BUIP098: Bitcoin Unlimited's Strategy for the November 2018 Hard Fork

"Run Bitcoin Unlimited to vote for compromise"

Date: 22 August, 2018
Proposer: Andrew Stone

There are 2 changesets proposed for the November 2018 hard fork that have a variety of supporters but can be summarized as coming from Bitcoin ABC and nChain. To review:

nChain:

1. Increase block size to 128MB
2. Re-activate additional opcodes (OP_MUL, OP_LSHIFT, OP_RSHIFT, OP_INVERT)
3. Remove restriction on the number of instructions executed per script (currently 200)

ABC (official announcement):

1. Include OP_CHECKDATASIG
2. Limit transaction size to > 100 bytes (to solve a possible but expensive attack detailed here)
3. Lexical transaction ordering
4. Consensus enforcement that scriptsig (spend scripts) contain only data push instructions

It is ironic that these changesets are mutually compatible, yet both groups reject the other's changes. There may be some specific critiques (see Appendix B) of various proposals, but the core the rationale behind the rejections seem to be the same used to block Group tokenization -- fewer changes are better because every change introduces risk. Additionally, there is concern that the blocking of certain features is happening due to undisclosed patented technologies that compete with the proposed features. By blocking the feature, the patent remains valuable.

Representatives of Bitcoin Unlimited have explored the idea of compromise with representatives from both groups with no success so far, even the smallest changes (like changing a constant to one better suited) have been rejected. Given the "no changes, no matter how reasonable, except mine" strategy being pursued by both of these organizations, I can only sadly conclude that this is again about power and ego not about technical merit and end user adoption.

I believe that the proponents of Bitcoin Cash need to stick together and come to a compromise, rather than fork and face another dispersion of economic activity. This is the essential conclusion of Metcalfe's law. With the 30 day median block size at 36.6Kb, I invite you to examine the above feature list and identify those whose inclusion will compensate for splitting the community due to the dramatic and rapid increase in adoption that the feature enables.

When I proposed the original plan of hard forks every 6 months the intention was to onboard as many people as possible by including many use cases, and accept the risk these changes implied. This strategy has been a failure. The periodic hard fork has not been used to activate any feature that resulted in significant new consumer-focused use cases. Such changes may modify just a few lines of code, but have large ramifications on the use of the blockchain and the community is concerned about this. Instead the periodic hard fork is

being used to “bundle” individual organizations’ favorite features into a single “swallow the sweet with the bitter” package.

I would like to propose a strategy for Bitcoin Unlimited for the near future. In essence, our message will be “run Bitcoin Unlimited to vote for compromise”. The Bitcoin Unlimited client will incorporate features from both organizations and allow these features to either be activated via BIP135 (a generalized form of BIP9 miner voting via version bits), explicit configuration, or (development time and feasibility permitting) emergent consensus. By allowing BIP135, we move to a miner voting process that allows individual features to gain agreement before activation. By allowing explicit configuration -- that is, allowing a user to force the feature “on” or “off” -- people running the BUcash full node can quickly react to any hash-power surprises.

Appendix A: A few notes on BIP135

BIP135 is a superset of BIP9. BIP9 has hard-coded activation thresholds and times and these are quite optimistic. For example, it proposed a 95% activation threshold, yet during the fight for larger blocks it became clear that well over 5% of the hash power actually had much larger investments in alt-coin hashing hardware. Although the economic model of Bitcoin assumes that 51% of the hash power wants what is “best” for the currency, its is a flawed to assume that for 100% of the hash power.

BIP135 allows activation thresholds and times to be configured. Note that these can be configured with the BIP9 values to make a particular activation bit backwards compatible with BIP9-only full nodes.

Note that BIP135 allows for a grace period after a feature is “locked in” and before it actually activates on mainnet. This period is used to allow clients that have not implemented the feature to actually implement it.

BIP135 also defines an end to the voting process, so failed initiatives can be removed from clients that pre-implemented them. It is our expectation that part of a reasonable path forward would be a common understanding that version bits voting should be used when at least one implementation has the corresponding feature set available and well-tested.

Finally, note that BIP135 implicitly allows feature obsolescence and removal some time after activation. The removal of a something can itself be defined as a “feature” and assigned a bit.

