What's New in Bitcoin Core 0.13.0?

Aug 22, 2016 02:46 PM by Aaron van Wirdum

Bitcoin Core 0.13.0, the thirteenth generation of Bitcoin's reference client as first launched by Satoshi Nakamoto almost eight years ago, has now been tagged for release. This is one of the final steps in the software release process and initiates the Gitian build process.

Bitcoin Core 0.13.0 was developed by some 100 contributors over a period of about five months. And while much of the development effort over this time has also been focused on Segregated Witness, which will be activated only in a future minor release of the software, Bitcoin Core 0.13.0 includes about a dozen notable improvements compared to Bitcoin Core 0.12.0.

These are the most important changes.

Child Pays for Parent

The number of transactions on the Bitcoin network has been steadily growing over time. As a result, more blocks have been filling up, and miners typically charge higher fees to include transactions into blocks. Transactions that don't include sufficient fees usually take longer to confirm, or perhaps even never confirm at all. This has proved to be somewhat problematic, especially in periods where so-called “stress tests” were conducted on the network, with spikes in the total number of transactions on the network and substantial transaction delays.

Individual users can solve this problem by including a higher fee in their transactions, incentivizing miners to prioritize these transactions. This is possible even after a transaction is sent, using Opt-in Replace-by-Fee (RBF); however, not many wallets include this option yet. Additionally, RBF is only an option for the sender of a transaction. Up till now, the receiver had no way to bump the fee for an incoming transaction to speed up confirmation.

This problem is effectively solved with a trick called “Child Pays for Parent” (CPFP). CPFP is a policy used by miners to select which transactions to include in blocks. With CPFP, miners don't necessarily pick the highest paying (and valid) transactions, but instead pick the most profitable *set* of transactions. In other words: they will select a low-fee transaction if a subsequent transaction that *relies* on the low-fee transaction offers a high enough fee to compensate. The miner will include both at the same time.

In practice, this means that the receiver of a low-fee transaction can “attach” a

In practice, this means that the receiver of a low-fee transaction can "attach" a high-fee transaction, spending the same coins to himself. Incentivized by the new, high-fee transaction, a miner will include the set of transactions. As such, the receiver won't have to wait as long for a confirmation, while the miner can increase his income.

Compact Block Support

Bitcoin's peer-to-peer protocol is currently somewhat inefficient. Nodes send each other most transaction data twice: once as a transaction as it is initially sent over the network, and once as part of a block when the transaction is confirmed.

This has some disadvantages. For one, sending transaction data twice requires more bandwidth than it really should, which adds to the cost of running Bitcoin Core. Second, and perhaps more importantly, forwarding new blocks to several peers at the same time can cause significant outbound bandwidth spikes. This potentially disrupts internet-usage each time a new block is found, which is potentially annoying for users. And perhaps, more importantly, it can slow down block propagation over the network as well. Slow block propagation can, in turn, favor bigger mining pools, thereby incentivizing a more centralized mining landscape.

Compact Blocks (BIP 152), developed by Bitcoin Core and Blockstream developer Matt Corallo, are designed to decrease excess data-transmission. When a new block is found, nodes initially only communicate very compact hashes of transaction data. As nodes have already received the full transaction data when it was originally sent over the network, they can use these hashes to figure out which transactions are included in the block and reconstruct the block themselves.

This trick does not always work out perfectly, however. If a node did not yet receive the initial transaction before receiving the hashes, that node, of course, can't select the transaction. Additionally, in rare cases a *wrong* transaction may hash into a *right* hash, fooling the node into believing it received the right transaction until it tries to reconstruct the block and finds it doesn't add up.

In both these cases of failure, the node simply requests the specific transaction data after all. Even with only some complete transactions in them, Compact Blocks will transmit over the network much faster, and require significantly less bandwidth.

Hierarchical Deterministic Key Generation

Up till now, Bitcoin Core generated a new and completely random public and private key pair for each new Bitcoin address. While this is important for security and privacy reasons, it can also be a bit of a burden for users. In order to secure all private keys against loss, they need to make regular backups.

