction as ready to be mined.

## Motivation

To make the best use of our current block propagation protocols (graphene, xthin, xthinner - all that is based on some kind of set reconciliation) it is important to not have too strong fragmentation between the mempools of the peers in the network. By not including new transactions before we have given them a reasonable chance to propagate we are creating beneficial conditions for the propagation of created blocks which should reduce the number of orphans produced when mining with Bitcoin Unlimited as the full node software of choice.

## Background

During a previous stress test on Bitcoin Cash, one of several network bottlenecks discovered related to the propagation of transactions. Not all nodes in the network will have the same performance, so we will always need to be able to work with nodes of varying ability to propagate transactions at low latency.

**Note:** In the discussion following this proposal, some of the assumptions that this proposal relies on have been broken. For example, it is said that the delay between accepting a transaction and mining on it is already ~30 seconds long. If this is the case both today and in the future, then this proposal to add extra delay is likely pointless. I suspect it is not, however, as from practical experience I have many times sent a TX and got included in a block within just a handful of seconds.