Understanding Segwit Block Size

Jimmy Song Follow

Jul 3, 2017 · 7 min read

After I wrote my last article, I was surprised by the protest about the 2MB part of the title (the title has since been changed). The protest stems from the fact that Segwit2x would allow block sizes that are larger than 2MB (it's possible to get very close to 8MB). This is often a point of confusion and also a point of very contentious debate.

In this article, I'm going to describe the difference between how Segwit determines block transaction capacity vs. how it is determined currently. If you've ever heard something like "Segwit is a block size increase" and didn't understand what that meant, this article is written with you in mind. I'll also describe how this affects Segwit2x and the block transaction capacity that we can expect to see before and after the 2x hard fork.

Block Size and Soft Fork

To talk accurately about block sizes, it's important to understand how block size is measured. Block size is simply the size, in bytes, of the serialized block (block header, number of txs and the txs themselves).

As part of the consensus rules, every node on the Bitcoin network currently checks that a block is less than 1,000,000 bytes. That is, a block that's greater than 1,000,000 bytes will be rejected by these nodes as a consensus rule.

Because legacy nodes (that is, nodes that don't upgrade) will reject a block that's greater than 1,000,000 bytes, any soft fork must keep this rule. But how can you increase block size and still keep this rule? Are larger blocks via a soft fork even possible?

Enter Segregated Witness

It was thought that the answer to the last question was "no" until Segregated Witness entered the picture. The fact that this can be done as a soft fork and allow for more transactions is an engineering breakthrough. The key insight is that a big part of the transaction, the scriptSig (signature, pubkey, etc), can essentially not be sent to Legacy nodes and still be counted as valid.

BIP143 defines new transaction outputs that do exactly this. p2wpkh and p2wsh are very similar to p2pkh and p2sh respectively, but move scriptSig data to the end of the transaction.

Non-Segwit transactions (defined as txs only spending non-Segwit output like p2pk, p2pkh and p2sh) put the scriptSig in the middle of the transaction. Segwit transactions (defined as txs spending at least one p2wpkh or p2wsh output) put the scriptSig at the end. The scriptSig part of Segwit transactions is called the "witness data". When Segwit transactions are sent to Legacy nodes *the witness data is stripped*. The key is that these "stripped" transactions are still valid transactions on Legacy nodes, which gives us a savings over non-Segwit transactions. Thus, more transactions can fit into the block sent to Legacy nodes without going over the 1,000,000 byte limit.

Segwit nodes get Segwit transactions and blocks that *include* the witness data using alternate network messages. The new network messages are defined in BIP144 as part of Segwit. The Segwit blocks *which include the witness data* can be over 1,000,000 bytes. Legacy nodes, as mentioned, receive the same blocks and transactions, but with the witness data stripped. This is a way to make Segwit a soft fork.

Restricting Segwit Blocks

The creators of Segwit could have let Segwit blocks be as large or as small as they wanted and Segwit would still have been a soft fork, provided the blocks sent to Legacy nodes are still 1,000,000 bytes or under. A 1MB Segwit block restriction would not increase block size at all and a 1GB Segwit block restriction would open up a very obvious attack vector. To restrict Segwit blocks, the creators of Segwit instead came up with a different restriction than size.

The Segwit blocks are restricted by something called *Block Weight*. Block Weight is a new concept introduced in Segwit, and it's calculated on a per-transaction basis. Each transaction has a "weight" which is defined this way:

(tx size *with witness data stripped*) * 3 + (tx size)

Non-Segwit transactions have zero witness data, so the weight for a non-Segwit transaction is exactly 4 times the size. Segwit transactions have some witness data so the weight is going to be less than 4 times the size. Note Segwit transactions are transmitted to Legacy nodes *without* witness data, so this formula will always result in blocks communicated to Legacy Nodes that are less than or equal to 1,000,000 bytes. Again, this is why Segwit is a soft fork.

Block Weight

Instead of the previous (legacy) consensus rule which stated that a block cannot be more than 1,000,000 bytes, the new, tighter, consensus rule is that the total block weight allowed for any block is 4,000,000. Notice that if you fill a block with only non-Segwit transactions, this is equivalent to a block size limit of 1,000,000 bytes (1,000,000 * 4 = 4,000,000).