Appendix B: General Arguments Against Various Features & BU Specific Notes

Note: I don’t necessarily agree or disagree with any of the arguments presented here. This is a general summary of arguments [and rebuttals in brackets] that I have heard. Actually, I think that quite a few of them are invalid, but make up your own mind. Please comment if you would like another argument or rebuttal added.

1. 128MB block size increase:
    1. There is no need at this time. Blocks are not even close to 1MB! It’s just for marketing. [when adoption comes it will be a tsunami]

    2. Dangerous. We have not even exercised 32MB blocks on mainnet. No pool has mined a block > 8MB.

    3. BU note: We already support 128MB blocks
2. Re-activation of opcodes

    1. Proposal came too late -- it missed the generally-agreed upon spec and code deadlines. [but spec and code does seem to be available within the date]

    2. BU note: Code here:
        1. D1592 - Add SCRIPT_ENABLE_MAGNETIC_OPCODES flag - https://reviews.bitcoinabc.org/D1592
        2. D1593 - Expand IsOpcodeDisabled() - https://reviews.bitcoinabc.org/D1593
        3. D1594 - rename monolith_opcodes.cpp to … - https://reviews.bitcoinabc.org/D1594
        4. D1631 - enable magnetic opcodes in .. - https://reviews.bitcoinabc.org/D1631
        5. D1598 - OP_MUL implementation - https://reviews.bitcoinabc.org/D1598
        6. D1606 - OP_INVERT implementation - https://reviews.bitcoinabc.org/D1606
        7. D1638 - OP_LSHIFT & OP_RSHIFT - https://reviews.bitcoinabc.org/D1638
        8. D1631 - enable magnetic upgrade in tests - https://reviews.bitcoinabc.org/D1631
3. Removal of instruction execution count restrictions

    1. Significant time is needed for people to think about attack scripts, consuming too much CPU or memory, for example.
    2. no use cases
    3. provided use cases can be done off chain with zero knowledge proofs
    4. The idea of freezing the instruction set and then using hundreds or thousands of instructions to implement general purpose primitives like EC multiplication is incredibly wasteful of UTXO and blockchain history space

    5. BU note: small LOC change (nchain commit: b47906926fe5b71549d1b422f2219ccdd10a5a0d)
4. OP_CHECKDATASIG

    1. Is not part of Bitcoin's original instruction set [we have already diverged from the original instruction set. Limiting changes to restoring the original set is the authority (nostalgia?) fallacy]

    2. Can be used to enable wagering which may be illegal in some jurisdictions [bitcoin itself is illegal in some jurisdictions, information is not generally illegal, and the blockchain should not and realistically cannot enforce legality on its participants just like cars cannot and do not enforce speed limits]

    3. BU note: We have significant experience here and can implement quickly
5. Limit transaction size to > 100 bytes (fixes this)

    1. Transaction != 64 bytes fixes the problem, why choose 100?

    2. Attack is extremely expensive and only tricks SPV wallets [but SPV wallet access is very important for bitcoin cash, and the attack will become cheaper]

    3. There may be a change to the MERKLEBLOCK protocol message that also fixes the problem -- that is, a non-consensus fix may exist (https://lists.linuxfoundation.org/pipermail/bitcoin-dev/2018-August/016298.html)

    4. BU note: small LOC change

6. Make scriptsig contain only data push instructions
    1. This is already enforced at the network level, so probably not contentious, yet not in the nChain list
7. Lexical Transaction Ordering
    1. All the value is available through voluntary, non-consensus-enforced, transaction ordering. [But by enforcing it, we ensure that the most efficient is always used]
    2. The stated original purpose (to enable parallel validation) is actually a feature of removing dependency ordering, not adding lexical ordering, and it even turns out that that's not true -- the proposed parallel validation algorithm can be implemented to efficiently verify dependency ordering.
    3. It is a major change, for little to no impact. [it prepares us for huge 1GB+ future blocks]
    4. There is no spec, just marketing documents
    5. There is no rush, so let's defer and investigate the possibilities (https://gist.github.com/gavinandresen/e20c3b5a1d4b97f79ac2#canonical-ordering-of-transactions)