BUIP056: Increase the Block Size Limit triggered by a support threshold

Proposer: Tomas van der Wansem
(Based on: BUIP055 by Peter R)
Submitted: 2017-05-12
edit: sponsored by @Zangelbert Bingledack

Abstract

This proposal allows nodes to be configured to change their block size limit when a support threshold is reached.

Motivation

BUIP056 is similar to BUIP055, but which I believe better may align with the requirements of miners:
As outlined in the motivation for BUIP055, the EB/AD configuration currently lacks a mechanism for miners to coordinate changes to the `max_block_size`.

BUIP055 solves this by allowing miners to preannounce a change to their `max_block_size` at a certain height. A drawback of that approach is that miners cannot predict whether there is enough support among miners for the change at the specified height.

This BUIP offers a way to change the `max_block_size` setting based on support for the new value among miners. By choosing a support threshold percentage, they can trigger a change of the block size limit guarded by what is for them the most relevant parameter for allowing the change.

Specification

max_block_size calculation

To determine the `max_block_size` used to verify a target block, a node will use the variables:

`current_limit` is the current size limit.

`new_limit` is the new size limit that will be activated when the threshold is reached.

`threshold` is the minimum number of supporting blocks in a difficulty period to trigger activation.

`new_limit` must be larger then `current_limit` and `threshold` must be a multiple of 5 between 50 and 100 inclusive.

For our calculation, we define a block set as a set of blocks in a single difficulty period at least 5 difficulty periods in the past; that is, a set of blocks before the target block with heights between `y * 2016` and `(y * 2016) + 2015` inclusive, for any non-negative integer `y` such that the height of the target block is larger than `(y * 2016) + 2015 + 10080`.

Given a target block, let X,N be a number pair with the following conditions:

* 2016 >= N >= `threshold`
* `current_limit` < X <= `new_limit`
* There exists a block set, in which at least N% of the blocks signal a `new_limit` equal to or larger than X and a `threshold` equal to or smaller than N.

If no number pair X,N exists, `current_limit` is used as `max_block_size` for the target block. Otherwise the highest value of X from all X,N pairs is used as the `max_block_size` for the target block.

Coinbase and user-agent signalling

Building on the format specified in BUIP005, the relevant variables are signalled
in the coinbase transaction as

"/EB<current_limit_MB>/FE<new_limit_MB>@<threshold>%/..."

and in the user-agent string as

"<user-agent>(EB<current_limit_MB>;FE<new_limit_MB>@<threshold>%...)/"

Rationale

* Only block sets at least 10080 blocks deep are considered to allow for an activation period in which more mining power can join.

* The block set is aligned with the difficulty period to minimize the risk of a chain split. Miners are strongly disincentives to stay on the minority as it will take at least ~8 weeks before difficulty adjustment with a 75% threshold.

Alternatives

* BUIP055 provides a coordinated EC upgrade using a target height. A drawback of this approach is that miners cannot predict future support. A fixed hight may result in retracting or postponing target heights due to insufficient support which can damage reputations and cause a loss of momentum.

* BIP135 Version bit signaling can also be used to schedule based on threshold. However version bit proposals cannot individually encompass a minimum threshold or a maximum threshold. This could be solved by using multiple proposals, but voting on multiple conflicting proposals can lead to complicated problems.

Changes

2015-05-13
* Changed to block count instead of percentage
* Changed EB=>FE
* Removed % from coinbase. Fixed typo
* Changed back to percentage, multiple of 5