Hierarchical Deterministic (HD) Key Generation (BIP 32), a cryptographic trick developed throughout 2012 and 2013 by Bitcoin Core developers Gregory Maxwell and Dr. Pieter Wuille, and Armory-developer Alan Reiner, solves this problem. With HD key generation, Bitcoin Core creates a completely new key pair for each new address, but all these keys are derived from a single, 12-word seed. As long as users remember this 12-word seed, they can re-generate all private keys and access all their funds.

It should be noted that HD Key Generation is not a new feature in the Bitcoin world. Many wallets already included the option for several years. It just never existed in Bitcoin's reference client — until now.

Performance & Security

And of course, Bitcoin Core 0.13.0 introduces a significant list of performance and security upgrades. The full extent of these improvements is beyond the scope of this article (see Bitcoin Core 0.13.0's release notes for all the details), but in short…

The database cache memory has been increased, which allows nodes to speed up transaction validation and more. The Bitcoin command line tool now allows users to type passphrases and other sensitive information interactively, improving security by not storing this information in plain text. The software is updated to use C++11 and Python 3, newer versions of the programming languages, that allow for more powerful features. ARM (a specific microprocessor architecture) binaries for Linux are now part of the release, so users don't have to compile this for themselves. Data concerning which transactions in a mempool rely on each other (as utilized with CPFP) can be communicated to external programs. Nodes on the network can request to receive only transactions that meet a certain fee threshold to prevent DoS attacks. And, lastly, there have been a lot of low-level improvements to the peer-to-peer, remote procedure call, and messaging system (ZMQ) protocols.

Thanks to Bitcoin Core lead maintainer Wladimir van der Laan and Bitcoin Core developer and Ciphrex CEO Eric Lombrozo for information and feedback.

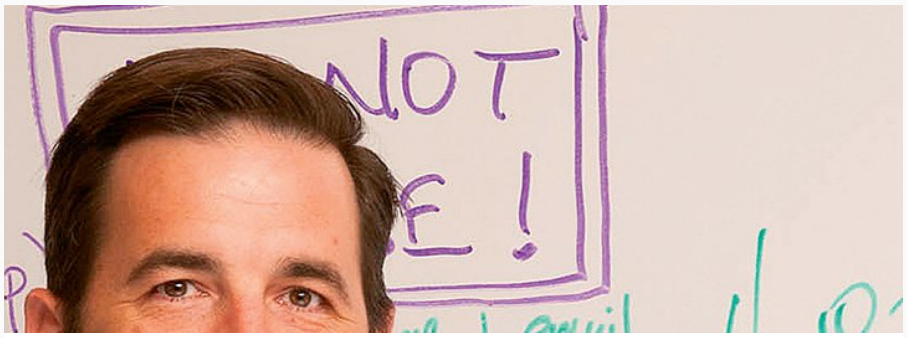
Note: Bitcoin Core 0.13.0 was officially released on August 23; the original title of this article, announcing its release on August 22, was incorrect. It had been tagged for release, which typically means it will be released within a couple of days. Bitcoin Core 0.13.0 can now be downloaded on bitcoincore.org and bitcoin.org. (But before downloading, note this safety warning on bitcoin.org.)

by Aaron van Wirdum

Aaron van Wirdum (Twitter / email / pgp / bitcoin) has a background in political history, with a specialization in the effects of new technology on societies. He discovered Bitcoin in early 2013, and has been writing his way through the rabbit hole since. Interested in decentralized consensus, FOSS, privacy in the digital age, censorship resistant payments, and more of that stereotypical Bitcoin stuff.

KEYWORDS: #bitcoin #core #transactions #data #users

Related Articles:



FEB 11, 2016

BitPay's Stephen Pair: Community Needs to Become Proficient at Managing Bitcoin Forks

#CORE-DEV

FEB 23, 2016

Bitcoin Core 0.12.0 Released: What's New?

#CORE-DEV

FEB 24, 2016

MIT Media Lab Director Joi Ito Speaks Up on Bitcoin Technical Development Environment

#CORE-DEV

FEB 25, 2016

Gavin Andresen: Bitcoin Core Is Not Listening to Its Customers

#CORE-DEV