```
BIP: 133
Layer: Peer Services
Title: feefilter message
Author: Alex Morcos <morcos@chaincode.com>
Comments-Summary: No comments yet.
Comments-URI: https://github.com/bitcoin/bips/wiki/Comments:BIP-0133
Status: Draft
Type: Standards Track
Created: 2016-02-13
License: PD
```

Abstract

Add a new message, "feefilter", which serves to instruct peers not to send "inv"'s to the node for transactions with fees below the specified fee rate.

Motivation

The concept of a limited mempool was introduced in Bitcoin Core 0.12 to provide protection against attacks or spam transactions of low fees that are not being mined. A reject filter was also introduced to help prevent repeated requests for the same transaction that might have been recently rejected for insufficient fee. These methods help keep resource utilization on a node from getting out of control.

However, there are limitations to the effectiveness of these approaches. The reject filter is reset after every block which means transactions that are inv'ed over a longer time period will be rerequested and there is no method to prevent requesting the transaction the first time. Furthermore, inv data is sent at least once either to or from each peer for every transaction accepted to the mempool and there is no mechanism by which to know that an inv sent to a given peer would not result in a getdata request because it represents a transaction with too little fee.

After receiving a feefilter message, a node can know before sending an inv that a given transaction's fee rate is below the minimum currently required by a given peer, and therefore the node can skip relaying an inv for that transaction to that peer.

Specification

1. The feefilter message is defined as a message containing an int64_t where pchCommand == "feefilter"
2. Upon receipt of a "feefilter" message, the node will be permitted, but not required, to filter transaction invs for transactions that fall below the feerate provided in the feefilter message interpreted as satoshis per kilobyte.

3. The fee filter is additive with a bloom filter for transactions so if an SPV client were to load a bloom filter and send a feefilter message, transactions would only be relayed if they passed both filters.
4. Inv's generated from a mempool message are also subject to a fee filter if it exists.
5. Feature discovery is enabled by checking protocol version >= 70013

Considerations

The propagation efficiency of transactions across the network should not be adversely affected by this change. In general, transactions which are not accepted to a node's mempool are not relayed by this node and the functionality implemented with this message is meant only to filter those transactions. There could be a small number of edge cases where a node's mempool min fee is actually less than the filter value a peer is aware of and transactions with fee rates between these values will now be newly inhibited.

Feefilter messages are not sent to whitelisted peers if the "-whitelistforcerelay" option is set. In that case, transactions are intended to be relayed even if they are not accepted to the mempool.

There are privacy concerns with deanonymizing a node by the fact that it is broadcasting identifying information about its mempool min fee. To help ameliorate this concern, the implementation quantizes the filter value broadcast with a small amount of randomness, in addition, the messages are broadcast to different peers at individually randomly distributed times.

If a node is using prioritisetransaction to accept transactions whose actual fee rates might fall below the node's mempool min fee, it may want to consider disabling the fee filter to make sure it is exposed to all possible txid's.

Backward compatibility

Older clients remain fully compatible and interoperable after this change. Also, clients implementing this BIP can choose to not send any feefilter messages.

Implementation

https://github.com/bitcoin/bitcoin/pull/7542

Copyright

This document is placed in the public domain.