```
BIP: 135
Title: Generalized version bits voting
Author: Sancho Panza <sanch0panza@protonmail.com>
Comments-Summary: No comments yet.
Comments-URI: https://github.com/bitcoin/bips/wiki/Comments:BIP-0135
              https://bitco.in/forum/threads/bip9-generalized-version-bits-voting-bip-genv
Status: Rejected
Type: Informational
Created: 2017-03-29
License: CC0-1.0
         GNU-All-Permissive
Post-History: https://lists.linuxfoundation.org/pipermail/bitcoin-dev/2017-April/013969.ht
Replaces: 9
```

Abstract

BIP9 introduced a mechanism for using the version bits to signal support for backwards-compatible changes (soft-forks) using a tally over the previous 2016 blocks computed at re-targeting intervals. It provided for a fixed threshold and non-configurable lock-in interval applicable to all deployments on a chain.

This document describes a generalized signaling scheme which allows each signaling bit to have its own configurable threshold, window size (number of blocks over which it is tallied) and a configurable lock-in period.

It extends the semantics of the signaling bits to cover arbitrary consensus changes, referred to under the general term 'forks'. The same range of version bits is used for signaling.

The states of the BIP9 state machine and its original parameters (name, bit, starttime, timeout) are retained. Some state transition conditions are extended by additional parameters ('threshold', 'windowsize', 'minlockedblocks', 'minlockedtime') to provide for fine-tuning of threshold and grace period.

Motivation

The Bitcoin protocol requires a flexible scheme for finding consensus on protocol changes, to ensure that it can adapt to the needs of the market and remain competitive as an electronic payment system.

While BIP9 has served the community well for previous deployments, there are some shortcomings in its approach:

- it specifically applies only to backward-compatible changes
- its fixed 95% threshold is not flexible enough to allow for a 'spectrum of contentiousness' to be represented

- small minorities can veto proposed changes, which can lead to undesirable stagnation

A generalized revision of the BIP9 specification can address these issues and satisfy the needs of the market for both soft and hard fork changes as well as more flexible activation thresholds and upgrade (grace) periods.

The proposal should allow more freedom of choice in activation strategies while remaining backward compatible with respect to existing BIP9-based deployments.

Terms and conventions

The version bits used by this proposal for signaling deployment of forks are referred to as 'signaling bits' or shortened to 'bits' where unambiguous.

All times in this specification are in seconds since the epoch [1]. Durations / time offsets are in seconds.

The term 'MTP' refers to the 'median time past' which is calculated as the median nTime of a block and its 10 predecessors. It is treated as a monotonic clock defined by a chain, and evaluated on the ancestor of a block, i.e.

MTP := **GetMedianTimePast(block.parent)**

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119.

Specification

Backward compatibility

This specification SHALL enable strict backward compatibility with existing BIP9-based deployments through suitable parameter configuration. Any part of the specification preventing full backward compatibility SHALL be considered as erroneous and amended.

As before, a set of configuration parameters SHALL exist for the version bits for each chain supported by an implementation. This permits each bit to be configured independently for each chain (mainnet, testnet, etc.)

Signaling bits

The signaling bits SHALL comprise the 29 least significant bits of the nVersion block header field. nVersion is a 32-bit field which is treated as a little-endian integer.

Signaling bits SHALL be assigned numbers from 0..28 ranging from the least significant (bit 0) to the most significant (bit 28) in the range.

The top 3 bits of nVersion MUST be set to 001 , yielding a range of possible nVersion values between [0x20000000...0x3FFFFFFF], inclusive.

If a block's nVersion does not have its top 3 bits set to 001, all its signaling bits MUST be treated as if they are 0 (see also: 'Tallying' section below).

Deployment states

With each block and fork, we associate a deployment state. The possible states are:

1. **DEFINED** is the first state that each fork starts out as. The genesis block for any chain SHALL by definition be in this state for each deployment.
2. **STARTED** for blocks past the starttime.
3. **LOCKED_IN** after STARTED, if at least threshold out of windowsize blocks have the associated bit set in nVersion, measured at next height that is evenly divisible by the windowsize.
4. **ACTIVE** for all blocks after the grace period conditions have been met.
5. **FAILED** if past the timeout time and LOCKED_IN was not reached.

In accordance with BIP9, a block's state SHALL never depend on its own nVersion; only on that of its ancestors.

