```
BIP: 125
Layer: Applications
Title: Opt-in Full Replace-by-Fee Signaling
Author: David A. Harding <dave@dtrt.org>
        Peter Todd <pete@petertodd.org>
Comments-Summary: No comments yet.
Comments-URI: https://github.com/bitcoin/bips/wiki/Comments:BIP-0125
Status: Proposed
Type: Standards Track
Created: 2015-12-04
License: PD
```

Abstract

Many nodes today will not replace any transaction in their mempool with another transaction that spends the same inputs, making it difficult for spenders to adjust their previously-sent transactions to deal with unexpected confirmation delays or to perform other useful replacements.

The opt-in full Replace-by-Fee (opt-in full-RBF) signaling policy described here allows spenders to add a signal to a transaction indicating that they want to be able to replace that transaction in the future. In response to this signal,

- Nodes may allow transactions containing this signal to be replaced in their mempools.
- The recipient or recipients of a transaction containing this signal may choose not to treat it as payment until it has been confirmed, eliminating the risk that the spender will use allowed replacements to defraud them.

Nodes and recipients may continue to treat transactions without the signal the same way they treated them before, preserving the existing status quo.

Summary

This policy specifies two ways a transaction can signal that it is replaceable.

- **Explicit signaling:** A transaction is considered to have opted in to allowing replacement of itself if any of its inputs have an nSequence number less than (0xffffffff - 1).
- **Inherited signaling:** Transactions that don't explicitly signal replaceability are replaceable under this policy for as long as any one of their ancestors signals replaceability and remains unconfirmed.

Implementation Details

The initial implementation expected in Bitcoin Core 0.12.0 uses the following rules:

One or more transactions currently in the mempool (original transactions) will be replaced by a new transaction (replacement transaction) that spends one or more of the same inputs if,

1. The original transactions signal replaceability explicitly or through inheritance as described in the above Summary section.

1. The replacement transaction may only include an unconfirmed input if that input was included in one of the original transactions. (An unconfirmed input spends an output from a currently-unconfirmed transaction.)

1. The replacement transaction pays an absolute fee of at least the sum paid by the original transactions.

1. The replacement transaction must also pay for its own bandwidth at or above the rate set by the node's minimum relay fee setting. For example, if the minimum relay fee is 1 satoshi/byte and the replacement transaction is 500 bytes total, then the replacement must pay a fee at least 500 satoshis higher than the sum of the originals.

1. The number of original transactions to be replaced and their descendant transactions which will be evicted from the mempool must not exceed a total of 100 transactions.

The initial implementation may be seen in Bitcoin Core PR#6871 and specifically the master branch commits from 5891f870d68d90408aa5ce5b597fb574f2d2cbca to 16a2f93629f75d182871f288f0396afe6cdc8504 (inclusive).

Receiving wallet policy

Wallets that display unconfirmed transactions to users or that provide data about unconfirmed transactions to automated systems should consider doing one of the following:

1. Conveying additional suspicion about opt-in full-RBF transactions to the user or data consumer.

1. Ignoring the opt-in transaction until it has been confirmed.

Because descendant transactions may also be replaceable under this policy through inherited signaling, any method used to process opt-in full-RBF transactions should be inherited by any descendant transactions for as long as any ancestor opt-in full-RBF transactions remain unconfirmed.

Spending wallet policy

Wallets that don't want to signal replaceability should use either a max sequence number (0xffffffff) or a sequence number of (0xffffffff-1) when they also want to use locktime; all known wallets currently do this. They should also take care not to spend any unconfirmed transaction that signals replaceability explicitly or

through inherited signaling; most wallets also currently do this by not spending any unconfirmed transactions except for those they created themselves.

Wallets that do want to make replacements should use explicit signaling and meet the criteria described above in the Implementation Details section. A Bitcoin Wiki page has been created to help wallet authors track deployed mempool policies relating to transaction replacement.

The initial implementation makes use of P2P protocol reject messages for rejected replacements, allowing P2P clients to determine whether their replacements were initially accepted by their peers. Standard P2P lightweight client practice of sending to some peers while listening for relays from other peers should allow clients to determine whether the replacement has propagated.

Motivation

Satoshi Nakamoto's original Bitcoin implementation provided the nSequence number field in each input to allow replacement of transactions containing that input within the mempool. When receiving replacements, nodes were supposed to replace transactions whose inputs had lower sequence numbers with transactions that had higher sequence numbers.

In that implementation, replacement transactions did not have to pay additional fees, so there was no direct incentive for miners to include the replacement and no built-in rate limiting that prevented overuse of relay node bandwidth. Nakamoto removed replacement from Bitcoin version 0.3.12, leaving only the comment, "Disable replacement feature for now".

Replacing transactions with higher-fee transactions provided a way for spenders to align their desires with miners, but by the time a Replace-by-Fee (RBF) patch was available to re-enable replacement, some receivers had begun to expect that the first version of a transaction they saw was highly likely to be the version of the transaction to be confirmed, and so some users advocated that replacement should be disallowed.

To address those concerns, a variation on RBF was created that required that the replacement transaction pay all of the same outputs as the original transaction in equal or greater amount. This was called RBF First Seen Safe (RBF-FSS), and the original RBF became known as full-RBF. Although agreeable to recipients who relied on the first-seen version of a transaction, each use of RBF-FSS required adding an extra input to a transaction, resulting in wallets being unable to use it if they had no spare inputs, a loss of privacy when inputs from different origins get used in the same transaction, and a wasteful increase in transaction byte size.

Opt-in full-RBF uses Nakamoto's original semantics (with a slight tweak to allow locktime users to opt-out) to signal that replacement is possible, providing first-seen users with the ability to ignore those transactions while also allowing for the efficiency benefits of full-RBF.

There are no known problematic interactions between opt-in full-RBF and other uses of nSequence. Specifically, opt-in full-RBF is compatible with consensus-enforced locktime as provided in the Bitcoin 0.1 implementation, draft BIP68 (Relative lock-time using consensus-enforced sequence numbers), and draft BIP112 (CHECKSEQUENCEVERIFY).

Deployment

Now, and since Bitcoin's first release, 100% of the network hash rate mines transactions using opt-in full-RBF semantics (sequence less than (0xffffffff - 1)).

Opt-in full-RBF as a default mempool replacement policy among nodes and miners is expected to become widespread as they upgrade to Bitcoin Core 0.12.0 (release expected Jan/Feb 2016) and similar node software such as Bitcoin LJR.

Actual replacement may be unreliable until two conditions have been satisfied:

1. Enough nodes have upgraded to support it, providing a relay path for replacements to go from spending wallets to miners controlling significant amounts of hash rate.

1. Enough hash rate has upgraded to support replacement, allowing for reasonable probability that a replacement can be mined.

Backwards compatibility

At the time opt-in RBF support was added/proposed, no known wallet created transactions by default with nSequence set below (0xffffffff - 1), so no known wallet explicitly signaled replaceability by default. Also no known popular wallet spent other users' unconfirmed transactions by default, so no known wallets signaled inherited replaceability.

See also

1. Transaction Replaceability on Bitcoin Wiki targeted at helping wallet authors use RBF

1. Tools for creating opt-in full-RBF transactions: https://github.com/petertodd/replace-by-fee-tools#replace-by-fee-tools

1. Reddit: Questions about opt-in RBF targeted at helping community members understand opt-in full-RBF

Copyright