SMA Algorithm Developer Spec

A complete explanation and motivation for the algorithm can be found here.

Constants

TargetBlockDelay = 1 (second)

DifficultyAdjustmentWindowSize = 2640 (blocks)

TimestampDeviationTolerance = 132 (block delays)

*MaxTimeOffsetSeconds = TimestampDeviationTolerance * TargetBlockDelay*

Algorithm

blockWindow(b block, size int):

1. return last *size* blocks in b.Past by Phantom order

calcBlockTarget(b block):

1. *bluestParent = b.parents.bluest()*
2. *DifficultyAdjustmentWindow = blockWindow(bluestParent, DifficultyAdjustmentWindowSize)*
3. *AdjustmentFactor* = (*DifficultyAdjustmentWindow.MaxTimestamp - DifficultyAdjustmentWindow.MinTimestamp*) / (*TargetBlockDelay * DifficultyAdjustmentWindowSize)*
4. return *DifficultyAdjustmentWindow.AvarageTarget * AdjustmentFactor*

Changes to block acceptance rules

Do not accept block *b* if any of the following is true:

1. Block in the future *b.Timestamp - systemClock.Now > MaxTimeOffsetSeconds*
    a. In this case, don't reject, but rather delay until it's time is acceptable
2. Block in the past: *b.Timestamp < blockWindow(b, 2 * TimstampDeviationTolerance - 1).MedianTimestamp*