Fork deployment parameters

Each fork deployment is specified by the following per-chain parameters:

1. The **name** specifies a very brief description of the fork, reasonable for use as an identifier. For deployments described in a single BIP, it is recommended to use the name "bipN" where N is the appropriate BIP number.
2. The **bit** determines which bit in the nVersion field of the block is to be used to signal the fork deployment. It is chosen from the set {0,1,2,...,28}.
3. The **starttime** specifies a minimum median time past (MTP) of a block at which the bit gains its meaning.
4. The **timeout** specifies a time at which the deployment is considered failed. If the MTP of a block >= timeout and the fork has not yet locked in (including this block's bit state), the deployment is considered failed on all descendants of the block.
5. The **windowsize** specifies the number of past blocks (including the block under consideration) to be taken into account for locking in a fork.
6. The **threshold** specifies a number of blocks, in the range of 1..windowsize, which must signal for a fork in order to lock it in. The support is measured when the chain height is evenly divisible by the windowsize. If the windowsize is set to 2016 (as in BIP9) this coincides with the 2016-block re-targeting intervals.
7. The **minlockedblocks** specifies a minimum number of blocks which a fork must remain in locked-in state before it can become active. Both

minlockedblocks and minlockedtime (see below) must be satisfied before a fork can become active.

8. The **minlockedtime** specifies a minimum grace time, an earliest time after lock-in at which the fork can become active. If the MTP of a block >= (minlockedtime + median time of the block that locked in the fork), then the fork becomes activated. Both minlockedtime and minlockedblocks (see above) must be satisfied before a fork can become active.

Tallying

If a block's nVersion does not have its top 3 bits set to 001, all its signaling bits MUST be treated as if they are '0'.

A signaling bit value of '1' SHALL indicate support of a fork and SHALL count towards its tally on a chain.

A signaling bit value of '0' SHALL indicate absence of support of a fork and SHALL NOT count towards its tally on a chain.

The signaling bits SHALL be tallied whenever the head of the active chain changes (including after reorganizations).

State transitions

The following diagram illustrates the generalized state machine:

NOTES:

The genesis block of any chain SHALL have the state DEFINED for each deployment.

A given deployment SHALL remain in the DEFINED state until it either passes the starttime (and becomes STARTED) or the timeout time (and becomes FAILED).

Once a deployment has STARTED, the signal for that deployment SHALL be tallied over the the past windowsize blocks whenever a new block is received on that chain.

A transition from the STARTED state to the LOCKED_IN state SHALL only occur when all of these are true:

- the height of the received block is an integer multiple of the window size
- the MTP is below the timeout time
- at least threshold out of windowsize blocks have signaled support

A similar height synchronization precondition SHALL exist for the transition from LOCKED_IN to ACTIVE. These synchronization conditions are expressed by the "mod(height, windowsize) = 0" clauses in the diagram, and have been

been added so that backward compatibility with BIP9's use of the 2016-block retargeting periods can be configured for existing deployments (see above 'Optional full backward compatibility' section).

A transition from LOCKED_IN to ACTIVE state SHALL only occur if the height synchronization criterion is met and two configurable 'grace period' conditions are fulfilled:

1. current height MUST be at least minlockedblocks above LOCKED_IN height
2. MTP must exceed LOCKED_IN time by at least minlockedtime seconds

NOTE: If minlockedtime and minlockedblocks are both set to 0, then the fork will proceed directly to ACTIVE state once the chain height reaches a multiple of the windowsize.

The ACTIVE and FAILED states are terminal; a deployment stays in these states once they are reached.

Deployment states are maintained along block chain branches. They need re-computation when a reorganization happens.

New consensus rules

New consensus rules deployed by a fork SHALL be enforced for each block that has ACTIVE state.

Optional operator notifications

An implementation SHOULD notify the operator when a deployment transitions to STARTED, LOCKED_IN, ACTIVE or FAILED states.

It is RECOMMENDED that an implementation provide finer-grained notifications to the operator which allow him/her to track the measured support level for defined deployments.

An implementation SHOULD warn the operator if the configured (emitted) nVersion has been overridden to contain bits set to '1' in contravention of the above non-signaling recommendations for DEFINED forks.

It is RECOMMENDED that an implementation warn the operator if no signal has been received for a given deployment during a full windowsize period after the deployment has STARTED. This could indicate that something may be wrong with the operator's configuration that is causing them not to receive the signal correctly.

For undefined signals, it is RECOMMENDED that implementation track these and alert their operators with supportive upgrade notifications, e.g.

- "warning: signaling started on unknown feature on version bit X"
- "warning: signaling on unknown feature reached X% (over last N blocks)"

- "info: signaling ceased on unknown feature (over last M blocks)"

Since parameters of these deployments are unknown, it is RECOMMENDED that implementations allow the user to configure the emission of such notifications (e.g. suitable N and M parameters in the messages above, e.g. a best-guess window of 100 blocks).

getblocktemplate changes

The getblocktemplate features introduced in BIP9 remain in effect unmodified.

Rationale

The timeout into FAILED state allows eventual reuse of bits if a fork was not successfully activated.

A fallow period at the conclusion of a fork attempt allows some detection of buggy clients, and allows time for warnings and software upgrades for successful forks. The duration of a fallow period is not specified by this proposal, although a conventional fallow period of 3 months is RECOMMENDED.