Thus, we can say the Block Transaction Capacity of non-Segwit transactions is the same as before Segwit. That is, you cannot fill Segwit blocks with non-Segwit transactions and be over 1,000,000 bytes. This is because non-Segwit transaction weight is directly proportional to size. 1000-byte non-Segwit transactions will always have a weight of 4000 no matter what the transaction is composed of.

This is not the case for Segwit transactions. 1000-byte Segwit transactions can have many different weights depending on how much of the transaction is taken up by Witness data. Consider two different transactions:

- A 1000 byte Segwit transaction with 500 bytes of witness data
- A 1000 byte Segwit transaction with 300 bytes of witness data

The first would have a weight of 500*3+1000=2500 whereas the second would have a weight of 700*3+1000=3100. From a miner's perspective, if both transactions had the same fee, they would make more money including the former transaction as it takes up less weight in a block despite the same size.

Similarly, transactions of different sizes can have the same weight. Consider 3 different transactions:

- An 800-byte Segwit transaction with 400 bytes of Witness data
- An 1100-byte Segwit transaction with 800 bytes of Witness data
- A 500-byte Non-Segwit transaction

All of these transactions have the same exact weight of 2000. When serialized to a Legacy node, they will be 400 bytes, 300 bytes and 500 bytes respectively (tx size without witness data). When serialized to a Segwit node, they will be 800 bytes, 1100 bytes and 500 bytes respectively.

At this point, you may be wondering how much Witness Data will be in a typical Segwit transaction. This depends on the number of p2wpkh or p2wsh inputs, but roughly speaking, Segwit Transactions are about 2/3 Witness Data.

So How Big Can Segwit Blocks Get?

Blocks received by legacy nodes will be less than 1,000,000 bytes. Blocks received by Segwit nodes, on the other hand, can be bigger, but how much bigger?

It turns out that if you made a pathologically large Segwit transaction, you can make a block with just the coinbase transaction and a

pathologically large Segwit transaction that's very close to 4MB. Essentially, this pathologically large Segwit transaction would be mostly Witness Data with the block weight just under 4,000,000. That block would be very close to 4MB, but way under 1,000,000 bytes when stripped of witness data. This is an extraordinary case and wouldn't be very profitable for a miner unless that transaction also had an extraordinarily high fee.

The normal case without pathologically large/giant fee Segwit transactions results in a block size of around 2MB, which is what the creators of Segwit designed for. When you hear someone say "Segwit is a block size increase", this is what they're referring to. The average Segwit block size will be roughly 2MB, though Legacy nodes will still receive blocks that are 1,000,000 bytes or lower due to stripped witness data.

Segwit2x

Which brings us back to the brouhaha over the title of my last article. What Segwit2x does is it increases the maximum Block Weight to 8,000,000. By doing so, this increases the non-Segwit Transaction Capacity in a given block to 2,000,000 bytes. That is, it basically upgrades the "Legacy" block size to 2,000,000. Now, "Legacy" anything doesn't really make sense in a hard-fork scenario (every node has to upgrade, thus there is no "Legacy"), but that's essentially the effect that the authors of Segwit2x were after and why I called it a 2MB hard fork in the first place.

But my protesters have a point. The Segwit Block size (the only block size really left after a hard fork as there are no Legacy nodes) has a maximum of close to 8MB after the planned hard fork in Segwit2x. The normal case is actually more like 4MB.

Conclusion

It's a little weird talking about the block size with Segwit since the size isn't the way blocks are restricted. Segwit blocks are restricted by weight and that's a related, but different calculation. If there are no Segwit transactions in a block, this weight calculation collapses to size, but in the more general case of blocks with Segwit transactions, miner profit does not strictly increase with block size.

For this reason, miners are not incentivized to make a block the maximum possible block size. This is because after a certain point (2mb without segwit2x hard fork, 4mb with), miners have to *remove* transactions in order to make the block size go up as they're likely already up against the block weight limit. That, of course, reduces fees and makes mining a larger block add cost for a miner.

In other words, you can expect miners to mine around 2MB blocks with Segwit only and 4MB blocks with Segwit and 2x hard fork. What you can be sure of, however, is that miners will maximize the weight of each block as miner profit strictly increases with block weight. Blocks, then will have close to 4,000,000 weight with Segwit only and 8,000,000 weight with Segwit and 2x hard fork.

Segwit makes block size a less relevant concept. Block Weight is the more accurate and useful metric to judge blocks by, though it certainly has a relationship to size.

Want to get curated Technical Bitcoin News? Sign up for the Bitcoin Tech Talk newsletter!