Due to the constraints set by BIP 34, BIP 66 and BIP 65, there are only 0x7FFFFFFB possible nVersion values available. This limits to at most 30 independent deployments. By restricting the top 3 bits to 001 we we are left with 29 out of those for the purposes of this proposal, and support two future upgrades for different mechanisms (top bits 010 and 011).

Guidelines

Parameter selection guidelines

The following guidelines are suggested for selecting the parameters for a fork:

1. **name** SHOULD be selected such that no two forks, concurrent or otherwise, ever use the same name.
2. **bit** SHOULD be selected such that no two concurrent forks use the same bit. Implementors should make an effort to consult resources such as [2] to establish whether the bit they wish to use can reasonably be assumed to be unclaimed by a concurrent fork, and to announce their use ('claim') of a bit for a fork purpose on various project mailing lists, to reduce chance of collisions.
3. **starttime** SHOULD be set to some date in the future, approximately one month after a software release date which includes the fork signaling. This allows for some release delays, while preventing triggers as a result of parties running pre-release software.
4. **timeout** is RECOMMENDED to be 1 year (31536000 seconds) after starttime.
5. **windowsize** SHOULD be set large enough to allow reception of an adequately precise signal. A good high-resolution value would be 2016 blocks

as used in BIP9. It is NOT RECOMMENDED to use a windowsize less than 100 blocks.

6. **threshold** SHOULD be set as high as possible to ensure a smooth activation based on the estimated support and the nature of the proposed changes. It is strongly RECOMMENDED that threshold >= windowsize / 2 (rounded up) to ensure that a proposal is only activated by majority support.
7. **minlockedblocks** is RECOMMENDED to be set >= windowsize, to ensure that a full window passes in LOCKED_IN state. Lower values will be ineffective as the transition from LOCKED_IN to ACTIVE is guarded by a synchronization based on the window size.
8. **minlockedtime** SHOULD only be set > 0 if a minimum LOCKED_IN time period needs be strictly enforced. It is permissible to set minlockedblocks to 0 and only specify minlockedtime, however the synchronization condition means the grace period can only expire once the time has passed AND the chain height is a multiple of the windowsize.

NOTE: If minlockedtime and minlockedblocks are both set to 0, then the fork will proceed to ACTIVE state when the chain height reaches a multiple of the windowsize.

A later deployment using the same bit is possible as long as the starttime is after the previous fork's timeout or activation, but it is discouraged until necessary, and even then recommended to have a pause in between to detect buggy software.

Signaling guidelines

An implementation SHOULD signal '0' on a bit if one of the following holds true:

- the deployment parameters are not DEFINED (not configured or explicitly undefined)
- the deployment is DEFINED and has not yet reached the STARTED state
- the deployment has succeeded (it has become ACTIVE)
- the deployment has FAILED

An implementation SHOULD enable the operator to choose (override) whether to signal '0' or '1' on a bit, once its deployment has at least reached the STARTED state.

An implementation SHOULD warn the operator if the configured (emitted) nVersion has been overridden to contain bits set to '1' in contravention of the above non-signaling recommendations.

A supporting miner SHOULD signal '1' on a bit for which the deployment is LOCKED_IN state so that uptake is visible. However, this has no effect on consensus rules. Once LOCKED_IN, a deployment proceeds to ACTIVE solely based on the configured grace period parameters (see 'Fork deployment parameters' above).

A miner SHOULD signal '0' on a bit if they wish to suspend signaling of support for a fork that is DEFINED in their software.

It is NOT RECOMMENDED to signal '1' for bits where the meaning is undefined (i.e. bits which are unclaimed by proposals).

Settings for BIP9 compatibility

This section lists parameter values which can be used to effect compatibility with the existing BIP9 versionbits state machine.

The following table describes mainnet compatibility options (95%, 2016 blocks):

Parameter
name
bit
starttime
timeout
windowsize
threshold
minlockedblocks
minlockedtime

The following table describes testnet compatibility options (75%, 2016 blocks):

Parameter
name
bit
starttime
timeout
windowsize
threshold
minlockedblocks
minlockedtime

Deployment

As this BIP is not itself consensus-relevant (Information like BIP9), it can be rolled out without the use of a BIP9 fork bit.

Backward compatibility through judicious fork configuration parameters should ensure that it does not interfere with existing known deployments.

By way of design it does not interfere with unknown (undefined) deployments.

Reference implementation

A working reference implementation, including tests, can be found in these Pull Requests:

- https://github.com/BitcoinUnlimited/BitcoinUnlimited/pull/458
- https://github.com/bitcoin/bitcoin/pull/10437

Existing unit tests and regression tests have been left active to demonstrate backward compatibility of the default settings with BIP9.

References

[1] http://pubs.opengroup.org/onlinepubs/9699919799/xrat/V4_xbd_chap04.html#tag_21_04_16

[2] List of existing BIP9 deployment proposals

Copyright